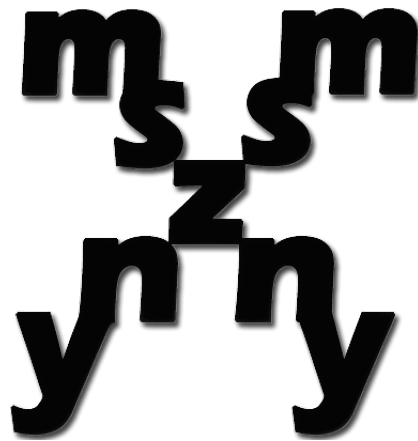


# XVIII. Magyar Számítógépes Nyelvészeti Konferencia



Szerkesztette:  
Berend Gábor  
Gosztolya Gábor  
Vincze Veronika

Szeged, 2022. január 27–28.

**Szerkesztette:**

Berend Gábor, Gosztolya Gábor, Vincze Veronika  
{berendg,ggabor,vinczev}@inf.u-szeged.hu

**Felelős kiadó:**

Szegedi Tudományegyetem  
TTIK, Informatikai Intézet  
6720 Szeged, Árpád tér 2.

**ISBN:** 978-963-306-848-9

**Nyomtatta:**

JATEPress  
6722 Szeged, Petőfi Sándor sugárút 30–34.

Szeged, 2022. január

**Az MSZNY 2022 konferencia szervezője:**

ELKH-SZTE Mesterséges Intelligencia Kutatócsoport

## Előszó

2022. január 27–28-án már tizennyolcadik alkalommal kerül sor a Magyar Számítógépes Nyelvészeti Konferencia megrendezésére. Idén azonban a tavalyi évhez hasonlóan, ismét a virtuális térben tartjuk meg konferenciánkat, a COVID-19 járványügyi helyzetre való tekintettel. Ugyanakkor bízunk benne, hogy a személyes találkozások és eszmecserek hiánya ellenére is sikeres, és szakmailag mindenkit gazdagító eseménynek nézünk elébe.

A konferencia fő célkitűzése a kezdetek óta állandó: lehetőséget biztosítani a nyelv- és beszédtechnológia területén végzett kutatások eredményeinek ismertetésére és megvitatására, ezen felül a különféle hallgatói projektek, illetve ipari alkalmazások bemutatására. A hagyományokat követve a konferencia idén is nagyfokú érdeklődést váltott ki az ország nyelv- és beszédtechnológiai szakembereinek körében. A 48 beküldött cikkből gondos mérlegelést követően 45-öt fogadott el a programbizottság, melyek témája számos szakterületre terjed ki a legújabb nyelvi modellek bemutatásától kezdve a beszédtechnológia eredményein keresztül a gépi fordításig.

Nagy örömet jelent számunkra, hogy Zódi Zsolt elfogadta meghívásunkat, aki plenáris előadásában megvilágítja a mesterséges intelligencia jogi vonatkozású kérdéseit.

Az idei évben is különállással jutalmazzuk a konferencia legjobb cikkét, mely a legjelentősebb eredményekkel járul hozzá a magyarországi nyelv- és beszédtechnológiai kutatásokhoz. Ezen felül immár negyedik alkalommal osztjuk ki a legjobb bíráló díját, amellyel a bírálók fáradságos, ugyanakkor nélkülözhetetlen munkáját kívánjuk elismerni.

Köszönettel tartozunk az ELKH-SZTE Mesterséges Intelligencia Kutatócsoportjának és a Szegedi Tudományegyetem Informatikai Intézetének helyi szervezésben segédkező munkatársainak. Végezetül szeretnénk megköszönni a programbizottság és a szervezőbizottság minden tagjának áldozatos munkáját, ami nélkül nem jöhetett volna létre a konferencia.

A szervezőbizottság nevében,  
Ács Judit, Berend Gábor, Gosztolya Gábor, Novák Attila, Sass Bálint, Simon Eszter, Sztahó Dávid, Vincze Veronika





# Tartalomjegyzék

<b>Nyelvmodellek</b>	<b>1</b>
3 Nyelvspecifikus transzformer modellek közötti megfeleltetéssel történő zero-shot jelentésegértelműsítés <i>Berend Gábor</i>	
15 BARTerezzünk! Messze, messze, messze a világtól - BART kísérleti modellek magyar nyelvre <i>Yang Zijian Győző</i>	
31 Okosabb vagy, mint egy XXXXXXXX? – Egy nyelvi játéktól a nyelvmodellek összehasonlításáig <i>Indig Balázs, Lévai Dániel</i>	
45 Charmen ELECTRA: Tokenizációmentes diszkriminatív nyelvi modellezés <i>Ficsor Tamás, Cserhádi Réka, Novák Attila, Mihajlik Péter, Zainkó Csaba, Berend Gábor</i>	
59 HuSpaCy: an industrial-strength Hungarian natural language processing toolkit <i>György Orosz, Zsolt Szántó, Péter Berkecz, Gergő Szabó, Richárd Farkas</i>	
<b>Szintaxis</b>	<b>75</b>
77 Igekötő-kapcsolás <i>Pethő Gergely, Sass Bálint, Kalivoda Ágnes, Simon László, Lipp Veronika</i>	
93 Adalékok a birtokos szerkezetekbe sűrített hasonlatok számítógépes azonosításához <i>Bajzát Tímea Borbála, Szemes Botond, Szlávich Eszter</i>	
105 Three-order normalized PMI and other lessons in tensor analysis of verbal selectional preferences <i>Márton Makrai</i>	
121 An introductory statistical study of Hungarian word order <i>Dávid Márk Nemeskey</i>	
<b>Beszédtechnológia</b>	<b>135</b>
137 Length Analysis of Speech to be Recorded in the Recognition of Parkinson’s Disease <i>Attila Zoltán Jenei, Dávid Sztahó</i>	

- 151 Sclerosis Multiplex hangalapú felismerése akusztikai alapú beágyazások használatával  
*Gosztolya Gábor, Tóth László, Svindt Veronika, Bóna Judit, Hoffmann Ildikó*
- 161 Hangkorverzió alkalmazása dysarthriás betegek beszédminőségének javítására  
*Terbe Dániel, Tóth László, Ivaskó Lúvia*
- 175 Enyhe kognitív zavar automatikus felismerése szekvenciális autoenkóder használatával  
*Vetráb Mercedes, José Vicente Egas-López, Balogh Réka, Imre Nóra, Hoffmann Ildikó, Tóth László, Pákáski Magdolna, Kálmán János, Gosztolya Gábor*
- 185 Deep learning alapú agyi jel feldolgozás és beszéd-szintézis előkészítő munkálatai  
*Arthur Frigyes Viktor, Csapó Tamás Gábor*
- 199 Releasing the BEAST - A BEA gépi beszédleiratozási feladat, megközelítések és eredmények  
*Mihajlik Péter, Balog András, Gráczi Tekla, Kohári Anna, Fegyó Tibor, Mády Katalin*

## **Alkalmazások**

**211**

- 213 Nyelvtanulást elősegítő feladatok automatikus előállítása finn és magyar nyelvekre  
*Ferenczi Zsanett*
- 227 Magyar nyelvű Fedőnevek-ágensek létrehozása nyers szövegek alapján épített gráf felhasználásával  
*Cserhádi Réka, Kolláth István, Kicsi András, Berend Gábor*
- 241 Absztraktív összefoglaló PreSumm módszerrel  
*Agócs Ádám, Yang Zijian Győző*
- 257 Twitter bot detection using deep learning  
*Ádám Kenyeres, György Kovács*
- 271 Forensic Authorship Classification by Paragraph Vectors of Speech Transcriptions  
*Dávid Sztahó, András Beke, György Szaszák, Attila Fejes*
- 281 Absztraktív összefoglalás arab nyelvre  
*Kahla Mram, Yang Zijian Győző*

## Szemantika, gépi fordítás

293

- 295 Mondatszintű szentiment analízis teljesítményének javítása adatkiterjesztéses eljárásokkal  
*Laki László János, Yang Zijian Győző*
- 311 Magyar páciensek narratív tapasztalatainak elemzése BERT témamoddellezéssel és szentimentelemzéssel  
*Osváth Mátyas, Yang Zijian Győző, Kósa Karolína*
- 325 Nagyot mondó képviselők? Fokozás a politikai kommunikációban  
*Szabó Martina Katalin, Vincze Veronika, Ring Orsolya, Guba Csenge*
- 343 Syntax-based data augmentation for Hungarian-English machine translation  
*Attila Nagy, Patrick Nanys, Balázs Frey Konrád, Bence Bial, Judit Ács*
- 357 Jobban fordítunk magyarra, mint a Google!  
*Laki László János, Yang Zijian Győző*

## Korpusz, szövegfeldolgozás

373

- 375 ELTE Verskorpusz – a magyar kanonikus költészet gépileg annotált adatbázisa  
*Horváth Péter, Kundráth Péter, Indig Balázs, Fellegi Zsófia, Szlávich Eszter, Bajzát Tímea Borbála, Sárközi-Lindner Zsófia, Vida Bence, Karabulut Aslihan, Timári Mária, Palkó Gábor*
- 389 NerKor 1.41e  
*Novák Attila, Novák Borbála*
- 403 Az NYTK-NerKor több szempontú kiértékelése  
*Simon Eszter, Vadász Noémi, Lévai Dániel, Nemeskey Dávid, Orosz György, Szántó Zsolt*
- 417 OCR hibák javítása neurális technológiák segítségével  
*Laki László János, Kőrös Ádám, Ligeti-Nagy Noémi, Nyéki Bence, Vadász Noémi, Yang Zijian Győző, Váradi Tamás*
- 431 HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából  
*Ligeti-Nagy Noémi, Ferenczi Gergő, Héja Enikő, Jelencsik-Mátyus Kinga, Laki László János, Vadász Noémi, Yang Zijian Győző, Váradi Tamás*

- 449 A kétrétegű dinamikus pragmatika implementációja  
*Szeteli Ádám, Friedszám Attila, Szeteli Anna, Kárpáti Laura, Hagymási Judit, Kleiber Judit, Alberti Gábor*
- 463 "Az invazív medvék nem tolerálják a szukis agressziót" – Magyar GPT2 kísérleti modell  
*Yang Zijian Győző*
- 477 Lehet-e automatikus családfeépítő programot készíteni?  
*Naszódi Mátyás*
- 491 A CLARIN ParlaMint magyar korpusza  
*Üveges István, Ring Orsolya*
- 505 Towards abstractive summarization in Hungarian  
*Márton Makrai, Ákos Máté Tündik, Balázs Indig, György Szaszák*
- 521 Szövegaugmentálási módszerek összehasonlítása politikai szövegek szentimentanalízise során  
*Üveges István, Csányi Gergely Márk, Ring Orsolya, Orosz Tamás*
- 535 Az ige helyhatározói bővítményeinek megkülönböztetése és az argumentumszerkezeti variánsok korpusz alapú szétválasztása  
*Szécsényi Tibor, Virág Nándor*
- 549 Lightweight diacritic restoration for V4 languages  
*Bálint Csanády, András Lukács*
- 561 Dinamikus szemantikai modell megszorított predikátumokkal  
*Szécsényi Tibor, Kovács Viktória*
- 573 Neme? Nemek közötti nyelvhasználati különbségek a Facebook bejegyzésekben  
*Kelemen Sára Bernadett, Yang Zijian Győző, Kmetty Zoltán*
- 587 Vállalati rendszerbe integrálható természetesnyelv-feldolgozó alkalmazás készítése digital-twin-distiller platformmal  
*Orosz Tamás, Csányi Gergely Márk, Gadó Krisztián, Üveges István, Vági Renáto, Vadász János Pál, Nagy Dániel*
- 603 A kis HIL-ELECTRA, HIL-ELECTRIC és HIL-RoBERTa - Magyar kísérleti nyelvi modellek tanítása kevés erőforrással  
*Yang Zijian Győző, Feldmann Ádám, Váradi Tamás*
- 619 Közérthetőség mint osztályozási probléma (?) - gépi tanulási kísérlet kézzel címkézett korpuszon  
*Üveges István*

- 633 Magyar nyelvű időpont-egyeztető dialógusrendszer v2  
*Nagy Soma Bálint, Herdinai Viktor, Pálfi Gellért, Fegyő Tibor,  
Mihajlik Péter, Farkas Richárd*

**Szerzői index, névmutató**

**645**



# NYELVMODELLEK





# Nyelvspecifikus transzformer modellek közötti megfeleltetéssel történő zero-shot jelentésegértelműsítés

Berend Gábor

Szegedi Tudományegyetem, Informatikai Intézet  
berendg@inf.u-szeged.hu

**Kivonat** Cikkünkben egy nyelvspecifikus transzformer modellekre támaszkodó, a jelentésegértelműsítési feladatot zero-shot módon elvégző eljárást mutatunk be. A javasolt módszer a nyelvközi tudástranszfert a tanítóadatokkal rendelkező forrás-, valamint a tanítóadatokkal nélküli célnyelv feldolgozására dedikáltan létrehozott egynyelvű előtanított modellekre épít. A nyelvek közötti kapcsolatot az egynyelvű transzformer modellek rejtett rétegei közötti megfeleltetést szolgáló leképezés tanulással érjük el. Eredményeink megmutatják, hogy az ilyen módon létrehozott, kizárólag angol nyelvű jelentésegértelműsített szövegeken tanuló modellek hatékonysága szignifikánsan javítható a többnyelvű maszkolt nyelvi modell alkalmazásához képest.

**Kulcsszavak:** jelentésegértelműsítés; zero-shot tanulás

## 1. Bevezetés

A jelentésegértelműsítés a természetesnyelv-feldolgozás egy régóta ismert, központi jelentőséggel bíró problémája (Weaver, 1949/1955; Lesk, 1986; Gale és mtsai, 1992; Navigli, 2009). A megoldási kísérletek között találkozhatunk tudásbázisokra támaszkodó, valamint felügyelt tanulást alkalmazó módszerekkel is, amelyek közül tipikusan az utóbbiak teljesítenek jobban. Mindkét fő megközelítésben közös, hogy komoly humán erőforrás-igénnyel rendelkeznek, hiszen mind a tudásbázisok, mind pedig a jelentésegértelműsítésen átesett tanító-, illetve kiértékelőszövegek létrehozása igen költséges folyamat. Noha angol nyelvre több viszonylag nagy (azonban a jelentések sokszínűségéből adódóan a kívánatosnál még így is elmaradó) jelentésegértelműsítési adatbázis is létezik (Miller és mtsai, 1994; Taghipour és Ng, 2015), a legtöbb nyelvre – köztük a magyarra is – korábban nem létezett megfelelő méretű és részletezettségű adatbázis.<sup>1</sup>

A nemrégiben közreadott XL-WSD adatbázis (Pasini és mtsai, 2021) ezen a helyzeten változtat, ezért is éreztük szükségesnek a különféle kurrens módszereket egymással komplexen összehasonlító jelentésegértelműsítési kiértékelési kísérletsorozat elvégzését, és bemutatását.

<sup>1</sup> Vincze és mtsai (2008) közreadott ugyan egy jelentésegértelműsített korpuszt, azonban abban mindössze 39 többértelmű szóalak különböző jelentéseinek előfordulásai voltak megtalálhatók.

## 2. Kapcsolódó munkák

A kontextualizált szóreprezentációk jelentéségyértelműsítésben való fölhasználhatóságára első ízben (Peters és mtsai, 2018) mutatott rá. Loureiro és Jorge (2019) kísérletei azt igazolták, hogy a BERT (Devlin és mtsai, 2019) maszkolt nyelvi modellből kinyerhető vektorok segítségével egy egyszerű, mégis nagy hatékonyságú 1-legközelebbi szomszédságon alapú osztályozó építhető.

Berend (2020a) egy olyan felügyelet nélküli megoldásra tett javaslatot, ami a kontextuális jelentésvektorokat olyan módon alakítja át, hogy azok a bennük található együttthatók többségében nulla értéket vegyenek föl, az azonos koordináták mentén nemnulla együttthatóval rendelkező szavak pedig tendenciózan azonos jelentéssel rendelkezzenek. A javasolt módszer segítségével létrehozott nagyfokú ritkasággal jellemzett jelentésvektorokat aztán a Loureiro és Jorge (2019) által is alkalmazott 1-legközelebbi módszerrel kombinálva szignifikáns javulások voltak elérhetőek.

Az eddigiekben bemutatott munkák mindegyike az angol nyelven történő jelentéségyértelműsítésre fókuszált, aminek hátterében az áll, hogy magyarra nem létezett korábban kellő méretű és diverzitású jelentéségyértelműsítésre létrehozott tanító-, illetve tesztelő adatbázis. Érdemes megemlíteni a Vincze és mtsai (2008) által megalkotott magyar nyelvű újsághíreket tartalmazó jelentéségyértelműsített HuWSD adathalmazt, azonban ez az erőforrás mindössze 39 többértelmű szóalak vonatkozásában tartalmaz annotációkat, így igazán reprezentatív kiértékelést ez az adatbázis nem tesz lehetővé. A adatbázis méreteiből fakadó limitációi ellenére is születtek többnyelvű transzformer architektúrákat alkalmazó eredmények a HuWSD vonatkozásában is (Berend, 2020b, 2021).

A Pasini és mtsai (2021) által megalkotott XL-WSD adatbázis az angolon kívül 17 további nyelven tartalmaz jelentéségyértelműsítésen átesett diverz szöveget, amelyek kísérleteink alapjául is szolgáltak.

## 3. Módszertan

Vizsgálataink során a transzformer modellekből kinyerhető, módosíthatatlan kontextuális reprezentációkat használó 1-legközelebbi szomszédságon alapuló módszert (Loureiro és Jorge, 2019), valamint a ritkításon átesett kontextuális reprezentációk (Berend, 2020a) használatát hasonlítjuk össze különféle esetekben.

### 3.1. Kontextuális modellek ritkítása

A kontextuális reprezentációk ritkítása során a Berend (2020a) által leírtak szerint jártunk el, azaz egy transzformer modell valamely rétegéből jövő  $d$  dimenziós rejtett reprezentációkat egy  $Y \in \mathbb{R}^{d \times n}$  mátrixban összegyűjtve, a

$$\min_{D \in \mathcal{C}, \alpha \in \mathbb{R}_{\geq 0}^{k \times n}} \frac{1}{2} \|Y - D\alpha\|_F^2 + \lambda \|\alpha\|_1 \quad (1)$$

feladatot oldottuk meg, ahol  $\mathcal{C}$  a legfeljebb 1 normájú oszlopvektorok alkotta  $d \times k$  méretű mátrixok konvex halmazát jelöli,  $\lambda$  a csupa nemnegatív értékből álló  $\alpha$  együtthatómátrix ritkaságát befolyásoló regularizációs együttható,  $k$  pedig a mátrixdekompozíció során alkalmazott atomok számára vonatkozó hiperparaméter.

A ritka kontextuális reprezentációkat a Berend (2020a) által javasolt módon használtuk föl, azaz minden lehetséges  $s_i$  jelentéshez társítottunk egy  $\phi_{s_i} \in \mathbb{R}^k$  vektort, amely vektor a  $D$  szótármátrixban található  $k$  jelentéskomponens és a jelentéségyértelműsített tanítókörpuszban  $s_i$  jelentésüként megjelölt szavak közötti kapcsolat erősségét fejezi ki a pontonkénti kölcsönös információ (PMI) segítségével. A tesztelés során egy  $\alpha_j \in \mathbb{R}^k$  ritkításon átesett kontextuális reprezentációval rendelkező szó kapcsán a modellünk azt az  $s^*$  jelentést választja ki az adott szóhoz, amelyre  $s^* = \max_{s \in S} s^\top \alpha_j$ , ahol  $S$  az adott szó lehetséges jelentéseinek halmazát jelöli.

### 3.2. Nyelvspecifikus transzformerek közötti leképezés

Kísérleteink során zero-shot tanulást alkalmaztunk, azaz úgy értékeltük ki a modelljeink jelentéségyértelműsítésben nyújtott teljesítményét, hogy a létrehozásuk során egyáltalán nem támaszkodtunk magyar nyelvű jelentéségyértelműsített szövegekre. Ez komoly előnyt jelent, hiszen a kellően nagy és jó minőségű tanítóadatbázis létrehozása nagyon költséges lenne. Nem véletlen, hogy az általunk használt XL-WSD adatbázisban is csupán a validációs- és teszhalmaz mondatai tekintendők valódi etalonként, a tanítóhalmaz mondatait gépi fordítás segítségével hozták létre a szerzők. Pasini és mtsai (2021) megmutatták, hogy a zero-shot módon, azaz csupán az angol jelentéségyértelműsített tanítókörpusz alapján, illetve az egyidejűleg több eltérő nyelv támogatására képes transzformer modellre (pl. mBERT, XLM-RoBERTa) támaszkodó modelljeik jobb eredmény elérésére voltak képesek, mint a nyelvspecifikus – ám a gépi fordításból adódóan jóval zajosabb – adatokon tanított alternatív modelljeik.

Az általunk vizsgált zero-shot modellek túlmutatnak a korábbiakban létrehozottaktól, ugyanis a nyelvek közötti tudástranszfert nem többnyelvű enkóderek segítségével kívánjuk kezelni, hanem a nyelvspecifikus modellek különböző rétegei mentén kialakuló rejtett reprezentációk közötti lineáris transzformáció alkalmazásával. A javasolt módszer előnye, hogy ezáltal lehetőségünk van kiaknázni a forrás-, valamint a cél nyelv feldolgozására specifikusan létrehozott transzformer modellek előnyeit, így elkerülhetővé válik a többnyelvű modellekre jellemző ún. *többnyelvűségi átok* (Conneau és mtsai, 2020).

Amennyiben a tesztelés során egy cél nyelvi mondat valamely szavához társuló kontextuális reprezentáció  $\mathbf{x}$ , a cél nyelvből a forrásnyelvbe vivő transzformáció pedig  $W$  által adott, úgy a forrásnyelvi szó ritka reprezentációját a

$$\min_{\alpha \in \mathbb{R}_{\geq 0}^k} \frac{1}{2} \|W\mathbf{x} - D\alpha\|_F^2 + \lambda \|\alpha\|_1, \quad (2)$$

szerint hoztuk létre. Érdemes észrevenni, hogy (1)-el szemben, (2) esetén az optimalizálás már csupán  $\alpha$ -ban történik, ami lehetővé teszi a ritka reprezentációk

hatékony meghatározását. Kísérleteinkben az RCSLS algoritmust (Joulin és mtsai, 2018) használtuk a  $W$  leképezés meghatározására, amelynek célfüggvénye a következők szerint alakul

$$\min_W \sum_{i=1}^n \left( -2x_i^\top W^\top y_i + \frac{1}{k} \sum_{y_j \in \mathcal{N}(Wx_i)} x_i^\top W^\top y_j + \frac{1}{k} \sum_{Wx_j \in \mathcal{N}(y_i)} x_j^\top W^\top y_i \right),$$

ahol az  $(x_i, y_i)$  kontextuális reprezentációk olyan párosait jelölik, amelyeket a célnyelvre, illetve a forrásnyelvre szabott nyelvi modellből nyertünk ki egy-egy azonos minőségben előforduló fordítási szópár vonatkozásában,  $\mathcal{N}$  pedig a tanítás során fölhasznált vektorok közül tér vissza az argumentumában szereplő vektor legközelebbi szomszédjaival.

## 4. Kísérletek

Azon kísérleteink során, amelyben (triviális eszközökkel) többnyelvű környezetben előtanított nyelvi modellek segítségével teremtettük meg a forrás-és a célnyelv közötti kapcsolatot, a 24-rétegből álló, több mint 100 eltérő nyelv feldolgozását támogató XLM-RoBERTa (Conneau és mtsai, 2020) modellre (a továbbiakban röviden XLM-R) támaszkodtunk. Azon esetekben, amikor a nyelvek közötti kapcsolatot utólagosan, egy leképezés tanulásával hoztuk létre, olyankor a 24 rétegből álló **bert-large-cased** modellt használtuk az angol tanítószövegek rejtett reprezentációinak meghatározására, míg a teszteléskor a 12 réteg alkotta **huBERT** modellt vettük igénybe. Az említett modelleket a **transformers** (Wolf és mtsai, 2019) könyvtárt használva értük el.

Mivel a transzformer architektúrán alapuló, nagy előtanított nyelvi modellek eltérő rétegei más típusú feladatok elvégzésére specializálódhatnak (Tenney és mtsai, 2019; Reif és mtsai, 2019) – és mivel a szemantikus viszonyok jellemzően a háló kései rétegeiben manifesztálódnak – a kísérleteink során az enkóderek utolsó négy rétegből (illetve a forrás- és célnyelv kezelésére dedikáltan létrehozott nyelvspecifikus enkódereket használó kísérleteink során ezek kombinációból) jövő reprezentációk alkalmazását vizsgáltuk.

A ritkítással létrehozott kontextuális reprezentációk megalkotása során a 3. fejezetben leírtak szerint jártunk el. Hiperparamétereinket (Berend, 2020a) nyomán  $k = 3000$ , valamint  $\lambda = 0,05$  értékekben határoztuk meg. A Berend (2020a) által alkalmazott módszertantól azon az egy ponton tértünk el, hogy mi nem alkalmaztuk a jelentésprototípusok reprezentációinak létrehozása során azt a normalizáló lépést (Bouma, 2009), amely előzetes vizsgálataink szerint a jelen felállításban minimálisan rontotta volna az eredményeket.

### 4.1. A huBERT és BERT közötti leképezés tanulása

Az RCSLS módszer alkalmazása során az azonos kontextusban álló, megegyező jelentéssel bíró szópárokat a nyelvtanulókat segítő Tatoeba platform alapján létrehozott korpuszból (Tiedemann, 2012) nyertük ki a **datasets** (Lhoest és mtsai,

2021) könyvtár segítségével. Az egyes fordított mondatpárokból származó azon  $(s, t)$  szópárok kontextuális reprezentációira tekintettünk a megfeleltetés tanulása során alkalmas horgonypontként, amelyekre teljesült, hogy a forrásnyelvi mondatból jövő  $s$  szó lehetséges fordításai között megtalálható volt a célnyelvi mondatban szereplő  $t$  szó, és ugyanez  $t$  irányából nézve is igaz (vagyis  $s$  a  $t$  szó egy lehetséges fordítása). Annak ellenőrzésére, hogy egy adott szó egy másik szó fordítása-e, a `word2word` erőforrást (Choe és mtsai, 2020) hívtuk segítségül. A leírtak alapján egyebek mellett az alábbi mondatpárból az aláhúzással és azonos színnel megjelölt szópárokat nyertük ki:

{'hu.': 'A *csigák lassan* másznak.', 'en': ' *Snails move slowly.*}

A nyelvek közötti kapcsolatot megteremtő  $W$  mátrixot húszezer azonos kontextusban szereplő fordítási pár, a forrás- és célnyelv utolsó négy rétegének valamelyikéből származó kontextuális reprezentációjának megfeleltetése mentén hoztuk létre.

## 4.2. A kiértékelő adatbázis

A kiértékelésünk során használt XL-WSD adatbázis magyarra vonatkozó teszt-halmazán 3484 különböző lemma (4138 különböző ragozott alakjának) összesen 4428 előfordulásának jelentéségyértelműsítését kell elvégezni a BabelNet (Navigli és Ponzetto, 2012) jelentéskészletével összhangban. A validációs halmazban 1021 egyedi lemma (1084 ragozott formájának) 1107 címkézett előfordulása található. A kiértékelés során alkalmazott minőségi mutató gyanánt a jelentéségyértelműsítés esetén megszokott F-mértéket használtuk. A benchmark adatbázisra vonatkozó bővebb statisztikák az adatbázist bemutató cikkben található (Pasini és mtsai, 2021).

## 4.3. Eredmények

Az 1. táblázatban az látható, hogy miként alakultak a többnyelvű XLM-R enkóder használata mellett kapott eredményeink a kontextuális reprezentációk módosíthatlanul hagyása, valamint a korábbiakban leírtak szerint végrehajtott ritkítása esetén az eltérő rejtett rétegek alkalmazása mellett. Amint az látható, mind a reprezentációk ritkítása, mind pedig az érintetlenül hagyása esetén a 21. réteg szolgáltatta a leghasznosabb információt, továbbá a ritkítás által hozott javulás mértéke jellemzően +5 pont körül mozgott (az utolsó, 24. réteg esetét leszámítva, ahol a változás mértéke csupán +0,5 volt).

A 2. táblázatban annak a megközelítésnek az eredményeit közöljük, amelyeket az általunk javasolt, az egynyelvű modellek rejtett rétegei közötti transzformáció alkalmazásának használatával értünk el. A vizsgálatainkat – a korábban említett módon – a forrásnyelvi, valamint a célnyelvi modellek utolsó négy rétegei közötti kapcsolat megteremtése mellett végeztük el, azaz a BERT és a huBERT modellekből a kísérleteink során fölhasznált rétegek a  $\{21, 22, 23, 24\} \times \{9, 10, 11, 12\}$  Descartes-szorzatból kerültek ki. A 16 kombináció közül a 2. táblázat azokat az eseteket tartalmazza, amelyek a legjobb teljesítményt voltak képesek nyújtani a

	Réteg	Módosítatlan	Ritkítás után
	21	65.42	<b>70.01</b>
	22	64.32	69.17
	23	63.69	68.13
	24	62.78	63.28

**1. táblázat.** A többnyelvű XLM-R enkóder (és abból származtatott ritka jelentésreprezentációk) használata mellett a háló eltérő rejtett rétegeiből kinyert kontextualizált reprezentációkkal kapott eredmények.

kiértékelő adatbázis validációra szánt részhalmazán. Ez a módosítatlan vektorok használata esetén a 22. és 12., míg a ritkításon átesett vektorok esetén a 23. és 12. rétegek használatát jelentette a forrásnyelv feldolgozására szolgáló BERT, valamint a tesztelés során látott magyar nyelvű szövegek feldolgozását végző hu-BERT vonatkozásában. A 2. táblázat eredményeinek az 1. táblázatban foglaltakkal való összehasonlításából egyértelműen kitűnik, hogy a többnyelvű modellek egynyelvű modellekre történő lecserélésével komoly javulásokat tudunk elérni. Érdeemes továbbá megjegyezni, hogy a Pasini és mtsai (2021) által a magyar részkorpuszon elért eredményei 47.29 és 68.36 F-mérték között mozognak, amely hatékonyságot a javasolt eljárásunkkal sikerült jelentősen meghaladni.

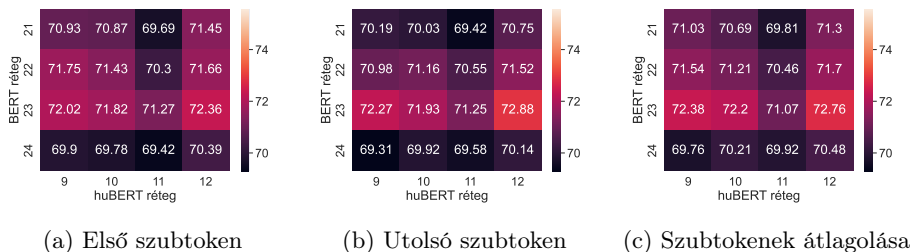
Módosítatlan		Ritkítás után	
Validációs halmaz	Teszt halmaz	Validációs halmaz	Teszt halmaz
73.44	72.76	<b>74.80</b>	<b>75.09</b>

**2. táblázat.** A leképezés tanulása mellett a validációs halmaz alapján kiválasztott legjobbban teljesítő rendszerek eredménye.

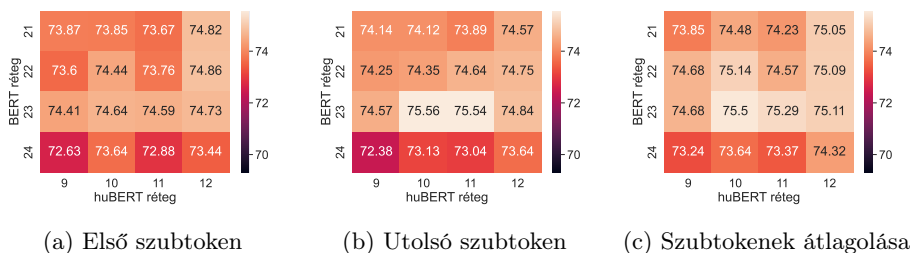
#### 4.4. Szótöredékek kezelése

A transzformer alapú modellek jellegzetessége, hogy az inputszekvenciákat a feldolgozásukat megelőzendő szótöredékek (szubtokenek) sorozatára bontják föl. Ebből adódóan ahhoz, hogy a szószintű rejtett reprezentációk megalkotására képessé váljunk, szükségünk van valamilyen aggregáló (pooling) eljárásra, ami az adott esetben több szótöredékre bontott szavak rejtett reprezentációjából kialakítja a szó egészéhez társítandó vektoros kontextuális reprezentációt. Az eddigiekben bemutatott kísérleteink során azt a gyakran használt módszert alkalmaztuk, amelyik a szószintű kontextuális vektorokat az azokat alkotó szótöredékek kontextuális vektorainak átlagolásával hozza létre.

Ács és mtsai (2021) azt vizsgálták, hogy a különféle transzformer alapú előtárolt nyelvi modellek magyar szövegeken elvégzett morfoszintaktikai osztályozásának hatékonysága mennyiben függ a tokenszintű kontextuális reprezentációk



1. ábra: A módosíthatlan kontextuális reprezentációk használata mellett a különböző forrás-, és célnyelvi enkóderrétegből jövő rejtett reprezentációkra támaszkodó megoldások eredményei eltérő pooling stratégiák alkalmazása esetén.



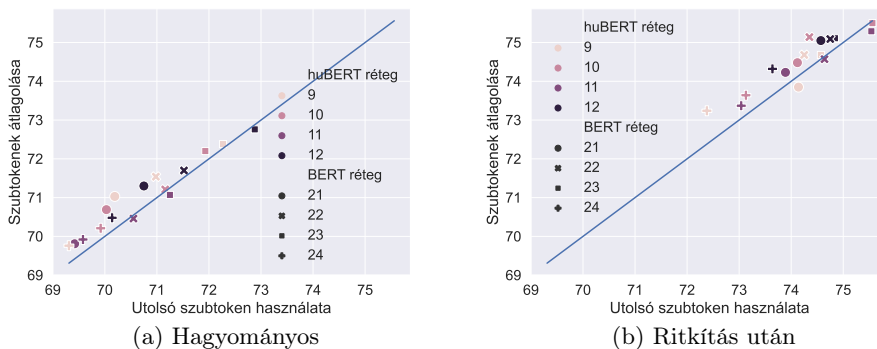
2. ábra: A ritkításon átesett kontextuális reprezentációk használata mellett a különböző forrás-, és célnyelvi enkóderrétegből jövő rejtett reprezentációkra támaszkodó megoldások eredményei eltérő pooling stratégiák alkalmazása esetén.

létrehozása során alkalmazott különböző alternatív aggregáló eljárások megválasztásától. További kísérleteink során a célnyelvünk vonatkozásában mi is az Ács és mtsai által vizsgált lehetőségek összehasonlítását végeztük el, amelyek a következők voltak:

- *első*: tokenen belüli első szubtoken rejtett vektorának használata,
- *utolsó*: tokenen belüli utolsó szubtoken rejtett vektorának használata,
- *átlag*: tokenen belüli szubtokenekhez tartozó rejtett vektorok átlagolása.

Az 1. ábra, valamint a 2. ábra a különféle stratégiák megválasztása esetén elért eredményeinket foglalják össze a BERT és a huBERT modell utolsó négy rétegeiből kinyert módosíthatlan, valamint a ritkításon átesett kontextuális vektorok (és a közöttük tanult leképezés) használata esetén. Mindkét ábrából kitűnik, hogy a forrásnyelvi BERT modell tekintetében az utolsó rétegre támaszkodó próbálkozások szerepeltek a legrosszabbul.

Ennek hátterében két magyarázat is állhat: az egyik, hogy a BERT utolsó rétegeből származó reprezentációk már a forrásnyelv esetében is kevésbé alkalmasak a jelentésértelműsítési feladat elvégzésére, a másik pedig, hogy ezek a



3. ábra: A forrás-és célnyelvspecifikus enkóderek különböző rétegekombinációi esetén kapott eredmények összehasonlítása eltérő pooling stratégiák alkalmazása esetén.

vektorok a forrásnyelv esetében még a többi vizsgált réteg használatával összemérhető eredmény elérésére képesek ugyan, a nyelvközi transzfer minősége azonban lerontja ezen modelleknek a zero-shot helyzetben való alkalmazásának eredményességét. A 3. táblázatban közölt eredmények az első magyarázatot valószínűsítik, az legalábbis mindenképp igaz, hogy a különböző módszerek által az egyes rétegek mentén az angol nyelvű tesztadatokon elért F1-mértékben kifejezett eredmények a BERT utolsó rétegének használata esetén mutatkoztak a legalacsonyabbaknak abban a helyzetben is, amikor a nyelvközi transzfer elvégzésére nem volt szükség, hiszen mind a tanítóadatok mind pedig a tesztelésre szánt adatok angolul álltak rendelkezésre.

Réteg	Módosítatlan	Ritkítás után
21	74.39	77.45
22	74.87	77.60
23	74.45	77.86
24	73.58	76.21

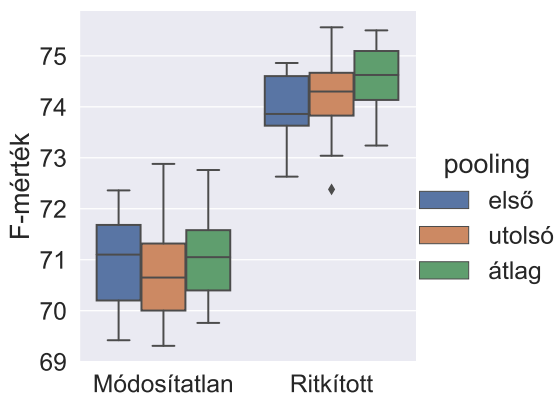
**3. táblázat.** A BERT egyes rétegeiből jövő kontextuális vektorok fölhasználásával elért jelentésegértelműsítési eredmények angol nyelv esetén.

Az 1. ábra, illetve a 2. ábra összevetéséből az is kitűnik, hogy a kontextuális vektorok ritkítása a többnyelvű modellek használatánál látottakhoz hasonlóan az eredmények nagymértékű javulását eredményezte a minden egyéb tekintetben azonosan létrehozott és kiértékelt, de ritkításon át nem esett vektorok használatához képest. Megfigyelhető továbbá, hogy míg a legjobb eredményt az utolsó szubtokenre támaszkodó aggregáló eljárással értük el, összességében nem jelent-



hető ki, hogy a kontextuális szóreprézntációk előállítása során az utolsó szótörödékvektor használata egyértelműen célravezetőbb lenne az egyes szavakhoz tartozó szótörödékvektorok átlagolásánál.

Mindezt a 3. ábra is alátámasztja, ahol egy-egy pont azt reprezentálja, hogy miként viszonyult azon rendszereknek az egymáshoz való eredménye, amelyek minden hiperparaméter vonatkozásában ugyanúgy lettek létrehozva a szövektorok aggregálása során alkalmazott stratégiát leszámítva. Páros t-próba alkalmazásával úgy találtuk, hogy az átlagolással nyert kontextuális szövektorok alkalmazása mellett kapott eredmények átlaga szignifikánsan magasabb az utolsó szótörödékből származó kontextuális szövektorokkal kapott eredményekhez képest mind a módosíthatatlan vektorok ( $p < 0.001$ ), mind pedig a ritkításon átesett vektorok használata esetén ( $p < 0.003$ ). A különböző vektoraggregálási stratégiák mentén kapott eredmények eloszlását a 4. ábrán is megfigyelhetjük.



4. ábra: A különböző tokenaggregálási stratégiák alkalmazásának hatásai a módosíthatatlan, valamint a ritkított esetben.

## 5. Konklúzió

Cikkünkben azt vizsgáltuk, hogy a specializáltan egy adott nyelv feldolgozására létrehozott neurális nyelvmodellek használata milyen előnyökkel jár a soknyelvű nyelvi modellek alkalmazásához képest olyan esetekben, amikor tanítóadatok nem állnak rendelkezésünkre a feldolgozni kívánt forrásnyelven. Mindehhez egy olyan lineáris transzformáció létrehozására tettünk javaslatot, ami az egymástól függetlenül tanított egynyelvű nyelvi modellek reprezentációi közötti kapcsolat megteremtését szolgálja. Az előzőeken túl bemutattuk azt is, hogy a kontextuális jelentésreprézntációk szótártanuláson alapuló, felügyelet nélküli módszerrel

történő ritkításával jelentősen javíthatók a zero-shot módon elvégzett jelentés-egyértelműsítési eredmények. Mindezekon felül megvizsgáltuk a különböző szub-tokenaggregáló stratégiákat is, és arra jutottunk, hogy az általunk vizsgált feladaton, illetve nyelvközi transzfer alkalmazása esetén a szavakat alkotó szóvektorok átlagolása teljesített a legjobban. A kísérleteink során használt forráskódok az [https://github.com/begab/sparsity\\_makes\\_sense](https://github.com/begab/sparsity_makes_sense) URL-ről érhetők el.

## Köszönetnyilvánítás

A dolgozatban szereplő kutatási eredmények létrejöttét az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- Ács, J., Kádár, Á., Kornai, A.: Subword pooling makes a difference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 2284–2295. Association for Computational Linguistics, Online (Apr 2021), <https://aclanthology.org/2021.eacl-main.194>
- Berend, G.: Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8498–8508. Association for Computational Linguistics, Online (Nov 2020a), <https://aclanthology.org/2020.emnlp-main.683>
- Berend, G.: Word sense disambiguation for Hungarian using transformers. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020). p. 3–13. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2020b)
- Berend, G.: Mitigating the knowledge acquisition bottleneck for Hungarian word sense disambiguation using multilingual transformers. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021). pp. 77–89. Szeged (2021)
- Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009. vol. Normalized, pp. 31–40. Tübingen (2009)
- Choe, Y.J., Park, K., Kim, D.: word2word: A collection of bilingual lexicons for 3,564 language pairs. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3036–3045. European Language Resources Association, Marseille, France (May 2020), <https://aclanthology.org/2020.lrec-1.371>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp.

- 8440–8451. Association for Computational Linguistics, Online (Jul 2020), <https://www.aclweb.org/anthology/2020.acl-main.747>
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://www.aclweb.org/anthology/N19-1423>
- Gale, W.A., Church, K.W., Yarowsky, D.: A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 26(5), 415–439 (Dec 1992), <https://doi.org/10.1007/BF00136984>
- Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., Grave, E.: Loss in translation: Learning bilingual word mapping with a retrieval criterion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2979–2984. Association for Computational Linguistics, Brussels, Belgium (Oct–Nov 2018), <https://aclanthology.org/D18-1330>
- Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation. pp. 24–26. SIGDOC '86, ACM, New York, NY, USA (1986), <http://doi.acm.org/10.1145/318723.318728>
- Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., Wolf, T.: Datasets: A community library for natural language processing. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 175–184. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021), <https://aclanthology.org/2021.emnlp-demo.21>
- Loureiro, D., Jorge, A.: Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5682–5691. Association for Computational Linguistics, Florence, Italy (Jul 2019), <https://www.aclweb.org/anthology/P19-1569>
- Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a semantic concordance for sense identification. In: HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8–11, 1994 (1994), <https://www.aclweb.org/anthology/H94-1046>
- Navigli, R.: Word sense disambiguation: A survey. *ACM Comput. Surv.* 41(2) (Feb 2009), <https://doi.org/10.1145/1459352.1459355>
- Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.* 193, 217–250 (Dec 2012), <https://doi.org/10.1016/j.artint.2012.07.001>

- Pasini, T., Raganato, A., Navigli, R.: Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence* 35(15), 13648–13656 (May 2021), <https://ojs.aaai.org/index.php/AAAI/article/view/17609>
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), <https://www.aclweb.org/anthology/N18-1202>
- Reif, E., Yuan, A., Wattenberg, M., Viegas, F.B., Coenen, A., Pearce, A., Kim, B.: Visualizing and measuring the geometry of bert. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems*. vol. 32, pp. 8594–8603. Curran Associates, Inc. (2019)
- Taghipour, K., Ng, H.T.: One million sense-tagged instances for word sense disambiguation and induction. In: *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. pp. 338–344. Association for Computational Linguistics, Beijing, China (Jul 2015), <https://www.aclweb.org/anthology/K15-1037>
- Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pp. 4593–4601. Association for Computational Linguistics, Florence, Italy (Jul 2019), <https://www.aclweb.org/anthology/P19-1452>
- Tiedemann, J.: Parallel data, tools and interfaces in OPUS. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. pp. 2214–2218. European Language Resources Association (ELRA), Istanbul, Turkey (May 2012), [http://www.lrec-conf.org/proceedings/lrec2012/pdf/463\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf)
- Vincze, V., Szarvas, Gy., Almási, A., Szauter, D., Ormándi, R., Farkas, R., Hatvani, Cs., Csirik, J.: Hungarian word-sense disambiguated corpus. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco (May 2008)
- Weaver, W.: Translation. In: Locke, W.N., Boothe, A.D. (szerk.) *Machine Translation of Languages*, pp. 15–23. MIT Press, Cambridge, MA (1949/1955), reprinted from a memorandum written by Weaver in 1949.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing (2019)

# BARTerezzünk!

## Messze, messze, messze a világtól, BART kísérleti modellek magyar nyelvre

Yang Zijian Győző

Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.  
yang.zijian.gyozo@nytud.hu

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport  
1083 Budapest, Práter u. 50/a.  
yang.zijian.gyozo@itk.ppke.hu

**Kivonat** A BART autoregresszív típusú modell, amely elsősorban szövegenerálási feladatokra alkalmas. A kutatásomban különböző BART modelleket tanítottam magyar nyelvre és azokat finomhangoltam különböző szövegenerálási feladatokra. A kísérleteimben BART base és large modelleket tanítottam magyar és angol-magyar nyelvekre. Az előtanított BART modelleket szövegosztályozás, absztraktív szövegösszefoglaló generálás, gépi fordítás és versgenerálás feladatokra finomhangoltam. Az eredmények alapján a BART kevésbé teljesít jól szövegosztályozás feladatára, de absztraktív szövegösszegzés feladatában „state of the art” eredményeket értem el. Érdekességként a kutatásom végén egy Petőfi versgenerátort mutatok be.

**Kulcsszavak:** BART, absztraktív összefoglaló generálás, szentiment analízis, szövegosztályozás, gépi fordítás, szövegenerálás, versgenerálás

## 1. Bevezetés

Kutatásomban különböző BART modellekkel kísérleteztem. A BERT alapú modellek, dekóder hiányában, kevésbé alkalmasak szövegenerálás feladataihoz, mint például a szövegösszefoglaló generálás vagy gépi fordítás. A BART egy enkóder-dekóder architektúrájú modell, ezért alkalmas szövegenerálásra. A kutatásomban különböző BART modelleket tanítottam be magyar és angol-magyar nyelvekre. Kísérleteztem base és large modellekkel egyaránt, majd az előtanított modelleket különböző nyelvtechnológiai feladatokra finomhangoltam. Kutatásom során kísérleteket végeztem a szövegosztályozás területén, továbbá létrehoztam különböző absztraktív összefoglaló generáló modelleket, gépi fordító modelleket és egy versgenerátort. A címben szereplő „Messze, messze, messze a világtól,” sort a versgenerátor generálta a „BARTelezzünk!” folytatásaként.

Modelljeim és szkriptjeim megtalálhatóak a Github<sup>1</sup> és Hugging Face<sup>2</sup> oldalakon.

## 2. Kapcsolódó irodalom

Jelen nyelvtechnológiai feladatok megoldásához az egyik alapvető megközelítés a nyelvi modellek előtanítása, majd azok tovább finomhangolása az adott specifikus feladatra. A konkrét természetes nyelvi feldolgozással kapcsolatos feladatok megoldására a különböző architektúrájú nyelvi modellek teljesítménye különböző. Az utóbbi években a nyelvtechnológia területén a transzformer (Vaswani és mtsai, 2017) architektúrájú modellek dominálnak. A BERT (Devlin és mtsai, 2019; Conneau és mtsai, 2020) típusú modellek bemutatták a maszkolt nyelvi modellezést, amelyek rendkívül magas pontosságot értek el a különböző token és mondatszintű osztályozásos feladatokban. Azonban ezek a modellek kevésbé alkalmasak szöveggenerálásra, mint például a szövegösszefoglalásra. Magyar nyelvre két BERT modell érhető el, a huBERT (Nemeskey Dávid Márk, 2021) és a HILBERT (Feldmann és mtsai, 2021).

A szöveggenerálás feladatára fejlesztették ki az autoregresszív típusú modelleket, mint például a GPT (Radford és Narasimhan, 2018), amelyek „balról jobbra” (left-to-right) modellek, azaz csak a szöveg bal oldalát látják a tanítás során, így rendkívül erősek abban, hogy kitalálják a még hiányzó részeket, a szöveg folytatását.

Az ELMo (Peters és mtsai, 2018) egy bal és egy jobb oldali reprezentációt konkatenál össze, azonban a bennük lévő jegyeket a tanítás során nem hangolja össze, így kevesebb összefüggést tudnak megtanulni.

A MASS (Song és mtsai, 2019) modell rendkívül hasonlít a BART modellhez. A bemeneti szövegből folytonosan kimaszkolnak tokeneket, majd ehhez a szöveghez a kimeneten hozzárendelik a hiányzó tokeneket. Ez a módszer azonban diszkriminatív feladatokra kevésbé hatékony.

Absztraktív összefoglalás területén a BART egyik legnagyobb ellenfele, a PEGASUS (Zhang és mtsai, 2020), amely az előtanítás során a fontosnak vélt egész mondatokat lemaszkolja a bementi dokumentumban.

Gépi fordítás területén a M2M100 (Fan és mtsai, 2020) modell egyetlen modellel képes 100 nyelvről 100 nyelvre fordítani. Tanításakor csak olyan párhuzamos korpuszokat használtak, ahol az angol a forrás vagy a célnyelv.

A Google a transzfer tanulás területén végzett kísérletet a T5 (Raffel és mtsai, 2020) modellel, amely egy nagy korpuszon tanított sztenderd enkóder-dekóder architektúrájú transzformer modell. A különbség más modellek finomhangolásától, hogy a T5 sokféle specifikus feladatot egy modellel tanít be, méghozzá szövegből szöveg (text-to-text) feladatként, legyen az gépi fordítás, vagy osztályozás.

<sup>1</sup> <https://github.com/nytud/neural-models>

<sup>2</sup> <https://huggingface.co/NYTK>

### 3. BART modell

A BART modell egy enkóder-dekóder architektúrán alapuló transformer modell, amelyet a Facebook fejlesztett<sup>3</sup>. Az enkóder kétirányú (Bidirectional), a dekóder autoregresszív (Autoregressive). A BART gyakorlatilag ötvöz egy BERT és egy GPT típusú modellt. A BART enkóder tanítása abban különbözik a BERT-től, hogy amíg a BERT veszteségfüggvényét arra optimalizálták, hogy megtanulja a kimaszkolt tokenek visszaállítását és azt, hogy két mondat egymást követő-e, addig a BART csak olyan feladatokat tanult, amelyek „zajtalanítanak”. A BART enkóder az alábbi feladatok alapján optimalizálja a veszteségfüggvényt: *token maszkolás* (Véletlenszerűen kimaszkolt tokenek visszaállítása), *textittoken törlés* (Véletlenszerűen kitörölt tokenek helyének meghatározása), *Szöveg kitöltés* (A SpanBERT (Joshi és mtai, 2020) módszerén alapszik, azonban itt Poisson eloszlás alapján számolják ki a hosszakat, majd a hossz alapján kerülnek kimaszkolásra szövegrészek. Ezzel azt tanulja meg a modell, hogy hány tokent maszkoltak ki. Az eredeti BERT-el ellentétben, nem önálló szövegelemeket maszkol ki, hanem egész szavakból álló folytonos szövegrészeket. Ezzel azt éri el, hogy a modell a szövegkörnyezet alapján nagyobb összefüggő szövegrészeket tud megtanulni.), *mondat permutáció* (Pont írásjel alapján mondatokra bontja a szöveget, majd véletlenszerűen megkeveri. A modell ezzel megtanulja, hogy milyen sorrendben voltak eredetileg a mondatok) és *dokumentum rotáció* (Véletlenszerűen kiválasztanak egy tokent, majd úgy forgatják a szöveget, hogy ez a kiválasztott token legyen az első token. A modell ezzel azt tanulja meg, hogy melyik tokenek lehetnek dokumentumkezdők.). A BART az előtanítás során egy dokumentumot ellát zajokkal, majd áteresztve az enkóder-dekóder architektúrán, a dekóder kiemenetére és az eredeti dokumentumra számolja ki a veszteséget. A BART egy nyelvi modellnek felel meg.

### 4. Előtanítás

A Facebook nem tette közzé az előtanítás szkriptjét, de a Hugging Face könyvtárai<sup>4</sup> tartalmazzak előtanítási kódokat. A BART előtanításához a BartForCausalLM függvényt használtam. A BartForCausalLM a BART modell dekóder önálló része, melynek a tetején egy nyelvmódel réteg (gyakorlatilag egy softmax, ami segít a következő token kiválasztásában) található. Ez alkalmas a következő szó prediktálására (causal language modeling). A modell tovább finomhangolható. A kutatásom során öt különböző BART modellt tanítottam elő:

- **BART-base-512**: Egynyelvű magyar BART base modell, 512 bemeneti hosszal.
- **BART-base-1024**: Egynyelvű magyar BART base modell, 1024 bemeneti hosszal.

<sup>3</sup> <https://github.com/pytorch/fairseq/tree/master/examples/bart>

<sup>4</sup> [https://huggingface.co/transformers/model\\_doc/bart.html](https://huggingface.co/transformers/model_doc/bart.html)

- **BART-large**: Egynyelvű magyar BART large modell. Erőforrás hiányában, csak részleges kiértékelés történt ezzel a modellel.
- **BART-base-enhu**: Angol-magyar kétnyelvű BART base modell.
- **BART-large-enhu**: Angol-magyar kétnyelvű BART large modell. Erőforrás hiányában, csak részleges kiértékelés történt ezzel a modellel.

#### 4.1. Felhasznált korpuszok

Az egynyelvű BART modellek tanításához a Webcorpus 2.0-t (Nemeskey, 2020) használtam. Az eredeti BART kutatás (Lewis és mtsai, 2020) alapján a korpuszból bekezdéseket nyertem ki, amelyek legalább egy darab pont írásjellel rendelkeztek.

Az angol-magyar kétnyelvű BART modell tanításához az angol WikiText-103 (Merity és mtsai, 2017) és a Webcorpus 2.0 magyar Wikipédia részét használtam. Hasonlóan az egynyelvű korpuszhoz, azokat a bekezdéseket hagytam meg, amelyek legalább egy darab pont írásjellel rendelkeztek.

Mind a három korpusz alapról tokenizálva volt. Az így létrejött korpuszok tulajdonságai az 1. táblázatban láthatóak.

	szegmens	token	type	bekezdés mondatszám (medián)	bekezdés tokenek száma (medián)
Webcopus 2.0	100.255.504	9.095.424.717	57.562.212	3	60
Angol WikiText-103	707.391	96.534.563	596.820	5	125
Magyar Wikipédia	1.098.156	90.349.849	3.137.980	4	69

1. táblázat. Előtanításhoz használt korpuszok jellemzői.

#### 4.2. Modellek tulajdonságai és tanítása

A 2. táblázatban láthatóak a főbb különbségek a tanított modellek hiperparamétereinek között. Főbb különbségek az enkóderek és dekóderek rétegeinek számában (Rétegek #), rejtett rétegeinek méretében (Rejtett), a figyelmi fejeinek számában (Fejek #), az előreccsatolt köztes rétegeinek méretében (FFN dim), valamint a bemeneti szöveg hosszában (Bemenet) és a szótár méretében (Szótár) mutatkoznak meg. A szótárak esetében a BART-base-enhu és a BART-large-enhu angol-magyar kétnyelvű szótárral, a többi modell magyar egynyelvű szótárral rendelkezik.

A 3. táblázatban láthatóak a tanítás szempontjából fontosabb tulajdonságok. A tanulási ráta mindegyik modell esetén  $2e-8$  volt. Az egyik kiemelendő információ a mentési pont. A modellek tanításai során egyszer sem konvergált a modell, de ez még nem jelenti azt, hogy nem tanult meg semmit. Ezért különböző mentési pontoknál kivettem egy-egy modellt és finomhangolással (osztályozás



	Réteg #	Rejtett	Fejek #	FFN dim	Bemenet	Szótár
BART-base-512	6	768	12	3072	512	30.000
BART-base-1024	6	768	12	3072	1024	30.000
BART-large	12	1024	16	4096	1024	30.000
BART-base-enhu	6	768	12	3072	512	40.000
BART-large-enhu	12	1024	16	4096	1024	40.000

2. táblázat. Modellek tulajdonságai.

és szövegkivonatolás) teszteltem, hogy a modelljeim finomhangolhatóak-e. A 3. táblázatban láthatóak azok a mentési pontok, valamint a hozzájuk tartozó veszteségi értékek (Loss), amelyek végül kiválasztásra kerültek, és alapot képeznek a jelen kutatás további részeihez. A mentési pontok variabilitása azzal magyarázható, hogy különbözőek a batch méretek, a hardver hátterek és az a tény, hogy nem mindig volt elegendő erőforrás a továbbtanításra.

	Gép (4 db)	Batch (per GPU)	Mentési pont (lépés)	Loss
BART-base-512	Tesla V100S - 32GB	50	150.000	1,14
BART-base-1024	Tesla V100S - 32GB	8	290.000	2,29
BART-large	Tesla V100S - 32GB	8	220.000	2,22
BART-base-enhu	GeForce GTX 1080 - 12GB	12	170.000	1,44
BART-large-enhu	Tesla V100S - 32GB	7	500.000	2,75

3. táblázat. Előtanítás tulajdonságai.

## 5. Finomhangolás

Az előtanított modelleket 4 különböző feladatra finomhangoltam: Mondatszintű szentiment analízis szövegosztályozás (SZENT), Absztraktív összefoglalás, szövegkivonatolás (SZUM), Gépi fordítás (GF) és Szöveggenerálás: Petőfi versgenerálás (VERS).

### 5.1. Felhasznált korpuszok

A finomhangoláshoz a különböző feladatokra az alábbi korpuszokat használtam fel, melynek tulajdonságai A 4. táblázatban láthatóak:

- **HI**: HVG korpusz + index.hu korpusz, amelyből a HVG korpusz online cikkeket tartalmaz 2012–2020 időszakból, az index.hu korpusz online cikkeket tartalmaz 1999–2020 közötti időszakból.
- **NOL**: Népszabadság online korpusz; a nol.hu online cikkeket tartalmazza a 1999–2016 közötti időszakból.

- **MARCELL** (Váradi és mtsai, 2020): Jogi szövegek (dok) és a hozzájuk tartozó egy soros leírások 1991–2019 közötti időszakból.
- **MTS**: Magyar Twitter Szentiment Korpusz<sup>5</sup>, a PrecognoX Kft.<sup>6</sup> jóvoltából. A korpusz 5 osztályos (MTS5), ahol 1 a legnegatívabb és 5 a legpozitívabb. Készítettem belőle egy 3 osztályos változatot (MTS3), ahol a 1-es és 2-es értékeket negatívként jelöltem, a 3-as értéket semlegesnek, valamint a 4-es és 5-ös értékeket pozitívként. Végül készítettem egy bináris változatot is (MTS2), ahol a 3-as értékű szegmenseket kihagytam, mert nem lehet eldönteni róluk egyértelműen, hogy pozitív vagy negatív.
- **SST**: Stanford Sentiment Treebank (Socher és mtsai, 2013), angol nyelvű szentiment analízis korpusz. Két változata van, a bináris osztályú SST-2 és az 1-5 likert skálájú SST-5.
- **OPUS**: OPUS (Tiedemann, 2012) korpuszból vett angol-magyar párhuzamos alkorpuszok gépi fordításhoz. Felhasznált alkorpuszok: ParaCrawl, OpenSubtitles, Tatoeba, DGT, WikiMatrix, EUbookshop, PHP manual, TED2020, KEDoc, KDE4.
- **PETŐFI**: Petőfi Sándor összes költeményei mű letöltve a Magyar Elektronikus Könyvtár oldaláról<sup>7</sup>.

	Feladat	Szegmens	Token #	Type #	Átlag token #
HI	SZUM	559.162	147.099.485 (cikk)	2.949.173 (cikk)	263,07 (cikk)
			16.699.600 (lead)	749.586 (lead)	29,87 (lead)
NOL	SZUM	397.343	153.003.164 (cikk)	2.482.398 (cikk)	384,52 (cikk)
			15.786.166 (lead)	623.445 (lead)	39,71 (lead)
MARCELL	SZUM	24.747	27.834.358 (dok)	444.352 (dok)	1124,82 (dok)
			277.732 (leírás)	29.189 (leírás)	11,59 (leírás)
MTS2	SZENT	2.737	42.797	13.713	15,62
MTS3, MTS5	SZENT	4.000	59.997	18.423	14,99
OPUS	GF	56.837.602	613.206.646 (en)	2.691.229 (en)	10,79 (en)
			507.702.362 (hu)	6.886.205 (hu)	8,93 (hu)
PETŐFI	VERS	<sup>854</sup> (költemény)	151.486	50.029	-

4. táblázat. Finomhangoláshoz használt korpuszok tulajdonságai.

## 5.2. Finomhangolás kísérletek

**A mondatszintű szentiment analízis szövegosztályozás** feladatához, a magyar és az angol-magyar modellek tanításához, a Magyar Twitter Szentiment Korpuszt használtam, míg az angol-magyar kísérletekhez az SST korpuszokat.

<sup>5</sup> <http://opendata.hu/dataset/hungarian-twitter-sentiment-corpus>

<sup>6</sup> <https://www.precognoX.com>

<sup>7</sup> <https://mek.oszk.hu/01000/01006/>

Az angol-magyar modell esetében mindamellett, hogy betanítottam a modelljeimet az eredeti SST korpuszokra és kiértékeltem, *zeroshot* kísérletet is végeztem, vagyis magyar szövegen való tanítás nélkül végeztem osztályozást az MTS tesztanyagban. Továbbá végeztem *transzfer* kísérletet (tf) is, ami az esetben azt jelentette, hogy az angol SST finomhangolás után, tovább finomhangoltam az angol-magyar modelletem a magyar MTS korpuszon. Mindegyik tanítást maximum 128 bemeneti szöveghosszal,  $2e-5$  tanulási rátán (learning rate), 4-es batch/GPU (4 db GeForce GTX 1080 - 12GB) méreten és 15 epoch számon tanítottam. A finomhangoláshoz a Huggingface Transformers githubján található példakódot<sup>8</sup> használtam fel.

**Az absztraktív összefoglalás, szövegvonatalás** feladatához a HI, a NOL és a MARCELL korpuszokat használtam. Összehasonlíthatóság végett, a NOL korpuszon betanítottam egy BERT alapú (huBERT modellel) absztraktív modellt a PreSumm eszközzel (Liu és Lapata, 2019), ugyanazokkal a beállításokkal, mint amit Yang és mtsai (2021) a kutatásaikban használtak. A BART base modelleket 512 / 1024 maximum bemeneti és 256 maximum kimeneti szöveghossz, 8-as batch/GPU (4 db GeForce GTX 1080 - 12GB) méret, 80 epoch,  $2e-5$  tanulási ráta, 15 ezer warmup lépés és fp16 beállítási hiperparaméterekkel tanítottam. Az egyetlen magyar large modelletem sikerült 4 db Tesla V100S - 32GB GPU-n finomhangolni. Azonban korlátolt erőforrás miatt, csak 20 epochon.

A sima absztraktív modellek tanítása mellett végeztem **transzfer** kísérleteket is, ami az esetben azt jelentette, hogy az angol-magyar base modelletem betanítottam az angol CNN/Daily Mail korpuszon (40 epoch), majd a betanított modellt továbbtanítottam a HI és a NOL korpuszon. Erőforrás hiányában nem tudtam az angol-magyar large modelletem finomhangolni. A finomhangoláshoz a Huggingface Transformers githubján található példakódot<sup>9</sup> használtam fel.

**A gépi fordítás** feladatához a BART-base-enhu modellel tanítottam angol-magyar (enhu) és magyar-angol (huen) gépi fordító modelleket. A tanításhoz az OPUS korpuszban található angol-magyar alkorpuszokat használtam. Angol-magyar nyelvre egy 512 maximum bemeneti és 512 maximum kimeneti szöveghosszú modellt és egy 128 maximum bemeneti és 128 maximum kimeneti szöveghosszú modellt tanítottam. Erőforrás és idő hiányában, magyar-angol nyelvre csak 128 maximum bemeneti és 128 maximum kimeneti szöveghosszú modellt tanítottam. Az 512 szöveghosszú modellt 4-es batch/GPU (4 db GeForce GTX 1080 - 12GB) mérettel és 1 epoch számmal, míg a 128 szöveghosszú modelleket 26-os batch/GPU (4 db GeForce GTX 1080 - 12GB) mérettel és 2 epoch számmal tanítottam. További fontosabb hiperparaméterek: 15 ezer warmup lépés, fp16,  $5e-5$  tanulási ráta. A finomhangoláshoz a Huggingface Transformers githubján található példakódot<sup>10</sup> használtam fel.

<sup>8</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification>

<sup>9</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization>

<sup>10</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/translation>

**A szövegenerálás** feladatához a Petőfi Sándor összes költeménye című kötetet használtam fel. A tanítóanyag létrehozásához kitöröltem a címekeket, keltezéseket (hely és dátum) és a tartalomjegyzéket. Két vers közé beraktam egy `<s>` címkét, ami jelzi a versek végét. Majd 3-as (sor) ablakot végigcsúsztatva a verseken generáltam forrás- és célnyelvi szövegeket. A forrásnyelvi szöveg lehetett 1 sor, 2 sor (egymást követő) és 3 sor (egymást követő), a kimeneti szöveg az 1, 2 vagy 3 sornak a következő sora. A BART modell jellege miatt, a forrásnyelvi szövegből véletlenszerűen kimaszkoltam 0-25% szót. Ily módon 99.453 sor tanítóanyag és 3000 sor validálási anyag keletkezett. A feladatot absztraktív generálás feladatként értelmeztem, ami egy szövegből-szöveg (seq2seq) generálási feladat. A forrásnyelvi szöveg egy hasonló méretű vagy hosszabb szöveg, mint a célnyelvi szöveg, ami a folytatása a forrásnyelvi szövegnek. Ezért a tanításhoz az absztraktív generáláshoz használt kódot használtam (ugyanazokkal a hiperparaméterekkel). Mivel a vers sorai nagyon rövidek, ezért maximum 128 bemeneti és maximum 128 kimeneti szöveghosszt használtam. A tanításhoz a BART-base-1024 modellt használtam fel. A vers generálásakor kézzel kell megadni az első sort, ez alapján generál a modell egy sor folytatást, majd az általunk megadott és az általa adott folytatásra generál egy újabb sort, a így keletkezett 3 sorra generál egy 4-dik sort. Ezután 3-as (sor) ablakkal tovább csúsztatva generálja a következő sorokat. A végső demóban egy minimális rímkényszert is próbáltam alkalmazni. A modell öt lehetséges folytatást generál, amelyeket sorrendbe raktam az alapján, hogy mennyire rímel a kettővel előtte lévő sorra. A rímet három magánhangzó mélységig vizsgáltam.

## 6. Eredmények

Az 5. táblázatban láthatóak a mondatszintű szentiment analízis osztályozás kísérlet eredményei. Egyértelműen látszik, hogy a BART modelljeim szignifikánsan alulmaradnak a huBERT-hez képest. Ez nem meglepő, hiszen az autoregresszív modelleknek nem erőssége az osztályozás, de még így is értékelhető minőségben lehet betanítani osztályozásos feladatokra. A legjobb eredményt a BART-base-512 adta, ez annak tudható be, hogy ezt a modellt sikerült a legnagyobb batch méret mellett tanítani. A kiértékeléshez a pontosság (accuracy) metrikát használtam. Mindegyik mérésnél 15 epochig tanítottam, azonban az eredmények táblázatba csak a legjobb eredmények kerültek be. Jellemzően a 3-5 epoch szám között érték el a legmagasabb pontosságot. Angol-magyar modellek esetén az SST korpuszból nem készítettem 1-3 likert skálájú alkorpust, ezért üresek ezek a mezők a zeroshot és a transzfer eredményeknél. A BART-base-enhu modell esetében az angol-magyar modellt finomhangoltam az MTS korpuszon A zeroshot esetében az angol-magyar modellt az SST korpuszon finomhangoltam, majd egyből kiértékeltem az MTS korpuszon, végül a transzfer esetében az SST korpuszon finomhangolt modellt továbbfinomhangoltam az MTS korpuszon. Az eredményekből az látszik, hogy a többnyelvű modellek gyengébben teljesítenek, mint az egynyelvű modellek, ami szintén várható volt, hiszen ezek csak Wikipédia anyagon tanultak és egyszerre kellett angolul és magyarul is tanulniuk.

A zeroshot eredmények meglehetősen gyengék, a transzfer kísérletek csak egy kicsivel tudták javítani a modell minőségét. Érdekes megfigyelés, hogy a large modellek nem teljesítenek jobban a base modelleknél. Továbbá az eredmények közé beillesztettem még az angol-magyar modelleknek az SST korpuszokon mért teljesítményét is.

	MTS2	MTS3	MTS5
huBERT	85.92	72.18	68.50
<b>BART-base-512</b>	<b>79,25</b>	<b>61,40</b>	<b>58,75</b>
BART-base-1024	76,66	56,89	57,75
BART-large-1024	76,29	54.88	58.75
BART-base-enhu	74,44	60,15	56,75
BART-base-enhu (zeroshot)	42,96	-	28,75
BART-base-enhu (transzfer)	74,81	-	57,25
BART-large-enhu	74,07	59,14	56,00
BART-large-enhu (zeroshot)	44,81	-	23,50
BART-large-enhu (transzfer)	72,59	-	56,74
	SST2	-	SST5
BART-base-enhu	79,01	-	36,72
BART-large-enhu	80,27	-	36,36

5. táblázat. Mondatszintű osztályozás eredménye.

A 6. táblázatban láthatóak az absztraktív összefoglalás kísérlet eredményei. A modell legnagyobb erőssége ebben a feladatban mutatkozik meg. Szignifikánsan jobb eredményt értem el a BART alapú modellekkel, mint a BERT alapú megoldással. Yang és mtsai (2021) munkájukban a fedés eredményeket publikálták. Azonban csak a PreSumm eszközre jellemző, hogy több, hosszabb szöveget generál kimenetnek. Összehasonlítva a HI korpuszon:

- Eredeti lead méretek: Átlag: 26,42, Medián: 24.
- PreSumm összefoglalók méretei: Átlag: 104,61, Medián: 105.
- BART-base-512 összefoglalók méretei: Átlag: 28, Medián: 24.

Az összehasonlításból észrevehető, hogy a PreSumm rendkívül hosszú összefoglalókat generál, ezért nem meglepőek a magas fedés mértékek, azonban így a pontosság mértékek drasztikusan csökkennek (látszik a 6. táblázat PreSumm F-mértékeiből). Az összehasonlításból látszik, hogy a BART törekszik a hossz megtanulására is, és közel olyan hosszúságú összefoglalókat generál mint az eredeti leadek (annak ellenére, hogy maximum 128 kimeneti hosszra van beállítva).

Azonban a MARCELL korpuszon már az esetek nagy részében aluteljesítenek a BART modellek a PreSummhoz képest. Ez annak tulajdonítható, hogy a rövidsége való törekvése most a hátrányára fordult. Egyedül a transzfer tanítással készült modell tudott magasabb eredményt elérni. A méretek összehasonlítva a MARCELL korpuszon:

- Eredeti lead méretek: Átlag: 11.59, Medián: 9.

- PreSumm összefoglalók méretei: Átlag: 11,466, Medián: 9.
- BART-base-512 összefoglalók méretei: Átlag: 9,97, Medián: 8.

Továbbá azt is figyelembe kell venni, hogy a PreSumm kísérletben a generált szövegből csak az első mondatot vették figyelembe, azonban a rendszer alapból több mondatot is generált.

Érdekes eredmény, hogy annak ellenére, hogy a MARCELL korpuszban rendkívül hosszú a bemeneti szöveg, a hosszú bemeneti szövegű BART-base-1024 teljesített a leggyengébben. Ebből arra tudok következtetni, hogy az egysoros leíráshoz a releváns információk inkább a bemeneti szöveg elején találhatóak, ezért a rendkívül hosszú szöveg csak megzavarja a generálást. Azonban az, hogy a transzfer tanítás ilyen mértékben tudta javítani a teljesítményt, azt jelentheti, hogy az angol tudásból olyan információt tudott kinyerni, ami segített neki a finomhangolásban.

A kiértékeléshez a ROUGE (Lin, 2004) metrikát használtam. A 6. táblázatban az F-mértékek láthatóak a következő formátumban: ROUGE-1/ROUGE-2/ROUGE-L.

Összehasonlíthatóság végett, a 6. táblázat végére beillesztettem az eredeti BART modell eredményét (a CNN/Daily Mail korpuszon)<sup>11</sup>. A kutatásom célja nem az angol eredmények felülmúlása volt, ezért csak 40 epoch számon tanítottam. Figyelembe véve, hogy kevesebb epoch szám mellett és csak Wikipédia szövegeken tanult elő a modell, mindössze 4% körüli értékkel marad csak le az eredeti BART modell eredményétől. A magyar modelleket nézve, magasabb epoch számon még jobb eredményt tudtam volna elérni. A tapasztalat az epoch számot illetően az, hogy a nagyobb epoch szám az összefoglaló generálás esetében nem eredményezett túltanulást.

A large modellelkel való kísérletek esetében, csak 40 epochig tanultak, kicsi batch méreten, így ők teljesítették a leggyengébben. Továbbá erőforrás hiányában a NOL korpuszon nem sikerült finomhangolni.

	HI	NOL	MARCELL
PreSumm (huBERT)	22,42/10,24/18,72	26,34/10,90/22,01	75,85/68,35/74,61
BART-base-512	30,18/13,86/22,92	46,48/32,40/39,45	71,25/62,79/69,75
BART-base-1024	<b>31,86/14,59/23,79</b>	<b>47,01/32,91/39,97</b>	71,01/62,58/69,42
BART-large	30,12/13,07/22,72	-	70,24/60,69/68,53
BART-base-enhu	31,36/14,34/23,48	42,71/27,59/35,38	71,47/63,04/69,93
BART-base-enhu-tf	31,76/14,47/23,47	45,05/30,46/37,64	77,06/70,64/75,96
	CNN/Daily Mail		
BART-base-enhu	40,07/17,61/27,35		
BART eredeti	44,16/21,28/40,90		

6. táblázat. Absztraktív összefoglaló generálás F-mérték eredmények.

<sup>11</sup> <https://paperswithcode.com/sota/abstractive-text-summarization-on-cnn-daily>

A 9. táblázatban látható egy példa arra, hogy a különböző modellek milyen összefoglalókat generáltak. A példa önmagában is összetett, kétféle készülékről is ír, ezért nehezen állapítható meg, hogy melyik is a fontosabb információ. Igyekeztem olyan példát mutatni, ami inkább a modellek határait, hátrányait mutatja. A példában egyértelműen látszik a ROUGE metrika egyik hátránya, miszerint az eredeti lead szövege meglehetősen szűkszavú, figyelemfelkeltő, de semmi hasznos információt nem szolgáltat, remélve, hogy megmozgatva az olvasó kíváncsiságát, bevonzza őt. Ez azonban torzít a ROUGE értékeken, hiszen a metrikával azt mérjük, hogy mennyire hasonlít a gép által generált szöveg a leadhez. Továbbá szembetűnő a PreSumm által generált szöveg hosszúsága. Ha a tartalmat nézzük a nagy része hű az eredeti cikkhez. Megfigyelhető még a BART-base-512 helyesírási hibája, ez más példákban is megjelenik, valamint a BART-large modell erősebb „hallucinációja”, amelyek más példákban is megmutatkoznak. Ezek a számokban is észlelhetőek, hiszen ezek teljesítettek a leggyengébben. A generált mondatok nyelvtanilag helyesek, tartalmilag viszont csak részlegesen felelnek meg az eredeti cikkeknek.

Az egynyelvű BART base modelljeim (HI<sup>12,13</sup> és NOL<sup>14,15</sup>) elérhetőek a Hugging Face oldalon.

A 7. táblázatban láthatóak a gépi fordítás kísérlet eredményei. Referenciaként a Google fordítót<sup>16</sup> választottam, két okból. Első ok, hogy érdekelt a Google fordító mai állapota, és hogy vajon jobbak-e nála az általam tanított modellek. A másik ok, hogy egy szabadon elérhető neurális alapokon működő gépi fordítót kerestem, mivel nem állt szándékomban külön másik neurális gépi fordító rendszert tanítani. A meglévő szabadon elérhető rendszerek közül az egyik legnépszerűbb rendszer mellett döntöttem. Az eredmények azt mutatják, hogy sikerült mindegyik esetben szignifikánsan felülmúlni a Google fordítót. Kiértékeléshez a BLEU (Papineni és mtsai, 2002) és a 3-gram chrF (Popović, 2015) metrikákat használtam. Érdekes, hogy az 512 bemeneti hosszal rendelkező modell 1 epoch alatt hasonló vagy jobb eredményt ért el, mint a 128 bemeneti hosszal rendelkező modell 2 epoch alatt.

Az 512 bemeneti hosszal rendelkező BART base fordító modelljeim<sup>17,18</sup> elérhetőek a Hugging Face oldalon.

Végül, de nem utolsó sorban a 8. táblázatban látható a szöveggenerálás kísérlet eredménye. A kvantitatív kiértékeléshez az absztraktív összefoglalónál használt ROUGE metrikát használtam. Azonban egy ilyen jellegű feladatnál az automatikus kiértékelési metrikák kevésbé relevánsak, sőt az emberi kiértékelés sem egyértelmű. Ezért inkább egy generált verset tettem be, ahol az első sort magam

<sup>12</sup> <https://huggingface.co/NYTK/summarization-hi-bart-hungarian>

<sup>13</sup> <https://huggingface.co/NYTK/summarization-hi-bart-base-1024-hungarian>

<sup>14</sup> <https://huggingface.co/NYTK/summarization-nol-bart-hungarian>

<sup>15</sup> <https://huggingface.co/NYTK/summarization-nol-bart-base-1024-hungarian>

<sup>16</sup> <https://translate.google.hu>

<sup>17</sup> <https://huggingface.co/NYTK/translation-bart-en-hu>

<sup>18</sup> <https://huggingface.co/NYTK/translation-bart-hu-en>

	BLEU	chrF-3
Google en-hu	25,30	54,08
BART-base-enhu (512, 1 epoch)	34,38	58,88
BART-base-enhu (128, 2 epoch)	33,59	58,23
Google hu-en	34,48	59,59
BART-base-huen (512, 1 epoch)	38,03	61,37
BART-base-huen (128, 2 epoch)	38,63	61,58

7. táblázat. Gépi fordítás eredményei.

adtam meg manuálisan. Mivel Petőfi Sándor költeményein tanult a modell, a generált szöveg „erősen Petőfi Sándor stílusú”.

---

Szegeden, január végén,  
 Lopott, koldult és magamért,  
 Lelkem reája...  
 Szeretlek téged, kedvesem,  
 Hol a boldogság mostanában?  
 Barátságos meleg szobába.  
 Sötétség volt, mint a hold,  
 S mint a hold, a csillag az éjben,  
 16.51/12.93/16.53

---

8. táblázat. Versgenerálás eredménye.

## 7. Összegzés

Kutatásomban különböző BART modelleket tanítottam magyar nyelvre, majd különböző nyelvtechnológiai feladatokra továbbtanítottam őket. A kísérletem során magyar nyelvű és magyar-angol nyelvű BART base és large modelleket tanítottam elő. Majd ezeket a előtanított modelleket szövegosztályozás, absztraktív generálás, gépi fordítás és szöveggenerálás feladataira finomhangoltam. Az eredmények azt mutatták, hogy a BART, mint autoregresszív modell elsősorban szöveggenerálás feladataira teljesít jól, azon belül is absztraktív szövegösszefoglaló feladatában. Ezen a területen magyar nyelvre „state of the art” eredményeket értem el.

## Hivatkozások

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020)



- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Çelebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., Joulin, A.: Beyond english-centric multilingual machine translation. ArXiv abs/2010.11125 (2020)
- Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: Improving Pre-training by Representing and Predicting Spans. Transactions of the Association for Computational Linguistics 8, 64–77 (01 2020)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
- Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (2019)
- Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. In: 5th International Conference on Learning Representations. Palais des Congrès Neptune, Toulon, France (2017)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)
- Nemeskey Dávid Márk: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002), <https://aclanthology.org/P02-1040>

- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)
- Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation. pp. 392–395. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), <https://aclanthology.org/W15-3049>
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020), <http://jmlr.org/papers/v21/20-074.html>
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), <https://aclanthology.org/D13-1170>
- Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mass: Masked sequence to sequence pre-training for language generation. In: International Conference on Machine Learning. pp. 5926–5936 (2019)
- Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)
- Váradi, T., Koeva, S., Yamalov, M., Tadić, M., Sass, B., Nitoń, B., Ogrodniczuk, M., Pezik, P., Barbu Mititelu, V., Ion, R., Irimia, E., Mitrofan, M., Păiș, V., Tufiș, D., Garabík, R., Krek, S., Repar, A., Rihtar, M., Brank, J.: The MARCELL legislative corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3761–3768. European Language Resources Association, Marseille, France (May 2020)
- Yang, Z.G., Agócs, Á., Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (2021)
- Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: Thirty-seventh International Conference on Machine Learning (2020)

## A. függelék: Példa az absztraktív modellek összegzéseiről

---

**Eredeti cikk szövege:**

Egyre többet hallani arról , hogy az okostelefon-gyártók olyan ujjlenyomat-olvasókkal kísérleteznek , amelyek be lennének építve a kijelzőbe . Korábban sokan úgy gondolták , hogy a Galaxy S8-aknál vagy legalábbis a Galaxy Note 8-nál jelentheti be az újdonságot a Samsung , de úgy tűnik , egyelőre még nem tudta ezt megoldani . A KGI neves elemzője , Ming-Chi Kuo szerint ez a funkció legkorábban a Galaxy Note 9-ben fog megjelenni , azaz még legalább egy évet várunk kell rá . A szakember egyébként jól ismert pontos jövődőléseiről , bár elsősorban az Apple és nem a Samsung termékeire összpontosít . A Samsung már a Galaxy S8-nál üveg alá tette az ujjlenyomat-olvasót ( aminek az elhelyezése amúgy nem váltott ki egyöntetű lelkesedést ) , de a kijelző alá tenni azt , egyelőre még senkinek sem sikerült . Pedig ha ez megvalósulna , az sok lehetőséget nyithatna meg a gyártók előtt . Kuo szerint a Samsung már eldöntötte , hogy nem változtat az ujjlenyomat-olvasón a Galaxy Note 8-nál , viszont a Note 9-nél már valószínűleg az új megoldást mutatja be . Kuo azt is megszelleztette , hogy a dél-koreai óriás lecseréli biometrikus szkener partnerét , a Synapticsot egy Egis nevű cégre .

---

**Eredeti lead szövege:**

Egy rég várt jellemző debütálását valószínűsítik a jövő évben megjelenő Galaxy Note 9-ben .

---

**PreSumm (huBERT) által generált szöveg:**

Ming-Chi Kuo , a KGI neves elemzője szerint a Samsung már a Galaxy Note 8-nál is bevezetheti az ujjlenyomat-olvasót , de egyelőre még nem tudni , mikor . A Samsung már eldöntötte , hogy nem változtat az ujjlenyomat-olvasón , viszont lecseréli a Synapticsot egy Egis nevű cégre . A szakember szerint ez sok lehetőséget nyithatna meg a gyártók előtt , ha megvalósulna az új funkció , akkor sok új lehetőséget nyitna meg a gyártó előtt . Ha ez megvalósulna , a Samsung lecseréli a Synapticsot egy Egisre .

---

**BART-base-512 által generált szöveg:**

A Ming-Chi Kuo szerint legkorábban egy év múlva jelenik meg a Samsung Galaxy Note 8 .

---

**BART-base-1024 által generált szöveg:**

A Samsung egyik neves elemzője , Ming-Chi Kuo szerint legalább egy évet kell várni rá , hogy a Galaxy S8-nál is megjelenhessen ujjlenyomat-olvasó .

---

**BART-large által generált szöveg:**

Ha igazak a híresztelések , jövőre egy új Samsung-csúcstelefon kerülhet a boltok polcaira .

---

**BART-enhu által generált szöveg:**

Az ismert Ming-Chi Kuo szerint a Samsung már eldöntötte , hogy nem változtat az ujjlenyomat-olvasón .

---

**BART-enhu-transzfer által generált szöveg:**

Ha hihetünk a pletykáknak , legkorábban a Galaxy Note 9-ben jelenik majd meg a Samsung új , ujjlenyomat-olvasós funkciója .

---

9. táblázat. Egy példa a modellek összegzéseiről.



# Okosabb vagy, mint egy XXXXXXXX? – Egy nyelvi játéktól a nyelvmodellek összehasonlításáig

Indig Balázs<sup>1,2</sup>, Lévai Dániel<sup>1,2</sup>

<sup>1</sup>Eötvös Loránd Tudományegyetem Bölcsészettudományi Kar  
TI Digitális Bölcsészet Tanszék

<sup>2</sup>Digitális Örökség Nemzeti Laboratórium  
{VEZETÉKNÉV.KERESZTNÉV}@btk.elte.hu

**Kivonat** A nyelvmodellek fejlődése az utóbbi években a nyelvtechnológiát sosem látott magasságokba repítette. A gépi algoritmusok egyre pontosabban tudják megoldani az összes elképzelhető nyelvi feladatot, holott a működésük igen szövevényes és kérdéses, hogy mennyire hasonlít az emberi gondolkodásra.

Cikkünkben többféle nyelvmodellet hasonlítunk össze. A feladatot a MorphoLogic Kft. 2000-ben bemutatott nyelvi játéka jelenti, melyben egy adott szó különböző környezetei, konkordanciája alapján kell kitalálni a kérdéses szót. Kitérünk a kísérleti rendszerhez használt adatok előfeldolgozására és a gépi modellek teljesítményére különböző beállítások mellett. Így a cikkben bemutatott keretrendszer a későbbiekben a rekreációs célok mellett pszicholingvisztikai vizsgálatokra is felhasználható a mentális nyelvi reprezentációk kutatásában.

**Kulcsszavak:** nyelvmodellek, BERT, KenLM, Word2Vec

## 1. Bevezetés

A nyelvtechnológia egyik alapfeladata a természetes nyelvek modellezése számítógép segítségével. A nyelv modellezésének információelméleti megalapozásából következő egyszerű n-gram modellektől (Shannon, 1948) eljutva az utóbbi évtizedet meghatározó szóbeágyazásra (Mikolov és mtsai, 2013b) és magasdimenzós vektorterekre épülő kontextuális nyelvmodellekig (Devlin és mtsai, 2019) a számítógépes nyelvmodellek a tudományterületet alapjaiban szabták át. A minőségük napjainkra olyan szintet ért el, hogy a modern GPT-3-as modell által előállított szöveg már nem különböztethető meg az ember által előállítottól (Elkins és Chun, 2020).

Cikkünkben a nyelvmodellek ipar által egy kevésbé fontosnak tartott jellemzőjét vizsgáljuk. A célunk, hogy bemutassunk többféle nyelvmodellet egy nyelvi játékon, amely az angolul *Cloze-test*nek nevezett játék, és létrehozunk egy nyílt hozzáféréstű és -forráskódú platformot, amely kísérleti keretrendszerként szolgálhat a későbbiekben. A létrehozott platformon a korpuszból vett nyelvi mintákra adott emberi és gépi reakció egyszerű vizsgálatára nyílik lehetőség, mely így számos későbbi pszicholingvisztikai kutatás alapjául szolgálhat.

## 2. Kapcsolódó irodalom

A szavak jelentését Harris (1954) a következőképpen definiálta: „If A and B have some environments in common and some not (e.g. occultist and lawyer) we say that they have different meanings, the amount of meaning difference corresponding roughly to the amount of difference in their environments.” Ez volt a disztribúciós szemantika kezdete. Később Firth (1957) kimondta az azóta szállóigévé vált mondatot: „You shall know a word by the company it keeps.”. Ekkor még nem volt szó arról, hogy az említett kontextusok is karakterizálva legyenek.



Új játék Szabad a gazda

Tipp Kérek még egy mondatot

...osztódott, és mindegyik részen xxxxxxx csillagzat szerkesztett ősz...  
 ...jesztett két szárnya csuklóján xxxxxxx /Deneb )4)1; eltátott ajkaira, ...  
 ...nagy böltnak négy szegeletében xxxxxxx edény van elásva, tele pénzze...  
 ...zer is találtak nyarantszak xxxxxxx kasban; - végre hogy az anya,...  
 ...lftotta Miskeit, ez pedig csak xxxxxxx Syllabáju szókkal igen rövi...  
 ...vány, és néptelen; imitt-amott xxxxxxx par [!] szeretseny Familiát...  
 ...sairól, máglyára kárhoztatnád? xxxxxxx példány ára három forint. De ...  
 ...en fogtunk ülést, mindegyikünk xxxxxxx ...  
 ...szerezni, mint a multba vetett xxxxxxx pillantás, mely elénk varázso...

Korábbi tipp  
 hosszas  
 mély

1. ábra: A szó a karakterekben vett hosszának megfelelő maszkolókarakterrel (nagy x) van kitakarva, és a jobb és bal oldalán a meghatározott hosszúságú szöveggörnyezete található. Az egyes sorok együtt konkordanciát alkotnak. A képen látható konkordancia megfejtése „egy-egy”.

2000 novemberében a MorphoLogic Kft. bemutatta a Nyelvi játék © nevű szoftverét<sup>1</sup> (1. ábra). Az egyszemélyes játék célja, hogy a játékos kitalálja a hiányzó szót, az úgynevezett KWIC-t (Keyword in Context) annak kontextusából. A szó a karakterekben vett hosszának megfelelő maszkolókarakterrel (nagy x) van kitakarva, és a jobb és bal oldalán a meghatározott hosszúságú szöveggörnyezete található. A felhasználónak lehetősége van tippelni, illetve egy újabb korpuszpéldát kérni. Az utóbbi esetben megjelenik egy új sorban egy újabb környezeti előfordulás a középen elhelyezkedő szó köré igazítva. Így a szó előfordulásainak példái konkordanciát alkotnak. A kitalálandó szóhoz tartozó példák

<sup>1</sup> Jelenleg nem ismert publikusan elérhető példány.

tárháza szükségszerűen véges, így annak kimerülése után a játék végét az jelenti, ha a játékos kitalálja a szót vagy feladja. A játék a későbbiekben csaknem teljesen elfelejtődött, de szavak és a környezetük kapcsolata tovább foglalkoztatta a számítógépes nyelvészeket itthon és külföldön egyaránt. A további fejezetekben néhány, kontextusokat érintő eredmény tárgyalása után bemutatjuk, hogy a játék elve miként alkalmazható különféle nyelvi jelenségek vizsgálatára ember és gép összehasonlításában.

### 3. A jobb- és bal oldali kontextus szerepe

A német nyelvű POS taggelés feladatát vizsgálva Ivanova és Kuebler (2008) felteszi a kérdést, hogy „milyen fontos a jobboldali kontextus?” A *memóriaalapú tanulást* (Lin és Vitter, 1994) alkalmazó modelljükben a klasszikus Markov-elvű bigram és trigram konfigurációt használtak: a kérdéses szótól balra található egy vagy két szó címkéjéből következtettek a szófaji címkére. Megvizsgálták továbbá azt az esetet, amikor egy vagy két szó lehetséges címkehalmazát<sup>2</sup> is bevették a jobb oldali környezetet alkotó szavakból. Ha csak a jobb oldali lehetséges címkéket vizsgálták, akkor 1,5%-os, ha pedig egy vagy két bal oldali már meghatározott címkét is bevettek, 2%-os javulást értek el a csak bal oldali címkék használatához képest. Habár eredményük elmarad 1%-kal a T'n'T rendszer (Brants, 2000) globális optimalizációt és a szövegeket is figyelembe vevő modelljéhez képest, jól látható az eredményeikből a jobb oldali környezet fontossága egy olyan merev szórendű nyelv esetén, mint a német<sup>3</sup>. Az n-gram modellek csúcsának számító KenLM (Heafield, 2011) a ritka n-gramokból adódó bizonytalanság leküzdésére különféle simítási eljárásokat használ, és alacsonyabb rendű n-gramokat vesz figyelembe rendkívüli sebességgel. Ebből következően a sima kontextusok vizsgálatának szerepe végleg háttérbe szorul, és a vektoros reprezentációk megjelenésével Mikolov és mtsai (2013b) gyakorlatilag teljesen megszűnik.

Az MTA-PPKE Magyar Nyelvtudományi Kutatócsoportban az ANAGRAMA projekt célja egy olyan szintaktikai elemzőrendszer létrehozása volt, amely az emberi olvasáshoz és halláshoz hasonlóan időben előre, „szigorúan balról jobbra” elemzi a bemenetet (Prószték és mtsai, 2014). A kutatócsoport munkatársai definiáltak egy három token hosszúságú ablakot, amely az elemzendő szóhoz képest jobbra helyezkedik el (Indig és mtsai, 2016). Így szemben a hagyományos Markovi feltételezéssel, a teljes bal oldali mellett három hosszú jobb oldali kontextus is figyelembe vehető az elemzés során az egyes nyelvi jelenségek kezelésekor, melyet az elvált igekötők kezelésével demonstráltak (Vadász és mtsai, 2017). A későbbiekben egyre nagyobb hangsúly helyeződött a jobboldali környezetre, melyet a kötöttebb szórendű magyar főnévi csoportok területén (Ligeti-Nagy és mtsai,

<sup>2</sup> A dekódolás balról jobbra történt, így nem lehetett a jobb oldali környezetet fixen meghatározni előre.

<sup>3</sup> Az eredmények hihetőek, annak tekintetében, hogy a német nyelvben a nyelvtani esetet kódoló névelő a kötőszó és az opcionálisan ragozott melléknévvel rendelkező főnév között helyezkedik el.

2018) igazoltak, a baloldalt teljesen nélkülözve. Nem minden nyelvi jelenségnek van szüksége jobboldali kontextusra, hiszen például a befejezett melléknévi igenév argumentumai tőle mindig balra helyezkednek el, és önmaga is a főnévi csoport fejtől balra foglal helyet a mondatban, vonzatainak azonosításához tehát elég a baloldali szomszédainak vizsgálata. Felmerül a kérdés, hogy az egyes nyelvi jelenségek feldolgozása milyen arányban igényel bal- és jobboldali környezetet, és ebből mi jelenik meg a nyelvmodellekben?

Az  $n$ -gram modellek adathiány-problémájának ellensúlyozásaképpen létrejöttek a szózsák modellek, melyek nem veszik figyelembe a szavak sorrendjét, így kisebb és átfogóbb modellt képesek adni, ami különösen a szabad szórendű nyelvek esetén fontos. A folytonos szózsák modell (CBOW) a szózsákok továbbfejlesztése, ahol az adott (kétoldali) kontextusai által leírt szó, nem számítva a kontextusbeli szavak sorrendjét, egy magasdimenziós térbe konvertálódik (Mikolov és mtsai, 2013b). Ez a gyengített feltételeket tartalmazó modell egyaránt jól alkalmazható a kötött szórendű nyelvek esetén, mint az angol, és a szabadabb szórendű nyelvek esetén is, mint a török. Kombinálva a globális optimalizációval teljesen maga mögé utasította a korábbi eljárásokat, és a számítási kapacitások bővülésével együtt nem tette szükségessé, hogy tovább vizsgálják, hogyan lehetne finomítani az igénybe vett szókörnyezetet, nagyobb ablakkal is kellően gyorsak voltak a számítások.

#### 4. Az adatbázis

A konkordanciákra alapuló játék elve nagyon egyszerű, ezért is kiemelt fontosságú a játék adatbázisaként használt korpusz. A reprodukálhatóság végett két szabadon elérhető korpuszt vettünk számításba. Halácsy és mtsai (2004) *Webkorpusz 1.0*-ját és Nemeskey (2021b) *Webkorpusz 2.0*-ját, melyből a későbbiekben felhasznált *huBERT* (Nemeskey, 2021a) is készült. Döntésünk indoka, hogy más szabadon elérhető, internetről származó szövegeket tartalmazó korpuszok nagy valószínűséggel e kettő korpusz valamelyikében szereplő szövegeket tartalmaznak, míg szinte biztosra vehető, hogy a két korpusz közötti átfedés nem zavaróan magas.

A kiválasztott korpuszokat meg kellett tisztítanunk a bennük szereplő szeméttől, illetve a számunkra nem megfelelő mondatoktól, hogy azok ne kerülhessenek a játékos elé. A mondatok tokenizálatlan változatait használtuk, kitöröltük és normalizáltuk a fehérszóközöket, és minden mondatból csak egy előfordulást vettünk számításba. Ezután a következő szűrősort definiáltuk és futtattuk:

- A 11 szónál rövidebb és 50 szónál hosszabb
- A több mint 25 karakter hosszú szót tartalmazó
- A 3 vagy több egymást követő szóban nagybetűs szavakat (névelemek) tartalmazó
- A legalább kettő darab, legalább négy karakter hosszú, teljesen nagybetűs szavakat (kiabálás) tartalmazó
- A három egymást követő, csak nem alfanumerikus karakterekből álló szavakat tartalmazó



- A három egymást követő, csak egybetűs szót (írógép stílus) tartalmazó
- Az összesen három vagy több visszaper (\) karaktert (escapelés) tartalmazó
- A *Unicode Replacement character*-t (U+FFFD) tartalmazó
- A „hullámos ő” (ő) és „hajtott ékezetes ú” (ú) (karakterkódolási probléma) betűket tartalmazó
- A *HTML escapelést* (&lt;, &gt;, &#12345;) tartalmazó

mondásokat kiszűrtük (1. táblázat).

	Előtte mondatok	Utána mondatok	%	Előtte szavak	Utána szavak	%
Webkorpusz 1.0	42 482 107	13 915 132	32,75	589 080 971	272 544 786	46,26
Webkorpusz 2.0	589 398 448	199 627 778	33,86	9 217 857 283	4 036 428 613	43,78

1. táblázat. A Webkorpusz 1.0 és 2.0 méretei a szűrés előtt és után.

A megszürt korpuszok mondataiból kontextusokat generáltunk a minimum 4, maximum 40 karakter hosszú kisbetűs szavakra, amelyeket megelemzett az emMorph (Novák és mtsai, 2016). A létrejött kontextusokat deduplikáltuk, és szűrtük a 30 darabnál ritkábban előforduló KWIC-eket, a gyakoribbakat pedig 30 előfordulásra mintavételeztük<sup>4</sup>. Az adatbázis méretének korlátozása miatt a maradék KWIC-ekből 8 000 darabos mintát vettünk 240 000 darab egyedi környezettel.

Érdekeséggéppen előállítottuk a játék „egyszerűbb változatát” is Kalivoda (2021) *PrevCons* adatbázisából, amelyben minden egyes igekötős ige egyszer fordul elő, de az igétet különválasztva az igekötők kitarakásával és konkordanciáival, 122 igekötővel élvezhető a játék.

## 5. Modellek

Két modellen végeztünk kísérleteket, az egyik egy BERT-alapú (Devlin és mtsai, 2019) modell, a másik egy KenLM-en alapuló modell. A modellek a Webkorpusz 2.0-n lettek feltanítva, viszont a korpusz nagy mérete miatt sok olyan szóalak fordul elő, melyek nagyon ritkák, így megszorítottuk a játék során a keresés állapotterét: mindkét modellnek azonos szóadatbázist adtunk meg, ami a Webkorpusz 2.0 3 millió leggyakoribb szava volt, ez a modellek szókincse, ezekből kellett választaniuk a modelleknek a Webkorpusz 1.0-án történő teszteléskor.

### 5.1. KenLM

A KenLM az n-gram nyelvmodellek közül az egyik leggyorsabbnak, legjobbnak számít. 5-gram modellt tanítottunk a Webkorpusz 2.0-n, és levágtuk az alacsony

<sup>4</sup> Az összes mintavételezés rögzített random seeddel történt, így reprodukálható.

frekvenciájú  $n$ -gramokat, hogy beleférjen a memóriába a modell: a 2-gramokra 4, a 3-gramokra 9, a 4-gramokra 16, az 5-gramokra pedig 25-ös alsó határt szabunk meg.

Egy adott kontextus esetén a modell úgy tippel, hogy a 3 millió szóból kiválasztja azokat, amelyeknek a hossza egyezik a hiányzó szó hosszával, ezek mindegyikére kiszámolja a szekvencia log-valószínűségét, több kontextus esetén ezeket összeadja (hiszen szorzás logaritmusos összeadás), majd az együttes log-valószínűségek közül kiválasztja a legnagyobbat, amit visszaad tippként.

## 5.2. BERT

A BERT (Devlin és mtsai, 2019) egy többszintű, kétirányú Transformer-alapú enkóder modell (Vaswani és mtsai, 2017). Enkóder lévén szóbeágyazást állít elő, minden szó–kontextus pároshoz hozzárendel egy vektort. A BERT-tanulása során kontextusokból próbálja a kimaszkolt szavakat kitalálni, hasonlóan a korábbi neurális nyelvmodellekhez (Mikolov és mtsai, 2013b; Bojanowski és mtsai, 2017; Pennington és mtsai, 2014), azzal a különbséggel, hogy a BERT tetszőleges méretű kontextust elfogad bemenetként. Nem sokkal ezelőttig nem volt elérhető magyar BERT, viszont tavaly óta elérhető a huggingface-hubról a *huBERT*<sup>5</sup> (Nemeskey, 2021b,a), amely a teljes Webkorporusz 2.0-n tanult<sup>6</sup>.

A BERT komplex, hosszútávú összefüggésekre érzékeny (Goldberg, 2019), így ideális választás hiányzó szó kitalálására. A BERT tokenizálásra a *WordPiece* (Schuster és Nakajima, 2012) algoritmust használja, ezáltal nem lehet egyszerűen karakterhossz-alapú tippelést használni BERT esetén. Amennyiben a tokenizáló egy szót például két subwordre bont, arról a BERT-nek tudnia kell, hogy két subwordöt tippeljen.

Azért, hogy optimalizáljuk a futási időt, a teljes adatbázist előre tokenizáltuk, így minden szóhoz tudjuk, hogy mely tokenekből épül fel. Ezekből építettünk egy fát, melynek a csúcaiban szavak vannak, a csúcsok között pedig tartalmazási reláció van. Amikor tippel a BERT egy több subwordből álló szót, akkor a fának kiválasztjuk az adott mélységhez tartozó összes csúcsát, a csúcsokhoz kiszámoljuk a csúcsot alkotó subwordök valószínűségét, majd ezeket csúcsenként összeszorozzuk (a BERT nem számol együttes valószínűséget).

Implementációként a PyTorch-alapú Huggingface modellt használtuk<sup>7</sup>.

## 5.3. Kontextusfüggetlen szóbeágyazások

A BERT-hez hasonlóan a korábbi neurális modellek is úgy tanulnak, hogy a hiányzó szót találják ki egy adott kontextus alapján, így ezeket a modelleket is kipróbáltuk a *Gensim* (Rehurek és Sojka, 2011) csomagban.

<sup>5</sup> <https://huggingface.co/SZTAKI-HLT/hubert-base-cc>

<sup>6</sup> Ezért is kellett egy olyan korpuszt választani teszteléshez, aminek a lehető legkisebb átfedése van a tanítóanyaggal.

<sup>7</sup> [https://huggingface.co/transformers/model\\_doc/bert.html#transformers.BertForMaskedLM](https://huggingface.co/transformers/model_doc/bert.html#transformers.BertForMaskedLM)

Több modellt kipróbáltunk: CBOW, skip-gram (Mikolov és mtsai, 2013b), és *FastText* (Bojanowski és mtsai, 2017) modelleket tanítottunk a Webkorpusz 2.0-n. A Gensim keretrendszerben található egy beépített függvény<sup>8</sup> a hiányzó szó tippelésére, viszont nagyon rossz tippeket adott.

Ez egybevág azokkal a korábbi tapasztalatokkal, miszerint a hasonló a modelleket nagyon nehéz kiértékelni (Faruqui és mtsai, 2016), és így a modellek ereje nem a tanítófeladaton jelenik meg leginkább. A jelenség kiküszöböléséppen mindig downstream feladatokon tesztelik őket, mint például szóanalógia (Mikolov és mtsai, 2013a) és szóhasonlóság.

Kétféle modellt tanítottunk: egy 100-dimenziós CBOW modellt és egy 100-dimenziós FastText modellt. A rossz teljesítmény miatt ezzel a modelleszáddal nem végeztünk hosszabb tesztek, viszont elérhető segédként a FastText modell: egy-egy tipphez meg tudja mondani, hogy mennyire hasonló a tipp a hiányzó szóhoz.

## 6. Mérések

Többféle mérést végeztünk, arra a kérdésre kerestük a választ, hogy mely oldali kontextus fontosabb mely modellnek, mik az általános tendenciák modellenként és modellek közt. Valamint a kontextusok számának szempontjából további két típusú méréssel vizsgáltuk a modelleket: egyrészt arra kerestük a választ, hogy mekkora az a kontextusméret, ami alapján egyetlen kontextusból kitalálja a modell, hogy mi a hiányzó szó, illetve hogy egy adott szóhoz és kontextusmérethez hány kontextus kell átlagosan, hogy a modell ki tudja találni a hiányzó szót.

### 6.1. Kétoldali kontextus

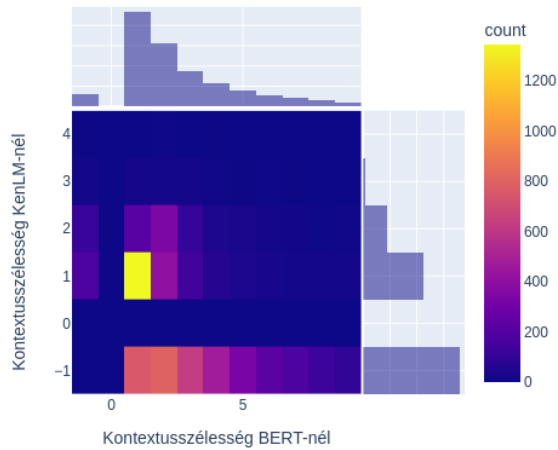
Az első típusú mérésünknel azt mértük ki, hogy egy adott szóhoz mekkora az a legkisebb kontextus, amiből a modellek kitalálják a szót, ez a 2. ábrán látható.

Az első és legszembetűnőbb észrevétel az, hogy a KenLM sokkal kevesebb szót talált el, mint a BERT, ráadásul nagyon kevés olyan eset volt (4,4%), ahol a KenLM kitalálta, a BERT pedig nem. Továbbiakban a BERT sokkal magasabb találati arányt ért el (65,0% vs. 32,1%), azaz ezekben az esetekben volt olyan kontextusszélesség, hogy a modell kitalálta a hiányzó szót. Látható továbbá, hogy amíg a KenLM esetén a harmadik, negyedik szó nem hozott sok új találatot (azaz a KenLM szempontjából nem rendelkezett információval), addig BERT esetén egy nagyon lassan lecsengő tendenciát látunk – még a kilencedik, tizedik szónál is plusz információt nyert ki a BERT a szövegből a hiányzó szóhoz kapcsolódóan.

A második típusú mérésben azt hasonlítottuk össze, hogy 10-széles kétoldali kontextus esetén hány tipp alatt jutnak el a modellek a helyes megoldásig úgy, hogy amit már egyszer tippeltek, azt nem tippelhetik újra. Minden próbálkozás után hozzáfűztünk a bemeneti mondatokhoz egy új mondatot, és a modellek az eddigi összes modell alapján tippeltek egy szót újra. A 3. ábrán látható, hogy a

<sup>8</sup> `gensim.models.word2vec.Word2Vec.predict_output_word`

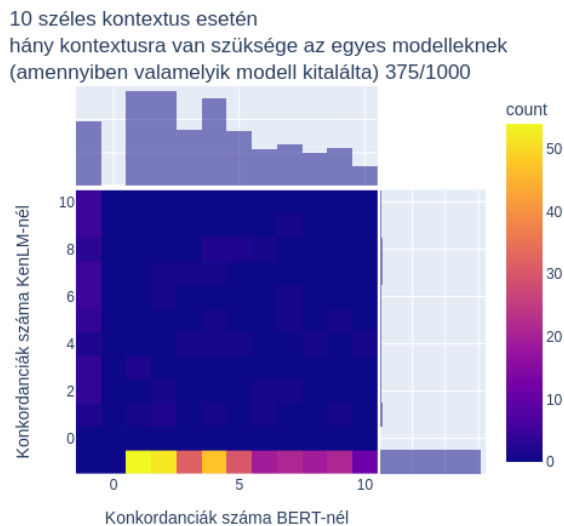
Mekkora KÉToldali kontextus kell a hiányzó szó kitalálásához (amennyiben valamelyik modell kitalálta) 6793/10000



2. ábra: Hőtérkép arról, hogy egy adott szóhoz mekkora kontextus kellett egyik-másik modellnek. A -1 azt jelenti, hogy a modell nem találta ki az adott szót 10-széles kontextussal sem, a címben szereplő 6793/10000 pedig azt jelenti, hogy 10000 szó-kontextus pár esetén 6793 olyan pár volt, ahol legalább az egyik modell kitalálta a hiányzó szót.

KenLM nem tudja jól használni a több kontextus által adott többletinformációt, összesen az esetek 6,9%-ában tudja kitalálni a hiányzó szót, szemben a BERT 33,6%-os eredményével.

Egy-egy esetben a 10 legvalószínűbb szót adják vissza a modellek – a BERT-nél egy-egy konkordancia esetén sokat változik ez a top 10 szó, KenLM esetén viszont 1-2 kontextus után „befagy”. Ezek alapján azt is megvizsgáltuk, hogy az a helyzet áll-e a fenn, hogy egy modell kimenete „befagy”, és csak azért találja-e el a hiányzó szót, mert elfogynak a tippek előre. Arra jutottunk, hogy KenLM esetén, ha az első tipp top 10-ében nem volt benne a hiányzó szó, akkor csak az esetek 1,3%-ában volt benne egy későbbi kör top 10 tippjében, BERT esetén pedig az esetek 17,5%-ában. Jól látható tehát, hogy a BERT-nek sokat segít a több kontextus, KenLM esetén viszont annyira kilapul a valószínűségi eloszlás, hogy a több kontextusból már nem nyer ki több információt a hiányzó szóra vonatkozóan.



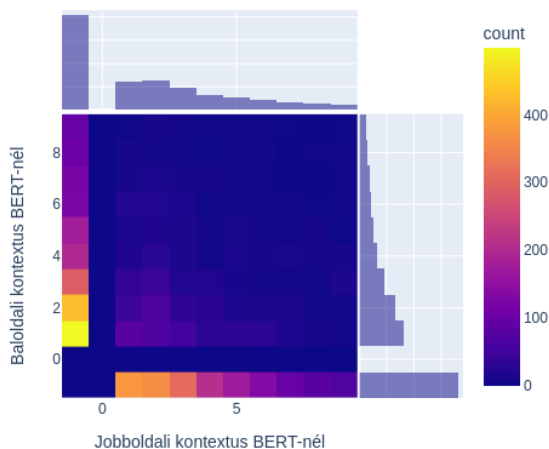
3. ábra: Hőtérkép arról, hogy egy konkordanciához mennyi kontextus kellett egyik-másik modellnek. A -1 azt jelenti, hogy a modell nem találta ki az adott szót 10 kontextussal sem, a címben szereplő 375/1000 pedig azt jelenti, hogy 1000 10 kontextus-szó pár esetén 375 olyan pár volt, ahol legalább az egyik modell kitalálta a hiányzó szót.

## 6.2. Egyoldali kontextus

Hasonlóan a kétoldali kontextushoz, az egyoldali kontextusokra is futtatunk egyrészt a *minimális kontextusméretet*, másrészt pedig a *kontextusok számát* vizsgáló kísérleteket.

Kontextusok minimális mérete szempontjából a két modellnél azonosak a tapasztalataink. A modellek teljesítménye oldalfüggetlen – noha a BERT körülbelül dupla annyi szót tudott kitalálni, mint a KenLM, mind a két modellnél oldaltól független az eredmény: a kitalált kontextusok fele bal oldali volt, a másik fele pedig jobb oldali, nagyon minimális átfedéssel. Az átfedés BERT esetén 23% (ha valamelyiket kitalálta), azaz a szavak 23%-a olyan, hogy egyaránt a bal oldali és jobb oldali kontextusból kitalálható, a KenLM viszont oldalfüggőbb: ott a kontextusoknak már csak a 9%-a olyan, hogy mindkét oldalról kitalálható.

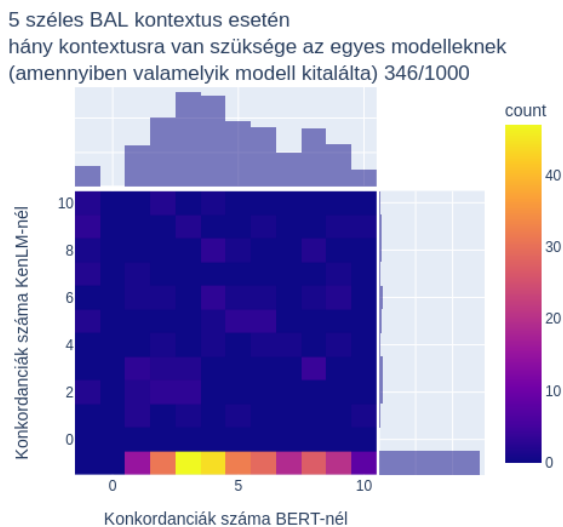
Mekkora kontextus kell a hiányzó szó kitalálásához  
(amennyiben ki lett találva) 5001/10000



4. ábra: Hőtérkép arról, hogy egy adott szóhoz mekkora kontextus kellett egyik-másik modellnek. A -1 azt jelenti, hogy a modell nem találta ki az adott szót 10-széles kontextussal sem. Nem nagyon van átfedés a kétoldali kitalálhatóság közt.

Az egyoldali konkordanciákat rövidebb kontextussal is megvizsgáltuk, hogy erősebb-e a több kontextus hatása a kétoldalihoz képest, a baloldali eset látható a 5. ábrán. Egyenletes lecsengés helyett azt látjuk, hogy a harmadik-negyedik kontextusnál éri el a BERT teljesítménye az inflexiós pontot – azaz a BERT szempontjából a harmadik-negyedik kontextusok adják a legtöbb információt a hiányzó szóra nézve, utána csökken az újonnan kitaláltak száma. Mindez azt mu-

tatja, hogy a BERT nemcsak a hosszú összefüggéseket, ezáltal a széles kontextusokat tudja jól kihasználni, hanem a több kontextusból érkező plusz információt is.



5. ábra: Hőtérkép arról, hogy egy konkordanciához mennyi kontextus kellett egyik-másik modellnek. A -1 azt jelenti, hogy a modell nem találta ki az adott szót 10 kontextussal sem. Látható, hogy amit a KenLM kitalál, azt a BERT is.

## 7. Összegzés

A MorphoLogic Kft. Nyelvi játék © elvei mentén létrehoztunk egy platformot, amelyben teljesen reprodukálható módon hasonlíthatók össze a különböző elméletekre épülő nyelvmodellek, és az emberi gondolkodás is vizsgálható különböző méretű környezeteken, valamint az eredeti játék funkciójára is alkalmas. Az eredetileg egyszemélyes játék gépi ellenféllel is játszható, valamint a FastText modell által a tippelt és a helyes szó közötti hasonlóság is megjelenik, segítve a játékoszt a további tippek megtételében. A kód és az adatbázis független a MorphoLogic Kft. implementációjától és szabadon elérhető<sup>9</sup>.

<sup>9</sup> A kódok és az adatbázis előállításának lépései elérhetők LGPL 3.0 licenc alatt: <https://github.com/ELTE-DH/word-guessing-game> és <https://github.com/ELTE-DH/BERTfluff>. A játék kipróbálható itt: <https://word-concordance-game.herokuapp.com/>. Valamint az igezőtős verzió itt: <https://prev-guessing-game.herokuapp.com/>

A mérési eredményeink bemutatják az egyes modellek korlátait: a kontextusfüggetlen szóbeágyazások nem alkalmasak a játék céljaira, de jól használhatóak a tippek analogikus hasonlóságának számszerűsítésére. A vizsgált n-gram modell a trigramoknál nagyobb környezetben nem volt képes javítani a tippjein, miközben a BERT alapú modell lassú lecsengéssel tovább javult. A fejlettebb kontextusfüggő szóbeágyazások esetén alig, míg az n-gramok esetén egyáltalán nem voltak képesek a modellek a többi példa figyelembe vételével javítani a kezdeti tippjeiken. A jobb- és bal oldali szósomszedság vitájában fej-fej melletti eredmények születtek, és érdekes módon az oldalak között kevés volt az átfedés: ha az egyik oldali környezet ki tudta találni a modellt a szót, a másikon szinte alig.

A problémakör ismertetett számos meglepő eredménye nem merítette ki a platform által megvalósítható kísérletek tárházát, így az szabadon tovább vizsgálható. A cikkünkben nem vizsgáltuk, hogy az emberi elemző miként reagál azokra az esetekre, amelyek a gépi nyelvmodellnek könnyen mentek vagy nehézséget okoztak. A későbbiekben ez terveink között szerepel, hogy megállapíthatjuk, hogy a mesterséges intelligencia ebben a játékban is legyőzi-e az embert, vagy az emberek nyelvi intuíciója a jelenlegi technológiai fejlettség mellett még nem modellezhető maradéktalanul.

## Köszönetnyilvánítás

A szerzők köszönetet mondanak a MorphoLogic Kft. egykori munkatársainak, akik 2000-ben létrehozták a Nyelvi Játék © szoftvert. Továbbá szeretnénk megköszönni a BME MOKK egykori munkatársainak és Nemeskey Dávidnak, hogy teljesen szabadon elérhetővé tették korpuszaikat. Külön köszönetet érdemel Kalivoda Ágnes, aki példamutatóan szabadon elérhetővé tette az általa létrehozott adatbázisokat.

## Hivatkozások

- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146 (2017), <https://aclanthology.org/Q17-1010>
- Brants, T.: TnT – A Statistical Part-of-Speech Tagger. In: *Sixth Applied Natural Language Processing Conference*. pp. 224–231. Association for Computational Linguistics, Seattle, Washington, USA (Apr 2000), <https://aclanthology.org/A00-1031>
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://aclanthology.org/N19-1423>
- Elkins, K., Chun, J.: Can GPT-3 Pass a Writer’s Turing Test? *Journal of Cultural Analytics* 1(1), 17212 (2020)



- Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C.: Problems with evaluation of word embeddings using word similarity tasks. In: Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP. pp. 30–35. Association for Computational Linguistics, Berlin, Germany (Aug 2016), <https://aclanthology.org/W16-2506>
- Firth, J.: A Synopsis of Linguistic Theory 1930-1955. In: Studies in Linguistic Analysis. Philological Society, Oxford (1957), reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.
- Goldberg, Y.: Assessing BERT’s Syntactic Abilities. ArXiv abs/1901.05287 (2019)
- Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04). European Language Resources Association (ELRA), Lisbon, Portugal (May 2004), <http://www.lrec-conf.org/proceedings/lrec2004/pdf/525.pdf>
- Harris, Z.S.: Distributional structure. WORD 10(2-3), 146–162 (1954)
- Heafield, K.: KenLM: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. pp. 187–197. Association for Computational Linguistics, Edinburgh, Scotland (Jul 2011), <https://aclanthology.org/W11-2123>
- Indig, B., Vadász, N., Kalivoda, Á.: Decreasing entropy: How wide to open the window? In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (szerk.) Theory and Practice of Natural Computing. pp. 137–148. Springer International Publishing, Cham (2016)
- Ivanova, S., Kuebler, S.: POS Tagging for German: how important is the Right Context? In: Calzolari, N. (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapias, D. (szerk.) Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08). European Language Resources Association (ELRA), Marrakech, Morocco (may 2008), <http://www.lrec-conf.org/proceedings/lrec2008/>
- Kalivoda, Á.: Az igekötők produktív kapcsolódási mintái. Argumentum 17, 56–82 (2021)
- Ligeti-Nagy, N., Vadász, N., Dömötör, A., Indig, B.: Nulla vagy semmi? In: XIV. Magyar Számítógépes Nyelvészeti Konferencia : MSZNY 2018. pp. 25–37 (2018)
- Lin, J.H., Vitter, J.S.: A theory for memory-based learning. Machine Learning 17(2-3), 143–167 (1994)
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR 2013 (01 2013a)
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 3111–3119. NIPS’13, Curran Associates Inc., Red Hook, NY, USA (2013b)
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). pp. 3–14 (2021a)

- Nemeskey, D.M.: Natural language processing methods for language modeling. Ph.D.-értekezés, Doctoral School of informatics, Eötvös Loránd University, Faculty of Faculty of Informatics (2021b)
- Novák, A., Siklósi, B., Oravecz, C.: A new integrated open-source morphological analyzer for Hungarian. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 1315–1322. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), <https://aclanthology.org/L16-1209>
- Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), <http://www.aclweb.org/anthology/D14-1162>
- Prószéky, G., Indig, B., Miháltz, M., Sass, B.: Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In: X. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2014). pp. 79–87 (2014)
- Rehurek, R., Sojka, P.: Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3(2) (2011)
- Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5149–5152 (2012)
- Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal 27(3), 379–423 (1948)
- Vadász, N., Kalivoda, Á., Indig, B.: Ablak által világosan – vonzatkeret-egyértelműsítés az igekötők és az infinitívuszi vonzatok segítségével. In: XI-II. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017). pp. 3–12 (2017)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is All you Need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>

## Charmen ELECTRA - Tokenizációmentes diszkriminatív nyelvi modellezés

Ficsor Tamás<sup>1</sup>, Cserhádi Réka<sup>1</sup>, Novák Attila<sup>2</sup>, Mihajlik Péter<sup>3</sup>, Zainkó Csaba<sup>3</sup>,  
Berend Gábor<sup>1</sup>

<sup>1</sup>Szegedi Tudományegyetem, Informatikai Intézet  
{ficsort,cserhatir,berendg}@inf.u-szeged.hu

<sup>2</sup>Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar  
novak.attila@itk.ppke.hu

<sup>3</sup>Budapesti Műszaki és Gazdaságtudományi Egyetem,  
Távközlési és Médiainformatikai Tanszék  
{mihajlik,zainko}@tmit.bme.hu

**Kivonat** Napjainkban a természetesnyelv-feldolgozás területén használt neurális modellek többsége előre definiált szótöredékekből kialakított szótárakkal dolgozik. A kötött szótár használatának eredményeképp az ezekre építő modellek érzékenyek a zajra, doménadaptációjuk költségesebb lehet, ezen felül többnyelvű modellek építése esetén a szótár mérete drasztikusan megnőhet. Ezen problémák orvoslására egy tokenizációmentes ELECTRA architektúrát mutatunk be, amely a Charformer blokkot alkalmazza a tokenizáló modul kiváltására. A modell  $\sim 17\%$ -kal kevesebb paramétert tartalmaz, mint a fix szótárral rendelkező társa. Továbbá azonos körülmények között tanított társánál szignifikánsabban jobb eredményt ér el az OpinHuBank adathalmazon. Kutatásunk további eredménye, hogy a huBERT modell finomhangolásával a szentimentosztályozás területén az OpinHuBank adatbázison a korábbi legjobb eredményt meghaladó teljesítményt értünk el.

**Kulcsszavak:** Transzformer, Tokenizáció, Vélemény és Érzelem Analízis

### 1. Bevezetés

A kontextusfüggő vektorokat előállító architektúrák (Peters és mtsai, 2018; Devlin és mtsai, 2019) megjelenése óta lényegében egyetlen komponenssel, end-to-end módon végezhető el a természetes nyelvi objektumok modellezése, dedikált elemző, illetve előfeldolgozó komponensek (pl. szófaji elemző, dependenciaelemző, stb.) alkalmazása nélkül. Az input szekvenciák teljesen end-to-end feldolgozásának a tokenizálás szab gátat. A tokenek előállításához szükséges szótárt előfeldolgozási lépésként határozták meg, általában valamilyen jól bevált statisztikai módszerrel (Kudo, 2018; Schuster és Nakajima, 2012; Sennrich és mtsai, 2016). Így a modell tanítása során lerögzített szótöredékek alakultak ki. Ezáltal viszont domain-specifikus feladatok (Gong és mtsai, 2018) és alacsony erőforrással rendelkező nyelvek (Hu és mtsai, 2020) modellezése nehézkessé vált. A

tokenizálók érzékennyé teszik a modelleket az inputban található esetleges elírásokra, illetve többnyelvű modellek esetén erőforrásigényes lehet ezen szótárak tárolása és tanítása.

A bemenet karakterszintű modellezésére voltak már korábbi törekvések, de azok csupán a meglévő tokenizálások karakterekre bontására szorítkoztak (Ma és mtsai, 2020). A közvetlenül karakterek lánculatán történő modellezés költséges feladat is lenne, mivel a bemeneti szekvenciahossz drasztikusan meg tudna növekedni ezáltal. Ennek a problémának a megoldására a Charformer (Tay és mtsai, 2021) ad egy lehetséges alternatívát, ahol kontextusfüggő byte szintű reprezentációt alkalmazunk alumintavételezéssel. Egy másik megközelítést a CANINE (Clark és mtsai, 2021) nyújt, ahol a bemeneti karaktereket hash-beágyazással reprezentálják.

Ebbe a kutatási trendbe illeszkedve cikkünkben bemutatunk egy új, az ELECTRA (Clark és mtsai, 2020) architektúrára épülő, tokenizálómentes modellt, ahol tokenizálás helyett a Charformer modult alkalmazzuk. Ezek együttes működéséhez még további strukturális változásokat eszközölünk a transzformer működésében. Az előtanítást a Hungarian Webcorpus 2.0 Wiki alkorpuszán végezzük, míg a kiértékelést az OpinHuBank adathalmazon. A tanításhoz szükséges kódot is közzéteszük a Github repozitóriumunkban<sup>1</sup> az előtanított súlyokkal együtt.

## 2. Kapcsolódó irodalom

A kontextualizált nyelvi modellek elterjedése az elmúlt években nagy mértékben előrelendítette a természetesnyelv-feldolgozási feladatok megoldását. Az egyik első ilyen nyelvi modell a transzformer architektúrára (Vaswani és mtsai, 2017) épülő BERT (Devlin és mtsai, 2019) volt. A BERT kiemelkedő teljesítményt ért el számos diverz nyelvfeldolgozási feladaton. A transzformerre épülő nyelvi modellek családja az évek során egyre bővül. Különböző megközelítéseket láthatunk: voltak, akik a meglévő BERT teljesítményén igyekeztek javítani a modell tanítási körülményeinek megváltoztatásával (Zhuang és mtsai, 2021), míg mások a tanítás erőforrásigényét igyekeztek csökkenteni (például „desztilláló modell-tömörítés” segítségével; Sanh és mtsai (2019)). Egy, a BERT-től eltérő működést képviselő modell az ELECTRA (Clark és mtsai, 2020). Clark és mtsai (2020) a BERT autoenkóder-stílusú generatív tanítása helyett azt javasolták, hogy a versengő modelleket alkalmazó paradigma szellemében egy kis generatív háló mellett egy diszkriminátort is tanítsunk, a diszkriminátor feladatául a tanítás során azt a bináris osztályozási feladatot szabva, hogy az minél pontosabban legyen képes beazonosítani a generátor által megváltoztatott részeit az inputnak. Ez a megközelítés gyorsabb és jobb konvergenciát mutatott a BERT által nyújtottaknál, tehát a tanítás erőforrásigényét sikerült csökkenteni, még ha a standardnak számító benchmark feladatokon nyújtott eredményessége némileg el is maradt elődjétől.

Nem kellett sokat várni a BERT alapú nyelvi modellek magyar nyelvű variánsainak megjelenésére sem. Nemeskey (2020, 2021) bemutatta a huBERT-et,

<sup>1</sup> <https://github.com/ficstamas/charmen-electra>

amely a Hungarian Webcorpus 2.0-n előtanított BERT Base konfigurációjú modell. Ezzel párhuzamosan jelent meg a HILBERT (Feldmann és mtsai, 2021) is, amely a huBERT-től eltérően a BERT Large konfigurációt használja. Ennek előtanításához egy eltérő adathalmaz-együttest használtak. Bár eddig is léteztek a magyar nyelvet támogató többnyelvű modellek (mBERT, Devlin és mtsai (2019); XLM, Conneau és Lample (2019); XLM-RoBERTa, Conneau és mtsai (2020)), egynyelvű változataik jobban teljesítenek az Ács és mtsai (2021) által vizsgált feladatokon.

Ezen modellek nagy hátránya, hogy fix szótártól függenek. A szótárak előállításához számos módszer létezik – Unigram (Kudo, 2018), WordPiece (Schuster és Nakajima, 2012), Byte-Pair Encoding (Sennrich és mtsai, 2016) – azonban előbbutóbb az alkalmazás során találkozni fogunk szótáron kívül eső szótöredékekkel (főleg doménspecifikus feladatoknál), ezenfelül ez megnöveli a nyelvi modell paraméterhalmazát is. További problémája még, hogy a bemeneti sztring kismértékű változtatása sokszor erősen hat a tokenszekvenciára, tehát a modell érzékeny az elírásokra. Ezen problémák többnyelvű modellek esetén sokkal nyilvánvalóbbak, de kialakulhatnak egynyelvű reprezentációk esetén is.

Számos egyéb megközelítés mellett Clark és mtsai (2021) ennek megoldására tettek javaslatot a CANINE architektúrával. Ezen modell bemeneti azonosítói a karakterek UTF-8 kódolása szerinti reprezentációk. Ebből a kontextuális karakterreprezentációkat további hash-beágyazások (Svenstrup és mtsai, 2017) alapján állítja elő és továbbítja a transzformernek. Egy másik megoldás a Tay és mtsai (2021) által bemutatott Charformer blokk. Az egyik legnagyobb különbség a két módszer között, hogy a Charformer byte-szinten kódolja a bemenetet, míg a CANINE karakter (hash) szinten. Továbbá a Charformer a karakterek fix kiterjedésű szomszédságát is figyelembe veszi.

### 3. Architektúra

A fejezetben bemutatandó új architektúra két koncepcióra épít. Az első, hogy a modell legyen tokenizációfüggetlen, amelyet a Charformer blokk (Tay és mtsai, 2021) alkalmazása biztosít. Ezenfelül a hatékony konvergencia érdekében a tanítás menete ne generatív, hanem diszkriminatív módszer mentén történjen, erre pedig az ELECTRA modell biztosítja az alapot. Az ezen alapokra ültetett modellünket Charmen ELECTRA-nak (**k**arakteralapú, tokenizáció**m**entes **ELECTRA**) kereszteljük, amelyet Charmen-E-nek rövidítünk az ábrákon.

#### 3.1. Charformer

Ebben a részben a Tay és mtsai (2021) által definiált Charformer architektúra működését mutatjuk be. Első lépésként lerögzítjük az  $\mathcal{E} \in \mathbb{R}^{V \times d}$  szótármatrixot, ahol  $V$  a tokenek száma,  $d$  pedig a dimenziók száma. Esetünkben  $V = 263$  mivel 256 byte-ot kódolunk, és fenntartunk további 7 vektort speciális tokenek eltárolására ( $[PAD]$ ,  $[MASK]$ ,  $[CLS]$ , további négy tokent végül mi nem hasznosítottuk). Így a bemenő szöveget a karakterek UTF-8-as reprezentációjával, mint byte-szekvenciával kódoljuk, amit  $X \in \mathbb{R}^{n \times d}$ -vel jelölünk.

Ezt követően a beágyazásunkra egydimenziós konvolúciót alkalmazunk a karakterek mentén, hogy globális információt szerezzünk az egyes pozíciókról. Az így kapott karakterbeágyazást tovább alakítjuk egy olyan látens reprezentációba, ahol a szótöredékeket reprezentáljuk legfeljebb  $M$  méretű kiterjedésben. Ehhez egy csúszóablakos átlagolófüggvényt vezetünk be  $F : \mathbb{R}^{b \times d} \rightarrow \mathbb{R}^d$ , ahol  $b \in \{1, 2, \dots, M\}$  a blokk mérete. Így

$$X_b = (F(X_{i:i+b}); F(X_{(i+s):(i+s)+b}); \dots),$$

definiálja az  $i$  karakterpozícióhoz és  $b$  méretű szótöredékblokkhoz tartozó reprezentációt,  $s = b$  lépésköz mentén. Ez alapján minden karakterpozícióhoz ( $i$ ) meg tudjuk mondani, hogy melyik szótöredék ( $b$  méretű) definiálja azt. Ehhez vegyük az  $F_R : \mathbb{R}^d \rightarrow \mathbb{R}$  lineáris transzformációt. Ebből adódik, hogy  $X_{b,i}$   $b$  blokkhoz tartozási értéke:

$$p_{b,i} = F_R(X_{b,i}).$$

Továbbá bevezetünk egy konszenzus modult is, hogy az egyes pozíciók közötti döntésről legyen információnk. Úgy tekinthetünk rá, mint egy blokkok közti self-attentionre, ami a következőképpen van formalizálva:

$$\hat{P} = \text{softmax}(PP^T)P.$$

A szótöredékek keverékét alkotó látens reprezentációt pedig a következőképpen tudjuk meghatározni:

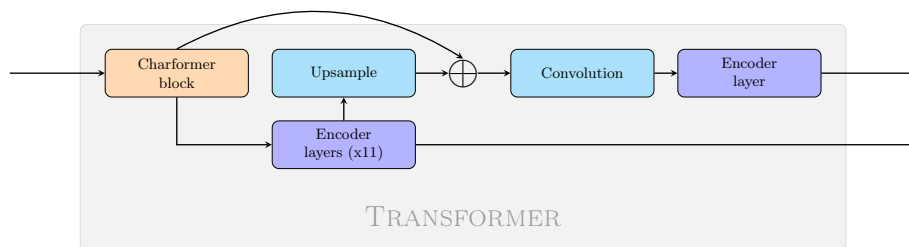
$$\hat{X}_i = \sum_{b=1}^M \hat{P}_{b,i} X_{b,i}. \quad (1)$$

Az így előállt reprezentáció redundáns információt tartalmazhat, mivel az információ alacsony lexikális szinten van reprezentálva. Ehhez egy  $F_D : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{\frac{n}{d_s} \times d}$  átlagoláson alapuló összegzőfüggvényt alkalmazunk, amely  $d_s$  faktorial fogja a bemenet hosszát csökkenteni. Az így előállt reprezentációt  $\tilde{X}$  fogja jelölni.

### 3.2. ELECTRA

Az eredeti ELECTRA modell egy generátor- és egy diszkriminátor hálóból áll. Mindkét háló transzformer architektúrájú. Lényeges különbség a komplexitásukban (paraméterszám) és a feladataikban van. A generátort jóval kevesebb paraméter alkotja, mint a diszkriminátort. Ennek egyik oka, hogy a generátort az előtanítás után eldobjuk, a másik, hogy komplexitásának növelése nagyon megnövelné a tanítás költségét. Ezenfelül az erre irányuló kísérletek során az az eredmény adódott (Clark és mtsai, 2020), hogy a generátor komplexitásának növelése nem vezet a diszkriminátor teljesítményének javulásához.

A generátor feladata megegyezik a maszkolt nyelvmodellezéssel (MLM). Bemenetként a generátormodell maszkolt szöveget kap, és a maszkolt fragmensek helyére jól illeszkedő szövegrészleteket javasol. A diszkriminátor bemenete egy olyan megváltoztatott szekvencia lesz, amelyet a generátor kimenetéből állítunk



1. ábra: A transzformer működésének javasolt módosítása, ahol az alsó kimenet egy alulmintavételezett reprezentációt ad, a felső pedig a bemeneti szekvenciával azonos méretűt. A  $\oplus$  a 3.3 szekcióban bemutatott konkatenációs művelet.

össze. Az eredetileg maszkolt elemek helyére a generátor által javasolt lehetséges jelöltekből a Gumbel-Softmax eloszlás szerint mintavételezünk. Az így előállított pozíciókat korrump pozícióknak nevezzük, amennyiben a korrumpált token nem egyezik meg az eredeti bemeneti tokennel. A diszkriminátor feladata eldönteni a szekvencia minden pozíciójáról, hogy az ott szereplő token korrumpálva lett-e vagy sem.

Feltűnhet a probléma, hogy ha a tokenizálást Charformer blokkal helyettesítjük, akkor (a bemenet és kimenet hosszának változása miatt) nem vagyunk képesek se a maszkolt nyelvi modellezés (MLM) hibáját meghatározni, se mintavételezni. Ennek megoldására a generátort byte-szintű maszkolt nyelvmodellezésre és a maszkolt byte-ok visszaállítására tanítjuk be, míg a diszkriminátor az egyes byte-ok korrumpáltságát fogja ellenőrizni. Ehhez helyre kell állítani a transzformer kimenetét az eredeti szekvenciahosszra, amihez a következő módosításokat eszközöljük.

### 3.3. Transzformer

Emlékeztetőképpen a  $\hat{X}$  jelölte a beágyazásunkat (1) a Charformer alkalmazásával,  $\tilde{X}$  pedig ennek a  $d_s$  faktorral arányosan alulmintavételezett változatát. A most bemutatandó folyamatot az 1. ábrán is szemléltetjük.

Elsőként az alulmintavételezett reprezentációnkat továbbadjuk  $L - 1$  encoder rétegnek. Ennek a kimenete használható egyéb feladatokhoz – például szekvenciaosztályozásnál – azonban az előtanításhoz nekünk nem megfelelő.

$$h_{\tilde{X}}^{(1)} = \text{Encoder}_1(\tilde{X}), \quad h_{\tilde{X}}^{(i)} = \text{Encoder}_i(h_{\tilde{X}}^{(i-1)}), \quad i \in \{2, \dots, L - 1\}$$

Ahhoz, hogy a szekvencia hosszát visszaállítsuk, a Clark és mtsai (2021) által javasolt módszert alkalmazzuk. Először a kiemenetet felülmintavételezzük olyan módon, hogy minden vektort  $d_s$ -sel arányos mennyiségben megismétlünk egymás után. Ezt követően az eredeti beágyazást ( $\hat{X}$ ) és a felülmintavételezett reprezentációt összekonkatenáljuk a  $\oplus : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times 2d}$  művelettel. A

felülmintavételezett rejtett reprezentáció a következőképpen adódik:

$$h_{up} = Conv(\hat{X} \oplus \text{UPSAMPLE}(h_{\hat{X}}^{(L-1)}, d_s), w),$$

ahol  $w$  a konvolúciós kernel szélessége, ami a mi esetünkben mindig azonos a maximális blokkmérettel ( $M$ ). Továbbá a konvolúcióra a következő teljesül:  $Conv : \mathbb{R}^{n \times 2d} \rightarrow \mathbb{R}^{n \times d}$ .

Erre a reprezentációra még egy utolsó Encoder réteget alkalmazunk, hogy a skálázott reprezentációt finomítsuk.

$$h_{up}^{(L)} = \text{Encoder}_L(h_{up})$$

## 4. Konfiguráció

### 4.1. ELECTRA

Ahhoz, hogy a modellünk teljesítménye összehasonlítható legyen a jelenlegi tokenizálóeljárásokat alkalmazó modellekével, a huBERT<sup>2</sup> tokenizálójával is előtanítunk egy ELECTRA modellt. A generátor<sup>3</sup> és a diszkriminátor<sup>4</sup> a Huggingface-en közzétett ELECTRA Base konfiguráció szerint lett inicializálva. Továbbá a maximális bemeneti szekvencia hossz 256-re van állítva. Erre a modellre egyszerűen csak *ELECTRA*-ként fogunk hivatkozni.

### 4.2. Charmen ELECTRA

Az 3. szekciókban bemutatott modellt az ábrákon csak *Charmen-E*-ként fogjuk említeni. Ezen kívül, ha a felülmintavételezett kimenetet használjuk, akkor azt külön jelezni fogjuk. A transzformer eredeti moduljait (encoder rétegeket, beágyazásokat) a Huggingface-en is található konfiguráció szerint inicializáltuk (beleértve mind a generátor-, mind a diszkriminátor blokkot). A tokenizáló egységen belül két paraméter változtatásának hatását vizsgáljuk: a maximális blokkméretét ( $M \in \{4, 6\}$ ), és az alulmintavételezési arányét ( $d_s \in \{2, 4\}$ ).  $d_s$ -sel arányosan a maximális szekvenciahossz is változik, méghozzá úgy, hogy az alulmintavételezett hossz megegyezzen az ELECTRA esetén alkalmazott értékkel (256).

## 5. Előtanítás

Az előtanítás során az ELECTRA modell két hibatag kombinációját veszi figyelembe. A generátor paramétereit a maszkolt nyelvi modellezési hiba visszatérjesztésével tanítjuk, ami a keresztentrópiával van kifejezve. A diszkriminátor tanításához pedig a korrumpált szövegrészek felderítésére vonatkozó hibát használjuk, ami bináris keresztentrópiával van meghatározva. Feltűnhet, hogy a generátor hibatagja könnyen elnyomhatja a diszkriminátor hibáját (hiszen míg a

<sup>2</sup> <https://huggingface.co/SZTAKI-HLT/hubert-base-cc>

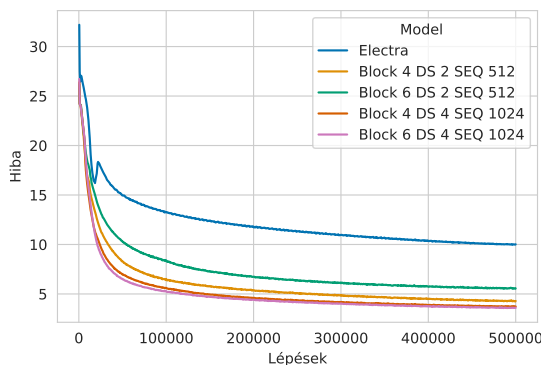
<sup>3</sup> <https://huggingface.co/google/electra-base-generator>

<sup>4</sup> <https://huggingface.co/google/electra-base-discriminator>



	Batch-méret	Lépések	Tanítási idő	$M$	$d_s$	Szekvenciahossz	Hiba	$ \theta $
ELECTRA	16	500 000	$\approx 150$ óra	–	–	256	10.02	120M
Charmen ELECTRA	16	500 000	$\approx 78$ óra	4	4	1024	3.73	101M
Charmen ELECTRA	16	500 000	$\approx 78$ óra	6	4	1024	3.56	101M
Charmen ELECTRA	16	500 000	$\approx 63$ óra	4	2	512	4.27	98M
Charmen ELECTRA	16	500 000	$\approx 63$ óra	6	2	512	3.59	98M

1. táblázat. Az előtanított modellek konfigurációja, és az előtanítással járó idő, illetve hiba.  $d_s$  az alulmintavételezési arányt,  $M$  pedig a maximális blokkméretet jelöli.



2. ábra: Az egyes modellek előtanítása során a hiba változása. ELECTRA a huBERT tokenizációs modellt takarja. A további 4 konfiguráció a Charmen ELECTRA paramétereit írja le. Block a maximális blokkméret ( $M$ ), DS az alulmintavételezési arány ( $d_s$ ), és SEQ a maximális bemeneti szekvencia hossz.

generátor egy sokosztályos, addig a diszkriminátor csupán egy bináris osztályozási feladatot lát el), ezért bevezetünk egy súlytényezőt az egyes hibategyekre, Clark és mtsai (2020) nyomán:

$$\mathcal{L}(x, \theta) = \lambda_1 \mathcal{L}_{Gen}(x, \theta_G) + \lambda_2 \mathcal{L}_{Disc}(x, \theta_D),$$

ahol  $(\lambda_1, \lambda_2)$  a két hibategyhez tartozó súly. Az eredeti cikkben javasolttal megegyező módon az  $(1, 50)$  értékeket használtuk a  $\lambda$  súlyparaméterekre.

A modelleket a Hungarian Webcorpus 2.0<sup>5</sup> (Nemeskey, 2020) Wikipedia-alkorpuszán tanítottuk, 500 000 lépésen keresztül összességében 16-os batch-mérettel (8-as batch-méret/GPU). A tanítás során  $8 \cdot 10^{-5}$  tanulási rátát használtunk AdamW optimalizálással ( $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda = 0.01$ ), és lineáris tanulásirata-ütemezőt 15 000 lépéses warmuppal. Az így előállított modellek konfigurációi részletesebben az 1. táblázatban láthatóak.

Az előtanítás során a hiba változását a 2. ábrán láthatjuk, ez magába foglalja együttesen a generátor és a diszkriminátor hibáját is. Az ábrán az ELEC-

<sup>5</sup> <https://hlt.bme.hu/en/resources/webcorpus2>

TRA és Charmen ELECTRA modellek hibája van feltüntetve, ahol a Charmen ELECTRA modellváltozatok a konfigurációs értékek szerint van megjelenítve. Az ábrán belül a `Block` a maximális blokkméretet ( $M$ ), `DS` az alulmintavételezési arányt ( $d_s$ ), a `SEQ` pedig a maximális bemeneti szekvenciahossz értékét jelöli. A Charmen ELECTRA hibaértéke első ránézésre jóval kedvezőbbnek tűnhet, azonban mivel az MLM feladat során az ELECTRA nagyjából 32 000 lehetséges kimenet közül választ, míg a Charmen ELECTRA csupán 263-ból, így az utóbbi feladata könnyebbnek is tekinthető. A Charmen ELECTRA modellek körében a nagyobb bemeneti szekvenciahossz kedvezőbbnek tekinthető (a modellezőképesség szempontjából), ugyanakkor az ábráról leolvasható, hogy a hibacsökkenés szempontjából a két vizsgált szekvenciahossz, és a két vizsgált blokkméret esetében ellentétesen viselkednek a modellek.

## 6. Finomhangolás

A Charmen ELECTRA esetén számos kérdést felvet, hogy mit tekinthetünk egy tokennek, hiszen a Charmen ELECTRA által használt egymás utáni karakteregyüttesek akár szóhatárokon is átívelhetnek. Ennél fogva a tokenszintű osztályozási feladatok megvalósítása nem egyértelmű. Figyelembe véve, hogy az ELECTRA és a Charmen ELECTRA más szinten reprezentálja az információt a kimeneten, így összehasonlítási alapnak a teljes szekvenciák osztályozására vonatkozó feladat használata mellett döntöttünk.

A kvalitatív elemzéshez az OpinHuBank<sup>6</sup> (Miháltz, 2013) adathalmazt választottuk. Az adathalmaz érzelmi attitűd (szentiment) osztályozási feladatot definiál. Összesen 10 000 mondat alkotja, melyekhez entitások tartoznak. Ezeket a mondat–entitás-párokat 5 annotátor pozitív, negatív és semleges kategóriába sorolta. A címkét többségi döntés alapján határozzuk meg, és ennek a meghatározása lesz a modell feladata.

A feladatot két felosztásban is megvizsgáljuk követve Hangya és mtsai (2015) munkáját. Az első esetben az összes címkét megtartjuk az osztályozáshoz, a második esetben pedig csak a nem semleges polaritást hordozó címkékkel dolgozunk. Az adathalmazt 70% tanító, 10% validációs és 20% tesztalmazra bontottuk fel.

Tekintve, hogy a modelljeink előtanítása során hiperparaméter-finomhangolást nem végeztünk, így egy stabilabb modell viselkedését is bevontuk az összehasonlításba. Ehhez a Webcorpus 2.0-n és a Wikipedia-alkorpuszon betanított huBERT modelleket is alkalmaztuk a kísérletekbe.

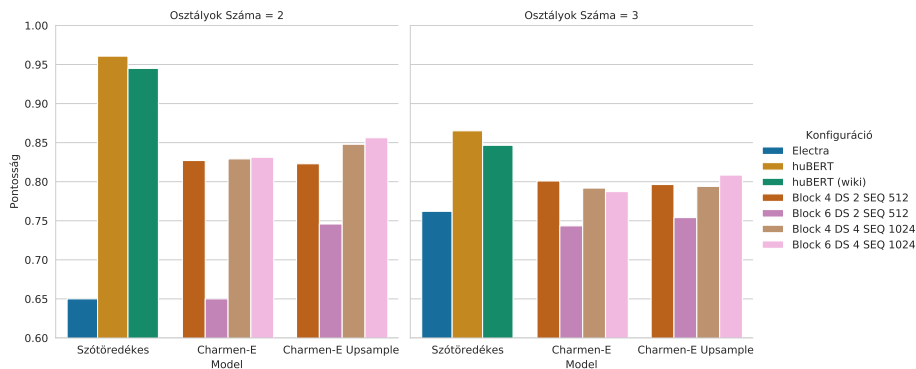
### 6.1. Hiperparaméterek

A modellek finomhangolását 3 epochon keresztül végeztük. Ez eddigiekhez hasonlóan AdamW optimalizálót használtunk ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ,  $\lambda = 0.1$ ),  $5 \cdot 10^{-5}$  induló tanulási rátával. A tanulási ráta értékét lineáris ütemben csökkentettük, és warmupot nem alkalmaztunk.

<sup>6</sup> [https://sites.google.com/site/mmihaltz/resources#h.p\\_ID\\_42](https://sites.google.com/site/mmihaltz/resources#h.p_ID_42)

#Lefagyasztott Rétegek	0	2	4	6	8	10	12
HuBERT	0.84	0.83	0.85	0.85	0.84	<b>0.85</b>	0.81
ELECTRA	0.74	0.74	0.74	0.73	0.74	0.75	<b>0.77</b>
Charmen ELECTRA	0.74	0.78	<b>0.81</b>	0.80	0.79	0.76	0.74
Charmen ELECTRA + Upsample	0.79	0.74	<b>0.81</b>	0.80	0.79	0.80	0.76

2. táblázat. A teljesítmény változása a validációs halmazon különböző számú lefagyasztott réteg alkalmazása mellett háromosztályos osztályozás esetén.



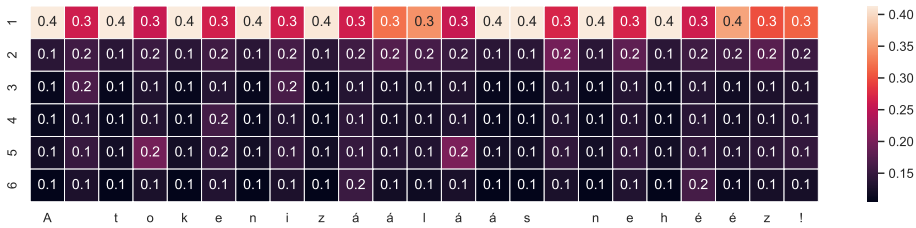
3. ábra: A modellek teljesítménye két- és háromosztályos osztályozási feladatokon. Az utolsó négy konfiguráció a Charmen ELECTRA használatának lehetséges konfigurációira vonatkozik. Block a maximális blokkméret ( $M$ ), DS az alulmintavételezési arány ( $d_s$ ), és SEQ a maximális bemeneti szekvencia hossz.

A kísérletek során a rétegek egy részét lefagyasztottuk ezzel csökkentve az túlillesztés esélyét és gyorsítva a tanítást, valamint csökkentve a memóriahasználatot. Az ehhez elvégzett kísérletek eredményeit a 2. táblázatban tüntettük fel. A táblázatban a három osztályt tartalmazó adathalmazon végzett eredmények láthatók a validációs halmazon kiértékelve. Az eredmények alapján a huBERT esetén 10, az ELECTRA esetén 12, a Charmen ELECTRA mindkét lehetséges kimenete esetén 4 réteg lefagyasztásával kaptuk a legjobb eredményeket.

## 6.2. Eredmények

Az OpinHuBank-en végzett kísérleteink eredménye a 3. ábrán látható. Itt van tüntetve a két- (bal) és háromosztályos (jobb) osztályozási eset is. Mindkét ábrán három részre osztva jelentettük meg az eredményeket. Az első csoportot a szótőredéken alapuló szótárat használó modellek, a másodikat a kimenet visszakalázása nélküli Charmen ELECTRA-modellek, a harmadikat pedig a kimenet felskalázását használó Charmen ELECTRA modellek alkotják.

A kizárólag a semlegestől eltérő címkéket használó kétosztályos feladat esetében a magyar BERT modellek teljesítettek a legjobban, egyenként  $\sim 96\%$  és



4. ábra: Charformer blokk által tanult  $\hat{P}$  mátrix, amely az egyes pozíciókhoz a blokkhoz tartozási valószínűségeket rendeli. Egyes karakterek az  $x$ -tengelyen duplán jelenhetnek meg. Ennek az oka, hogy két byte-on vannak reprezentálva.

~ 94%-os teljesítménnyel. Ezeket követi a Charmer ELECTRA-modellek többsége 82% és 86% közötti teljesítménnyel, amelyek mögött kiemelkedően rossz teljesítményt nyújtva kullog az 512-es szekvenciahosszt és 6-os blokkméretet használó modell. A szótöredékeket használó ELECTRA teljesítménye pedig a legrosszabb mind közül. Ugyanezt a tendenciát vehetjük észre a hámosztályos esetben is, annyi különbséggel hogy az ELECTRA modell relatív teljesítménye valamivel jobb lett. Első ránézésre meglepő, hogy a három egyébként legrosszabbul teljesítő modell abszolút teljesítménye javult a nehezebb feladaton, míg a többi modellel érthető módon romlott. A magyarázatot abban láthatjuk, hogy a mondatok ~ 74%-a többségi döntés szerint semleges az OpinHuBank-ben, és a gyengébb modellek a leggyakoribb címkét preferálják.

Bináris és hámosztályos esetben a két huBERT modell teljesített a legjobban, meghaladva az OpinHuBank adatbázison eddig közölt korábbi legjobb eredményeket (Hangya és mtsai, 2015). A tokenizálómmentes Charformer-modelleknek a huBERT-énél valamivel rosszabb teljesítménye nem igazán okozott meglepetést. Egyrészt a huBERT egy nagyon jól tanított modell, másrészt az irodalomban korábban vizsgált esetekben (Tay és mtsai, 2021; Clark és mtsai, 2021) is a tokenizálómmentes modellek valamivel mindig a fixen töredékesített társaik mögött végeztek. Cserébe viszont – mint ahogy az 1. táblázatban is látható – gyorsabb és kisebb modellt kapunk, ami maga alakítja ki a befoglaló kontextus függvényében a szótöredékeket. A fixen tokenizált ELECTRA-val összevetve, amely azonos körülmények között lett tanítva, mint a Charmer ELECTRA, sikernek könyvelhető el a teljesítménye.

## 7. Charformer belső reprezentációja

A Charformer blokk által kialakított belső tokenizáció, valamilyen szinten visszafejthető a modellből. Amennyiben visszatekintünk az (1) egyenletre, észrevehetjük, hogy  $\hat{P}$  a pozíciók egyes blokkokhoz tartozási valószínűségét reprezentálja. Ennek a  $\hat{P}$  mátrixnak a reprezentációját láthatjuk a 4. ábrán. Ezt a mátrixot az 1024 hosszú bemeneti szekvenciahosszal, 6-os blokkmérettel, és 4-es aluminta-

vételezési faktorral rendelkező Charmen ELECTRA modell adta. Látható, hogy a modell a byte-szintű reprezentációt preferálja, ami nem feltétlen megfelelő. Az előtanítás során byte-szinten maszkoltunk, és a diszkriminátor is byte-szinten hozott döntést. Valószínűleg ennek köszönhetően alakult ki ez a klaszterezettség.

### 7.1. Charformer alapú tokenizálás

Amennyiben a  $\hat{P}$  mátrixot átmeneti valószínűségeknek tekintjük, úgy képesek vagyunk fix szegmenseket felismerni a szövegben. Ezzel előállítva egy valószínű szegmentációját a bemeneti szövegnek. Erre a Viterbi algoritmust alkalmazzuk, ahol az állapotok valószínűsége az adott korpuszon kiszámított  $n$ -gram valószínűségek lesznek. A mi esetünkben  $n = \{1, 2, 3, 4, 5, 6\}$  mivel 6 blokkunk van.

A fenti modell segítségével létrehozott reprezentáció alapján az említett algoritmussal előállított tokenizálásra az alábbiakban mutatunk be néhány példát:

- 'A toke', 'nizál', 'ás n', 'ehéz!'
- 'A', ' kutya', ' szav', 'unk er', 'edeté', 'r', 'e töb', 'b', ' felt', 'étel', 'ezés', ' van.'
- 'A', ' biol', 'ógia', ' szót', ' elős', 'zör M', 'icha', 'el Ch', 'r', 'istoph', ' Hanov', ' ném', 'et fil', 'ozófu', 's', ' hasz', 'n', 'álta', ' 1766', '-ban ', 'egyik ', 'k', 'önyve', ' cím', 'ében.'

Vegyük észre, hogy a blokkméretek merev korlátozó tényezőként hatnak az egyes szótöredékek kialakulása során. Tehát a Viterbi-algoritmus nem tud hosszabb szótöredéket összerakni, mint például hogy 'ehéz!' (az é karakter 2 byte-ot tesz ki az UTF-8 szerinti kódolása miatt).

## 8. Konklúzió

Cikkünkben bemutattuk egy tokenizálásmentes megközelítését az ELECTRA modellnek. Bár teljesítménye nem éri el a huBERT modellt, tanítása a huBERT tanításához szükséges erőforrásigényének töredéke volt csupán. Így jóval kisebb és gyorsabb modellt kaptunk, amely az azonos körülmények között tanított merev tokenizáláson alapuló ELECTRA modellnél jobban teljesít. Kutatásunk további eredménye, hogy a huBERT modell finomhangolásával az Opin-HuBank adatbázison a korábbi legjobb eredményeket meghaladó osztályozási teljesítményt kaptunk.

## Köszönetnyilvánítás

A dolgozatban szereplő kutatási eredmények létrejöttét az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- Ács, J., Lévai, D., Nemeskey, D.M., Kornai, A.: Evaluating contextualized language models for hungarian. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 15–28. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021), [http://acta.bibl.u-szeged.hu/73354/1/msznykonf\\_017\\_015-028.pdf](http://acta.bibl.u-szeged.hu/73354/1/msznykonf_017_015-028.pdf)
- Clark, J.H., Garrette, D., Turc, I., Wieting, J.: CANINE: Pre-training an efficient tokenization-free encoder for language representation (2021), <https://arxiv.org/abs/2103.06874>
- Clark, K., Luong, M.T., Le, Q.V., Manning, Ch.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: International Conference on Learning Representations (2020), <https://openreview.net/forum?id=r1xMH1BtvB>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020), <https://www.aclweb.org/anthology/2020.acl-main.747>
- Conneau, A., Lample, G.: Cross-lingual language model pretraining. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E.B., Garnett, R. (szerk.) NeurIPS. pp. 7057–7067 (2019), <http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining>
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://aclanthology.org/N19-1423>
- Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang Zijian, Gy., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021), <http://real.mtak.hu/120856/1/feldmann21.pdf>
- Gong, C., He, D., Tan, X., Qin, T., Wang, L., Liu, T.Y.: FRAGE: frequency-agnostic word representation. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (szerk.) NeurIPS. pp. 1341–1352 (2018), <http://papers.nips.cc/paper/7408-frage-frequency-agnostic-word-representation>
- Hangya, V., Farkas, R., Berend, G.: Entitásorientált véleménydetekció webes híryananyagokból. In: XI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2015), [http://acta.bibl.u-szeged.hu/58936/1/msznykonf\\_011\\_227-234.pdf](http://acta.bibl.u-szeged.hu/58936/1/msznykonf_011_227-234.pdf)
- Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., Johnson, M.: XTREME: A massively multilingual multi-task benchmark for evaluating cross-

- lingual generalisation. In: III, H.D., Singh, A. (szerk.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 4411–4421. PMLR (13–18 Jul 2020), <https://proceedings.mlr.press/v119/hu20b.html>
- Kudo, T.: Subword regularization: Improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 66–75. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), <https://aclanthology.org/P18-1007>
- Ma, W., Cui, Y., Si, C., Liu, T., Wang, S., Hu, G.: CharBERT: Character-aware pre-trained language model. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 39–50. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020), <https://aclanthology.org/2020.coling-main.4>
- Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2013), [http://acta.bibl.u-szeged.hu/58859/1/msznykonf\\_009\\_343-345.pdf](http://acta.bibl.u-szeged.hu/58859/1/msznykonf_009_343-345.pdf)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020), [https://hlt.bme.hu/en/publ/nemeskey\\_2020](https://hlt.bme.hu/en/publ/nemeskey_2020)
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). pp. 3–14. Szeged (2021), [http://acta.bibl.u-szeged.hu/73353/1/msznykonf\\_017\\_003-014.pdf](http://acta.bibl.u-szeged.hu/73353/1/msznykonf_017_003-014.pdf)
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), <https://aclanthology.org/N18-1202>
- Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108 (2019), <http://arxiv.org/abs/1910.01108>
- Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: ICASSP. pp. 5149–5152. IEEE (2012), <https://doi.org/10.1109/ICASSP.2012.6289079>
- Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany (Aug 2016), <https://aclanthology.org/P16-1162>
- Svenstrup, D., Hansen, J.M., Winther, O.: Hash embeddings for efficient word representations. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (szerk.) NIPS. pp. 4928–4936 (2017), <http://papers.nips.cc/paper/7078-hash-embeddings-for-efficient-word-representations>

- Tay, Y., Tran, V.Q., Ruder, S., Gupta, J., Chung, H.W., Bahri, D., Qin, Z., Baumgartner, S., Yu, C., Metzler, D.: Charformer: Fast character transformers via gradient-based subword tokenization (2021), <https://arxiv.org/abs/2106.12672>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc. (2017), <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. pp. 1218–1227. Chinese Information Processing Society of China, Huhhot, China (Aug 2021), <https://aclanthology.org/2021.ccl-1.108>



# HuSpaCy: an industrial-strength Hungarian natural language processing toolkit

György Orosz, Zsolt Szántó,  
Péter Berkecz, Gergő Szabó, Richárd Farkas  
gyorgy@orosz.link  
{szantozs,rfarkas}@inf.u-szeged.hu

Institute of Informatics, University of Szeged  
2. Árpád tér, Szeged, Hungary

**Abstract.** Although there are a couple of open-source language processing pipelines available for Hungarian, none of them satisfies the requirements of today’s NLP applications. A language processing pipeline should consist of close to state-of-the-art lemmatization, morphosyntactic analysis, entity recognition and word embeddings. Industrial text processing applications have to satisfy non-functional software quality requirements, what is more, frameworks supporting multiple languages are more and more favored. This paper introduces HuSpaCy, an industry-ready Hungarian language processing toolkit. The presented tool provides components for the most important basic linguistic analysis tasks. It is open-source and is available under a permissive license. Our system is built upon spaCy’s NLP components resulting in an easily usable, fast yet accurate application. Experiments confirm that HuSpaCy has high accuracy while maintaining resource-efficient prediction capabilities.

## 1 Introduction

Basic natural language processing tasks such as tokenization, sentence splitting, part-of-speech tagging, lemmatization, dependency parsing and named entity recognition are amongst the most widely studied problems in natural language processing. Several text analysis applications have been developed during the last decades for both English and other less-resourced languages such as Hungarian. However, a large majority of them solely focus on achieving high scores on artificial benchmarks and ignore the importance of practical usability.

In this paper we introduce HuSpaCy, an industry-strength Hungarian text processing pipeline capable of parsing and tagging texts with high accuracy on limited computational resources. Our system is built upon spaCy’s<sup>1</sup> NLP components, which means that it is fast, has a rich ecosystem of NLP applications and extensions, comes with extensive documentation and a well-known API.

<sup>1</sup> <https://spacy.io/>

First, we give an overview of the underlying models, then rigorous evaluation is presented using various datasets. Finally, experiments are presented confirming that HuSpaCy has high accuracy in many subtasks while maintaining resource-efficient prediction capabilities.

## 2 Background

### 2.1 Demands for a language processing pipeline in the 2020s

Starting from the release of the Penn Treebank (Marcus et al., 1993) in 1992, the research community developed language processing tools for particular tasks, like tokenization, part-of-speech tagging etc. These tools are usually run in a sequence and form a pipeline. In the 2000s, many language-specific corpora and treebanks were developed along with such pipelines. Hungarian was among the best supported languages (Simon et al., 2012) ten years ago.

In the early 2010s, Universal PoS (Petrov et al., 2012) and Universal Dependency (Nivre et al., 2016) labeling schemata were developed with the goals of "cross-linguistically consistent treebank annotation for many languages" and "facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective." Many language-specific pipelines changed their representations to these universal annotation schema, but most of them stayed in their own software architecture. Industrial NLP applications are frequently multi-lingual, i.e. the same NLP task has to be solved in several languages. The demand for standardization over languages is high in commercial partners. Beyond universal PoS and dependency annotations, companies who are not NLP experts but want to apply language processing tools prefer multilingual software frameworks and make business decisions to support Hungarian, based on the availability in multilingual frameworks.

The last five years of NLP are dominated by neural language models (NLM) and the applications based on them (Young et al., 2018). Academic research has introduced various deep learning methods outperforming previous state of the art in many areas. Such systems usually employ a single neural network providing end-to-end NLP solutions without the need for specific pipeline steps. Also, several pre-trained multilingual NLMs are becoming available which provide standardized solutions for many languages. Regardless of the multilinguality and the high accuracy of deep learning solutions, a lot of critiques have been raised by real-world industrial NLP projects recently.

These are as follows: deep learning solutions often require far more computational resources compared to classic solutions. They heavily rely on GPU acceleration along with significant memory consumption. What is more, their running costs are usually 10 or even 100 times higher than that of alternative solutions. These drawbacks are questioning the commercial return of the accuracy gain. We also note that modern NLP pipelines consist of static word embedding representations and use deep learning for individual pipeline steps as well, hence, the advantage of large neural end-to-end systems might be very small.

Another industrial demand about language processing systems is to provide human-readable output. Most of the industrial applications are fully or partially rule-based solutions, as (enough) training data for a pure machine learning solution is not available. And there is no free lunch! Each and every real-world application has its own requirements. Rule-based components of these real-world applications require language-specific representations which can be used for defining rules. Such human-readable representations consist of tokens, lemmata, part-of-speech tags, morphological features, dependency parse trees and named entities. Static word embeddings are often integral parts of industrial applications, as many practical algorithms (e.g. semantic textual similarity methods) heavily rely on them.

## 2.2 Requirements for industrial-strength language processing pipelines

Considering decades of experience of practical NLP applications, developing an “industrial-strength” text processing system is a challenging task. First of all, such a tool should tackle the most important text preprocessing tasks including tokenization, sentence splitting, PoS tagging, lemmatization, dependency parsing, named entity recognition and word embedding representation.

Next, the application has to be accurate enough for real world scenarios while it should be resource conscious at the same time. Furthermore, an industry focused system should be developer friendly, customizable and easy-to-integrate, as NLP modules are integral parts of a larger system in practical applications. These requirements imply that solid documentation should be available as well. Moreover, it is often desired that the underlying machine learning model(s) should be reproducible and controllable.

Last but not least, modern NLP applications are usually multilingual, thus compatibility with international annotation standards (Petrov et al., 2012; Nivre et al., 2016; de Marneffe et al., 2021; McCarthy et al., 2020) is necessary. Moreover, it is also preferred to be easily usable through a well-known multilingual toolkit.

## 2.3 The landscape of Hungarian language processing pipelines

Up until recently, only a few text processing applications were focused on meeting these criteria even for English. When spaCy was released in 2015 (Honnibal, 2015), it was one of the first tools targeting industrial applications mainly. Its authors created an unprecedented tool which offers near state-of-the-art accuracy while being an order of magnitude faster than other tools available. SpaCy also comes with an intuitive API, has detailed documentation, and also fits well into the Python ecosystem of machine learning tools. What is more, it is easily deployable and offers built-in syntax and entity visualization tools as well.

The landscape of the Hungarian text processing systems is similar to that of English before the “industrial NLP revolution”. There are a number of standalone

text analysis tools<sup>2</sup> (Simon et al., 2012) capable of performing individual text processing tasks, but they often do not play well with each other.

In contrast, there are only a few attempts at providing industrial Hungarian pipelines. One of them is `magyarlanc` (Zsibrita et al., 2013) which is a Java based system consisting of state-of-the-art pipeline steps, which were adapted and extended from various libraries. It was designed to serve industrial applications, what is more, a lot of effort has been made on software quality, speed, memory efficiency and customizability. It performs tokenization, sentence boundary detection (SBD), PoS tagging, lemmatization and dependency parsing, but lacks entity recognition and word embeddings. It uses the version 1 of the Universal Dependency (UD) annotations. Although the tool is still used in real world commercial applications, it is not maintained for years.

There is only one other attempt to provide a unified framework for Hungarian text processing tools: `emtsv` (Simon et al., 2020; Indig et al., 2019a,b) (and its predecessor `e-magyar` (Váradi et al., 2018; Váradi et al., 2017)) is a result of a multi-institute collaboration project aiming to integrate existing NLP toolkits into a single application. Unfortunately, neither computational efficiency nor developer ergonomics were amongst the main goals of the project. Although `emtsv` can yield Universal morphosyntactic annotations through conversion, it is rather inaccurate. What is more, it is not designed to efficiently deal with word vectors, therefore no such facility is available in the system.

Talking of Hungarian-specific pipelines, we must mention the contenders of the recent multilingual CoNLL text parsing competitions (Zeman et al., 2017, 2018). There were numerous submissions, but Stanza (Qi et al., 2020) and UD-Pipe (Straka, 2018) are by far the most popular freely available off-the-shelf applications. These tools provide morphological and syntactical analysis of raw texts for many languages, but lack entity annotations. Accuracy scores vary across tools, but all of them are limited by the small size of the publicly available UD annotated gold standard corpora.

	NER	Word embeddings	High throughput	Part of a multilingual pipeline	Free for commercial usage
<code>magyarlanc</code>	-	-	✓	-	✓
<code>emtsv</code>	✓	-	-	-	-
UDPipe	-	-	✓	✓	-
Stanza	✓	✓	-	✓	✓

**Table 1.** Hungarian language processing pipelines evaluated with regards of requirements of industrial applicability.

<sup>2</sup> cf. <https://github.com/oroszgy/awesome-hungarian-nlp>

Table 1 summarizes the landscape of the most important Hungarian language processing pipelines and show how they meet the requirements of today’s NLP applications. We must note that `emtsv` does not have any restriction for using it in commercial applications, although some of its most important components have very restrictive licenses (e.g. `emMorph`). All in all, we can say that none of them is easily applicable in industrial settings.

We present `HuSpaCy`, a new industry-ready Hungarian natural language processing toolkit. It provides all the aforementioned basic text processing modules with high accuracy. The underlying models are optimized to be light on memory consumption and CPU usage. The presented tool is open source<sup>3</sup> and is available under the permissive CC-BY-SA-4.0 license. Our system is built on top of `spaCy`’s infrastructure, thus extensive documentation, debugging tools, an ergonomic API and a flourishing ecosystem are already provided.

### 3 HuSpaCy internals

This section introduces the NLP algorithms behind the presented tool. As our system is built on `spaCy`’s architecture, we mainly relied on its symbolic and ML-based text processing infrastructure. The following paragraphs give a high-level overview of the framework utilized and also describes the contributions of this work.

#### 3.1 Tokenization

`HuSpaCy` builds on `spaCy`’s (Honnibal, 2021) tokenization infrastructure which works as follows: first the input text is split on whitespaces, then token boundaries are identified by splitting prefixing or suffixing character sequences. To make this algorithm viable for Hungarian, we extended it with language specific prefix and suffix splitting rules. Furthermore, we had to deal with the ambiguity of tokens around full stops, thus an extensive abbreviation list has been incorporated to increase the module’s accuracy. During this process we mostly relied on the test cases of `HunToken` (Németh and Zséder, 2013) to fine-tune the algorithm.

#### 3.2 Morphosyntactic tagging, sentence splitting and parsing

Sentence boundaries, dependency parse trees, PoS tags, and the corresponding morphosyntactic features are predicted by a multitask deep learning model of the underlying NLP framework. `SpaCy`’s machine learning approach can be summarized as ”embed, encode, attend, predict” (Honnibal, 2016) which our system adapts for its tagging and parsing components.

<sup>3</sup> <https://github.com/huspace/huspace>

Tokens are embedded using the concatenation of static (pretrained) word vectors and ones learned during the task-specific training process. We use a publicly<sup>4</sup> available 300d word embedding which has been trained on the Hungarian Webcorpus (Halácsy et al., 2004) and a snapshot of the Hungarian Wikipedia with CBOW methodology (Mikolov et al., 2013). Task specific word vectors are 256 wide consisting 64 dimensional embeddings of the tokens' prefixes, suffixes, shapes and the lowercase forms. To make such computation efficient, feature hashing is extensively applied to all kinds of input strings.

During the encoding part, vectors are passed through a four deep stacked CNN encoder (Lecun et al., 1998) which uses residual connections and is accompanied with maxout pooling<sup>5</sup> (Honnibal, 2017). Efficient prediction is guaranteed by the underlying greedy tagger consisting only of a linear and a softmax layer. As for the dependency parsing, an arc-eager transition system (Honnibal et al., 2013) is utilized, which shares weights with the tagger model through multitask learning.

Finally, sentence boundary recognition is formalized as a sequence tagging problem where tokens are tagged with a binary label indicating the first token of a sentence. This component is an integral part of the multitask architecture, thus it also shares its neural model with the parser and the morphosyntactic tagger.

### 3.3 Lemmatization

SpaCy's default lemmatization model is mainly designed for English. It is not suitable for morphologically complex languages such as Hungarian as it only uses lookup tables. Hence, we decided to look for a more sophisticated solution and adapted the Lemmy toolkit (Kristiansen, 2019) which is an open-source Python implementation of the CST rule-learning engine (Jongejan and Dalianis, 2009). To improve its accuracy we incorporated three minor modifications. First, prefixing numbers of numeric tokens are masked to help the engine in case of inflected numbers. (For example the masked token of '2021-ben' becomes '0000-ben'.) Second, we enforce lowercasing of sentence starting tokens if they are not proper nouns. Finally, if there are multiple lemma candidates available for a given (word, tag) pair, we pick the one with the highest frequency on the training dataset.

### 3.4 Named entity recognition

SpaCy's entity recognizer is built on the transition-based parser architecture described in Section 3.2 (similarly to Lample et al. (2016)). However, there are two key differences compared to the system of Lample et al. (2016). The first is that the set of possible transition actions reflects the BILOU tagging scheme.

<sup>4</sup> [https://github.com/orozgy/hunlp-resources/releases/tag/webcorpuswiki\\_word2vec\\_v0.1](https://github.com/orozgy/hunlp-resources/releases/tag/webcorpuswiki_word2vec_v0.1)

<sup>5</sup> The pooling step is considered to be the "attention" mechanism.

This trick allows the model to have better discrimination ability between different entity classes, furthermore it makes the learning problem easier. Second, the state vector computation includes clues not just from the surrounding words but the tokens of previous entities as well. The sequence tagger model uses BILOU tags for encoding entity boundaries and the decoder is built on a greedy softmax layer similar to that of the morphosyntactic tagger.

## 4 Experiments and results

### 4.1 Text parsing

In order to benchmark HuSpaCy, we performed a series of experiments comparing its performance with the most popular off-the-shelf pipelines available. Evaluation is carried out on the test set of the Hungarian Universal Dependencies Corpus (de Marneffe et al., 2021) by using the evaluation script of the CoNLL 2018 Shared Task<sup>6</sup>.

Three popular text processing tools have been selected for comparison. `emtsv` is a Hungarian specific pipeline integrating state-of-the-art NLP components, UDPipe is used as a baseline system in CoNLL competitions, while Stanza has high scores on parsing UD corpora. All systems are used as black boxes meaning they have not been retrained or fine-tuned.

Up until now, there has only been a single Hungarian corpus (Csendes et al., 2004) having both morphosyntactic and dependency parse annotations. What is more, UD annotations are available only in a rather small subcorpus of it (de Marneffe et al., 2021). As PoS and morphosyntactic labels can be transcribed automatically from the Hungarian-specific formalism to UD with high accuracy, additional silver standard data can be utilized to train taggers.

In case of HuSpaCy, we applied a two-step learning strategy<sup>7</sup> to best utilize all available training data. In the first step, the tagger and the SBD components are pre-trained on the whole transcribed SZC<sup>8</sup>. This is followed by a fine-tuning step on the gold standard UD dataset where dependency annotations are also learned. To allow fair comparison with Stanza and UDPipe, a single step model relying solely on the UD data is also involved in the evaluation.

For similar reasons, the lemmatizer has been trained with two configurations. First we used only the training set of the Hungarian UD corpus, then we allowed the tool to learn from the whole transcribed Szeged Corpus (except the sentences overlapping with either our test or development sets).

<sup>6</sup> [https://universaldependencies.org/conll18/conll18\\_ud\\_eval.py](https://universaldependencies.org/conll18/conll18_ud_eval.py)

<sup>7</sup> Hyperparameters of the models are available in the tool's repository (tag v0.4.2) as configuration files.

<sup>8</sup> When we refer to the Szeged Corpus as a training set, we mean all the sentences that are not part of the development or the test set of the Universal Dependencies corpus.

The authors of Stanza<sup>9</sup> and UDPipe<sup>10</sup> have already published their tools' accuracy on the Hungarian UD corpus, however the same is not true for `emtsv`. To evaluate the latter toolkit we used the following (default) configuration to produce an UD-compatible output: `emToken`, `emMorph`, `emLem`, `emTag`, `emmorph2ud`, `emDep`, `emConll`. While `emtsv` can provide parse trees, their annotation schema is not compatible with that of the Universal Dependencies, hence, its output is not evaluable. We must also note that comparison with the `emtsv`'s tagger and lemmatizer might not be fair, as this tool was trained on a different train-test split which might conflict with ours. (There is a high chance that its training data overlaps with the sentences of our test set.)

	Tokenization	Sentence splitting
Stanza	99.87%	97.00%
UDPipe	99.80%	95.90%
<code>emtsv</code>	99.77%	98.67%
HuSpaCy (UD)		97.66%
HuSpaCy (SZC)	99.89%	97.54%

**Table 2.** Tokenization and sentence boundary detection F1 scores on the test of the Hungarian UD Corpus.

F1 scores in Table 2 suggest that tokenization is easily handled by all of the systems, although HuSpaCy is marginally better compared to the rest of the tools. Sentence boundary detection is a more complex task, where language specific knowledge is necessary. This can be either built into the system (as it is the case with `emtsv`) or learned by a ML model. Numbers show that both approaches can yield satisfactory SBD components, although the rule-based solution of `emtsv` stands out followed by the tagging approach of our pipeline.

	PoS acc.	Morph. acc.	UAS	LAS
Stanza	96.03%	93.76%	83.62%	78.86%
UDPipe v1	90.60%	88.50%	72.80%	67.20%
<code>emtsv</code>	89.19%	89.12%	–	–
HuSpaCy (UD)	94.70%	89.03%	79.03%	73.17%
HuSpaCy (SZC)	96.58%	93.23%	79.39%	74.22%

**Table 3.** Comparison of tagging accuracy and attachment scores of the benchmarked pipelines on the test set of the Hungarian UD Corpus.

<sup>9</sup> <https://stanfordnlp.github.io/stanza/performance.html>

<sup>10</sup> <https://ufal.mff.cuni.cz/udpipe/1/models>



Tagging accuracy and attachment scores are presented in Table 3. Results show that Stanza is a clear winner in dependency parsing while the PoS tagging score of HuSpaCy (the one using additional training data) is the highest one. It can be seen that the usage of the extra silver standard data yields better performance for our models both during tagging and dependency parsing. UDPipe and `emtsv` have relatively low scores: the results of UDPipe are not surprising (cf. Zeman et al. (2018)), but `emtsv`'s scores are unexpected given that it is built upon state-of-the-art morphosyntactic tagging facilities (Orosz and Novák, 2013).

	Accuracy
Stanza	94.25%
UDPipe v1	88.50%
<code>emtsv</code>	94.94%
HuSpaCy (UD)	94.82%
HuSpaCy (SZC)	95.53%

**Table 4.** Lemmatization accuracy of NLP pipelines measured on the test set of the Hungarian UD Corpus. HuSpaCy (UD) uses the same setting as its contenders, while HuSpaCy (SZC) builds on the whole Szeged Corpus for training.

Lemmatization results in Table 4 show that all the systems except UDPipe are accurate enough. HuSpaCy trained on the full Szeged Corpus stands out, its score is more than 0.5% higher than the second best system (`emtsv`). The best configuration of HuSpaCy scores more than 0.5% higher than the one trained solely on the UD dataset.

## 4.2 Named entity recognition

Comparing NER components is not as straightforward as it is for the parsing subtasks. There are multiple evaluation datasets, but there is no consensus between researchers on their usage. NYTK-NerKor (NerKor) (Simon and Vadász, 2021) is a relatively new corpus consisting of 1 million tokens, while SzegedNER (Szarvas et al., 2006a) is a 200,000 token subset of the Szeged Corpus. Simon et al. (2022) uses the former dataset to benchmark some of the most popular tools, while previous work mainly rely on the latter one. UDPipe does not have a NER component, thus we cannot include it in this investigation. As for `emtsv`, its entity recognizer was trained using the whole SzegedNER corpus, its comparison against other tools would not be fair.

HuSpaCy's entity recognition capabilities are benchmarked in this work on both corpora using the same train-test splits as Szarvas et al. (2006b) and Simon et al. (2022) suggest. As Hungarian entity recognition datasets share the same

tagset and rely on similar annotation guides, it is possible to train models using both corpora. In this regards, we follow the work of [Simon et al. \(2022\)](#) and evaluate HuSpaCy on the combined corpus as well. We also include results of previous entity recognition attempts so as to put our results in context. One of the first systems was developed by [Szarvas et al. \(2006b\)](#), which utilizes decision trees for tackling the problem. Next, there is HunTag ([Recski and Varga, 2009](#); [Simon, 2013](#)), which is a statistical tagger utilizing a linear model combined with Hidden Markov models. [Simon \(2013\)](#) also showed that it is possible to improve on the F1 score of the base system by incorporating silver standard data. Most recently, [Nemeskey \(2020a\)](#) developed an entity recognizer on top of Hungarian BERT models ([Nemeskey, 2020b](#)) achieving state-of-the-art results.

	SzegedNer	NerKor	Combined
<a href="#">Simon (2013)</a>	95.06%	–	–
<a href="#">Szarvas et al. (2006b)</a>	94.77%	–	–
<b>emBERT</b>	97.40%	92.09%	92.99%
Stanza	91.78%	80.53%	83.75%
HuSpaCy	95.31%	80.75%	83.46%

**Table 5.** Comparison of entity recognition F1 scores on the SzegedNER test set ([Szarvas et al., 2006b](#)), on the NerKor test set and on the combined test.

Table 5 contains F1 scores of all the entity recognizers mentioned above. It can be seen that the BERT-based model achieves the highest scores on all of the datasets with a large margin. However, these models are also well-known for their enormous computational costs. HuSpaCy is the second best contender on SzegedNER, although its performance is on par with Stanza on other datasets. **emBERT**'s results are outstanding when NerKor is involved in the comparison. As [Simon et al.](#) concludes these measurements are in accordance with similar English NER benchmarks (cf. [Qi et al. \(2020\)](#)). Pretrained transformer-based models often yield significantly higher performance scores compared to other sequence tagging approaches due to the underlying attention mechanism and the their model's increased capacity. But there is no free lunch, higher accuracy comes with significantly increased prediction costs.

The final model of HuSpaCy builds on the weights of a pretrained neural tagger (using the strategy described in Section 4.1) yielding 84.56% F1 on the combined dataset. This result is a significant improvement compared to Stanza's score and also confirms the usefulness of additional silver standard training data usage for spaCy's multitask neural model.

### 4.3 Resource usage

Resource usage such as memory consumption and processing speed is an important aspect of practical text processing systems, thus we benchmarked<sup>11</sup> text parsing pipelines (cf. Section 4.1) in this respect. In order to have a fair comparison, we configured all systems to perform only tokenization, sentence splitting, PoS tagging, lemmatization and dependency parsing. As timing measurements should ignore model loading times, Stanza and UDPipe were used by their Python interfaces, while `emtsv` was utilized through its REST API. We used the UD test set to measure throughput and peak memory consumption.

	Throughput (tokens/sec)	Memory usage (GB)
Stanza	222	0.9
UDPipe	1741	0.4
<code>emtsv</code>	122	3.9
HuSpaCy	2612	2.1

**Table 6.** Throughput (measured in tokens/second) and peak memory consumption of benchmarked NLP pipelines.

Table 6 presents computational efficiency measures suggesting that our system has the highest throughput amongst all the tools. HuSpaCy is almost 50% faster than UDPipe, while producing significantly better parses. As regards Stanza, there is a huge tradeoff on having the best dependency parser: it is almost 8 times slower than UDPipe and more than 10 times slower compared to our system.

Memory consumption of the pipelines are acceptable as all of them could easily fit in a modern computer’s RAM. Our tool has the highest memory usage which is due to its 300-dimensional word vectors. In comparison, Stanza is the only other tool having word embeddings, but its vectors’ sizes are limited to 100d.

## 5 Conclusions

We presented HuSpaCy, a new industry-ready Hungarian language processing pipeline that is open source and is freely available. While previous approaches have failed to provide a tool which can be easily used to solve practical text processing problems, our system builds on the solid foundations of an industrial NLP framework. We presented how our toolkit utilizes spaCy’s underlying ML

<sup>11</sup> All experiments were performed on a computer having an Intel Core i7-8750H CPU and 16 GB RAM running Ubuntu Linux 20.04 LTS.

models to provide all the basic language analysis components. We performed various experiments proving that our system has high accuracy in many text processing tasks while using only moderate computation resources.

As results show, the accuracy of HuSpaCy's dependency parser needs further improvements. Further advancement opportunities lie in fine-tuning the NER model and in using a new neural lemmatizer.

In summary, this study described a new freely available tool which is suitable for real-world industrial applications.

## Acknowledgements

The authors would like to thank Dávid Nemeskey and Dániel Lévai for their help in benchmarking `emBERT` and Stanza. HuSpaCy research and development is funded by the Ministry of Innovation and Technology NRD Office within the framework of the Artificial Intelligence National Laboratory Program.

## Bibliography

- Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004. pp. 19–23 (2004)
- Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). European Language Resources Association (ELRA), Lisbon, Portugal (May 2004), <http://www.lrec-conf.org/proceedings/lrec2004/pdf/525.pdf>
- Honnibal, M.: Introducing spaCy (Feb 2015), <https://explosion.ai/blog/introducing-spacy>
- Honnibal, M.: Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models (Nov 2016), <https://explosion.ai/blog/deep-learning-formula-nlp>
- Honnibal, M.: Multi-task cnn for parser, tagger and ner (issue #1057) (May 2017), <https://github.com/explosion/spaCy/issues/1057>
- Honnibal, M.: Tokenization - spaCy Usage Documentation (Nov 2021), <https://spacy.io/usage/linguistic-features>
- Honnibal, M., Goldberg, Y., Johnson, M.: A non-monotonic arc-eager transition system for dependency parsing. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. pp. 163–172. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013), <https://aclanthology.org/W13-3518>

- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – the `emtsv` pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (aug 2019a), <https://www.aclweb.org/anthology/W19-4018>
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` – egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019b)
- Jongejan, B., Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In: Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. pp. 145–153 (2009)
- Kristiansen, S.L.: `lemmy`: Lemmy a lemmatizer for Danish and Swedish (Apr 2019), <https://github.com/sorenlind/lemmy>
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition (2016)
- Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
- Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Comput. Linguist. 19(2), 313–330 (jun 1993)
- de Marneffe, M.C., Manning, C.D., Nivre, J., Zeman, D.: Universal Dependencies. Computational Linguistics 47(2), 255–308 (07 2021), [https://doi.org/10.1162/coli\\_a\\_00402](https://doi.org/10.1162/coli_a_00402)
- McCarthy, A.D., Kirov, C., Grella, M., Nidhi, A., Xia, P., Gorman, K., Vylomova, E., Mielke, S.J., Nicolai, G., Silfverberg, M., Arkhangelskiy, T., Krizhanovsky, N., Krizhanovsky, A., Klyachko, E., Sorokin, A., Mansfield, J., Ernštreits, V., Pinter, Y., Jacobs, C.L., Cotterell, R., Hulden, M., Yarowsky, D.: UniMorph 3.0: Universal Morphology. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3922–3931. European Language Resources Association, Marseille, France (May 2020), <https://aclanthology.org/2020.lrec-1.483>
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
- Nemeskey, D.M.: Egy `emBERT` próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2020). pp. 409–418. Szeged (2020a)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020b)
- Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal Dependencies v1: A multilingual treebank collection. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16). pp. 1659–1666. European Language Resources Association

- tion (ELRA), Portorož, Slovenia (May 2016), <https://aclanthology.org/L16-1262>
- Németh, L., Zséder, A.: huntoken: word and sentence tokenizer (2013), <https://github.com/zseder/huntoken>
- Orosz, G., Novák, A.: PurePos 2.0: a hybrid tool for morphological disambiguation. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013). p. 539–545. INCOMA Ltd. Shoumen, Hissar, Bulgaria (2013)
- Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). pp. 2089–2096. European Language Resources Association (ELRA), Istanbul, Turkey (May 2012), [http://www.lrec-conf.org/proceedings/lrec2012/pdf/274\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf)
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020)
- Recski, G., Varga, D.: A Hungarian NP Chunker. The Odd Yearbook. ELTE SEAS Undergraduate Papers in Linguistics pp. 87–93 (2009)
- Simon, E., Lendvai, P., Németh, G., Olasz, G., Vicsi, K.: A magyar nyelv a digitális korban – The Hungarian Language in the Digital Age. Georg Rehm and Hans Uszkoreit (Series Editors): META-NET White Paper Series, Springer (2012)
- Simon, E.: Approaches to Hungarian Named Entity Recognition. Ph.D. thesis, PhD School in Cognitive Sciences, Budapest University of Technology and Economics (2013)
- Simon, E., Indig, B., Kalivoda, Á., Mittelholcz Iván, S.B., Vadász, N.: Újabb fejlemények az e-magyar háza táján. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–42. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2020)
- Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (eds.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)
- Simon, E., Vadász, N., Lévai, D., Nemeskey, D., Szántó, Z., Orosz, G.: Az NYTK-NerKor több szempontú kiértékelése. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 375–388 (2022)
- Straka, M.: UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 197–207. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), <https://www.aclweb.org/anthology/K18-2020>
- Szarvas, G., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.: A highly accurate named entity corpus for Hungarian. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06). Euro-

- pean Language Resources Association (ELRA), Genoa, Italy (May 2006a), [http://www.lrec-conf.org/proceedings/lrec2006/pdf/365\\_pdf.pdf](http://www.lrec-conf.org/proceedings/lrec2006/pdf/365_pdf.pdf)
- Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In: International Conference on Discovery Science. pp. 267–278. Springer (2006b)
- Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholtz, I., Novák, A., Indig, B., Prószéky, G., Vincze, V.: Az e-magyar digitális nyelvfeldolgozó rendszer. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017). pp. 49–60. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2017)
- Váradi, T., Simon, E., Sass, B., Mittelholcz, I., Novák, A., Indig, B., Farkas, R., Vincze, V.: E-magyar – A Digital Language Processing System. In: chair, N.C.C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., Tokunaga, T. (eds.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 7-12, 2018 2018)
- Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine 13(3), 55–75 (2018)
- Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., Petrov, S.: CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 1–21. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), <https://aclanthology.org/K18-2001>
- Zeman, D., Popel, M., Straka, M., Hajič, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinková, S., Hajič jr., J., Hlaváčová, J., Kettnerová, V., Urešová, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C.D., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.C., Sanguinetti, M., Simi, M., Kanayama, H., de Paiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, Ç., Sulubacak, U., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Alcalde, H.F., Strnadová, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonça, G., Lando, T., Nitisaroj, R., Li, J.: CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 1–19. Association for Computational Linguistics, Vancouver, Canada (Aug 2017), <https://aclanthology.org/K17-3001>
- Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of Recent Advances in Natural Language Processing 2013. pp. 763–771. Association for Computational Linguistics, Hissar, Bulgaria (2013)





# SZINTAXIS



## Igekötő-kapcsolás

Pethő Gergely, Sass Bálint, Kalivoda Ágnes, Simon László, Lipp Veronika

Nyelvtudományi Kutatóközpont

pagstudium@gmail.com

{sass.balint,kalivoda.agnes,simon.laszlo,lipp.veronika}@nytud.hu

**Kivonat** Ahogy ebben a mondatban is látjuk, a magyarban az igekötő el tud válni az igéjétől. A korpuszok alapvető egysége a szó, a token, emiatt a magyar nyelvű korpuszokban hagyományosan mindig külön tokenként jelenik meg az elvált igekötő. Azt az alapvető információt tehát, hogy melyik igéhez tartozik az adott igekötő, ezek a korpuszok nem tartalmazzák. Jelen tanulmányban egyrészt (1) módszert adunk az igekötők alapigéhez kapcsolására, másrészt (2) egy sémát arra, hogy ezt hogyan jelenítsük meg a korpusz annotációjában a korpuszban való keresést leginkább támogató módon. Az eszközt az **e-magyar** rendszer moduljaként implementáltuk, olyan funkcionalitást valósít meg, amelyre számos kutatásban merül fel igény. Az elkészült **emPreverb** modul és a kézzel annotált fejlesztő és tesztkorpuszok szabadon felhasználhatók.

**Kulcsszavak:** igekötő, elvált igekötő, alapige, keresés korpuszban, korpusz, korpusznyelvészet, lexikográfia

### 1. Bevezetés

Korpuszaink alapstruktúrája a legtöbb esetben: szavak sorozata. Ez nagyon leegyszerűsítve szóközökkel elválasztott karaktorsorozatokat sorozatát jelenti. A tokenizálás művelete ezt sok ponton finomítja. A szavakat, írásjeleket, azaz a korpusznak a tokenizáló által szolgáltatott alapegységeit nevezzük tokeneknek.

Amit a tokenizáló sosem szokott megcsinálni, hogy egy tokenként értelmezze két olyan elem együttesét, melyek közé közbeékelődik egy további elem. Ennek következtében tokenizálás után az elvált igekötő külön token lesz, azaz a tapadó igekötős ige egy token lesz, az elvált igekötő és az igéje viszont kettő. Mondhatjuk, hogy ez nem a tokenizáló feladata, ezt majd megoldják a magasabb elemzési szintek, de attól még a tény tény marad, hogy egy morfológiailag elemzett korpuszban nagyságrenddel nehezebb lesz megkeresni egy igekötős ige összes alakját, mint mondjuk egy főnévét.

Valóban, érdemes úgy tekinteni a dolgot, hogy a ‘*megkeres*’ és a ‘*keres meg*’ ugyanannak a lexémának két alakja, azaz gyakorlati szempontból – lexikográfiai és korpusznyelvészeti megfontolásokat szem előtt tartva – a ‘*keres meg*’ is „egy szó”. Azontúl, hogy „van benne szóköz”, nem is nagyon van jó érv amellet, hogy ez két lexéma lenne.

A magyar szótárakban az igekötős igék hagyományosan saját szócikket kapnak. Ezek a szócikkek egyaránt szólnak a tapadó és az elvált alakokról, egyaránt

tartalmaznak olyan példaszövegeket, amelyekben az igekötő elvált az igétől, meg olyanokat, amelyekben nem: mindkét esetre ugyanannak a lexémának az adataként tekintünk. A *magyar nyelv nagyszótára* (Ittész és mtsai, 2006–2021) eddig elkészült kötetében a címszavak 14%-a igekötős ige. Ha pedig az igekötős ige korpusznyelvészeti vizsgálatáról van szó, akkor is beleértik mindkét alaktípust, sokszor éppen ezek eloszlása, használata, fajtái a kutatás tárgya.

Azokban a korpuszvizsgálatokban, amikor nemcsak példákat keresünk, hanem átfogóan, statisztikailag akarunk elemezni egy jelenséget, az alapvető kiindulópont az összes releváns korpuszhelyet tartalmazó találati lista, illetve az ebből készített gyakorisági lista. Az *‘elmegy’* gyakorisági listáján nyilvánvalóan a *‘megy el’*, *‘el sem fog tudni menni’* stb. találatoknak is szerepelniük kell.

Az általunk ismert magyar nyelvű, elemzett korpuszok ennek a követelménynek nem felelnek meg. Bár a probléma az elmúlt évek számos olyan lexikográfiai (Lipp és Simon, 2021) és nyelvészeti projektje kapcsán előkerült, amely a korpuszalapú vagy korpuszvezérelt megközelítést vallotta magáénak, eddig nem született rá átfogó megoldás. Ebben a tanulmányban ezt a megoldást keressük két aspektusból: bemutatunk (1) egy *igei természetű szavak* (igék, igenevek és deverbális főnevek) és hozzájuk tartozó igekötők összerendelésére szolgáló eszközt és (2) egy sémát arra, hogy hogyan érdemes ezt a korpuszban reprezentálni úgy, hogy az igekötős szavakat érintő korpuszalapú munkát minél kényelmesebbé, könnyebbé tegyük. Azt reméljük, hogy eredményeinket széles körben lehet majd hasznosítani. A továbbiakban az igekötőket a [/Prev], az igéket a [/V] kóddal fogjuk jelölni szükség esetén.

## 2. Kapcsolódó munkák

### 2.1. Igekötő-kapcsolás

A magyar nyelvű korpuszokat nézve: az MNSZ-ben (Oravecz és mtsai, 2014) és a TMK-ban (Dömötör és mtsai, 2017) nincsen igekötő-kapcsolás, az Ómagyar korpuszban (Simon és Sass, 2012) – mely a kezdetektől a majdani szintaktikai elemzés igényével készült – viszont van, a feladatot itt manuális munkával oldották meg.

Az első automatikus, szabályalapú módszert a Mazsola adatbázis készítéséhez fejlesztették (Sass, 2011, 38–39. oldal), ez egyszerűen a morfológiai kódok alapján kapcsolja össze az igekötőt és az igét, és azt sem ellenőrzi, hogy volt-e eredetileg igekötő az igealakon. Tudomásunk szerint a Mazsola adatbázisa egyben az egyetlen kutatási célra elérhető nagyméretű magyar nyelvű korpusz, melyben automatikus igekötő-kapcsolás van. Természetesen a Szeged Treebank (Csendes és mtsai, 2005) is tartalmaz – manuális munkával készített – igekötő-kapcsolást. A második módszer (Recski, 2011, 3.1 rész) szintén szabályalapon működik, négy viszonylag triviális szabály alkalmazásával 0,964-es  $F_1$ -pontszámról számol be. Ez a módszer morfológiailag elemzett korpuszon dolgozik, az eredménye révén a szintaktikai elemzéshez kíván hozzájárulni. Ugyancsak szabályalapú módszert alkalmaz a VFRAME eljárás (Vadász és mtsai, 2017, 2018), amely az igekötő-

kapcsolásnak egy részproblémáját oldja meg: a finit ígéhez, valamint az infinitívusozhoz tartozó igekötők megtalálását az ige szűk kontextusában. Novák és mtsai (2019) részfeladatként utalnak a témára, megjegyzik, hogy az elvált esetben az igék lemmája nem tartalmazta az igekötőt, „ezért ezt a hibát is kijavították,” de a hogyanról nem nyilatkoznak. Megemlíjtjük, hogy az eredeti *e-magyar* rendszerben (Sass és mtsai, 2017) létezett egy segédmodul, ami a szintaktikai elemzés alapján kapcsolja az ígéhez az igekötőt. Így elindulva a feladat nyilván triviális.

A németben a magyarhoz hasonlóan létezik elvált igekötő, azaz felmerül a szóban forgó feladat, és az utóbbi időben többen foglalkoztak is vele (Batinić és Schmidt, 2018; Köper, 2018). Volk és mtsai (2016) célkitűzése hasonló a jelen tanulmányéhoz, módszere triviálisabb: egyszerűen az igekötőhöz *balra* legközelebbi finit ígéhez kapcsolják az igekötőt, ha az igekötős ige benne van a szótárban. A németben marginális jelentőségű topikalizált igei partikulát (tehát ami az ígétől balra esik) az egyszerűség kedvéért nem is igei partikulának, hanem határozószónak tekintik; ezek a magyar nyelv esetében nem intézhetők el ilyen könnyen.

## 2.2. Reprezentáció a korpuszban

Az MNSZ (Oravecz és mtsai, 2014) csupán elkülöníti az igekötőtlen igéket az igekötősektől (IGE vs IK.IGE), a TMK-ban (Dömötör és mtsai, 2017) viszont a tapadás a morfológiai kódon túl az ige lemmájában is jelölődik. Az elvált igekötő hovatarozásáról egyik sem ad információt.

Az Ómagyar korpuszban (Simon és Sass, 2012) viszont már megjelenik ez: az alapigénél külön attribútumként szerepel a hozzá tartozó elvált igekötő lemmája. Ezt az információt meglehetősen nehéz előhívni a keresés során, de a törekvésből látszik, hogy mennyire fontos ügy ez – szintaktikai és magasabb szintű elemzésekhez mindenképpen.

A Mazsolában (Sass, 2009) az elvált igekötő az alapige lemmája elé van kapcsolva, ez nagy előrelépés, ugyanakkor ebben a reprezentációban az elválás/tapadás információ elvész: a *‘keres meg’* is egyszerűen *‘megkeres’*-ként jelenik meg. Novák és mtsai (2019) is így jár el, azaz az igekötőt *beleírja* az igelemmába.

Az eredeti *e-magyar* rendszer (Sass és mtsai, 2017) kicsit szofisztikáltabb, de nem feltétlenül jobb megközelítéssel él. Szintén „az igekötő és az igealak szótövének egybeírásaként az igekötős szótövet adja meg elvált esetben is”, de az így létrejött lemmát egy külön attribútumban tárolja (`lemmaWithPreverb`) a lemmától elkülönítve, ami a keresés szempontjából kevésbé jó megoldás.

Volk és mtsai (2016) esetében a módszer lényegében ugyanez, Batinić és Schmidt (2018) azonban továbblép: nemcsak hogy odailleszti az elvált igekötő lemmáját a hozzá tartozó ígéhez, hanem *törli* az igekötő lemmáját az eredeti helyéről. Emögött az a megfontolás húzódhat meg, hogy a korpusz semmilyen részletét ne duplikáljuk az eredetihez képest. Amint látni fogjuk, jelen tanulmányban még tovább lépünk a keresést megkönnyítő explicit annotáció felé.

Elmondhatjuk, hogy még a szintaktikai elemzőkre épülő keresőkben (Zsibrita és mtsai, 2017) sem valósul meg feltétlenül a könnyű kereshetőség. Pusztán attól, hogy az ige-igekötő él be van húzva, nem lehet minden további nélkül egyben kezelni az elvált és a tapadó alakokat.

### 3. Módszer

#### 3.1. Igekötő-kapcsolás

Az igekötő-kapcsolás megvalósításához el lehet indulni nyers, morfológiailag elemzett vagy akár szintaktikailag elemzett korpuszból. Egy olyan gazdag morfológiájú nyelvben, mint a magyar, az elsőt nem látjuk célravezetőnek. Az igekötő-kapcsolásban nagyon gyakoriak az egyszerű esetek (pl. *‘vesszenek össze’*), viszont számos fajta nemtriviális eset is előfordul (pl. *‘el lehetne nem-költői módon is mondani’*). Szintaktikailag elemzett korpuszból kiindulva a feladat triviális lenne – egyetlen dependenciakapcsolatot kell kifejteni –, ehhez azonban fel kell tennünk a szintaktikai elemzés hibátlanágát.

Azt gondoljuk, hogy ezt általában nem tehetjük meg, mert a szintaktikai elemzők főként az egyszerű eseteket kezelik jól. Ha például nagyon távolra kerül egymástól az ige és az igekötő, gyakrabban tévesztenek. Úgy tűnik – és ezt a 4.1. részben olvasható méréseink is alátámasztják –, az igekötő-kapcsolás nem olyan természetű feladat, amiben segít a szintaktikai elemzés. Emiatt, valamint mások hasonló (Sass, 2009; Recski, 2011; Batinić és Schmidt, 2018) megközelítéseire is építve, illetve hogy egy viszonylag kis erőforrásigényű megoldást találjunk, a morfológiailag elemzett korpuszból való kiindulás mellett döntöttünk.

A feldolgozás „irányát” tekintve két lehetőség van. Kívülről befelé esetben a (tag)mondatból kiindulva szűkítjük a szóba jövő tokenek körét, majd mikor már csak egy ige és egy igekötő marad, akkor összekapcsoljuk őket. Belülről kifelé esetben pedig az ige és az igekötő közötti elemeket vizsgáljuk, és ha megfelelő kombinációt találunk, akkor kapcsolunk. Négy ok miatt választottuk az utóbbit: (1) a segédigés esetek (pl. *‘haza akarok menni’*) kezelése könnyebb; (2) nem szükséges külön tagmondatszegmentálás; (3) az *‘el kell, hogy menjek’* típusú példák esetében – legalábbis látszólag – tagmondathatáron átívelő igekötő-ige kapcsolat van; (4) könnyebben kezelhető az árva igekötő, akár ellipsis (*‘Megnézted? – Meg.’*), akár elemzési hiba (*‘meg[/Prev] várnak[/V] se vár’*) miatt fordul elő.

A Volk és mtsai (2016)-hoz hasonló olyan megoldások, amelyek csak adott szótárban előforduló igekötő-ige kombinációkat fogadnak el, a magyarban nem alkalmazhatók. Ennek az az oka, hogy produktív szóalkotással tetszőleges számú igekötős igét hozhatunk létre (pl. *megpirospaprikázzák, lekezítcsokolomoztak*). Ezzel kapcsolatban ld. többek között: Ladányi (2007, 2012); Kalivoda (2021a,b).

A megvalósítás során az **e-magyar** rendszer (Indig és mtsai, 2020) tokenizáló, morfológiai elemző és szófaji címkéző (**tok,morph,pos**) moduljának eredményéből indultunk ki. A módszerünkben egyszerűen azokat a lexikai elemeket tekintjük igekötőnek, amelyek az automatikus morfológiai elemzés során **[/Prev]** címkét kapnak, ezt a címkét nem bíráljuk felül, így a ragos névmásként is értelmezhető igekötők (pl. *‘rá’, ‘hózzá’*) hibás elemzés esetén kimaradnak. (Az igekötő-állomány meghatározásának problémáiról ld. többek között Jakab (1976); Komlósy (1992); Forgács (2005); Kerekes (2011); Kalivoda (2021b) munkáit.) A feladat megoldásakor a gépi tanulás alkalmazását az ehhez szükséges mennyiségű gold standard annotált tanítóadat hiánya eleve kizárta, de egyébként is alkalmasnak látjuk a szabályalapú megközelítést esetünkben, így emellett döntöttünk.

A módszert `xtsv`-modulként implementáltuk. Ennek köszönhetően integráns része lehet a sokak által használt **e-magyar** rendszernek, így szélesebb körben is ismertté és alkalmazhatóvá válhat. Jelenleg a modul `xtsv`-modulként önállóan működik, és tervezzük az **e-magyar** rendszerbe való közvetlen integrációját.

Az algoritmus kialakításának kiindulópontját az MNSZ2-ben (Oravecz és mtsai, 2014) végzett igekötős keresések alkották. A munka első szakaszában iteratívan jártunk el: a minél jobb fedés érdekében minden lépésben a még nem kezelt esetek körében leggyakoribbnak látszó szórendi mintázatot próbáltuk megragadni egy CQL-kifejezéssel. Ha a kapott találatok tisztasága megfelelő volt, az adott keresőkifejezést igekötő-kapcsolási szabállyá fogalmaztuk át; ellenkező esetben finomítottuk a keresőkifejezést további, az `msd` (morfológiai elemzés) és a `word` (szóalak) attribútumokra vonatkozó feltételek hozzáadásával. Egyszerű példával illusztrálva: egy tipikus igekötőre (pl. *‘szét’*) indított keresés találatain jól látszik, hogy ennek a leggyakoribb szórendi mintázata az `ige+igekötő`, pl. *‘szedte szét’*. Erre a mintázatra rákeresve (`[msd="IGE.*"] [word="szét"]`) azonban azt látjuk, hogy a találatokban sok olyan van, jelesül sok *‘volna szét’*, amelyekben az igekötőt nem helyes a tőle közvetlenül balra álló igéhez kapcsolni. Ha a keresési feltételünket ennek megfelelően finomítjuk (`[msd="IGE.*" & word!="volna"] [word="szét"]`), akkor a kapott találatok már eléggé tiszták, így megfogalmazhatjuk azt a kapcsolási szabályt, hogy az igekötővel mindig kapcsoljuk össze a közvetlenül tőle balra álló igét, kivéve, ha az a *‘volna’* token. Amint a megfelelő keresések találati számai mutatják, ez a szabály lefedi adott esetben a *‘szét’* igekötő MNSZ2-beli összes előfordulásának 77%-át úgy, hogy nagyon ritkán eredményez hibás kapcsolást. Ezzel az iteratív eljárással eljutottunk egy általános korpuszpéldákon mérve már „viszonylag jó” pontossággal és fedéssel működő kiinduló algoritmushoz, amelyet az 1. ábra első három pontjában foglaltunk össze.

A második szakaszban új stratégiával javítottuk az algoritmust: mivel az igekötős szórendi mintázatok túlnyomó részét már kezeltük, összeállítottunk egy olyan adathalmazt (ld. 4.1. rész), amely nagy arányban tartalmazott olyan „nehéz” eseteket, amelyeket még nem fedtünk le. Ezt az adathalmazt fejlesztő-validáló adatként használva, az algoritmusunk által itt vétett kapcsolási hibákat (hiányzó és téves kapcsolásokat) elemezve, valamint Kalivoda (2021b) átfogó szórendi elemzéseire támaszkodva bővítettük az algoritmust új, nagyobb fedést biztosító szabályokkal úgy, hogy ezek a szükséges mértékben meg legyenek szorítva, ezáltal a kapcsolat pontosságát ne ronghassák. Ezt az eljárást addig folytattuk, amíg a fejlesztő-validáló adathalmazunkban már csak olyan hibásan kapcsolt esetek maradtak, amelyek kezeléséhez már nem célravezető pusztán morfológiai címkéket használni. A végső algoritmus az 1. ábrán látható. Elsősorban rossz központoszású szövegekben fordul elő, hogy a szabályok alapján egy igekötő egy tőle balra és egy tőle jobbra lévő igéhez is kapcsolható lenne. Ilyenkor a következő szabállyal döntünk: ha balra és jobbra ugyanolyan távolságra van a két ige, akkor a jobbra lévő igéé lesz az igekötő, különben a közelebbié.

A létrejött `emPreverb` modul Linux környezetben futtatható Python-kódja elérhető a <https://github.com/ril-lexknowrep/emPreverb> címen, ahol az algoritmus működése részleteiben tanulmányozható.

Az igei természetű szavakon (igék, igenevek és deverbális főnevek) végighaladva, háromszavas ablakban vizsgálódva a következőket hajtjuk végre.

Az előkészítő lépésben kezeljük az eleve igekötős alakokat:

0. ‘*ráordít*’

Ezután az alábbi szabályoknak megfelelő esetekben kapcsoljuk az igekötőt.

A kiinduló algoritmus szabályai:

1. ‘*meg* kell *próbálni*’
2. ‘*szivárogtatta* *ki*’
3. ‘*tudtam* csak *meg*’

A végső algoritmus további szabályai:

4. ‘*meg* van *győződve*’
5. ‘*be* nem *tartásának*’
6. ‘*meg* lehetett volna *küldeni*’
7. ‘*miért* *szólna* ő ebbe *bele*’
8. ‘*fel* kell, hogy *vállalja*’
9. ‘*be* *kászálódott* az *ülésre*’ (rossz helyesírás kezelése)

Egyéb esetben nincs kapcsolás, nem igekötős az ige.

1. ábra: Az igekötő-kapcsolás algoritmus. Az *igekötők* és az *igék* jelölése. A kiinduló algoritmus a 0–3. lépéseket foglalja magában, a végső az összes szabályt. Az előkészítő lépésben csupán az történik, hogy betoldunk a morfológiai annotáció elejére egy [/Prev] kódot. Erre azért van szükség, mert az *e-magyar* nem jelöli a morfológiai kódban az igekötősséget.

### 3.2. Reprezentáció a korpuszban

Az igekötő-kapcsolás korpuszbeli annotációjának kidolgozásakor a *kényelmes kezelhetőséget* tartottuk szem előtt elsődleges szempontként. Ezt az segíti legjobban, ha minél explicitebben fogalmazzuk meg az egyes információkat az annotációban, akár redundáns módon. A megoldásunk hasonlít Batinić és Schmidt (2018)-éhoz, amennyiben az igekötő lemmáját az alapige lemmája elé írjuk, és az igekötő tokenjéből töröljük az eredeti lemmát. Emellett azonban az általunk javasolt annotáció részletesebb, informatívabb. Az ‘*elő se jön*’ példa esetében az algoritmus (1. ábra, 1. szabály) megállapítja, hogy kapcsolnunk kell az igekötőt az igehez. Nézzük, hogyan jelenik ez meg a kidolgozott annotációs sémában.

1. Az iginél a ‘*jön*’ lemma helyett ‘*előjön*’-t szerepeltetünk: lemma=e1őjön.
2. A morfológiai kód elejére egy [/Prev] kód kerül, ami az igekötősséget jelöli: xpostag=[/V]... → xpostag=[/Prev] [/V]...
3. Az szóösszetételt megjelenítő attribútumba kerül a lemma az igekötőhatár megjelölésével: compound=e1ő#jön
4. A prev nevű új mezőben az iginél elkönyveljük, hogy az igekötő eredetileg tapadó (pfx) volt vagy elvált (sep), illetve az igekötőnél azt, hogy sikerült-e igehez kapcsolni (conn). Példánkban az ‘*elő*’ prev=conn, a ‘*jön*’ prev=sep lesz.
5. A kapcsolt igekötőnek töröljük az eredeti lemmáját (lemma="").
6. Végül az erre szolgáló previd mezőben az igt és az igekötőt is ellátjuk egy közös indexszel: previd=n, ahol n egy folyamatosan növekvő egész szám; a



`prevpos` mezőben pedig feltüntetjük az igekötő igétől tokenben számított pozícióját előjeles formában.

Az annotációt az 1. táblázatban foglaljuk össze. Megjegyezzük, hogy a fejlesztés alatt álló `emCompound` modulról, melynek a feladata az, hogy az összetételei határokat megjelölő annotációt adjon a szavakhoz, jelen tanulmányban nem számolunk be részletesen, ugyanakkor egy verzióját közzétesszük. Most nekünk csak azért fontos, mert ez a modul adja meg az igekötő-ige határt a tapadó esetben, ez után következik az `emPreverb` modul futtatása, ami az összes többi feladatot végzi, beleértve az igekötő-ige határ megjelölését az elvált esetben.

form	lemma	xpostag	compound	prev	previd	prevpos
<b>1.</b>						
eloldozódott	eloldozódik	[/V][...]				
→						
eloldozódott	eloldozódik	[/Prev][V][...]	el#oldozódik	px		
<b>2.</b>						
tér	tér	[/V][...]				
vissza	vissza	[/Prev]				
→						
tér	visszatér	[/Prev][V][...]	vissza#tér	sep	7	+1
vissza	∅	[/Prev]	∅	conn	7	

**1. táblázat.** Az igekötő-kapcsolás korpuszannotációjának bemutatása. Az eredeti korpuszpélda, majd → után az igekötő-kapcsolás utáni állapot látható. Az igekötő-kapcsolás során az ismertetett `compound`, `prev`, `previd` és `prevpos` mezőkkel bővül az annotáció. Az első példa a tapadó igekötő esete, a második pedig az elválté.

## 4. Kiértékelés

### 4.1. Az igekötő-kapcsolás teljesítménye

Az igekötő-kapcsoló algoritmus kiértékelését különböző tényezők nehezítették:

1. Nem állt rendelkezésünkre előzetesen olyan annotált és *hibátlannak feltételezhető* adathalmaz, amelyet *kifejezetten az igekötő-kapcsolás* problémáira tekintettel állítottak volna össze. Ez azért okoz gondot, mert – amint már utaltunk rá – az igekötő-előfordulások túlnyomó többségének az illesztése majdnem triviális, a legegyszerűbb algoritmusokkal is viszonylag nagy pontossággal megoldható. Az ilyen megoldásokkal nem kezelhető „nehezebb” és egyszersmind ritkább szintaktikai mintázatokban szereplő igekötő-ige párok kapcsolásának sikeressége nehezen mérhető az általános korpuszokon, mivel a nagyszámú „könnyű”, így szempontunkból irreleváns eset elnyomja a kis számú releváns esetet a mérési eredményekben.
2. Mivel az *e-magyar* rendszer által generált morfológiai elemzések szolgálták az igekötő-kapcsoló algoritmus bemenetét, foglalkozni kellett azzal a problémával, hogy ezek az elemzések hibásak lehetnek, így például a rendszer igekötőket tévesen határozószóként (pl. ‘oda’) vagy ragozott főnévként (‘végig’) elemzett az adott szövegkörnyezetben.

3. Mivel nem egyes tokenek, hanem tokenpárok automatikus annotálását kellett kiértékelni, nem teljesen kézenfekvő, hogy mi számítson helyes, illetve helytelen címkzésnek, valamint az utóbbiakat miként értékeljük ki.

Ezeket a problémákat az alábbiak szerint oldottuk meg.

**Tesztadatok** Fontosnak tartottuk, hogy olyan statisztikailag minél reprezentatívabb korpuszt használjunk mind az igekötő-kapcsoló algoritmus fejlesztésére, mind teljesítményének mérésére, amely a megoldandó feladattal kapcsolatban kezelendő szintaktikai mintázatokat a tényleges előfordulásaik arányában tartalmazza. Ezt szem előtt tartva két kis tesztkorpuszt állítottunk össze az MNSZ2-ből véletlen mintavétellel. A minta elemeiként egész mondatokat kértünk le. Ugyan az alábbiakban a tömörség kedvéért mindig igékről, illetve ige-igekötő párokról beszélünk, valójában a tesztkorpuszainkban nemcsak igéket, hanem más igei természetű szavakat (igeneveket és deverbális főneveket) is annotáltunk az igekötőhöz tartozó „ige”-ként, ha az igekötő éppen ilyentől vált el.

1. Egy kb. 500 mondatot tartalmazó „*általános*” véletlen minta, amelyet az [msd="IK"] keresés eredményeként kaptunk, majd úgy szűrtünk, hogy mondatonként pontosan egy ige-igekötő pár szerepeljen benne. Terjedelme mintegy 13000 token. Az ige-igekötő párokat a példák ellenőrzését követően kezel, koindexálással megjelöltük összetartozóként a nyers szövegben.
2. Egy kb. 600 mondatot tartalmazó, mintegy 19000 token terjedelmű „*nehéz*” véletlen minta, amelyet az [msd!=".\*IGE.\*"] [msd="IK"] kereséssel kaptunk. Ez tehát olyan mondatokból áll, amelyek tartalmaznak legalább egy olyan igekötő-előfordulást, amelytől közvetlenül balra nem ige áll. A több igekötő-ige párt tartalmazó mondatokat megtartottuk, és a bennük található összes összetartozó igekötő-ige párt kézzel annotáltuk a nyers szövegben, kivéve a triviálisnak tekintett igekötő-ige párokat, ahol az igekötőtől közvetlenül balra áll az igéje (pl. '*nézett ki*'). Ha a találatként kapott mondatokban szerepelt olyan (kevésbé triviálisan kezelhető) segédigés szerkezet is, amelyet a keresési feltétel alapján nem kaptunk volna találatként (pl. '*nézett volna ki*'), akkor ezekben is összetartozóként jelöltük meg az igekötőt és az alapigét. Az elliptikus szerkezetben szereplő, igéhez nem kapcsolható igekötőket külön jelöléssel láttuk el. Mellérendelő szerkezetekben megengedtük, hogy tapadó igekötős igéhez is tartozhasson másik, elvált igekötő (pl. '*át- meg átjártá*'); illetve hogy egy igéhez több elvált igekötő tartozzon (pl. '*mutat vissza a város középkori történetére, vissza a ködszerű múltra*'). Az utóbbi esetben az azonos igéhez tartozó igekötők azonos indexet kaptak, ugyanakkor feltételezzük, hogy egy igekötő mellérendelő szerkezetben sem tartozhat több mint egy igéhez.

Mindkét korpusz esetében nagyobb mintából indultunk ki eredetileg, és ebből eltávolítottuk (1) a különbözőképpen hibás vagy szempontunkból irreleváns (pl. ékezetek nélküli, hibásan beszűrt szóközöket tartalmazó, archaikus nyelvezetű vagy írásmódú stb.) mondatokat, amelyek esetében az **e-magyar** morfológiai elemzése vagy szófaji címkzése várhatóan kudarcot vallott volna. (2) Emellett

szintén töröltük az igekötőt valójában nem tartalmazó, azaz az MNSZ2-ben tévesen igekötősként szereplő mondatokat, ahol nem volt mit kapcsolni. (3) Töröltük továbbá a koindexálást azokról az igekötő-ige párokról, amelyek esetében az **e-magyar** szófaji elemzője tévesen nem igekötőként címkézte fel a tényleges igekötőt. Mindezeket a lépéseket az indokolja, hogy a kiértékelés során *célzottan az igekötő-kapcsoló algoritmusok helyességét* akartuk megítélni önmagában, és ennek érdekében irrelevánsként ki akartuk zárni az olyan kapcsolási hibák lehetőségét, amelyek az **e-magyar** elemzésének hibájából fakadnak. Nem ellenőriztük ugyanakkor, hogy az **e-magyar** helyes morfológiai és szófaji címkéket rendel-e a mondat egyéb, különösen az igekötő és az ige közé beékelődött szavaihoz, ugyanis úgy véljük, hogy az igekötő-kapcsoló algoritmusnak lehetőleg elég robusztusnak kell lennie ahhoz, hogy az ilyen elemzési hibák jelenlétében is minél pontosabban működjön.

A kétféle tesztadathalmaz együttes használatát az indokolja, hogy *egyfelől* olyan megoldást keresünk, amely a kapcsolás tekintetében nehéz szintaktikai mintázatokat is helyesen kezeli – ezek csak a „nehéz” halmazon mérhetőek hatékonyan –, *másfelől* az „általános” halmazon történő kiértékelés révén biztosítjuk, hogy az eljárás az egyszerű esetekben gazdag, sima magyar szövegen is működjön.

A „nehéz” eseteket tartalmazó adathalmazt két diszjunkt részre osztottuk. Az egyik részt ( $N = 1292$  igekötővel) az igekötő-kapcsoló algoritmus fejlesztése, a szabályok finomhangolása, hibaelemzése során fejlesztő-validáló halmazként használtuk, míg a másik részt ( $N = 376$  igekötővel) tesztalmazként félretettük, és csak az alább ismertetett mérésekre használtuk. Az „általános” adathalmazt kizárólag tesztalmazként használtuk, a fejlesztés során semmilyen módon nem használtuk fel.

**A teljesítménymérés elvei** Mind a saját algoritmusunk, mind a vele összehasonlításképpen vizsgált egyéb eljárások kiértékelése során az alábbi definíciókat alkalmaztuk. Nem ige-igekötő párokat számoltunk, hanem *igekötőket*. Pontosan azokat a tokeneket tekintettük igekötőnek, amelyeket a tesztadatainkban a fentebb összefoglalt elveket követve kézzel igekötőként jelöltünk meg. Egy igekötőt *pozitívnak* tekintünk, ha az algoritmus kapcsolt hozzá igét. Ezen belül *helyes pozitívnak* tekintettük, amennyiben pontosan azt az igei természetű tokent kapcsolta hozzá, amellyel koindexáltuk a kézi annotáció során, és *hamis pozitívnak*, ha bármely más tokent kapcsolt hozzá. Megfordítva egy igekötőt *negatívnak* tekintünk akkor, ha az algoritmus *nem* kapcsolt hozzá más tokent. Ezen belül *helyes negatívnak*, ha valóban nem kellett hozzá kapcsolni semmit (ez elliptikus szerkezetekben áll fenn), és *hamis negatívnak*, ha a kézi annotáció szerint kellett volna kapcsolni hozzá tokent. Az említett négy mérőszám alapján számítottuk a szokásos pontosság, fedés és  $F_1$  teljesítménymutatókat. Beszámolunk továbbá az accuracy értékről is, ugyanis ezt hasonlóan relevánsnak gondoljuk az adott feladat összefüggésében.

A saját algoritmusunkat összehasonlítottuk két egyszerűbb igekötő-kapcsoló algoritmussal mint baseline-nal:

- **nearest-verb**: Az igekötőket kapcsoljuk mindig a legközelebbi igéhez, *kivéve* a közvetlenül tőlük jobbra álló igéhez.
- **max2**: Minden igére vizsgáljuk meg, hogy van-e tőlük balra pontosan két token távolságban (tehát egy közbeeső tokennel), illetve jobbra pontosan egy vagy két token távolságban igekötő, és ha igen, akkor kapcsoljuk hozzá.

A **nearest-verb** algoritmus hasonlít Recski (2011, 3.1.) baseline-jához, amely egyszerűen a legközelebbi igéhez kapcsolja az igekötőt minden korlátozás nélkül. Úgy ítéltük meg, hogy ez utóbbi mesterségesen rossz baseline, ugyanis az igekötőtől közvetlenül jobbra álló igéhez a helyesírási normát viszonylag követő korpuszokban (mint amilyen az MNSZ2 túlnyomó része) *szinte soha* nem helyes az igekötőt kapcsolni. Ott ugyanis általában csak a szerző által további lépésekben kizárt segédige jellegű igék állhatnak, amelyek pedig soha nem az igekötő alapigéi (pl. ‘*ki tudja nyitni*’; ‘*be lenne zárva*’). Legegyszerűbbnek tehát azt láttuk, hogy a **nearest-verb** baseline-t a fenti nagyon egyszerű korlátozó feltétellel alkalmazzuk.

A **max2** baseline alapötlete, hogy az igétől közvetlenül balra – az imént kifejtett okból – *szinte soha* nem áll az igekötője, és – amint Kalivoda (2021b, 14., 15., 18., 20. és 21. táblázat) összegzi – az igekötő az összes eset elenyésző részében távolodik el az igéjétől több mint két token távolságra.

A két baseline mellett algoritmusunkat összehasonlítottuk az **e-magyar** rendszer **emStanza** moduljának outputjával is, amely a szintaktikai elemzés részeként (tehát ellentétben a mi módszerünkkel és a két baseline-nal nem pusztán morfológiai információkra támaszkodva) kapcsolja az igekötőket az igéjükhöz. Ennek kapcsán értelemszerűen arra a kérdésre kerestük a választ, hogy a szintaktikai elemzéssel megbízhatóan jobb igekötő-kapcsolást kapunk-e, mint anélkül. A Stanza (Qi és mtsai, 2020) szintaktikai elemző outputjában az igekötő-kapcsolás a **compound:preverb** függőségi reláció formájában közvetlenül megjelenik, így ez az információ további feldolgozás nélkül rendelkezésre áll, és pontossága kézenfekvően kiértékelhető a fenti elveket követve.

		a) auto címke				b) auto címke			
		pozitív	negatív			pozitív	negatív		
annotáció	helyes	489	0	489	helyes	346	6	352	
	hamis	1	10	11	hamis	3	21	24	
		490	10	500			349	27	376

**2. táblázat.** Az a) „*általános*”, illetve a b) „*nehéz*” esetek konfúziós mátrixa.

**A mérés eredményei** A két tesztadathalmaz mérete  $N_{\text{általános}} = 500$ , illetve  $N_{\text{nehéz}} = 376$ . Algoritmusunk értékeléséhez érdekes lehet egy pillantást vetni az „*általános*” és „*nehéz*” esetek konfúziós mátrixára (2. táblázat). A kiértékelés eredményei a 3. táblázatban láthatók.

**Összefoglaló észrevételek** Amint az algoritmusunk (vö. 1. ábra) ismertetése kapcsán hangsúlyoztuk, kiemelt célkitűzésünk volt a minél magasabb pontosság biztosítása. A pontosság előtérbe helyezése az eredményekben jól tükröződik, az **emPreverb** ebben a tekintetben mindkét tesztadathalmazon látványosan jobb

	pontosság (%)	fedés (%)	$F_1$	accuracy (%)	pontosság (%)	fedés (%)	$F_1$	accuracy (%)
emPreverb	<b>99,80</b>	98,00	<b>0,9889</b>	<b>97,80</b>	<b>99,14</b>	<b>94,28</b>	<b>0,9665</b>	<b>93,62</b>
nearest-verb	96,56	<b>98,76</b>	0,9765	95,40	76,62	93,79	0,8434	73,14
max2	97,94	96,94	0,9744	95,00	92,28	79,48	0,8540	75,00
emStanza	96,04	90,66	0,9328	87,40	90,83	89,24	0,9003	81,91

(a) Eredmények az „*általános*” mintán. (b) Eredmények a „*nehéz*” mintán.

### 3. táblázat. A vizsgált algoritmusok eredményei.

eredményt ér el a három másik megoldáshoz képest. A fedés nem közelíti ezt meg, de összességében nem rosszabb az alternatívákhoz képest. Vegyük észre, hogy a **nearest-verb** baseline fedése triviális módon lesz magas, mivel ez a módszer minden igekötőhöz igyekszik kapcsolni valamit, bármilyen messze is legyen tőle. Ez persze a pontosság rovására megy, ami összességében alacsonyabb  $F_1$  pontszámot eredményez. Ugyan az „*általános*” számok ezt – a fentebb részletezett okok miatt törvényszerűen – kevésbé adják ki, a „*nehéz*” adatok világosan mutatják, hogy az **emPreverb** meggyőzően jobban teljesít az összes többi megoldáshoz képest.

Figyelemreméltóak a Stanza eredményei: ugyan a „*nehéz*” adatokon messze a két baseline felett teljesít (bár hasonlóan messze elmaradva az **emPreverb** számaitól), az „*általános*” adatokon még a „buta” baseline-okhoz képest is minden szempontból nagyon rosszak az eredményei, így praktikusán nem használható az igekötő-kapcsolás automatikus annotálására korpuszokban.

A <https://github.com/ril-lexknowrep/hungarian-preverb-corpus> oldalon megtalálhatók a kézzel annotált fejlesztő és tesztkorpuszok az annotálási útmutatóval együtt, valamint a teljes kiértékelési környezet, melynek segítségével a fenti kiértékelés reprodukálható.

## 4.2. A reprezentáció hasznossága

Ebben a fejezetben az igekötő-kapcsolás jelen tanulmány keretében kidolgozott korpuszbeli reprezentációjának, annotációjának hasznosságát mutatjuk be példákon keresztül a 3.2. rész pontjai szerint haladva. Javaslatunk a harmadik, negyedik és hatodik pont tekintetében tud többet a Batinić és Schmidt (2018, 1. táblázat) által bemutatott modellhez képest.

1. Talán a legfontosabb eredmény, hogy az igekötős igék *összes* korpuszbeli találatát megkaphatjuk – tapadástól/elválástól függetlenül – egy egyszerű, pusztán a lemmára irányuló CQL lekérdezéssel: `[lemma="e1őjön"]`, és ennek köszönhetően gyakorisági listát is egyszerűen készíthetünk a találatokból.
2. Azáltal, hogy bővítjük az igekötős igék morfológiai kódját a `[/Prev]` elemmel, könnyen hozzájuthatunk az összes igekötőtlen ígéhez: `[xpostag="\[/V\].*"]` akár összes igekötős ígéhez: `[xpostag="\[/Prev\]\[/V\].*"]` ismét csak függetlenül attól, hogy az igekötő elválik-e. Utóbbi kitélt minden további megállapításunkhoz is értsük hozzá.
3. A `[compound="meg#.*" & xpostag="\[/Prev\].*"]` lekérdezéssel megkapjuk az összes ‘*meg*’ igekötős ígét. Megjegyezzük, hogy ennek megvalósulásához a tapadó esetben az **emCompound** modulra, az elvált esetben pedig az **emPreverb** modulra van szükség.
4. A `prev` mezőre támaszkodva lekérdezhetjük adott igekötős ige összes tapadó: `[lemma="e1őjön" & prev="pfx"]` vagy összes elvált: `[lemma="e1őjön" &`

`prev="sep"]` alakját, vizsgálhatjuk az igék elválási hajlandóságát. Általánosan, azaz minden igére egyben, a `[xpostag="\[/Prev\]\[/V\].*"]` lekérdezéssel, majd a belőle `lemma+prev` szerint készített gyakorisági lista révén kaphatjuk ezt meg.

5. Az odakapcsolt igekötőket – az üres lemmának köszönhetően – a `[lemma="" & xpostag="\[/Prev\]"]` lekérdezéssel, az árván maradt igekötőket pedig a `[lemma=".+" & xpostag="\[/Prev\]"]` lekérdezéssel gyűjthetjük össze. Az árván maradt igekötők általában ellipszis vagy hibás morfológiai annotáció – például egy igekötőként azonosított határozószó – eredményei, ahogy erről már volt szó.
6. A `prevpos` mezőből készített gyakorisági lista által megállapíthatjuk az igekötő eltávolodásának eloszlását adott/összes ige és adott/összes igekötő vonatkozásában is.

Ha arra vagyunk kíváncsiak, hogy adott igekötő mennyire szeret elválni, akkor ez a lekérdezés: `[prev="(sep|pfx)" & compound="meg#.*"]` és a `compound+prev` szerint belőle készített gyakorisági lista lesz segítségünkre. Ha pedig úgy tesszük fel a kérdést, hogy a magyar nyelvben általában mennyire szeretnek elválni az igekötők, akkor a lekérdezést a következőre cserélve: `[prev="(sep|pfx)"]` és `prev` szerinti gyakorisági listát készítve azt kapjuk eredményül, hogy 60-65% a tapadó és 35-40% az elvált igekötő – vö. (Kalivoda, 2021b) 157. oldalán szereplő 41,2%-os adattal.

Ahogy korábban utaltunk rá, a könnyű lekérdezhetőséghez az explicit annotáció visz közelebb. Ezt támasztják alá a fentiek. Az annotáció hasznosságához hozzátartozik, hogy az elmondottak mind egy az egyben implementálhatók a NoSketchEngine korpuszkezelő alatt, mi is így jártunk el a fenti lekérdezések kipróbálásakor.

## 5. Össze#foglalás

Tanulmányunkban (1) szabályalapú megoldást adtunk a magyar elváló igekötőknek és az igéjének az automatikus egymáshoz rendelésére, és (2) bemutattunk egy modellt az igekötők és igék viszonyára vonatkozó információknak a korpusz annotációjában való olyan megjelenítésére, mely a korpuszban történő keresést a lehető legnagyobb mértékben megkönnyíti. Válogatott nehéz példák alapján készített automatikus módszerünk teljesítménye ( $F_1 = 0,9889$ ) meggyőzően jobb a korábbiakénál, és a kiértékelés során azt is megmutattuk, hogy a feladat megoldásakor nem érdemes automatikus szintaktikai elemzésre támaszkodni, a morfológiai elemzésre építő szabályalapú módszer jobb eredményt ad. Számos példával illusztráltuk, hogy a javasolt reprezentációs séma milyen új lekérdezési lehetőségekre nyit kaput. Minden eszköz és erőforrás elérhető a <https://github.com/rillexknowrep> oldalról: az igekötő-kapcsoló modul az `emPreverb`, az összetételi határokat megjelölő modul az `emCompound`, az annotált korpuszok és a kiértékelő környezet pedig a `hungarian-preverb-corpus` repozitóriumban. Eredményeink lényegében minden magyar nyelvű lexikográfiai projektben és korpuszalapú nyelvészeti kutatásban hasznosulhatnak a jövőben, ehhez hozzájárulhat az is, hogy az implementáció a közismert `e-magyar` rendszerbe illeszkedik.

## Hivatkozások

- Batinić, D., Schmidt, T.: Reconstruction of Separable Particle Verbs in a Corpus of Spoken German. In: Rehm, G., Declerck, T. (szerk.) *Language Technologies for the Challenges of the Digital Age*. pp. 3–10. Springer International Publishing, Svájc, Cham (2018)
- Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (szerk.) *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD 2005)*. pp. 123–131. Springer LNAI, Berlin, Heidelberg (2005)
- Dömötör, A., Gugán, K., Novák, A., Varga, M.: Kiútkeresés a morfológiai labirintusból – korpuszépítés ó- és középmagyar kori magánéleti szövegekből. *Nyelvtudományi közlemények* 113, 85–110 (2017)
- Forgács, T.: Grammatikalizálódás az igekötők körében. In: Oszkó, B., Sipos, M. (szerk.) *Uráli grammatizáló*. pp. 88–116. MTA Nyelvtudományi Intézet, Budapest (2005)
- Indig, B., Sass, B., Mittelholcz, I.: The xtsv Framework and the Twelve Virtues of Pipelines. In: Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (szerk.) *Proceedings of the 12th Language Resources and Evaluation Conference*. pp. 7044–7052. European Language Resources Association, Marseille (2020)
- Ittzés és mtsai: A magyar nyelv nagyszótára (2006–2021), <https://nagyszotar.nytud.hu>
- Jakab, I.: A magyar igekötők állományi vizsgálata. No. 91 in *Nyelvtudományi Értekezések, Akadémiai Kiadó, Budapest* (1976)
- Kalivoda, Á.: Az igekötők produktív kapcsolódási mintái. *Argumentum* 17, 56–82 (2021a)
- Kalivoda, Á.: Igekötős szerkezetek a magyarban. Doktori (PhD) értekezés, Pázmány Péter Katolikus Egyetem, Bölcsész- és Társadalomtudományi Kar, Nyelvtudományi Doktori Iskola, Budapest (2021b)
- Kerekes, J.: Az igekötők meghatározásának problémái. In: Gécseg, Zs. (szerk.) *LingDok10: Nyelvész-doktoranduszok dolgozatai*. pp. 109–131. Szegedi Tudományegyetem Nyelvtudományi Doktori Iskola, Szeged (2011)
- Komlósy, A.: Régenek és vonzatok. In: Kiefer, F. (szerk.) *Strukturális magyar nyelvtan 1., Mondattan*, pp. 299–527. Akadémiai Kiadó, Budapest (1992)
- Köper, M.: Computational approaches for German particle verbs: compositionality, sense discrimination and non-literal language. Doktori PhD értekezés, Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart, Stuttgart (2018)
- Ladányi, M.: Produktivitás és analógia a szóképzésben: elvek és esetek. Tinta Könyvkiadó, Budapest (2007)
- Ladányi, M.: Igekötős igék kapcsolódási mintázatai. Vizsgálati lehetőségek. In: Tolcsvai Nagy, G., Tátrai, Sz. (szerk.) *Konstrukció és jelentés. Tanulmányok a magyar nyelv funkcionális kognitív leírására*, pp. 71–84. Eötvös Loránd Tudományegyetem, Budapest (2012)

- Lipp, V., Simon, L.: Towards a new monolingual Hungarian explanatory dictionary: overview of the Hungarian explanatory dictionaries. *Studia lexicographica* 15(29), 83–96 (2021)
- Novák, A., Laki, L.J., Novák, B., Dömötör, A., Ligeti-Nagy, N., Kalivoda, Á.: Egy magyar nyelvű kérdezőrendszer. In: XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 83–95. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2019)
- Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014). pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik, Iceland (2014)
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020)
- Recski, G.: A sekély mondattani elemzés további lépései. In: Tanács, A., Vincze, V. (szerk.) VIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011). pp. 113–118. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2011)
- Sass, B.: „Mazsola” – eszköz a magyar igék bővítményszerkezetének vizsgálatára. In: Váradi, T. (szerk.) Válogatás az I. Alkalmazott Nyelvészeti Doktorandusz Konferencia előadásáiból. pp. 117–129. MTA Nyelvtudományi Intézet, Budapest (2009)
- Sass, B.: Igei szerkezetek gyakorisági szótára – egy automatikus lexikai kinyerő eljárás és alkalmazása. Doktori PhD értekezés, Pázmány Péter Katolikus Egyetem, Budapest (2011)
- Sass, B., Miháltz, M., Kundráth, P.: Az e-magyar rendszer GATE környezetbe integrált magyar szövegfeldolgozó eszközlánca. In: Vincze, V. (szerk.) XI–II. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017). pp. 79–90. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2017)
- Simon, E., Sass, B.: Nyelvtechnológia és kulturális örökség, avagy korpuszpépítés ómagyar kódexekből. *Általános Nyelvészeti Tanulmányok* 24, 243–264 (2012)
- Vadász, N., Kalivoda, Á., Indig, B.: Ablak által világosan – vonzatkeret-egyértelműsítés az igekötők és az infinitívuszi vonzatok segítségével. In: Vincze, V. (szerk.) XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017). pp. 3–12. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2017)
- Vadász, N., Kalivoda, Á., Indig, B.: Egy egységesített magyar igei vonzatkerettől építése és felhasználása. In: Vincze, V. (szerk.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018). pp. 3–15. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2018)
- Volk, M., Clemenide, S., Graën, J., Ströbel, P.: Bi-particle Adverbs, PoS-Tagging and the Recognition of German Separable Prefix Verbs. In: Dipper, S., Ne-



ubarth, F., Zinsmeister, H. (szerk.) Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016). Bochumer Linguistische Arbeitsberichte, vol. 16, pp. 3–10 (2016)

Zsibrita, J., Farkas, R., Vincze, V.: Függőségi elemzésen alapuló magyar nyelvű keresőrendszer. In: Vincze, V. (szerk.) XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017). pp. 363–369. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2017)



## Adalékok a birtokos szerkezetekbe sűrített hasonlatok számítógépes azonosításához

Bajzát Tímea Borbála<sup>1,2</sup>, Szemes Botond<sup>1</sup>, Szlávich Eszter<sup>1</sup>

<sup>1</sup>Eötvös Loránd Tudományegyetem, Digitális Örökség Nemzeti Laboratórium  
1088 Budapest, Múzeum krt. 4., 4–6.

boboszemes@gmail.com

szlavich.eszter@btk.elte.hu

<sup>2</sup>ELKH Irodalomtudományi Intézet, Digitális Örökség Nemzeti Laboratórium  
1118 Budapest, Ménesi út 11–13.

bajzat.timea@abtk.hu

**Kivonat:** A dolgozat a birtokos szerkezetekbe sűrített hasonlatok automatikus azonosítására kifejlesztett módszerünk bemutatását, valamint az azonosított esetek alapján a konstrukció mint poliszém hálózat elemzését tartalmazza. Ez a különböző típusú sűrített hasonlatok kvantitatív és kvalitatív vizsgálatára irányuló kutatásunk megalapozó lépésének tekinthető. Egy ilyen kutatást a szerkezetek sajátos stílusértéke motivál, hiszen azokra egyszerre jellemző a díszítettség és a tömörség a több tagmondat által kidolgozott hasonlatokhoz képest. A dolgozatban bemutatott konstrukció formai elkülönítése komoly nehézséget jelent, amelyet egy többlépcsős, reguláris kifejezéseken alapuló, valamint a szavak grammatikai tulajdonságait is figyelembe vevő módszer kifejlesztésével kívánunk megoldani.

### 1. Bevezetés

Az alábbi dolgozat egy tágabb, kvantitatív kutatás megelőző lépéseiről számol be. Ez a kutatás a névszavakból álló szókapcsolatokba sűrített hasonlatok azonosítását tűzi ki célul, hogy összevesse ezek gyakoriságát a több tagmondat által kidolgozott, formailag konvencionálisabb, „hagyományos” hasonlatok gyakoriságával különböző korok irodalmi és hétköznapi szövegeiben. A kutatásunk nyelvelméleti kiindulópontja a funkcionális kognitív elméleti keret (Langacker 1987, 2008, Tolcsvai Nagy szerk. 2017). A több tagmondatból álló, konvencionálisabbnak nevezett hasonlatok alatt azokat értjük, amikor a hasonlítást kidolgozó mellékmondat a főmondat által nyelvileg színre vitt elemi jelenettel (elemi jelenet vö. Imrényi 2017: 666) „párhuzamba állítható” (Kugler 2017: 850).

A jelen tanulmány a birtokos szerkezetbe sűrített hasonlatok számítógépes azonosítására kifejlesztett módszerünk bemutatását, valamint a konkrét nyelvi példányok vizsgálatával a konstrukció jelentéshálózatának leírását kezdeményezi. Ez a különböző típusú sűrített hasonlatok kvantitatív és kvalitatív vizsgálatára irányuló kutatásunk megalapozó lépésének tekinthető. A kutatást a szerkezetek sajátos stílusértéke motiválja, hiszen azokra egyszerre jellemző a figurák részletezése és a tömörség a több tagmondat által kidolgozott hasonlatokhoz képest. A dolgozatban

bemutatott konstrukció formai elkülönítése összetett kihívást jelent, amelyet egy többlépcsős, reguláris kifejezéseken alapuló, valamint a szavak grammatikai tulajdonságait is figyelembe vevő módszer kifejlesztésével kívánunk megoldani. A hasonlítás tömörsége önmagában is fontos, stilisztikai potencialitással bíró elem a vizsgált szövegekben, amit tovább erősítenek a szerkezetekben egymás mellé kerülő jelzős főnevek nagy száma. Példaként említhetők a következő mondatpárok:

- (1a) A puha takaró egy meghitt, adventi ölelésként vár a nehéz nap után.  
(1b) Úgy vár a puha takaró a nehéz nap után, mint egy meghitt, adventi ölelés.

- (2a) A színpadra belibbenő balett-táncosok üde méltóságával kínálta helytel az idős hölgyet.  
(2b) Olyan üde méltósággal kínálta helytel az idős hölgyet, mint a színpadra belibbenő balett-táncosok.

A vizsgálat egy szöveg fontos stílusértékének tekinti, ha ezek (lásd (1a) és (2a)) a szerkezeti mintázatok nagy számban fordulnak elő benne, vagy ha gyakorinak tekinthetők a tagmondatkapcsolatokkal kidolgozott hasonlatokhoz viszonyítva – továbbá rákérdez az egyes korszakokban való eloszlásukra is. Ezen eloszlás kvantitatív, számítógépes vizsgálata lehetőséget nyújt számunkra ahhoz, hogy a feltárt nyelvi mintázatok alapján hozzájáruljunk a magyar prózairodalom stílustörténeti vizsgálatához.

A fent idézett második mondatpár ((2a) és (2b)) rávilágíthat az egyes konstrukciók automatikus azonosításának nehézségeire is – ami pedig elengedhetetlen a későbbi kvantitatív vizsgálatok számára. A birtokos szerkezetekbe sűrített hasonlatok ugyanis – amelyek sajátos stílusértékére (a konstrukció tömörségére és a jelzőhalmozás lehetőségére) már Herczeg Gyula is felfigyel (Herczeg 1958, 1981) – nehezen különíthetők el más birtokos szerkezetektől, amelyek nem dolgoznak ki hasonlítást: „*A színpadra belibbenő balett-táncosok öltözőjében kínálta helytel az idős hölgyet.*” Az eddigi munkánk során egyértelműen ez jelentette a legnagyobb akadályt a többi típushoz (a *-ként* essivus rag és *-szerű* produktív képzőszerű utótag, valamint a *gyanánt* névutóval alkotott névszói szerkezetekhez) képest. Az alábbiakban ezért elsősorban a hasonlítást kifejező birtokosok azonosítására kifejlesztett módszerünket ismertetjük, illetve – a kereshetőségnek köszönhetően – a konstrukció poliszémiájának szemléltetésére is hozunk a korpuszunkból példányokat, hogy felvázolhassuk a konstrukció jelentéshálózatát.

A kutatásunk célja tehát egy olyan funkcionális kognitív nyelvészeti kiindulópontot érvényesítő módszer kidolgozása, amellyel formai és szemantikai kritériumok alapján nagymennyiségű szövegből bizonyos megszorításokkal konstrukciók nyerhetők ki reguláris kifejezések alkalmazásával. Az általunk kidolgozott metódus és a meghatározott megszorítások korpuszadatok elemzésével, s Herczeg Gyula megfigyelései alapján, dinamikusan, több fázisban szerveződtek. A referenciakorpuszunk egy olyan 100 szépirodalmi regényből álló szövegtömeg volt, amely összeállításánál során törekedtünk arra, hogy a magyar nyelvű regényirodalom történetiségére vonatkozóan reprezentatívnak bizonyuljon (a korpuszépítés elvét az 3. fejezet mutatja be részletesebben). De a későbbiekben az általunk kialakított vizsgálati metódust alkalmazni kívánjuk az ELTE regénykorpusz adatbázisának szövegeire

(<https://regenykorpusz.elte-dh.hu/>) is, mivel ezen korpusz csaknem 400 regény nyelvileg annotált, kereshető elérését biztosítja. A kutatáshoz azért használtunk szépirodalmi szövegeket, mert ez a vizsgált konstrukció – a fent is megfogalmazott feltevéseink alapján – stílusspecifikus, az irodalmi nyelv eszközeként funkcionálhat. A vizsgálatot az R programozási nyelven és szoftverkörnyezetben végeztük.

A célzott konstrukciót sűrített hasonlatnak neveztük el (Herczeg 1958: 313 alapján), ezzel a terminussal fogunk hivatkozni rá; a konstrukció felépítése pedig: DET – N<sub>GEN</sub> – N<sub>INS</sub> – V, azaz egy névelő; egy birtokos esetű főnév; egy birtokos személyjeles, instrumentalis esetű főnév és egy ige alkotja: *az előkelők magabiztosságával megállt*. A konstrukció tetszőlegesen bővíthető a főnevek jelzőivel: *az orosz táncosnők finom eleganciájával lépett be*. A szerkezetet azért tartjuk sűrített hasonlatnak, mert a legtöbb esetben parafrázálhatók hasonlító összetett mondatokkal: *olyan magabiztosan állt meg, mintha előkelő lenne / olyan magabiztosan állt meg, mint egy előkelő; olyan finom eleganciával lépett be, mintha orosz táncosnő lenne / olyan finom eleganciával lépett be, mint egy orosz táncosnő*.

Ha a sűrített hasonlatokat a hasonlító összetett mondat parafrázisokkal rokonítjuk, akkor a hasonlító összetett mondat semlegesebb stílusértékű. Továbbá a sűrített hasonlatok esetében kiemelt jelentőségű az, hogy a tömörítés eszközeként funkcionálnak, hiszen egy tagmondatban teszik elérhetővé a hasonlító és a hasonlított közötti viszonyt integráltabb kidolgozásban. A hasonlítás műveletét végrehajtó nyelvi szerkezetek forma–jelentés pár szimbolikus kapcsolata alapján történő elemzésének kezdeményezését támogatja a funkcionális kiindulópont is, hiszen a nyelvi konstruálás során az eltérő kidolgozási formák a használat és a jelentés felől motiváltak (Langacker 1999, Tolcsvai Nagy 2017b: 224 Langacker 2008: 55–91 alapján).

A tanulmányunk 5. fejezetében az általunk sűrített hasonlatnak nevezett séma konstrukciójának poliszém hálózatával foglalkozunk, ugyanis amellet, hogy ezen szerkezet képes egy integráltabb jelentésviszonyban végrehajtani a hasonlítás műveletét, további jelentéssűrítő potenciállal is bír, amely stilisztikailag szintén jelöltebb (azaz nemsemleges) szerkezeteket hoz létre. Ezen funkciója korántsem periferikus az irodalmi korpuszból hozott eredmények alapján, továbbá nehezen határolható el a hasonlítás végrehajtó példányoktól, azonban kevésbé parafrázálhatók a hasonlító összetett mondatokkal, valamint a kompozitum birtokos esetű főneve rendszerint valamilyen absztrakt entitás (lásd (3)).

(3) [...], *akkor sokkal kiméletesebben és főleg a megfontoltság ravaszságával kell eljárnia.*<sup>1</sup>

A forma–jelentés pár szimbolikus viszonyából kiindulva motivált az, hogy az előzőekben tárgyalt két jelentést/jelentésárnyalatot nem élesen elválasztva egymástól tárgyaljuk. Az efféle integrált, jelentéssűrítő funkcióit aktiváló szerkezetekre Herczeg Gyula (1958: 313) is felfigyelt, elkülönítve kezelte őket a sűrített hasonlatoktól. Az előbbi parafrázálhatóságát abban ragadja meg, hogy két összekapcsolt módhatározóvá alakítja a komponensszerkezeteket (Herczeg 1958: 313). (Az előző példát átalakítva: “...*megfontoltan és ravaszsan kell eljárnia.*” A két típus nem szigorú elhatárolását mutatja azonban, hogy – bár kissé erőltetetten hat –, ám nem tűnik teljesen

<sup>1</sup> Nadas Péter – Emlékiratok könyve (1986)

hibásnak e példamondat hasonlattá alakítása sem: *“olyan ravaszan, mint a megfontoltság.”*) Azonban fontos megjegyeznünk azt, hogy a parafrázisalkotás mind ebben az esetben, mind pedig a sűrített hasonlat jelentésben pusztán csak részleges, hiszen az integráció miatt sajátos jelentésárnyalattal (és stíluspotenciállal) bírnak ezen példányok.

## 2. A vizsgálat módszere

A kutatás ezen fázisa tehát egy olyan keresési eljárást kíván létrehozni, amely lehetővé teszi azoknak a birtokos szerkezeteknek az automatikus azonosítását, amelyek a fent vázolt módon sűrített jelentést dolgoznak ki. Mint említettük, ezek elkülönítése más birtokos szerkezetektől komoly kihívást jelent, amennyiben nem határozható meg egy egyértelmű formai különbség közöttük. A különböző eljárások kézi kiértékelése után, egy többlépcsős, reguláris kifejezéseken alapuló módszert hoztunk létre, amely hatékonyan azonosította a kívánt szerkezetet a keresések során. Fontos, hogy a hatékonyság alatt azt értjük, hogy az általunk felépített reguláris kifejezés nagy arányban teszi szűrhetővé és ezáltal korpuszalapúan vizsgálhatóvá a birtokos szerkezettel kidolgozott jelentéssűrítő kifejezéseket, viszont számolnunk kellett azzal is, hogy az így lekért mintákban olyan példányok is előfordultak, amelyek ugyan formailag illeszkedtek a keresett mintázatra, de nem jelentéssűrítő funkcióban fordultak elő. A módszer segítségével nagyszámú példát tudtunk hozni a vizsgált konstrukcióra a magyar regényirodalom több, mint 170 évéből, ami a szerkezet jobb megértését és leírását biztosítja. Az így nyert belátások pedig a jövőben integrálhatók lehetnek a keresést meghatározó szempontrendszerbe, amitől a módszer hatékonyságának növelését reméljük.

A keresések az e-magyar (Indig és mtsai 2019, Simon és mtsai 2020) nyelvi elemzővel feldolgozott szövegeken futnak le. Az, hogy szófaji és morfológiai elemzésnek vetjük alá a korpuszunkat alkotó szövegtömeget, azért indokolt, mert később, a reguláris kifejezés futtatásakor tudjuk csökkenteni ezzel a számunkra nem kedvező példányok előfordulási számát, mivel figyelembe tudjuk venni a tokenek és azok lemmáinak grammatikai tulajdonságait. Az emtsv-vel való feldolgozás kimenete alapján a szövegeket táblázatos formában kezeljük, amely táblázatokban az oszlopokat 1. a szöveg szavai, 2. a lemmák, 3. a szófajcímkék és 4. a morfológiai elemzés alkotják. A keresések egyes fázisai az adott szempontok szerint más-más oszlopra vonatkoznak, ám a találatok pozíciója (azaz a sorok száma) alapján ezek könnyen összehangolhatók. Első lépésként az elvont jelentéstartalmat kifejező és a *-val/-vel* instrumentális esetraggal ellátott, birtokos személyjeles főneveket azonosítottuk a korpuszban. Az elvont jelentéstartalom (absztraktumok vö. Paradis 2005) fontos megkülönböztető jegye ugyanis a konstrukciónak, hiszen ez a szempont kizárja a konkrét főnevet szerepeltető mondatokat a keresésből, amelyek általában nem sűrített hasonlatok kidolgozásában vesznek részt (pl.: *„A színpadra belibbenő balett-táncosok öltözőjében kínálta helytel az idős hölgyet.”*) Az elvont jelentéstartalmat a formai oldalról tipikusan

---

<sup>2</sup> Az alábbi linken keresztül elérhető a nyilvános Github repozitórium, ahol az általunk fejlesztett programkód jelenlegi állapota található: <https://github.com/SzemesBotond/s-r-tett-hasonlatok>

az alábbi képzők megléte jelezheti: *-alom/-elem; -ság/-ség; -ás/-és*. Azaz a szóalakra vonatkozó keresés az alábbi végződésekre vonatkozott: *-almával, -elmével, -ságával, -ségével, -ásával, -ésével*, majd ezek közül a találatok közül csak a főnévi, illetve a birtokos paradigmát jelölő tagekkel ellátott szavakat őriztük meg, és ezek képezték a további elemzési metódus bemenetét. A kutatás során ellenőriztük, hogy a szerkezetben milyen gyakorisággal fordulnak elő többes számban az említett absztrakt főnevek, ám azt találtuk, hogy így kapott találatok közül csupán egy, meglehetősen speciális eset vonatkozott a keresett konstrukcióra: *“Száz arany négy krajcárért! kiáltják az ő piaci élelmességükkel, és a tömegben kitör a röhej és vihanolás.”*<sup>3</sup> Ez az eredmény már önmagában a konstrukció részletesebb megismeréséhez vezetett, hiszen kijelenthetjük, hogy prototipikusan egyes számban állnak benne az absztraktumok. Meg kell még jegyeznünk, hogy a konkrét toldalékalakokra való keresést azért részesítettük előnyben az emtsv emMorph moduljával szemben, amely bár lehetővé teszi a képzőkre vonatkozó információk kinyerését, mert jelen esetben egyértelműen meghatározható a keresett főnevek végződése formai úton, így a legpontosabb találatokat ezen keresztül kaphatjuk. Azonban amennyiben hasonulással vagy egyéb formai bizonytalansággal is számolnunk kellene, nem tudnánk megkerülni a nyelvi elemző alkalmazását.

Következő lépésként az azonosított főneveket megelőző szavak szófaja alapján szűkítettük a találatokat. Azokat a találatokat tartottuk meg, amelyeket főnév (*egy balett-táncos méltóságával*) vagy főnév-melléknév szófaji kombináció (*egy balett-táncos üde méltóságával*) előz meg, valamint balra tőle négy token keresési ablakban egy névelő található az elemi mondatban. Meg kell említenünk, hogy így a módszer a birtokos főnév előtti bizonyos jelzős mellérendelői szerkezeteket nem tudja azonosítani (*egy balett-táncos üde, de elszánt méltóságával*), mivel ezen eseteket a komponensek közötti kötőszó előfordulása miatt eleve eldobja a metódus. Ahogy látható, ezen lépés potenciálisan csökkentheti a jó találatok számát, azonban a fals találatok nagyobb számú kiszűrése végett fontos volt megtartanunk. Mindazonáltal kísérleteket folytatunk arra nézve is, hogy ezeket a szerkezeteket az emtsv emChunk moduljával azonosítsuk, amely ugyanis éppen a főnévi csoportok (frázisok) automatikus felismerését teszi lehetővé. Ugyanakkor tapasztalataink szerint ez sem működik hibátlanul az olyan esetekben, amelyeket az említett módszerrel sem tudunk befogni, így további kutatásokra van szükség, hogy emChunk hatékonyságát lemérjük. Biztató azonban, hogy az előző példában (*egy balett-táncos üde, de elszánt méltóságával*) a kifejezéshez tartozó részeket helyesen azonosítja a modul.

Ezt követően a szűkített szólista szótöveivel dolgoztunk tovább, pontosabban azt kívántuk meghatározni, hogy milyen szófajú szavakból képezték a fentebb is listázott képzők az elvont főneveket. Ez azért fontos, mert kutatásunk alapján a melléknévi tő felülreprezentált a keresett konstrukciókban (pl. *méltó-ságával*). Az e-magyar által lemmaként azonosított formák a birtokos és a *-vall/-vel* ragot nélkülöző eseteket, azaz tulajdonképpen magukat az elvont főneveket mutatják (pl. *méltóság*). Ezért ebben a fázisban a képzők eltávolításával hoztuk létre a kívánt tövet, amelyen ismételtlen lefutattuk az e-magyar elemzését. Így tudtuk csak azokra a szavakra korlátozni a keresést, amelyekben melléknévi alapból jönnek létre az elvont birtokos főnevek. Ezek képezik a végső találataink kiindulópontját, ezt a listát azonban kiegészítettük olyan nem melléknévi tövű, esetleg nem az említett képzőkkel képzett főnevekkel, amelyek

<sup>3</sup> Mikszáth Kálmán – Új Zrinyiász (1898)

mégis nagy számban fordulnak elő a konstrukcióban. Ezeket elsősorban Herczeg Gyula szintén erre irányuló szövegei (Herczeg 1958: 314), valamint a tövekre vonatkozó szűkítést megelőző keresések találatai alapján határoztuk meg. A nem melléknévből képzett szavak: *jóérzés, érzés, remegés, derengés, pezsgés, feszültség, ragyogás, kétségbeesés, megjelenés, ragaszkodás, hanghordozás, odaadás, szójárás, tájszólás, lángolás, észjárás, arckifejezés, illetve: nyugalom, izgalom, aggodalom, fájdalom, szánalom, léleknyugalom, unalom, vasfegyelem, buzgalom, értelem, félelem, szorgalom, riadalom, ijedelem; a nem az említett módon képzett főnevek: kedvével, könyörületével, dühével, rajongásával, kitartásával, pompájával, türelmével, fölényével, lendületével.*

Ezáltal létrejött azoknak a szavaknak a listája, amelyek nagy eséllyel szerepelnek a keresett konstrukcióban. Utolsó lépésként azokból a mondatokból, amelyekben ezek a szavak szerepelnek, kiszűrtük azokat az eseteket, amikor a konstrukció közelében olyan szó áll, amelynek kötelező vonzata a *-val/-vel* instrumentális ragos eset, és így nem a sűrített hasonlat meglétét jelzi: *szembe(n), visszaél, helyettesít, összevet, versenyez, küzd, vetekszik, ellenkezik, ellentétben, megeléged, áthidal, rendelkezik, kapcsolatos, tisztába(n), együtt, rendelkezik, összefügg, összetéveszt, összeegyeztet, arány, felruház.*

### 3. A referenciakorpusz<sup>4</sup>

A kereséseket 100 kanonikus magyar regényen végeztük el, ez a korpusz több mondatstilisztikai kutatás alapját képezte már – ezek közül is kiemelendő a tagmondatkapcsolatok kvantitatív elemzését célzó tanulmány (Szemes 2021), amely többek között a hasonlatok gyakorisági vizsgálatát is elvégezte a korpuszban található regényeken. Ezért a későbbiekben a jelen kutatás eredményei jól összevethetők lehetnek ennek a tanulmánynak az eredményeivel, további adalékot szolgáltatva a magyar regény stílustörténetének feltárásához.

A korpusz időbeli kiindulópontja a kiforrott magyar regényirodalom kezdete, azaz az 1830-as évek (Szilágyi és Vadera 2010: 370). A végpontot Nadas Péter *Párhuzamos történetek* című, 2005-ben megjelent regénye jelenti, kizárva a kutatásból a kortárs magyar irodalom elmúlt évtizedeit, mivel nehéz és vitatható lenne olyan szempontot találni, amelynek segítségével csupán pár kanonikus mű a napjaink magyar irodalmából kiválasztható. Kanonikus művekre azért van szükség, mert a 19. század első feléből jóformán csak ezek érhetők el digitalizált formában, ami így meghatározta a korpusz egészének szerkezetét. Ugyanígy az ebből a korszakból elérhető alkotók száma szabta meg azt is, hogy későbbi időszakokból hány szöveg került a kutatásba, mivel arányos időbeli eloszlásra törekedtünk a regények első megjelenését illetően. Ezáltal minden évtizedből legkevesebb 3, legfeljebb 8 szöveg került a korpuszba. Ez összesen 100 regényt és 173 évet, azaz átlagosan 1,7 évenként egy regényt jelent. Szintén az arányosság miatt egy írótól maximum 4, de inkább kevesebb regényt

<sup>4</sup> Az alábbi linken keresztül elérhető a korpuszt képező regények listája és azok bibliográfiai adatai: <https://github.com/SzemesBotond/s-r-tett-hasonlatok/blob/main/A%20reg%C3%A9nyek%20felhaszn%C3%A1lt%20kiad%C3%A1sa%20%C3%A9s%20el%C3%A9rhet%C5%91s%C3%A9ge.xlsx>



válogattunk; ez összesen 58 szerzőt eredményezett, mindegyiknek átlagosan 1,7 regényét. Egy szerzőtől abban az esetben szerepel 4 írás, ha azok között jelentős az időbeli távolság, hogy ezáltal vizsgálható legyen az is, hogy az egyes tendenciákat követi-e az adott alkotó életműve, vagy inkább időszaktól független szerzői „ujjlenyomatokról” (Baayen 2002) beszélhetünk.

A szövegek összetételét az irodalomtörténeti konszenzuson túl a Magyar Elektronikus Könyvtár (<https://mek.oszk.hu/>) és a Digitális Irodalmi Akadémia (<https://pim.hu/hu/dia>) adatbázisai határozták meg. Mivel egyik sem érvényesít határozott irodalomtörténeti koncepciót a digitalizáció során, ezért az ELTE BTK Digitális Bölcsészeti Tanszék által egy nemzetközi projekt (*Distant Reading for European Literary History COST Action CA16204* vö. Schöch et al. 2020) keretében összeállított regénykorpuszhoz fordultunk, amely az 1840 és 1920 között megjelent magyar nyelvű prózairodalomból előre meghatározott szempontok szerint válogat (Bajzát és mtsai 2021), valamint az Akadémia Kiadó *Magyar irodalom* című kézikönyvéhez (Gintli szerk. 2010) és egy, a magyar prózastílus történetét tárgyaló korábbi monográfiához (Herczeg 1981). A korpuszt ezenkívül egy-két népszerű szerzővel egészítettük még ki (például Rejtő Jenő, Gárdonyi Géza).

#### 4. A vizsgálat eredményei

A második fejezetben ismertetett eljárást futtattuk a harmadik részben bemutatott referenciakorpuszon. Összesen 561 darab példányt azonosítottunk a módszerrel, azonban, mint ahogyan azt korábban is felvázoltuk, a módszer nem képes a kedvezőtlen találatokat maximálisan kiszűrni. Előnye viszont az, hogy a keresett jelentésszerkezetre vonatkozóan olyan nagy mintát képes biztosítani, amely a további nyelvi elemzésekhez elengedhetetlen, valamint hozzájárul azon törekvésekhez, hogy a magyar irodalom stílustörténete kvantitatívan, mintázatok azonosításával jobban hozzáférhetővé váljon. Az 561 darab példányt manuálisan elemeztük annak érdekében, hogy felmérjük azt, hogy hogyan oszlanak meg a mintában a jelentéssűrítő műveletet végrehajtó szerkezetek és a mintázatra ugyan formailag illeszkedő, de nem az általunk megcélzott adatok. Mivel a kézi elemzés során nem minden esetben volt eldönthető egy-egy példány kategóriába illesztése, ezért mindhárman egymástól függetlenül végeztük a feldolgozást, majd összevetettük egymással az eredményeinket, és a kérdéses eseteket megvitattuk. Mivel törekedtünk arra, hogy a funkcionális kiindulópontot folyamatosan érvényesítsük, ezért nem lehetett mindig igen-nem oppozícióra leszűkíteni a döntéseinket. Úgy döntöttünk, hogy ahelyett, hogy csak a sűrített hasonlatokat címkézzük találatként, a vele szoros hasonlóságot (prototípuselvű kategorizáció vö. Langacker 1987: 16–17, Taylor 1995; Tolcsvai Nagy 2013, 2017b: 36) mutató, jelentéssűrítő szerkezeteket (lásd 1. fejezet) is megtartottuk. Az 561 példányból összesen 43,67%-a volt azon előfordulások száma, amely nem az általunk keresett szerkezet volt (245 db adat), és 316 volt az, amelyet mindhárman elfogadtunk a kézi elemzés során (56,3%).

A jelen dolgozatnak nem célja az eredmények részletes és szisztematikus bemutatása a referenciakorpuszként használt szövegtömeg összes regényének tükrében, ám pár megfigyelésünket és az ezekből fakadó következtetéseinket közöljük. Fontos, hogy a

találatok alacsony száma miatt a számokat ezen a ponton nem arányítjuk a szövegek tokanszámához. A kutatás későbbi fázisában azonban, amikor más típusú sűrített hasonlatok (*-ként*, *-szerű* ragok és a gyanánt névutó) eloszlását is vizsgáljuk, és így több találatot együttesen kezelünk, már mindenképpen relatív gyakorisággal érdemes dolgoznunk.

Herczeg Gyula stílustörténeti koncepciója szerint a keresett szerkezet a 19. század végétől jelenik meg és válik érzékelhető stíluseszközzé a magyar irodalomban (Herczeg 1981:314–16). Eredményeink ezt annyiban igazolják, hogy kiugró értékeket csak 20. században keletkezett regények vesznek fel a keresések és az eredmények kézi ellenőrzése alapján. A kiugró értéket itt úgy értjük, hogy míg regényenként átlagosan 1,92 db ilyen sűrített hasonlat fordul elő a korpuszként alkalmazott szövegtömegben, addig az ettől jelentősen eltérő szövegstruktúrákban legalább 5 ilyen példány található. Kiemelkedik a korpuszból Határ Győző *Héliáne* című műve (32 darab sűrített hasonlat), amely szövegterében amúgy is gyakran dolgozódnak ki hasonlatok (vö.: Szemes 2021) és jellemző rá a figurális tér nagyobb fokú részletezettsége, amelyet a szerkezetek élénk jelzőhasználatára és a hasonlatok sűrítettsége tovább fokoz. Szintén jellemzőnek tekinthető a konstrukció Kosztolányi Dezső konkrétságra és pontos megfogalmazásra törekvő prózájára, amely viszont éppen nem a díszítettség, hanem vélhetően a tömörség miatt fordulhat ehhez a szerkezethez. Nádas Péter *Emlékiratok könyve* című művében szintén gyakrabban adathatunk sűrített hasonlatokat (13 darab), mint a referenciakorpusz jelentős hányadában, viszont érdekes lehet azt megjegyeznünk, hogy a *Párhuzamos történetek* című művében már nem mutatható ki hasonló tendencia (3 darab sűrített hasonlat) – ahogyan azt is jeleznünk kell, hogy a közölt adatok az abszolút gyakoriságra vonatkoznak és nem a szövegek hosszával arányos relatív gyakoriságra.

Azon szerkezeteket tekintve, amelyek nem parafrázálhatók az összehasonlító összetett mondatokkal (lásd 1. fejezet), szintén kimutathatók mintázatok (regényenként átlagosan 1,65 példány fordul elő belőlük). Ezek magasabb előfordulásai összefüggést mutatnak bizonyos esetekben a sűrített hasonlatok nagyobb arányú meglétével. Például Szomory Dezső *A párizsi regény* című művében ezen példányok szintén gyakrabban adathatók (14 darab előfordulás), és a sűrített hasonlatok is viszonylag nagyobb arányban fordulnak elő benne (5 darab). De van arra is példa, hogy míg a sűrített hasonlatokkal történő nyelvi konstruálás jellemzőbb az adott regény szövegterében, addig a jelentéssűrítő, módhatározói kapcsolatra parafrázálható szerkesztésmód egyáltalán nem fordult elő a módszerünkkel generáltatott találati listában (pl. Márai Sándor *A gyertyák csonkig égnek* című regényének esetében, vagy pedig Krúdy Gyula *A vörös postakocsi* című prózájának tekintetében).

## 5. A vizsgált konstrukció jelentéshálózatának szerveződése

A funkcionális kiindulópontból végzett elemzés a vizsgált, DET – N<sub>GEN</sub> – N<sub>INS</sub> – V konstrukció poliszém (poliszémia vö. Cuykens és Zawada 2001, Tolcsvai Nagy 2012, 2017b: 268–288, 2021: 113–121) hálózat azon hálózatrészének elemzésére fókuszál, amely közvetlenül összekapcsolódik a sűrített hasonlatként konvencionizálódott funkcióval, s az ezen csomópont körül szerveződő jelentéseket/jelentésárnyalatokat

elemzi az előzőekben bemutatott módszerrel feltárt konkrét nyelvi példányok alapján. A poliszémia tárgyalásakor a Tolcsvai Nagy Gábor (2012: 288) által összeállított tételeket tekintjük kiindulópontnak: a nyelvi jelentés szerkezetességét és fogalmi alapúságát, a dinamikus relációkon alapuló műveleti működést, a kognitív motiváltságot és a jelentésszerkezetek közötti összefüggést, illetve a prototípuselv alapján megképződő hálózatosságot.

Az általunk fókuszba helyezett jelentésszerkezet és az ahhoz kapcsolódó formai megvalósulás szorosan összefügg más konstrukciós jelentésekkel, ezen eltolódások bizonyos esetekben a szemantikai oldalon történnek meg, másfelől lehetnek a formai oldalról is kimutatható elkülönbözések. Ez utóbbira kevésbé tudunk a jelen eljárásban példányokat hozni, mivel a reguláris kifejezés alapján működő mintavételezési módszerünket szigorú megkötésekkel hoztuk létre. A konstrukciótípus jelentéseinek/jelentésárnyalatainak elkülönítésében segítségünkre lehet a már korábban is említett (részleges) parafrazeálhatóság, illetve a birtokos (genitivusi) szerkezettag és a *-val/-vel* instrumentalis esetragos elvont főnév feldolgozása által aktivált fogalmi tartományok (Lakoff 1987) közötti szemantikai távolság mértéke.

(4) Arnótné az öregasszonyok bölcsességével megérzi, hogy a jövevény, ha lopott vagy ölt.<sup>5</sup>

(5) [...], 2-3000 pengőt is elkért egy műtétért, amelyet a szenvedély ihletével s egy finommechanikai műszerész biztonságával és higgadságával vezetett le, a legnehezebb epeműtettel is altatás [...]<sup>6</sup>

A (4)–(5) példányok esetében jól megfigyelhető az általunk fókuszpozícióba helyezett sűrített hasonlatok tipikus, konvencionalizálódott működése. Akár több tagmondatból álló hasonlító szerkezetté is képesek vagyunk őket (részlegesen) parafrazeálni (pl. *olyan bölcsen, mint amilyenek az öregasszonyok*). Ezen parafrazeálhatóság egyfelől eredeztethető abból, hogy a birtokos (genitivusi) szerkezet által aktivált fogalmi tartomány részeként tudjuk feldolgozni a birtokos által aktivált tulajdonságot, és ezen keresztül a két fogalom között létrejövő integrált reláció (*blend* vö. Tolcsvai Nagy 2021: 109–113) könnyedén összeegyeztethető a világról való tudásunkkal, a mentális begyakorlottsága nagyobb. Az ÖREGSÉG fogalmával könnyedén asszociálható a BÖLCSESSÉG, illetve a finommechanikai műszerészek munkavégzéséhez fűzött elvárásaink között szerepel(het) mind a biztonságra, mind pedig a higgadságra való törekvésük. A (4) példamondat feldolgozása azonban elkülönbözést mutat az (5)-től, hiszen a határozott névelő működése révén a feldolgozás során elérhetővé válik számunkra az az információ, hogy az elsődleges figura (*Arnótné*) maga is birtokosok által kidolgozott csoportba tartozik, míg az (5)-ben feltehetőleg az orvosok és a finommechanikai műszerészek precízsege között válik hozzáférhetővé az analógia.

---

<sup>5</sup> Darvasi László – A könyvmutatványosok legendája (1999)

<sup>6</sup> Déry Tibor – Felelet (1952)

(6) [...], hangja és arcza élénken emlékeztetett a mentoréra az imént. A gyöngéd szív nemességével nyomott el minden szemrehányást, mely a házat ért csapás [...]

A (6) példány esetében képesek vagyunk a több tagmondatból álló hasonlító mintázatához illesztve parafrázálni a birtokos személyjeles kompozitumszerkezetet (“*nemesen, mint a gyöngéd szív*”). Azonban megfigyelhetővé válik az, hogy a konstruálás és a feldolgozás pillanatában jön létre az integrált szemantikai tér a komponensek jelentésstruktúrái között. Ugyan (feltehetőleg) gyengébb asszociatív viszony fennáll a *gyöngéd szív* és a *nemesség* elvontabb jelentésű alszerkezetek között, de ez korántsem olyan könnyen hozzáférhető és feldolgozható, mint az előzőekben idézett példányok esetében ((4)-(5)). A (6) példány továbbá szemlélteti azt, hogy a konstrukció hálózatosan szerveződő jelentései közötti átmenetiséget a kontinuum-elv (Tolcsvai Nagy 2017a: 56) érvényesítésével ragadhatjuk meg. Ahogyan haladunk az egyre absztraktabb integrált relációt kidolgozó kompozitumok felé, úgy csökken a több tagmondatból álló hasonlatra való parafrázálhatóság valószínűsége.

(7) De a háztetőkön is megmozdult a nép: a szabadulás örömeinek dühével szórták az utcán nyüzsgő ellenségre a követ és gerendákat. [...]

A (7) példányban jól kirajzolódik az, hogy azon esetek, amelyeknél a vizsgált kompozitum komponensei maguk is absztrakt entitások, illetve a közöttük létrejövő integrált tér feldolgozhatósága nehezebb, fogalmilag feltűnőbbek, hiszen a birtokos tag (*örömeinek*) és a birtok (*dühével*) eltérő szemantikai tereket aktiválnak. A poétikai stíluspotenciálja a konstrukcióknak vélhetően ezen tulajdonságból fakad. A konstruáló szubjektum ráadásul a fent idézett példány esetében kiaknázza azt, hogy az öröm és a düh egymással látszólagosan ellentétes érzelmi állapot, viszont az általuk aktivált fogalmi tartományokban a FELFOKOZOTT ÉRZELMI INTENZITÁS jelen van, tehát kimutatható részleges érintkezés a szerkezet elemeinek jelentéskomponensei között. Viszont azon példányokat, amelyek a (7)-hez hasonlóan dolgozódnak ki, azokat már nem a sűrített hasonlat címkével láttuk el a manuális elemzés során. A tipikus sűrített hasonlatoktól való viszonylag nagyobb elmozdulást jelöli az, hogy ezen jelentéssűrítő kompozitumok nem parafrázálhatók a több tagmondatból álló hasonlatok mintázatára. A (7)-ben a formai oldalról tekintve további elkülönbözés figyelhető meg, hiszen összetettebb birtokviszony konstruálódik meg (DET – N<sub>GEN</sub> – N<sub>INS</sub> – V), maga a birtokos is egy másik komponens birtokaként képződik meg.

Az ötödik fejezet célja volt, hogy betekintést nyújtson az általunk vizsgált konstrukcionális jelentés szerveződésére hálózatelvűen, kognitív szemantikai kiindulópontot érvényesítve. Megfigyelhettük a vizsgált adatok tükrében, hogy a szemantikai oldalon történő elmozdulás során a kompozitum jelentésárnyalatai variábilisan szerveződnek. Az elmozdulás mögött kimutatható, hogy a komponensszerkezetek absztrakciója, illetve az általuk aktivált szemantikai terek közötti nagyobb távolság és nehezebb összeegyeztethetőségük állnak. Természetesen nem törekedtünk a jelen dolgozatban a teljes jelentéshálózat szisztematikus feltárására, csupán néhány csomópont bemutatására vállalkoztunk.

---

<sup>7</sup> Gárdonyi Géza – A láthatatlan ember (1902)

## 6. Összefoglalás

A tanulmányunk megkísérelte bemutatni a birtokos szerkezetekbe sűrített hasonlatok automatikus, számítógépes azonosítására kidolgozott eljárásunkat, illetve tartalmazza a vizsgált konstrukció poliszém hálózatának funkcionális kognitív szemantikai elemzését. A kutatás során szembesülnünk kellett azzal, hogy a vizsgálat fókuszába helyezett nyelvi mintázat és a vele asszociálódó hasonlító funkció formai elkülönítése a korpuszban korántsem problémamentes folyamat. Az általunk kialakított, reguláris kifejezésen alapuló módszert igyekeztünk a nyelvi anyag kézi elemzéséből származó tapasztalatok nyomán létrehozni. Az így létrejött adatszűrési metódus hatékonysága abban rejlik, hogy a viszonylag kis számban előforduló nyelvi szerkezet a módszerünk segítségével jól detektálhatóvá válik a bemenetet képező szövegtömegben. A jelen vizsgálat referenciakorpuszaként egy olyan 100 kanonikus magyar regényből álló szöveggyűjtemény szolgált, amely egészen az 1830-as évektől kezdve 2005-ig bezárólag tartalmaz műveket. A jelenlegi eredményeink alapján visszacsatolhatunk Herczeg Gyula azon stílustörténeti koncepciójára, miszerint az ilyen, birtokos szerkezetbe sűrített hasonlatok megjelenése a 19. század végére tehető a magyar prózairodalomban. A kimutatásaink alapján ezen szerkezet már korábban adatolható a vizsgált regényekben, viszont kiugró mennyiségbeli érték ténylegesen a 20. századi művek egy részében válik hozzáférhetővé (pl. Határ Győző *Heliáne*).

A funkcionális kognitív nyelvelméleti keretbe illeszkedő poliszémia-elemzésünk arra vállalkozott, hogy a vizsgált konstrukció jelentéshálózatának azon szegmensét elemezze, amely közvetlenül összekapcsolódik a sűrített hasonlatként konvencionálizálódott funkcióval, valamint bemutassa azt, hogy a kontinuum-elv alapján történő jelentésfeltárás gyümölcsöző eredményeket szolgáltathat a jelen esetben.

A bemutatott vizsgálat egy olyan hosszabb távú kutatás alapját képezi, amely a sűrített hasonlító szerkezetek (pl. főnév + *-ként*; főnév + *-szerű*) kvantitatív és kvalitatív vizsgálatára törekszik. Egy ilyen kutatást egyfelől motivál az, hogy a mintázatok nagy mennyiségű adathalmazon való kimutatása a stílustörténeti tudásunkhoz járul hozzá, másfelől pedig a szerkezet működésének használatalapú megközelítését támogatja.

## Hivatkozások

- Baayen, H., van Halteren, H., Neijt, A., Tweddle, F., An experiment in authorship attribution. JADT 2002: 6es Journées internationales d'Analyse statistique des Donn'ees Textuelles. Conference Paper. (2002)
- Bajzát, T., Szemes, B., Szlávich, E. Az ELTE DH Regénykorpusz és lehetőségei. In: Online térben az online térért: Networkshop 30: országos online konferencia. 2021. április 6-9. pp. 63-72. HUNGARNET Egyesület. Budapest (2021)
- Cuyckens, H., Zawada, B. (szerk.) Polysemy in Cognitive Linguistics. John Benjamins, Amsterdam, Philadelphia. (2001)
- Gintli, T. (szerk.) Magyar irodalom. Akadémiai Kiadó. Budapest. (2010)
- Herczeg, Gy. Határozóként álló, -val, -vel ragos utótagú birtokos szerkezetek. Magyar Nyelvőr 84, pp. 311–573. (1958)
- Herczeg, Gy. A régi magyar próza stílusformái. Tankönyvkiadó Vállalat. Budapest. (1981)

- Imrényi, A. Az elemi mondat viszonyhálózata. In: Tolcsvai Nagy, G. (szerk.) *Nyelvtan. A magyar nyelv kézikönyvtára* 4. pp. 663–761. Osiris. Budapest. (2017)
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N. emtsv – egy formátum mind felett. In: XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 235–247. Szegedi Tudományegyetem TTIK, Informatikai Intézet. Szeged. (2019)
- Kugler, Nóra. Az összetett mondat. In: Tolcsvai Nagy, G. (szerk.) *Nyelvtan. A magyar nyelv kézikönyvtára*. 4. pp. 806–895. Osiris. Budapest. (2017)
- Lakoff, G. *Women, Fire, and Dangerous Things*. The University of Chicago Press, Chicago, London (1987)
- Langacker, R. W. *Foundations of Cognitive Grammar. Volume I Theoretical Prerequisites*. Stanford University Press. Stanford–California. (1987)
- Langacker, R. W. *Grammar and conceptualization*. Mouton de Gruyter. Berlin, New York. (1999)
- Langacker, R. W. *Cognitive Grammar: A basic introduction*. Oxford University Press. Oxford. (2008)
- Paradis, C. Ontologies and construals in lexical semantics. *Axiomathes* 15, pp. 541–573. (2005)
- Schöch, C., Eder, M., Rosari, A., Pieter, F., Primorac, A. *Foundations of Distant Reading. Historical Roots, Conceptual Development and Theoretical Assumptions around Computational Approaches to Literary Texts*. (2020) <https://dh2020.hcommons.org/>
- Szilágyi, M., Vaderna G. A klasszikus magyar irodalom (kb. 1750-től kb. 1900-ig). In: Gintli, T. (szerk.) *Magyar irodalom*. pp. 313–637. Akadémiai Kiadó. Budapest. (2010)
- Simon, E., Indig, B., Kalivoda, Á., Mittelholcz, I., Sass, B., Vadász, Noémi. Újabb fejlemények az e-magyar háza táján. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–42. Szegedi Tudományegyetem TTIK, Informatikai Intézet. Szeged. (2020)
- Szemes, B. Kidolgozott viszonyok. A tagmondatkapcsolatok automatikus azonosításának hasznosíthatósága a stilsztikában és az irodalomtörténet-írásban, *Digitális Bölcsészet* 4 (2021) - megjelenés alatt.
- Taylor, J. R. *Linguistic categorization*. Oxford University Press. Oxford. (1995)
- Tolcsvai Nagy, G. A poliszémia hálózati modellje. *Nyelvtudományi Közlemények* 108, pp. 287–344. (2012)
- Tolcsvai Nagy, G. Bevezetés a kognitív nyelvészetbe. Osiris. Budapest. (2013)
- Tolcsvai Nagy, G. Bevezetés. In: Tolcsvai Nagy G. (szerk.) *Nyelvtan. A magyar nyelv kézikönyvtára* 4. pp. 23–71 Osiris. Budapest. (2017a)
- Tolcsvai Nagy, G. Jelentés. In: Tolcsvai Nagy Gábor (szerk.) *Nyelvtan. A magyar nyelv kézikönyvtára* 4. pp. 207–466. Osiris. Budapest. (2017b)
- Tolcsvai Nagy, G. (szerk.) *Nyelvtan. A magyar nyelv kézikönyvtára* 4. Osiris. Budapest. (2017c)
- Tolcsvai Nagy, G. *Kognitív szemantika*. Eötvös Kiadó. Budapest. (2021)

# Three-order normalized PMI and other lessons in tensor analysis of verbal selectional preferences

Márton Makrai<sup>1,2</sup>

<sup>1</sup> Institute of Cognitive Neuroscience and Psychology  
Centre for Natural Sciences, Budapest

<sup>2</sup> Research Group for Language Technology  
Hungarian Research Centre for Linguistics  
makrai.marton@ttk.hu

**Abstract.** We investigate several questions in transitive verb structure representation by decomposing tensors populated with different subject-verb-object association measures, including a novel generalization of normalized pointwise mutual information to the higher-order ( $>2$ ) case. Which association measure works the best in modeling verb structures? Should we include occurrences with unfilled arguments in our statistics? We also investigate qualitatively the latent dimensions, and the difference between each noun as a subject versus an object.

**Keywords:** selectional preferences, tensor decomposition, association measures

## 1 Introduction

Verbs have been characterized on the basis of how frequently various syntactic constituents occur in various grammatical relations to them, which is, not surprisingly, related to the meaning of the verb (Levin, 1993). These selectional preferences have been analyzed with machine learning tools (Van de Cruys, 2009). Verb structures include collocations, whose syntactic modifiability or semantic compositionality is reduced: their linguistic distribution may be idiosyncratic or the sense of the combination may be habitual or even fixed (Bouma, 2009).

*Tensors* ( $>2$ -dimensional arrays) generalize matrices; while matrices contain numbers aligned in two dimensions, rows and columns, tensors have more of these dimensions, also called *axes* or *modes*<sup>1</sup> Singular value decomposition (SVD) of a co-occurrence matrix is a natural tool to compute generalizations about the interactions between two modes, like words and documents (LSA, Landauer and Dumais (1997)), target and context words (words embeddings, Mikolov et al. (2013b); Levy and Goldberg (2014b); Pennington et al. (2014)), or words and dependency contexts (Levy and Goldberg, 2014a). Four ways of looking at SVD (in LSA) can be distinguished (Turney and Pantel, 2010): the goal can be the modeling of some latent meaning, noise reduction, indirect aka. high-order co-occurrences (when two words appear in similar contexts), or sparsity reduction. Intuitively, language features multi-mode interactions: *the turntable playing the piano* can be strange (Van de Cruys, 2009), while the two-mode relations

<sup>1</sup> The term *mode* is preferred when data from different modalities are fused.

$\langle \text{play, SUBJ, turntable} \rangle$  and  $\langle \text{play, OBJ, piano} \rangle$  are perfect. Tensor generalizations of matrix decomposition (Kolda and Bader, 2009), especially *low-rank factorizations*, open the way for the analysis of such interactions.

It seems that, after intensive early research (Van de Cruys, 2009; Van de Cruys et al., 2013; Polajnar et al., 2014; Fried et al., 2015; Hashimoto and Tsuruoka, 2015), results obtained with skip-gram and related word embedding methods outshone tensor methods for verb argument structure. Yet at the same time, tensor decomposition has developed remarkably, and NLP test-beds in the domain of verb argument structure have been involved in cutting-edge scalable, noise-robust tensor works (Sharan and Valiant, 2017; Bailey et al., 2018; Frandsen and Ge, 2019). The data-driven linguistic understanding of word ambiguity and especially that of verb selection is still immature. Here we try to make progress in the linguistic direction by further research on tensorial analysis of verb argument structure.

Tensor decomposition provides embedding vectors for each mode (in our case, nouns as subjects, verb, and nouns as objects) analogous to word embeddings in (shallow or deep) neural networks. In this paper, we compute different association measures between subjects, verbs, and objects, populate tensors with these measures, decompose the tensors with different algorithms, and investigate the resulting word embeddings quantitatively and qualitatively to answer the following questions.

Our first four questions will be answered quantitatively in the modeling of English subject-verb-object triple similarity, while the last two questions are qualitative.

- Which *association measure* yields the best representations? We experiment with several measures, including our novel generalization of normalized pointwise mutual information to the higher-order ( $>2$ ) case.
- Should we include *empty argument fillers* (subjects or objects) in our co-occurrence statistics? Ideally, including them may help generalization over the transitive and the intransitive uses of the same verb, while discarding them may help focusing on transitive structures cleanly as a separate phenomenon.
- The two tensor decomposition algorithms, CPD and Tucker, which we will introduce in Section 3, have very different time-complexity: Tucker is much faster. Tensor decomposition has hyper-parameters like the decomposition rank and the frequency cutoff. Both are related to memory limitation, especially the latter. It would be beneficial, *if the two algorithms reached the best results with the same hyper-parameters*, because then a fast parameter tuning with Tucker would also benefit CPD. Is this the case?
- How does the trade-off between the three hyper-parameters related to the *size of the decomposition* (i.e. the decomposition rank, the inclusion of empty fillers, and the frequency cutoff) look like?
- Do latent dimension of our word embeddings reflect lexical knowledge?
- Can the difference between each noun as a subject versus an object correspond to some intuitive difference between subjecthood and objecthood?

Section 2 describes the linguistically motivated association measures between subjects, verbs, and objects we apply. These measures include ones that are novel to the best



of our knowledge. Section 3 offers an introduction to tensor decomposition. Section 4 describes our experiments. Our code is available online.<sup>2</sup>

## 2 Counts, weighting, and associations

Word co-occurrences form *sparse* arrays, as most words do not occur empirically with most words, and frequencies span many orders of magnitude (*Zipf* or power law distribution, Manin (2008); Gittens et al. (2017)). In order to scale to large data, linguistic tensor decomposition methods have to be based on sparse tensors populated with more sophisticated scores than frequency. Now we turn to these weighting functions and especially to linguistically motivated association scores.

The simplest choice is the logarithm of the co-occurrence frequency (Pennington et al., 2014; Sharan and Valiant, 2017). Jenatton et al. (2012) places the modeling of the ⟨subject, verb, object⟩ triples in the context of multi-relational learning, and apply a weighting function related to the log-bilinear model (Mnih and Hinton, 2007; Mikolov et al., 2013a).

Van de Cruys (2009, 2011); Van de Cruys et al. (2013), and Bailey et al. (2018) use three-mode generalizations of the information-theoretic association measure (*Positive Pointwise Mutual Information* ((P)PMI). Positivity is related to sparse inputs: in order to attribute higher scores to actual co-occurrences than unattested ones, PMI and the lexicographic association scores introduced in the following paragraph, *positive* variants of the association measures have to be used, e.g. PPMI, which replaces negative PMI entries with zero. We discuss the two types of three-variable generalization of PPMI in Section 2.2: the more standard total correlation (that we still call PMI) and interaction information.

We also experiment with generalizing Log Dice (Rychlý, 2008) to three axes

$$\log \frac{3f(x, y, z)}{f(x) + f(y) + f(z)} + c,$$

where  $c$  is chosen so that the Log-Dice values are non-negative. (While 3 in the nominator is redundant, because it is subsumed under  $c$ , we keep it in the formula to make it more reminiscent of the established 2-variable case.) The use of Log Dice as well as salience introduced in the next paragraph has, to the best of our knowledge, mainly been limited so far to lexicography.

### 2.1 Salience and normalized PPMI

PPMI, despite of its nice information-theoretic interpretability, is biased towards rare events (Turney and Pantel, 2010; Levy et al., 2015; Zhuang et al., 2018). This motivates the Sketch Engine lexicographic software (Kilgarriff et al., 2004) to multiply vanilla PPMI by  $\log f$  (in our case, by  $\log(f(x, y, z))$ ), to get the measure of *salience*. We apply similar modifications to every score introduced in Section 2 so far. Denoting vanilla PPMI, interaction information and Log Dice by `pmi-vanl`, `iact-vanl`, and

<sup>2</sup> <https://github.com/makrai/verb-tensor>

Dice-vanl, respectively, we get pmi-sali, iact-sali, and Dice-sali by multiplying the vanilla score by  $\log f(x, y, z)$ .

There is a theoretically better motivated way of transforming PMI to some measure which is less biased towards rare combinations. In Bouma (2009)’s approach, *normalization* is related to boundedness. He looks for measures whose absolute value is pointwise larger than that of PMI. Entropy and negative log probability are two of those measures, and we follow the literature in choosing the latter. In our experiments, we apply this normalization to the two multi-mode generalizations of PMI which will be introduced in Section 2.2, interaction information and the one which we will still call PMI. While normalized interaction information does not excel in our experiments, tree-variable normalized PMI, which is to the best of our knowledge the novelty of the present paper, proves the best among the alternatives considered. Empirically, when divided by  $-\log p(x, y, z)$ , positive interaction information and the more standard 3-mode PPMI is upper-bounded by 1 and 2, respectively.

## 2.2 Higher-order PMI

One would think that it’s obvious that the 3-variable generalization of Pointwise Mutual Information (PMI) is

$$\log \frac{p(x, y, z)}{p(x)p(y)p(z)}, \quad (1)$$

but it turns out that this is only one of the possible generalizations. Van de Cruys (2011) introduces two pointwise association measures, whose expected values are two different multivariate generalizations of mutual information (Shannon and Weaver, 1949): interaction information (McGill, 1954) and total correlation (Watanabe, 1960).

Pointwise *interaction information* is based on the notion of conditional mutual information.<sup>3</sup>

$$\log \frac{p(x, y)p(x, z)p(y, z)}{p(x, y, z)p(x)p(y)p(z)}$$

*Total correlation* on the other hand quantifies the amount of information that is shared among the variables, with a pointwise variant defined by the formula in Equation 1. Following the literature (Villada Moirón, 2005; Van de Cruys, 2009; Van de Cruys et al., 2013; Bailey et al., 2018), when we speak about (*multivariate Positive*) *Pointwise Mutual Information* in this paper, we will mean (pointwise) total correlation.

Van de Cruys (2011) reports that, in their Dutch experiments, both methods are able to extract salient subject verb object triples (prototypical SVO combinations like *poll represents opinion* and fixed expressions). Narrowing the scope to the word *play*, they find that interaction information picks up on prototypical SVO combos e.g. *orchestra plays symphony*, while the more established one (which he calls specific correlation) picks up on *play a role* and salient subjects that go with the expression.

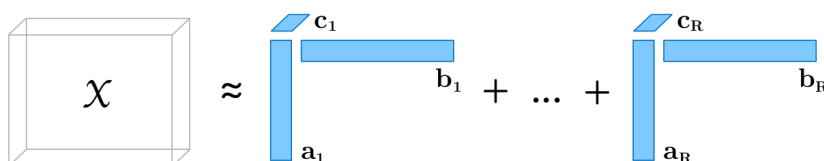
<sup>3</sup> Mnemonically, the formula of the pointwise variant generalizes the 2-mode case along the inclusion and exclusion principle, except it has the numerator and the denominator swapped to ensure a proper set-theoretic measure.

### 3 Tensor decomposition

The main entry point to tensor computation is Kolda and Bader (2009), but Rabanser et al. (2017) is also worth consulting.

There is no single generalization of the SVD concept, the two most popular extensions, Canonical Polyadic Decomposition and the more general Tucker, feature different generalized properties. Sidiropoulos et al. (2017) discuss the interpretation of these two different ways of decomposition in signal processing and machine learning points of view.

#### 3.1 Canonical Polyadic Decomposition



**Fig. 1.** Canonical Polyadic Decomposition, figure from Rabanser et al. (2017).

Canonical Polyadic Decomposition (CPD, aka. CanDecomp, Parallel Factor model, CanDecomp, rank decomposition, or Kruskal decomposition, (Carroll and Chang, 1970)) expresses a tensor as a minimum-length linear combination of rank-1 tensors. A rank-1 tensors is the tensor product of a collection of vectors, just as the dyadic product of two vectors is a 1-rank matrix, see Figure 1.

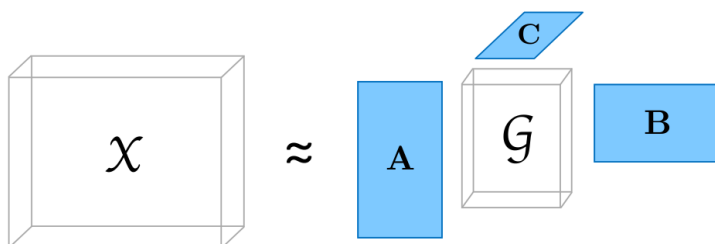
The alternating least squares algorithm (ALS, Carroll and Chang (1970); Harshman (1970)) is an iterative method for CPD. In each iteration, all but one of the modes are fixed and the remaining one is fitted. ALS does not guarantee convergence, and even if it converges, this cannot be detected in a trivial way. Orth-ALS (Sharan and Valiant, 2017) improves on ALS.

#### 3.2 Tucker decomposition

While CPD seems more relevant for linguistics representation, we also discuss Tucker decomposition, because it can be computed much more efficiently. Tucker decomposition (aka. Higher Order SVD, Tucker (1966)) factorizes a tensor into a core tensor  $\mathcal{G}$  multiplied by a matrix along each mode, see Figure 2. In the case of

$$\text{subject} \times \text{verb} \times \text{object}$$

tensors, rows of the three matrices contain embedding vectors of entities (subjects or objects) and those of verbs (“relation”), and entries of the core tensor  $\mathcal{G}$  determine the



**Fig. 2.** Tucker Decomposition, figure from (Rabanser et al., 2017).

levels of interactions between the former three. Tucker decomposition is not unique, because we can transform  $\mathcal{G}$  without affecting the fit if we apply the inverse of that transformation to the factor matrices. Uniqueness can be improved (Kolda and Bader, 2009) by imposing e.g. sparsity, making the elements small, or making the core “all-orthogonal”. Other priors and constraints in tensor learning involve non-negativity and independence (Lahat et al., 2015).

## 4 Experiments

In this section, we report our experiments. After introducing, in Section 4.1, the corpus that serves as the basis of our empirical investigations, Section 4.2 compares association measures, the two alternatives for treating missing arguments, the two decomposition algorithms, and some other hyper-parameters (the decomposition rank and the frequency cutoff) in the classical task of predicting the similarity of English subject-verb-object triples (Kartsaklis and Sadrzadeh, 2014). Then in Section 4.3, we investigate the latent dimensions qualitatively. Section 4.4 compares the embedding vector of each noun as a subject versus an object, to see how differently nouns behave in the two roles.

### 4.1 Experimental setting: the corpus and the similarity task

In our experiments, we took the occurrence counts of ⟨subject, verb<sup>4</sup>, direct object⟩ triples from the automatically dependency-parsed (Nivre et al., 2016) English corpus DepCC (Panchenko et al., 2018), irrespectively of whether there were other arguments or adjuncts. Regarding empty fillers, we investigated two alternatives: including them (represented by a fixed string) or discarding them from our statistics. `tensorly` (Kosaiji et al., 2016) was used for CPD and (general and non-negative) Tucker decomposition of tensors. For tensor population in `COordinate` format, we use the `sparse` Python library.

Our quantitative tests are based on a classical similarity data-set for English transitive verb structures (SVO triples) by Kartsaklis and Sadrzadeh (2014, KS14). The data-set contains triples with gold (human) similarity scores. We represent SVO triples

<sup>4</sup> *Verb* means, in UD terms, that the `upos` starts with VB.

by concatenating the corresponding subject, verb, and object embedding vector (we experimented with normalizing the vectors, but we did not find it useful), and computed the Spearman correlation between the cosine similarities of the (long) vectors in each pair with the human scores.

cutoff	shape with unfilled	shape without unfilled
1	(324 196, 90 606, 287 967)	(206 488, 41 075, 188 619)
10	(160 629, 37 427, 129 694)	(109 432, 19 824, 92 635)
100	(92 999, 20 937, 69 536)	(71 768, 13 907, 57 420)
1000	(44 168, 10 444, 32 359)	(40 309, 8 838, 30 280)
10000	(13 765, 5 070, 12 313)	(13 610, 4 895, 12 115)
100000	(3 474, 2 313, 4 120)	(3 463, 2 308, 4 108)
1000000	(546, 814, 981)	(545, 813, 980)
10000000	(36, 194, 87)	(35, 194, 86)

**Table 1.** The length of each axis, i.e. the number of subjects, verbs, and objects, at different frequency cutoffs.

## 4.2 Quantitative results in transitive structure similarity

We populated tensors with the association measures introduced in Section 2. The statistics were based on either including empty argument fillers (i.e. treating all arguments “optional”) or excluding these occurrences. We took different cutoffs and computed non-negative or general CPD or Tucker decompositions in different ranks. Table 1 shows the length of each axis, i.e. the number of subjects, verbs, and objects, at different frequency cutoffs.

Correlations we obtain in the subject-verb-object task are shown in Table 2. The properties of the original sparse tensor (the association measure, whether empty fillers are included, and the frequency cutoff) are shown on the left of the vertical line, while those of the decompositions (non-negative or general CPD or Tucker decompositions to the specified rank) are shown on the right. The table shows, in addition to the best setting, each setting obtained by changing one meta-parameter. The best result is obtained by non-negative CPD. The horizontal line shows the place of our best general Tucker, general CPD, and non-negative Tucker decompositions, which we discuss later in this subsection. In Tucker decompositions, we use the same rank among all axes.

We obtained the best correlation, 0.7360, from the decomposition of a tensor populated with salience-weighted PMI values, including empty fillers, and setting the frequency cutoff to 1 million, i.e. restricting the axes of the tensor to the subjects, verbs, and objects that appear at least 1 million times. This best correlation was obtained with non-negative CPD in rank 64. This correlation value is in the same range as 0.76 obtained by Hashimoto et al. (2014) with a much more complex system that used to be the state-of-the-art, when this task was fashionable.

The table shows the correlation obtained by changing each (meta-)parameter. While the results seem to be relatively robust with respect to the decompositions *rank*, it may

assoc measure	unfilled	cutoff	non-negative	decomp algo	rank	corr
pmi-sali	included	1 000 000	non-neg	parafac	64	0.7359
pmi-sali	included	1 000 000	non-neg	parafac	128	0.7097
pmi	included	1 000 000	non-neg	parafac	64	0.6857
pmi-sali	included	1 000 000	non-neg	parafac	32	0.6773
pmi-sali	included	300 000	non-neg	parafac	64	0.6630
npmi	included	1 000 000	non-neg	parafac	64	0.6602
dice-sali	included	1 000 000	non-neg	parafac	64	0.4709
pmi-sali	excluded	1 000 000	non-neg	parafac	64	0.4578
pmi-sali	included	1 000 000	general	parafac	64	0.4560
ldice	included	1 000 000	non-neg	parafac	64	0.4409
log-freq	included	1 000 000	non-neg	parafac	64	0.4322
iact-sali	included	1 000 000	non-neg	parafac	64	0.4112
niact	included	1 000 000	non-neg	parafac	64	0.4068
pmi-sali	included	3 000 000	non-neg	parafac	64	0.3936
iact	included	1 000 000	non-neg	parafac	64	0.3248
pmi-sali	included	1 000 000	non-neg	tucker	64	0.2989

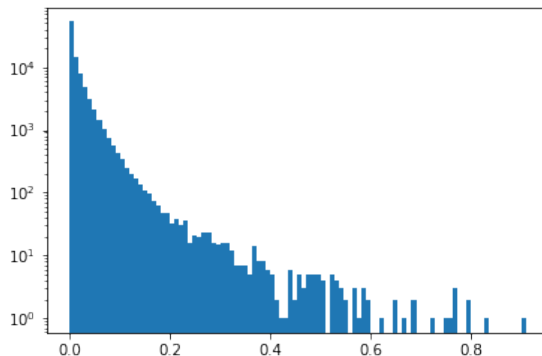
**Table 2.** Quantitative results: correlations in the subject-verb-object triple similarity task (Kartsaklis and Sadrzadeh, 2014) obtained with word embeddings of tensor decompositions.

be interesting that when we concatenate the subject, the verb, and the object embedding vectors, 64 dimensional each, we get a vector in the famous range of a couple of hundreds of dimensions, which proved to work well in many different scenarios like LSA and static word embeddings (see the introduction).

As for our *association measures*, different weighted variants (salience, vanilla, or normalization) of PMI work the best, followed by log-Dice and log frequency. Variants of interaction information performs the worst.

The inclusion of empty fillers, the frequency cutoff, and the decomposition rank are all related to the *size of the tensors*. While we have already seen that the decomposition rank does not have a great influence on the results, if we exclude empty fillers, a more generous frequency cutoff may theoretically lead to better results than if we change only one of these two parameters. It turns out, that we can indeed get relatively good result (0.694181) this way, but with general Tucker decomposition (instead of non-negative CPD) and log-Dice (instead of salience-weighted). The cutoff is 1 million.

Non-negative decomposition is advantageous from the interpretational point of view, because in our experiments, they resulted in embedding matrices which are *sparse* in the broad sense that most coordinates are low. Figure 3 shows a histogram of the matrix elements. Note that the vertical axis, which corresponds to the histogram count in each bin, is logarithmic. The figure suggests that frequency decreases faster than exponentially as larger weights are considered. The good performance of non-negative CPD suggests that non-negativity introduces meaningful structure. Sparsity raises the hope that coordinates are interpretable, i.e. they correspond to concepts or properties.



**Fig. 3.** The histogram of the verb embedding matrix elements. Note that the vertical axis, which corresponds to the histogram count in each bin, is logarithmic. The figure suggests that frequency decreases faster than exponentially as larger weights are considered.

CPD has the advantage that it maps the *modes in the same space*. In our case, this is the most interesting for subjects and objects: we can compare the same noun in the two roles. We return to this in Section 4.4.

While our best results have been obtained with non-negative CPD, we discuss general Tucker and CPD and non-negative Tucker as well. Results with general decompositions and non-negative Tucker are shown in Table 3 and Table 4, respectively. General Tucker and CPD and non-negative Tucker all prefer normalized PMI as the association measure, disfavor interaction information, and results with log frequency and log Dice vary. General and non-negative Tucker obtains the best results with the same rank as non-negative CPD, and the two non-negative decomposition algorithms also share the value for a best cutoff. It is inconclusive whether it is advantageous to include occurrences with unfilled arguments in our statistics.

### 4.3 Qualitative analysis of latent dimensions

Now we investigate the latent dimensions obtained by tensor decomposition. We experimented with non-negative and general CPD and Tucker decomposition with the hyper-parameters that reached the best result in the SVO-similarity task.

The latent dimensions are shown in Tables 5 to 7. (Dimensions with general Tucker are degenerate, and they are omitted to save space.) Each line corresponds to a latent dimension. Dimensions are visualized by the words with the greatest coordinates in the dimension. Blocks represent dimension triples.  $\emptyset$  denotes that the corresponding grammatical function is unfilled. Some latent dimensions, like the first one in our non-negative CPD are dominated by (the empty filler and) pronouns. In these cases we *emphasize* the first contentful filler. `-rrb-` stands for right round brackets, and its appearance may be an artifact of the corpus.

assoc measure	unfilled	cutoff	rank	correlation	assoc measure	unfilled	cutoff	rank	correlation
npmi	included	100 000	64	0.7191	npmi	excluded	300 000	256	0.6383
pmi-sali	included	100 000	64	0.7049	pmi-sali	excluded	300 000	256	0.6166
log-freq	included	100 000	64	0.6883	pmi	excluded	300 000	256	0.5811
pmi	included	100 000	64	0.6759	npmi	excluded	1 000 000	256	0.5754
npmi	included	30 000	64	0.6729	npmi	excluded	100 000	256	0.5713
ldice	included	100 000	64	0.6685	npmi	excluded	300 000	512	0.5677
ldice-sali	included	100 000	64	0.6666	npmi	excluded	300 000	128	0.5290
npmi	included	300 000	64	0.6598	npmi	excluded	30 000	256	0.5239
npmi	included	100 000	128	0.6540	npmi	included	300 000	256	0.5070
npmi	included	100 000	32	0.6042	log-freq	excluded	300 000	256	0.2465
npmi	excluded	100 000	64	0.5207	ldice	excluded	300 000	256	0.2093
iact-sali	included	100 000	64	0.5059	iact-sali	excluded	300 000	256	0.1280
niact	included	100 000	64	0.4632	niact	excluded	300 000	256	0.0726
iact	included	100 000	64	0.4316	iact	excluded	300 000	256	0.0615

**Table 3.** Results with general Tucker (left) and general CPD (right).

assoc measure	unfilled	cutoff	rank	correlation
npmi	excluded	1 000 000	64	0.5186
npmi	excluded	1 000 000	128	0.5102
npmi	excluded	300 000	64	0.4814
pmi	excluded	1 000 000	64	0.4563
pmi-sali	excluded	1 000 000	64	0.4387
npmi	excluded	1 000 000	32	0.3753
npmi	excluded	3000 000	64	0.3366
npmi	optional	1 000 000	64	0.2889
iact	excluded	1 000 000	64	0.0989
log-freq	excluded	1 000 000	64	0.0763
ldice	excluded	1 000 000	64	0.0698
ldice-sali	excluded	1 000 000	64	0.0619
niact	excluded	1 000 000	64	0.0454
iact-sali	excluded	1 000 000	64	0.0064

**Table 4.** Results with non-negative Tucker.

dim	words
0	∅, that, which, it, <i>story</i> , he, they, who, what, one, she, work, event, -rrb-, this, you. . .
0	catch, attract, draw, pay, deserve, capture, gain, grab, get, receive, focus, require. . .
0	attention, eye, crowd, interest, fire, visitor, audience, conclusion, breath, people, . . .
1	∅, who, we, he, I, you, she, they, -rrb-, <i>student</i> , member, people, group, Center, parti. . .
1	attend, host, hold, organize, schedule, enjoy, join, arrange, cancel, miss, watch, pla. . .
1	meeting, event, conference, session, party, show, school, class, dinner, church, tour, . . .
2	that, which, it, this, ∅, <i>change</i> , factor, they, choice, condition, decision, issue, -rr. . .
2	affect, impact, influence, improve, hurt, reflect, benefit, change, damage, enhance, a. . .
2	ability, performance, health, outcome, life, quality, result, business, development, e. . .
3	file, which, page, site, that, it, book, report, section, document, collection, websit. . .
3	contain, include, provide, have, list, feature, display, show, comprise, present, give. . .
3	information, link, material, number, list, datum, name, content, statement, reference, . . .

**Table 5.** Latent dimensions with Non-negative ParaFac.



dim words	
5	court, Court, judge, panel, official, we, he, it, authority, government, -rrb-, Board,...
10	reject, dismiss, deny, grant, hear, consider, decide, accept, throw, resolve, sustain,...
7	motion, appeal, claim, request, argument, case, challenge, application, complaint, att. ...
4	revenue, sale, share, price, stock, production, cost, rate, order, volume, number, fut. ...
3	rise, fall, increase, jump, drop, decline, climb, decrease, grow, gain, slip, represen. ...
1	percent, %, \$, increase, point, most, rate, level, average, less, matter, value, cost,...
11	hotel, property, room, restaurant, home, Center, house, location, facility, House, are. ...
8	offer, boast, feature, have, provide, include, enjoy, serve, accommodate, occupy, prep. ...
9	room, pool, accommodation, access, facility, restaurant, variety, service, view, range. ...
6	board, Council, Board, Commission, Committee, member, committee, Congress, Court, cour. ...
2	approve, adopt, reject, pass, consider, review, endorse, propose, award, recommend, ac. ...
2	resolution, request, budget, plan, proposal, contract, change, application, project, i. ...

**Table 6.** Latent dimensions with Non-negative Tucker.

In the case of CPD, the dimensions are enumerated in the order as returned by the algorithm. With Tucker, the values  $g_{ijk}$  in the core tensor  $\mathcal{G}$  represent the interaction between the  $i$ th latent dimension for subjects, the  $j$ th one for verbs, and the  $k$ th one for objects. We sorted the triples of SVO latent dimensions in our best non-negative and general Tucker decomposition by this interaction strength. The index of each dimension, as returned by the algorithm, is also shown in the table. E.g. the first block in non-negative Tucker shows that the strongest interaction is between the 5th latent dimension of subjects, the 10th one for verbs, and the 7th one for objects. Note that in the non-negative case,  $g_{ijk} \geq 0$ , so we do not have to take the absolute value. Dimensions obtained with the two *non-negative algorithms* seem semantically interpretable, while those from general decomposition are less convincing.

dim words	
0	Israel, group, government, Foundation, Association, company, -rrb-, military, army, Cl. ...
0	launch, wage, suspend, mount, begin, run, fund, organize, sponsor, administer, carry, ...
0	campaign, attack, program, initiative, operation, strike, programme, website, effort, ...
1	user, you, application, customer, developer, visitor, client, processor, device, User, ...
1	access, select, specify, upload, view, enter, edit, browse, click, create, retrieve, m. ...
1	file, datum, content, document, page, parameter, site, folder, node, Internet, informa. ...
2	device, assembly, means, structure, system, element, plate, section, interface, unit, ...
2	comprise, include, contain, have, utilize, employ, represent, say, mean, control, enab. ...
2	layer, element, device, tube, housing, spring, electrode, pump, plate, container, memb. ...
3	attorney, plaintiff, defendant, party, respondent, prosecutor, State, lawyer, governme. ...
3	file, receive, oppose, make, give, present, withdraw, handle, publish, drop, provide, ...
3	motion, notice, petition, appeal, response, answer, objection, charge, request, submis. ...

**Table 7.** Latent dimensions with General ParaFac.

#### 4.4 Comparing subject and object vectors

Tensor decomposition can shed light on how differently nouns behave as subjects and as objects. This question is related to symmetric factorization (Bailey et al., 2018), which imposes symmetry constraints between the embeddings of the same entities in different modes (in our case, between the embeddings of the same noun as a subject or an object). Our approach is complementary, based on that CPD maps nouns as subjects and objects in the same space.

In our experiments, we consider (non-negative) CPD decomposition with the hyper-parameters that proved best in English SVO-similarity. We computed the (unnormalized) dot product similarity between the subject and object vector of each noun, and sorted all the nouns by this similarity. The largest distance is found with  $\emptyset$ , *he*, *she*, *they*, *I*, *device*, *system*, *that*, *you*, *it* . . . , while the most symmetric nouns are *doubt*, *reality*, *future*, *same*, *hope*, *feeling*, *mine*, *reason*, *consumer*, *plenty* . . . A possible explanation is that the former, especially personal pronouns, are much more frequent in agentive roles than other nouns, while they are infrequent in patient roles. Words in the second group can be framed in language both as animate and as inanimate. *Future* or *hope* are not alive in the biological sense, but they are often attributed agentive roles (what can be called a metaphorical use of language but being metaphorical does not mean that the usage is peripheral, as it has been noted by linguists).

## 5 Conclusion and future work

Weighted variants of positive pointwise mutual information proved better than the considered alternatives in modelling subject-verb-object structure similarity. It does not matter, whether we include occurrences with unfilled arguments in our statistics. Our best results were obtained with non-negative CPD. The best frequency cutoff and the decomposition rank is the same for the two non-negative decomposition algorithms, which raises the hope that these hyper-parameters of non-negative CPD can be fine-tuned based on the much faster non-negative Tucker, but this needs to be tested in other setups. Our experiments provided lexically interpretable latent dimensions and verb clusters, and the difference between subject and object embeddings can be related to animacy, at least in the case of non-negative CPD.

Tensor decompositions offer a direction orthogonal to the mainstream (Rogers et al., 2020) in the data-driven understanding of linguistic structure. Our line of research can be extended cross-lingually (Vulić et al., 2017; Majewska et al., 2018; Sun et al., 2010).

## Bibliography

- Bailey, E., Meyer, C., Aeron, S.: Learning semantic word representations via tensor factorization (2018), <https://openreview.net/forum?id=BlkIr-WRb>, arXiv:1704.02686
- Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: GSCL 2009: International Conference of the German Society for Computational Linguistics and Language Technology (2009)

- Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika* 35, 283–319 (1970)
- Van de Cruys, T.: A non-negative tensor factorization model for selectional preference induction. In: *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. pp. 83–90. Association for Computational Linguistics, Athens, Greece (3 2009), <https://www.aclweb.org/anthology/W09-0211>
- Van de Cruys, T.: Two multivariate generalizations of pointwise mutual information. In: *Proceedings of the Workshop on Distributional Semantics and Compositionality*. pp. 16–20. Association for Computational Linguistics, Portland, Oregon, USA (6 2011), <https://www.aclweb.org/anthology/W11-1303>
- Van de Cruys, T., Poibeau, T., Korhonen, A.: A tensor-based factorization model of semantic compositionality. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 1142–1151. Association for Computational Linguistics, Atlanta, Georgia (6 2013), <https://www.aclweb.org/anthology/N13-1134>
- Frandsen, A., Ge, R.: Understanding composition of word embeddings via tensor decomposition. In: *7th International Conference on Learning Representations, ICLR 2019 (5 2019)*, <https://openreview.net/forum?id=H1eqjiCctX>, arXiv preprint arXiv:1902.00613
- Fried, D., Polajnar, T., Clark, S.: Low-rank tensors for verbs in compositional distributional semantics. In: *ACL (2015)*
- Gittens, A., Achlioptas, D., Mahoney, M.W.: Skip-gram – zipf + uniform = vector additivity. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 69–76. Association for Computational Linguistics (2017), <http://aclweb.org/anthology/P17-1007>
- Harshman, R.A.: Foundations of the parafac procedure: Models and conditions for an “explanatory” multi-modal factor analysis. *UCLA Working Papers in Phonetics* 16, 1–84 (1970), <http://publish.uwo.ca/~harshman/wpppfac0.pdf>
- Hashimoto, K., Stenetorp, P., Miwa, M., Tsuruoka, Y.: Jointly learning word representations and composition functions using predicate-argument structures. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. p. 1544–1555 (2014)
- Hashimoto, K., Tsuruoka, Y.: Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In: *3rd Workshop on Continuous Vector Space Models and their Compositionality (2015)*
- Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*. pp. 3167–3175. NIPS’12, Curran Associates Inc. (2012), <http://dl.acm.org/citation.cfm?id=2999325.2999488>
- Kartsaklis, D., Sadrzadeh, M.: A study of entanglement in a categorical framework of natural language. In: *The 11th workshop on Quantum Physics and Logic (6 2014)*, arXiv:1412.8102

- Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: Sketch engine. In: Williams, G., Vessier, S. (eds.) *Proceedings of Euralex*. pp. 105–116. Lorient, Université de Bretagne-Sud, Faculté des lettres et des sciences humaines (7 2004)
- Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM review* 51(3), 455–500 (2009)
- Kossaifi, J., Panagakis, Y., Anandkumar, A., Pantic, M.: Tensorly: Tensor learning in python. *Journal of Machine Learning Research (JMLR)* 20, 1–6 (2016), arXiv preprint arXiv:1610.09555
- Lahat, D., Adali, T., Jutten, C.: Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE* 103(9), 1449–1477 (2015)
- Landauer, T.K., Dumais, S.T.: A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review* 104(2), 211 (1997)
- Levin, B.: *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press (1993)
- Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pp. 302–308. Association for Computational Linguistics, Baltimore, Maryland (06 2014a), <http://www.aclweb.org/anthology/P14-2050>
- Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems 27*. pp. 2177–2185 (2014b)
- Levy, O., Remus, S., Biemann, C., Dagan, I.: Do supervised distributional methods really learn lexical inference relations? In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 970–976. Association for Computational Linguistics, Denver, Colorado (2015), <https://www.aclweb.org/anthology/N15-1098>
- Majewska, O., Vulić, I., McCarthy, D., Huang, Y., Murakami, A., Laippala, V., Korhonen, A.: Investigating the cross-lingual translatability of VerbNet-style classification. *Language Resources and Evaluation* 52(3), 771–799 (2018)
- Manin, D.Y.: Zipf’s law and avoidance of excessive synonymy. *Cognitive Science* 32, 1075–1098 (2008)
- McGill, W.: Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory* 4(4), 93–111 (1954)
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings (05 2013a)*, <http://arxiv.org/abs/1301.3781>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc. (2013b), <https://bit.ly/39HikH8>
- Mnih, A., Hinton, G.: Three new graphical models for statistical language modelling. In: *Proceedings of the 24th international conference on Machine learning*. pp. 641–648. ACM (2007)

- Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: A multilingual treebank collection. In: Proc. LREC 2016. pp. 1659–1666 (5 2016)
- Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S., Biemann, C.: Building a web-scale dependency-parsed corpus from common crawl. In: Proceedings of LREC 2018. ELRA (2018)
- Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics (2014), <http://www.aclweb.org/anthology/D14-1162>
- Polajnar, T., Rimell, L., Clark, S.: Using sentence plausibility to learn the semantics of transitive verbs. In: NIPS Learning Semantics Workshop (2014), in arXiv, some minor errata fixed.
- Rabanser, S., Shchur, O., Günnemann, S.: Introduction to tensor decompositions and their applications in machine learning (11 2017), <http://arxiv.org/abs/1711.10781v1>, arXiv:1711.10781 [stat.ML]
- Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in bertology: What we know about how bert works. arXiv preprint arXiv:2002.12327 (2020)
- Rychlý, P.: A lexicographer-friendly association score. In: Proceedings of Recent Advances in Slavonic Natural Language Processing. pp. 6–9 (2008)
- Shannon, C.E., Weaver, W.W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1949)
- Sharan, V., Valiant, G.: Orthogonalized ALS: A theoretically principled tensor decomposition algorithm for practical use. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017. pp. 3095–3104 (8 2017), <http://proceedings.mlr.press/v70/sharan17a.html>
- Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on signal processing 65(13), 3551–3582 (7 2017), <https://doi.org/10.1109/TSP.2017.2690524>
- Sun, L., Korhonen, A., Poibeau, T., Messiant, C.: Investigating the cross-linguistic potential of VerbNet: style classification. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 1056–1064. Association for Computational Linguistics (2010)
- Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika 31(3), 279–311 (1966)
- Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
- Villada Moirón, M.B.: Data-driven identification of fixed expressions and their modifiability. Ph.D. thesis, University of Groningen (2005)
- Vulić, I., Mrkšić, N., Korhonen, A.: Cross-lingual induction and transfer of verb classes based on word vector space specialisation. arXiv preprint arXiv:1707.06945 pp. 2546–2558 (9 2017), <https://www.aclweb.org/anthology/D17-1270>
- Watanabe, S.: Information theoretical analysis of multivariate correlation. IBM Journal of research and development 4(1), 66–82 (1960)

XVIII. Magyar Számítógépes Nyelvészeti Konferencia    Szeged, 2022. január 27–28.

Zhuang, Y., Xie, J., Zheng, Y., Zhu, X.: Quantifying context overlap for training word embeddings. In: EMNLP (2018)

# An introductory statistical study of Hungarian word order

Dávid Márk Nemeskey<sup>1,2</sup>

<sup>1</sup>Department of Digital Humanities – Eötvös Loránd University

<sup>2</sup>National Laboratory for Digital Heritage

nemeskey.david@btk.elte.hu

**Abstract** Hungarian is often cited as a language with free word order. While this is not strictly true, the rules that govern the sentence structure are derived from pragmatics and are thus much more flexible than they are for analytical languages such as English. This paper presents an introductory statistical study into Hungarian word order. We report the order of verbal arguments in simple sentences in two corpora: the Hungarian Wikipedia and TrendMiner. An experimental method for ordering adjectives in noun phrases is also presented.

**Keywords:** word order, argument structure, order of adjectives

## 1 Introduction

Hungarian is frequently called a free word order language. It is true that, being an inflecting language with a rich case system, Hungarian does not need a set word order to distinguish between the main constituents of a clause (i.e. the verbal arguments). This poses a challenge for foreign learners of the language and influences the language acquisition process of Hungarian children as well (Nagyházi, 2013; Pléh, 1981).

It must be noted that most of the time the term *word order* comes up in the literature, it does not refer to the order of each individual word, but that of the *constituents* of the sentence; more specifically, the order of the verb and its arguments. The language has parts where the order of words is determined by syntactic rules, such as the structure of noun phrases (NPs; although see Section 4). These usually cause little confusion and will not be discussed in this paper.

It is not only learners of the language who are baffled by the seemingly random word order. Initially, linguists also had a hard time explaining it, and even advised forming sentences following German rules (Márton, 1805)<sup>1</sup>. It was Brassai (1852–53) (following earlier attempts by Táncsics (1833) and Fogarasi (1838)) who first suggested that constituents in the Hungarian sentence are ordered based on their discourse functions.

---

<sup>1</sup> Although this was probably only because the book was written to a German audience.

É. Kiss (1981, 1994) proved that the Hungarian sentence has an invariant structure, which marks constituents from a pragmatics point of view<sup>2</sup>. The main communicative roles, the *topic* (already known background information) and the *focus* (new information introduced by the sentence) occupy the two “slots” before the verb; the postverbal part serves no discourse function. The order of constituents within the topic and postverbal slots is free, while the focus may only host a single constituent.

The original discussion has since been extended to complex sentences (Kenesei, 1984b), exclamatives (Lipták, 2006) and discontinuous phrases (Barta et al., 2004). Behavior of constituents have been given further thought in both the preverbal (Puskás, 2000) and postverbal parts (Szalontai and Surányi, 2020). The word order of certain constituents, such as PPs, have also received attention (Dékány and Hegedűs, 2015). Competing theories also exist, such as Kenesei (1984a), but they argue against the particular grammatical framework used in É. Kiss (1981), not the general observations.

The interest in word order is not purely academic, however. Aside from the aforementioned impact on language acquisition, there are certain natural language processing (NLP) tasks where the order of sentence constituents is important. One example is dependency parsing, where determining the correct (or a valid) order of head and dependents can be a challenge. The problem also comes up in data generation or augmentation for machine learning in domains where training corpora are scarce or nonexistent.

Unfortunately, the studies above do not provide practical answers to these challenges, for two reasons. First, our NLP pipelines sorely lack any form of semantic, let alone pragmatic processing capabilities. Second, the studies are highly theoretic and are not data driven, and hence, cannot serve as bases for a machine learning system.

In this paper, we aim to conduct an initial statistical study into Hungarian word order. In the long run, we aim to create a framework for acquiring statistical data that could serve as the basis for the tasks mentioned above. Here, as a proof of concept, we focus on the methodology and examine a few simpler aspects of word order:

1. the order of verb arguments in simple sentences;
2. the tendency of oblique arguments to occupy the topic or focus position;
3. the order of adjectives in a noun phrase.

## 2 Methodology

### 2.1 Data

We run our experiments on two Hungarian corpora, which were selected to be markedly distinct in their style. We hypothesize that the word order statistics are not homogeneous across the language but depend on the style of the corpus.

<sup>2</sup> We do not aim at presenting the full timeline of the field in this paper. For a historical overview of theories on word order, the reader is referred to Nagyházi (2013)



**Wikipedia (WP)** The Hungarian Wikipedia<sup>3</sup> is a semi-edited corpus, created by volunteers. As an encyclopedia, most of the articles use formal, written language that aims to convey factual information in a neutral tone. For our study, we used the version available in the Hungarian Webcorpus 2.0 (Nemeskey, 2020).

**TrendMiner (TM)** The TrendMiner corpus is a collection of 1.9 million political Facebook comments, harvested over the course of 3 months in 2013 and 2014 (Miháltz et al., 2015). As expected of such material, the language is highly informal, often rude, and as close to spoken Hungarian as the medium allows. Compared to Wikipedia, it contains a high amount of questions, exclamatives and imperatives.

TrendMiner contains about 46M tokens in 4M sentences; Wikipedia is about 3 times larger at 175M tokens in 13.8M sentences.

Both corpora were processed with `emtsv` (Indig et al., 2019) using the morphological analyzer, lemmatizer and dependency parser modules. For Wikipedia, we only needed to run the latter as the rest of the annotations is already available in Webcorpus 2.0. TrendMiner also comes fully annotated, but it uses different tagsets and so had to be re-processed to make the two sources compatible.

## 2.2 Tooling

In order to quickly find sentences with a certain argument structure, we loaded the data into the treebank search tool `dep_search`<sup>4</sup> (Luotolahti et al., 2015). `dep_search` allows the user to index and query data in the CoNLL-U format<sup>5</sup> using a custom query language<sup>6</sup>. The language enables the specification of a dependency subgraph complete with morphological constraints that is matched against the indexed treebank. Matching sentences are returned in their original CoNLL-U format, with the single target token of the query marked as such.

We indexed and queried our two corpora separately to allow for comparison between the two. Since `dep_search` returns sentences as-is, we developed our own scripts to extract the argument structure from the query results. Both our code<sup>7</sup> and a patched version of `dep_search`<sup>8</sup> that fixes a few issues with the original are available on GitHub.

## 3 Argument structure

### 3.1 Preprocessing

In this initial study, we concentrate on the simplest sentences: namely,

<sup>3</sup> <https://hu.wikipedia.org/wiki>

<sup>4</sup> [https://github.com/fginter/dep\\_search](https://github.com/fginter/dep_search)

<sup>5</sup> <https://universaldependencies.org/format.html>

<sup>6</sup> <http://bionlp.utu.fi/searchexpressions-new.html>

<sup>7</sup> [https://github.com/DavidNemeskey/word\\_order](https://github.com/DavidNemeskey/word_order)

<sup>8</sup> [https://github.com/DavidNemeskey/dep\\_search](https://github.com/DavidNemeskey/dep_search)

1. the sentence is declarative;
2. it has a single finite verb as its ROOT;
3. the verb is not negated;
4. it has exactly one subject and object, each noun( phrase)s.

As Hungarian is a pro-drop language (certainly as far as subjects are concerned), we repeated our measurements also for the case when the SUBJ relation is not realized on the surface.

To satisfy the first condition, we simply dropped all sentences that did not end with a single period “.” token. The second to fourth conditions were implemented by the `dep_search` query<sup>9</sup>

```
VERB&Mood=Ind&VerbForm=Fin
!< _ >SUBJ NOUN >OBJ NOUN !>OBL _ !>COORD _ !>NEG _
```

Due to a limitation of the `dep_search` query language, we could not narrow the query down for it to discard sentences containing multiple subjects and/or objects. Instead, these were filtered later from the final statistics. As a result, the percentage values in Tables 2, 3 and 5 do *not* sum to 100%; the remaining mass is accounted for by these more complicated sentences.

Corpus	No obliques (see 3.2)		With obliques (see 3.3)		Retained
	SUBJ-OBJ	OBJ	SUBJ-OBJ	OBJ	
Wikipedia	276 788	126 794	398 541	350 805	8.37%
TrendMiner	20 487	34 004	14 890	15 399	2.07%

**Table 1.** Number of sentences in the filtered collections

Table 1 reports the sizes of the filtered corpora. As shown in the last column, only a small fraction of the original corpora passed our filters; in fact, the number of sentences we ended up from TrendMiner is so low that the feasibility of a more in-depth study than the one presented below is questionable. The stylistic differences between the two corpora are also reflected in that the sentences in WP tend to be more complex (including a subject and/or obliques most of the time), while subject dropping is more frequent in TM.

### 3.2 The simple sentence

Table 2 lists the relative frequencies (in percentages) for all possible orderings of the VERB and its SUBJ and OBJ dependencies in our simple sentences. Looking at the top, we can see that the most frequent orderings are SVO and SOV, although TrendMiner clearly prefers the former and Wikipedia the latter. While the order of the rest of the orderings is the same for both corpora, the distributions

Structure			Counts	%	Structure			Counts	%
SUBJ	OBJ	VERB	77 673	28.06	SUBJ	VERB	OBJ	6901	33.68
SUBJ	VERB	OBJ	64 841	23.43	SUBJ	OBJ	VERB	4571	22.31
OBJ	SUBJ	VERB	58 851	21.26	OBJ	SUBJ	VERB	1837	8.97
OBJ	VERB	SUBJ	15 950	5.76	OBJ	VERB	SUBJ	1636	7.99
VERB	OBJ	SUBJ	1328	0.48	VERB	OBJ	SUBJ	474	2.31
VERB	SUBJ	OBJ	992	0.36	VERB	SUBJ	OBJ	375	1.83

**Table 2.** SUBJ-VERB-OBJ word order variations in WP (left) and TM (right)

are decidedly different: TM’s has a “fatter” tail, whereas in WP, VERB-initial sentences are virtually nonexistent, and OSV is almost as prevalent as SVO.

Table 3 shows the results for the SUBJ-drop sentences. Again, the two corpora differ in their preferred orderings: Wikipedia prefers OV by a large margin, while TrendMiner contains the two possible ordering of VERB and OBJ in almost equal measure, slightly preferring VO. While the order of the first two rows are the same as in Table 2 with respect to VERB and OBJ, their distribution is not: the difference between SVO and SOV was bigger in TM than in WP, while here it is the opposite. We are going to show, however, that the disparity is only skin-deep.

Structure			Counts	%	Structure			Counts	%
OBJ	VERB		82 905	65.39	VERB	OBJ	17 309	50.90	
VERB	OBJ		35 807	28.24	OBJ	VERB	15 376	45.22	
OBJ	OBJ	VERB	3525	2.78	OBJ	VERB	OBJ	763	2.24
OBJ	VERB	OBJ	3135	2.47	OBJ	OBJ	VERB	356	1.05
VERB	OBJ	OBJ	1191	0.94	VERB	OBJ	OBJ	183	0.54

**Table 3.** VERB-OBJ word order variations in WP (left) and TM (right)

We can interpret all sentences with no overt subject as having a virtual pronoun subject that was dropped from the realized sentence. (Although explicitly realizing the pronoun changes the pragmatics of the sentence. Example (1) from Wikipedia illustrates this: the two sentences are semantically equivalent, but (1b) emphasizes the overt subject, while the covert one in (1a) is neutral.) If we drop the SUBJ relation from all structures in Table 2 and merge the (now) identical rows, we end up with 55.08% OBJ VERB to 24.27% VERB OBJ in Wikipedia and 39.27% to 37.82% in TrendMiner, which is much closer to the relative ratios in Table 3.

<sup>9</sup> For an explanation of the query, see the documentation linked above.

- (1) a. *Bevallottan Robert Bresson stílusát követi.*  
 admittedly Robert Bresson 's style.ACC follow.3SG
- b. *Ő bevallottan Robert Bresson stílusát követi.*  
 he admittedly Robert Bresson 's style.ACC follow.3SG
- ‘He admits to following Robert Bresson’s style.’

The bottom half of Table 3, which lists statistics for sentences with two objects, paints the same picture. In TrendMiner, the “neutral” (although rather unusual) OBJ VERB OBJ is the most frequent ordering, with the other two being much rarer, while in Wikipedia, OBJ OBJ VERB not only outnumbers VERB OBJ OBJ 3 to 1, but is also the most numerous variant overall.

These results confirm our earlier hypothesis that the two corpora would differ in their word orders. Yet there might be other factors at play that may explain the differences. The two corpora might simply contain different sets of verbs. In addition to that, the verb distributions might also exhibit a bias not present in the language in general. In the following, we are investigating these hypotheses.

VERB, OBJ				SUBJ, VERB, OBJ			
Wikipedia		TrendMiner		Wikipedia		TrendMiner	
Verb	Percent	Verb	Percent	Verb	Percent	Verb	Percent
<i>nevez</i>	5.77	<i>kíván</i>	7.44	<i>tartalmaz</i>	3.15	ad	2.53
kap	2.00	<i>köszön</i>	4.00	ad	2.36	kap	2.22
ír	1.89	lát	2.21	jelent	2.15	jelent	1.41
ad	1.78	kap	1.89	kap	2.02	tesz	1.21
tart	1.61	<i>kíván</i>	1.78	alkot	1.56	lát	1.19
végez	1.50	kér	1.74	mutat	1.56	hoz	1.13
használ	1.38	vár	1.63	vesz	1.12	jár	0.84
talál	1.20	ad	1.59	okoz	1.07	okoz	0.79
készít	1.13	hoz	1.41	hoz	1.03	fizet	0.71
vezet	1.11	szeret	1.18	képez	0.99	vesz	0.69
épít	1.09	olvas	1.16	biztosít	0.99	mutat	0.68
hív	1.09	tesz	1.11	használ	0.99	tűr	0.67
tekint	0.96	ismer	1.08	tart	0.95	megszáll	0.66

**Table 4.** Relative frequency of the top words in both corpora for the VERB OBJ and SUBJ VERB OBJ frames

Table 4 lists the most frequent verbs found in the two corpora for the VERB OBJ and SUBJ VERB OBJ frames. What is clear at first glance is that the two frames have widely different verb usage characteristics. VERB OBJ is dominated by a few words in both corpora: “*nevez*” (*call*) in WP and “*kíván*” (*wish*, also misspelled as “*kíván*”)<sup>10</sup> and “*köszön*” (*greet*) in TM. These verbs seem to be

<sup>10</sup> With many occurrences in expressions like “*Jó reggelt kívánok.*” (*Good morning.*)

idiosyncratic of each corpus, as they are outliers to the linear functions that can be fitted to the frequencies of the rest of the words on the lists (with a  $R^2$  of 0.94 in both cases). Consequently, removing them from the data might improve the accuracy of our statistics.

The updated statistics are listed in Table 5. Comparing them to Table 3, we can see that removing *nevez* from WP has left the relative frequencies of the structures unchanged. On the other hand, the frame distribution in TM has become closer to that WP, proving the existence of the bias hypothesized above and demonstrating the importance of filtering idiosyncratic word (usage)s from the data.

Structure	Counts	%	Structure	Counts	%
OBJ VERB	76 881	60.64	VERB OBJ	15 836	46.56
VERB OBJ	34 517	27.22	OBJ VERB	12 355	36.34

**Table 5.** VERB-OBJ word order variations in WP (left) and TM (right) without corpus-specific words

With the bias (largely) out of the way, we can turn our attention to the vocabularies of the two datasets and see if they explain the differences in the argument structure distributions. Table 5 shows a mixed picture. In the top 13 verbs for VERB OBJ, there are only two words (“*kap*” (*receive*) and “*ad*” (*give*)) common to both list; however, for SUBJ VERB OBJ, about half of the words (6) fall in the intersection. In the longer tail (up to the top 50 verbs), the ratio of words common to both corpora are around 33% for VERB OBJ and 50% for SUBJ VERB OBJ. How much of the difference can be explained by this is left for future study.

### 3.3 Obliques

For oblique arguments, a similar study could be conducted as for SUBJ and OBJ. However, the resulting table would be hard to interpret and would tell very little, as OBL is an umbrella relation that covers any of 15 noun cases (nominative and accusative excluded). Instead, we opted to compare oblique argument types based on their inclination to move to a preverbal position.

As explained in Section 1, the preverbal positions in a Hungarian sentence are taken by the pragmatically most important parts of the information structure: the topic and the focus. Unfortunately, without pragmatological analysis, we cannot determine which role the argument takes, so we simply check if the argument precedes the verb.

Looking at the results in Table 6, we can see that half of the cases (namely All, Cau, Ill, Ins, Sub, Ter, Tra) have very similar (within 3%) relative frequencies in both corpora, while some of them (Ab1, De1, El1) differ by more

Case	Suffix	Wikipedia		TM	
		Frequency	Preverbal	Frequency	Preverbal
Abl	-től	5029	55.72%	419	38.90%
Ade	-nál	2414	70.30%	114	64.91%
All	-hoz	4333	46.83%	188	44.68%
Cau	-ért	1355	43.03%	244	45.08%
Dat	-nak	6513	55.43%	253	42.59%
Del	-ről	6227	62.15%	294	36.39%
Ela	-ból	6642	64.57%	652	46.17%
Ess	-ként	7225	73.74%	221	82.35%
Ill	-ba	7019	48.17%	690	44.93%
Ine	-ban	65 659	71.54%	2234	62.67%
Ins	-val	25 504	63.30%	1421	62.14%
Sub	-ra	21 226	53.24%	1715	56.15%
Sup	-n	44 772	75.19%	1457	65.41%
Tem	-kor	1404	85.04%	49	71.43%
Ter	-ig	2757	77.73%	142	74.65%
Tra	-vá	1178	63.67%	41	60.98%
Nom		193 458	89.91%	9218	87.31%
Acc	-t	125 399	58.28%	4561	43.20%

**Table 6.** The total number of oblique arguments with specific cases and the percentage they occur in a preverbal position. Nominative and accusative included for reference. Only sentences with a single SUBJ, OBJ and OBL were taken into account.

than 15%. On the whole, Wikipedia seems to employ more preverbal obliques; the difference is only 4% on the type level, but significantly larger (67.15% vs 57.65%) on the token level. The difference makes sense intuitively, as the goal of Wikipedia is to convey factual information and new information enters the sentence in the focus position, which, as we have seen, immediately precedes the verb in Hungarian.

Why certain oblique arguments prefer the preverbal position more than others is an interesting topic for future work. Looking at the directional cases, TO-type cases (All, Ill, Sub) are around 50% in both corpora, while AT-types (Ade, Ine, Sup) are the “most preverbal” with 75-75% in WP and 60–65% in TM. FROM types (Abl, Ela, Del) tend to be between the two in WP, but have the lowest percentages of all cases in TM. Again, the explanation of these tendencies requires further research.

## 4 Adjective order

It has long been theoreticized that the (neutral) order of attributive adjectives in an NP is subject to restrictions based on their semantic category. Dékány (2021) mentions the variations below:

- a. value > dimensions > physical property > speed > human propensity > age > colour (Dixon, 1982)

- b. cardinal > ordinal > quality > size > shape > colour > nationality (Cinque, 1994)
- c. ordinal > cardinal > size > length > height > speed > width > weight > temperature > wetness > age > shape > colour > origin > material (Scott, 2002)
- d. subjective comment > evidential > size > length > height > speed > depth > width > weight > temperature > ?wetness > age > shape > colour > nationality/origin > material (Laenzlinger, 2005)

, and posits that these restrictions are also in place for Hungarian.

Similarly to prior theoretical work on word order in general, the category lists above make sense intuitively, but are not readily usable for NLP. One reason is that there is no mapping defined to actual words. The lists are also incomplete, so even if the mappings existed, many adjectives would be left without a category. In this section, we investigate if similar, but more complete, ordering rules can be extracted from a corpus in a data-driven fashion.

#### 4.1 Categories

We started out by collecting all noun phrases with at least two adjectives in them:

```
NOUN >ATT@L ADJ&Case=Nom >ATT@L ADJ&Case=Nom
```

A custom script was then used to retain only those where the adjectives immediately precede the noun. This allowed us to concentrate on the core issue without having to deal with numerals (NUM) and coordinating conjunctions (CCONJ) for now. We then deleted all tokens other than the adjectives and added the dummy tokens DET and NOUN to the beginning and the end of each adjective group, respectively. This gave us a list of 1.6M virtual NPs from Wikipedia; we decided against using TrendMiner in this experiment so as to have a cleaner dataset. The total number of adjective types is 132 795.

A word2vec (Mikolov et al., 2013) embedding was trained on the concatenated virtual NPs using gensim<sup>11</sup> (Řehůřek and Sojka, 2010). We used a CBOW model of 25 dimensions with a window size of only 2, so that adjectives that belong to separate NPs do not see each other over the dummy tokens. To allow us to actually evaluate our results, we only trained embeddings for adjectives that occurred in at least 4000 NPs; 113 in total.

As a last step, we ran the  $k$ -means algorithm (MacQueen et al., 1967) in Scikit-learn<sup>12</sup> (Pedregosa et al., 2011) on the normalized vectors to obtain our candidate categories. Table 7 shows the results of a clustering with  $k = 8$ .

As can be seen, the clusters are remarkably consistent, aside from the few odd words (*in italics*). It is also not difficult to assign a name to the clusters, apart from the last one, which plays the role of the “kitchen sink”. Some of the

<sup>11</sup> <https://github.com/RaRe-Technologies/gensim>

<sup>12</sup> <https://github.com/scikit-learn/scikit-learn>

Cluster	Words
<b>Nationality</b>	magyar amerikai német francia brit angol <i>saját</i> olasz japán orosz
<b>Importance</b>	nagy legnagyobb jelentős teljes ismert fontos kisebb önálló fő
<b>Position</b>	első egyik <i>című</i> egyes különböző második utolsó további elleni másik
<b>Cultural</b>	katolikus katonai politikai gazdasági zenei televíziós tudományos
<b>Affiliation</b>	nemzetközi nemzeti válogatott állami egyetemi városi műszaki
<b>Age</b>	új római királyi kis régi <i>jános</i> helyi erdélyi századi egykori modern
<b>Sports</b>	évi olimpiai nyári női legjobb országos európai budapesti téli bajnoki
<b>Misc.</b>	<i>nevű álló magyarországi lévő hagyományos található egész ún.</i>

**Table 7.** Adjective categories generated by an 8-way k-means clustering

clusters, such as Nationality and Position, clearly correspond to well-established categories. Given the capacity of the model and the very limited input data, we conjecture that what the embedding learned are indeed the adjective’s category (i.e. role or place in the NP) and not its full semantic representation. However, this requires formal validation.

Curiously, most categories do not appear at all, which is due to the frequency limit we employed to make our data interpretable. When no such limit is enforced, various other categories manifest, such as color, year, ordinals, etc. These usually cluster together really well, but we have yet to find the right cluster number or algorithm to make sure that each cluster is meaningful.

## 4.2 Ordering

To discover if an ordering exists for our clusters, we converted our virtual NPs to a graph. Each token was mapped to a vertex, and a directed edge  $e : v_1 \rightarrow v_2$  was put between vertices  $v_1$  and  $v_2$  iff the token corresponding to  $v_1$  directly precedes the one corresponding to  $v_2$  in any of the NPs. The weight of  $e$  equals to the number of NPs in which the connection was found.

Ideally, this graph would be a directed acyclic graph (DAG) with DET as the source and NOUN as the sink. In reality, the graph is full of cycles: for 132 797 vertices it has about 15k cycles of length 2 and 610k of length 3. An example of a 2-cycle is *idegen írású*, from the NPs<sup>13</sup>

- (2) a. *latin betűs írású idegen nyelveknek*  
 Latin letter writing foreign language.PLUR.DAT  
 ‘of foreign languages written in Latin script’

<sup>13</sup> As one reviewer rightly pointed out, having both “*idegen*” and “*írású*” on the same level in (2b) is a parsing error. Unfortunately, the dependency parser is far from perfect and it affects our results as well. Another class of errors we discovered with it is that sometimes the morphological and the dependency labels contradict, e.g. a [N] [Acc] word gets the SUBJ label.



- b. *idegen írású alakok*  
 foreign writing form.PLUR  
 ‘forms written in foreign writing (style)’

We took this “raw” graph and replaced each vertex with the cluster to which the associated token belongs; vertices not part of any of the 8 clusters above were removed, along with the edges connected to them. This resulted in a much simpler graph with 8 nodes. Unfortunately, this graph still contains cycles, so we iterated through the cycles and dropped the edge with the least weight from each until we were left with a DAG (all graph processing steps were done in NetworkX<sup>14</sup> (Hagberg et al., 2008)).

Topologically sorting the final graph (and disregarding *Misc.*) yielded the order *Position > Importance > Age > Nationality > Sports > Affiliation > Cultural*. Where the categories are comparable, our list matches the theories above. For instance, *Position > Importance > Age > Nationality* roughly corresponds to *ordinal > size > age > origin* in (Scott, 2002).

Hungarian	English
harmadik európai irodalmi	third European ... of literature
újabb híres	another famous
jelentős római	significant Roman
másik olasz állami	another Italian state
saját legjobb	own best
<i>japán olimpiai központi</i>	<i>Japanese Olympic central</i>
kisebb lengyel téli kereskedelmi	smaller Polish winter commercial

**Table 8.** Adjective sequences generated randomly from our categories

Table 8 presents a few adjective combinations generated randomly according to our ordering. All but one of the examples feel valid. The error comes from the *Sports > Affiliation* pair in our ordering, which is probably not entirely consistent. For instance, the sequence in question should be *központi olimpiai*, but the ordering is correct for *országos egyetemi*. We leave the task of finding out whether better or more fine-grained clustering could solve such problems for future work.

## 5 Conclusion and future work

In this paper, we have conducted an initial, “proof of concept” statistical study into various aspects of Hungarian word order. We suggested a basic machinery for acquiring frequency data on word order variations from a corpus. We have

<sup>14</sup> <https://networkx.org/>

shown that word order is dependent on the type of text (formal or informal) and the corpus. Our results show that the study of word order should not be confined to the field of theoretical linguistics. Our method could be used to collect statistics for e.g. text generation in a domain with no preexisting textual data, or data augmentation for machine learning.

We have devised an experimental method based on clustering of word embeddings for determining the order of adjectives in noun phrases. The method shows promising results, but the coverage and the quality of the clustering needs improvement.

There are several open avenues for further research. In this paper, we only considered very simple sentences with a single verbal predicate; future work should broaden the focus on one hand and produce more fine-grained statistics on the other. In particular, individual properties of verbs (such as the presence of preverbs, definiteness, or its semantics, i.e. position in an embedding space) might have a significant effect on the word order. Another possible future direction is linking our data to other fields of NLP such as language modeling or verb frame databases (e.g. Mazsola (Sass, 2018)), which would enable us to evaluate the impact of verbal constructions on word order.

We also intend to try and bridge the gap between our statistical approach and the theoretical work on the field. This would allow us to experimentally validate some of the theories or even to incorporate some of their predictions to our NLP toolchains.

## Acknowledgments

The author would like to thank the reviewers for their insightful comments.

## Bibliography

- Barta, Cs., Dormeyer, R., Fischer, I.: Word order and discontinuities in a dependency grammar for Hungarian. In: Proceedings of the 2nd Conf. on Hungarian Computational Linguistics (MSZNY), Szeged Hungary, Juhász Nyomda. pp. 19–27 (2004)
- Brassai, S.: Tapogatózások a magyar nyelv körül. Pesti Napló (1852–53)
- Cinque, G.: On the evidence for partial N-movement in the romance DP. In: Cinque, G., Koster, J., Pollock, J.Y., Rizzi, L., Zanuttini, R. (eds.) Paths towards universal grammar. Studies in honor of Richard S. Kayne, pp. 85–110. Georgetown University Press, Washington, DC (1994)
- Dékány, É.: The Hungarian Nominal Functional Sequence. Springer Nature (2021)
- Dékány, É., Hegedűs, V.: Word order variation in Hungarian PPs. Approaches to Hungarian 14, 95–120 (2015)
- Dixon, R.M.: Where have All the Adjectives Gone?: And Other Essays in Semantics and Syntax. De Gruyter Mouton (1982)

- É. Kiss, K.: Structural relations in Hungarian a ‘free’, word order language. *Linguistic Inquiry* 12, 185–213 (1981)
- É. Kiss, K.: Sentence structure and word order. In: *The syntactic structure of Hungarian*, pp. 1–90. Brill (1994)
- Fogarasi, J.: Euréka! *Atheneum* II 13, 193–198; 16, 241–249; 19, 289–297 (1838)
- Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference*. pp. 11–15 (2008)
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – the `emtsv` pipeline for Hungarian. In: *The 13th Linguistic Annotation Workshop* (8 2019)
- Kenesei, I.: On the logic of word order in Hungarian. In: Abraham, W., de Mey, S. (eds.) *Topic, Focus, and Configurationality*. Benjamins (1984a)
- Kenesei, I.: Word order in Hungarian complex sentences. *Linguistic Inquiry* 15(2), 328–342 (1984b)
- Laenzlinger, C.: French adjective ordering: Perspectives on DP-internal movement types. *Lingua* 115(5), 645–689 (2005)
- Lipták, A.: Word order in Hungarian exclamatives. *Acta Linguistica Hungarica* 53(4), 343–391 (2006), <http://www.jstor.org/stable/26190105>
- Luotolahti, J., Kanerva, J., Pyysalo, S., Ginter, F.: SETS: Scalable and efficient tree search in dependency graphs. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. pp. 51–55. Denver, Colorado (2015), <https://aclanthology.org/N15-3011>
- MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
- Miháltz, M., Váradi, T., Csertő, I., Fülöp, É., Pólya, T.: Beyond sentiment: Social psychological analysis of political facebook comments in hungary. In: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2015)*. ACL (2015)
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc. (2013), <https://bit.ly/39HikH8>
- Márton, J.: *Ungarische Grammatik, wodurch der Deutsche die ungarische Sprache richtig erlernen kann*. Wien (1805)
- Nagyházi, B.: Az egyszerű mondat szórendjének egy lehetséges tanítási modellje a magyar mint idegen nyelv oktatásában. Ph.D. thesis, Pécsi Tudományegyetem (2013)
- Nemeskey, D.M.: *Natural Language Processing Methods for Language Modeling*. Ph.D. thesis, Eötvös Loránd University (2020)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J.,

- Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
- Pléh, Cs.: The role of word order in the sentence interpretation of Hungarian children (1981)
- Puskás, G.: Word Order in Hungarian: The syntax of  $\bar{A}$ -positions. *Linguistik Aktuell/Linguistics Today*, John Benjamins Publishing Company (2000), <https://books.google.hu/books?id=aZY9AAAAQBAJ>
- Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. pp. 45–50. ELRA, Valletta, Malta (5 2010), <http://is.muni.cz/publication/884893/en>
- Sass, B.: Mazsola-mindenkinek. In: Vincze, V. (ed.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018). Szegedi Tudományegyetem Informatikai Tanszékcsoport (2018)
- Scott, G.J.: Stacked adjectival modification and the structure of nominal phrases. *Functional structure in DP and IP: The cartography of syntactic structures* 1, 91–120 (2002)
- Szalontai, Á., Surányi, B.: Word order effects of givenness in Hungarian. In: Hegedűs, V., Vogel, I. (eds.) *Approaches to Hungarian: Volume 16: Papers from the 2017 Budapest Conference*, pp. 138–163 (2020)
- Táncsics, M.: *Nyelvészet*. Pest (1833)

# BESZÉDTECHNOLÓGIA



# Length Analysis of Speech to be Recorded in the Recognition of Parkinson's Disease

Attila Zoltán Jenei<sup>1</sup>, Dávid Sztahó<sup>1</sup>

<sup>1</sup> Department of Telecommunications and Media Informatics,  
Budapest University of Technology and Economics,  
Magyar tudósok körútja 2, 1117 Budapest, Hungary  
jeneia@edu.bme.hu, sztaho.david@vik.bme.hu

**Abstract:** Parkinson's disease is an incurable neurodegenerative disease to the present clinical knowledge. It is diagnosed mostly by exclusion tests. Numerous studies have confirmed that speech can be promising to suspect the presence of the disease. On the other hand, just a few researches discuss the appropriate length of the speech sample or the contribution of parts of the full-length recordings in the classification. Hence, we partitioned each original recording into four shorter samples. We trained linear and radial basis function (rbf) kernel Support Vector Machine (SVM) models separately for original recordings, each partitioned group and all partitioned samples together. We found no significant difference between the results of the rbf kernel models. However, we obtained significantly better results with a portion of the entire speech using linear kernel models. In conclusion, even a shorter piece of a longer speech may be adequate for classification.

## 1 Introduction

Parkinson's disease (PD) is one of the most common neurodegenerative diseases described first by James Parkinson (Parkinson, 1817). The prevalence of PD is about 1-2 cases per 1000 worldwide. Nonetheless, cases in the 60+ population can exceed 100 per 1000 (Tysnes & Storstein, 2017). The aetiology of the disease is unknown however predisposing environmental and genetic factors may play a role in its development (Lindgren et al., 2005).

Pathologically, the emergence of PD is caused by the death of dopamine-producing neurons in the Substantia Nigra brain region. In addition, abnormal aggregation of alpha-synuclein protein (Lewy-bodies) is also observed (Simon et al., 2020). Dopamine is a neurotransmitter that serves as a messenger substance among nerve cells in the brain. It plays an important role in many everyday behaviours, including how we move, feel, or eat. It regulates movement and also supports the reward system (Wise, 2004).

The importance of PD connected researches is given by the fact that it is incurable according to current clinical knowledge. With therapy, medication, or deep brain stimulation, symptoms can be relieved and progression can be slowed. Therefore, it means a life-long procedure for the patient (Armstrong & Okun, 2020). As a result, it has significant cost implications for both the patient and the treating institution (Denisova et al., 2020).

In Parkinson's disease, both motor and non-motor symptoms may occur. Motor symptoms may include bradykinesia, muscle rigidity, resting tremor, and postural instability. The first three of these are considered to be the primary indicator of the disease. In addition, non-motor symptoms such as olfactory impairment, sleep dysfunction, and cognitive impairment may occur (Armstrong & Okun, 2020).

Because of these symptoms and the reasons listed above, it is crucial to recognize the disease as soon as possible. The appropriate therapy and medication could maintain the patient's quality of life and limits the progression of his disease. This is exacerbated by the fact that many other diseases exist with similar symptoms. Furthermore, many times it is not easy for a patient to get to a neurological examination. Because no clear diagnostic procedure is available to detect the disease, in most cases, the patient's history, laboratory tests, and other examinations can help rule out other diseases (Reichmann, 2010; Tolosa et al., 2006).

Several kinds of research focus on non-invasive modalities for the recognition of Parkinson's disease, such as imaging procedures, movement, drawing/handwriting, and speech analysis. From these, our present research focuses on text read aloud (bounded speech). Furthermore, there is a lot of research on recognizing Parkinson's disease from different types of speech types (e.g., persistent vowels, sentences, or spontaneous speech). However, there is less analysis on parts of longer speech recordings (detailed in Section 2). Within this scope, the effect of different parts of the read text on PD recognition is examined in this article. These results are also compared to the full-length recordings.

The structure of the article is the following: in Section 2 the literature related to the research is described, in Section 3 the methodology applied to the research is presented, in Section 4 the results are given, and in Section 5 conclusions are drawn from the results.

## 2 Related Work

Due to the reduced amount of dopamine, nerve conduction has a limited ability to function. Based on this, PD also affects the process of speech production (dysphonia) (Rusz et al., 2011). The voice of people with Parkinson's disease is typically low-volumed, with a tremor-like character, sudden stops, and starts may be present (Schulz & Grant, 2000).

There has been and there still is ongoing research into the use of these phenomena in speech. These are done with four types of speech databases: 1) persistent vowel, 2) pronunciation of words, syllables, 3) reading of bound texts, 4) spontaneous speech.

The significance of the sustained vowel was given by the fact that the formation of a vowel requires the active work of the muscles of the vocal cords. In Parkinson's disease, there is uncertainty/stuttering in muscle movement. One of the most commonly used vowels are /a/ (Tsanas et al., 2012), (Hemmerling & Sztahó, 2019), /e/ and /i/ (Vaiciukynas et al., 2017).

The use of syllables and words is widespread in the study of imprecise consonant formation (for example, the pronunciation of /pa-ta-ka/). This can be used to examine the sudden stops/starts of speech (Novotný et al., 2014).



One disadvantage of the former modalities is that they do not provide information about the continuous movement and functioning of the articulation. This is why the use of reading text and spontaneous speech has become widespread (Frid et al., 2014), (Kiss et al., 2018).

Based on these, it can be seen that many speech modalities and speech lengths are used in the recognition of Parkinson's disease. In the following research, several sustained vowels, words, and sentences were recorded and examined with the same model setting (Vadovsky & Paralic, 2017). With a Random Forest approach, 52.5%, 57.5% and 45% accuracy were achieved with sustained vowel /a/, /o/ and /u/, respectively. 67.2% of accuracy was obtained with words and 65% with sentences. The examinations were performed on 20 Parkinson's disease patients and 20 healthy subjects using 26 speech-based features.

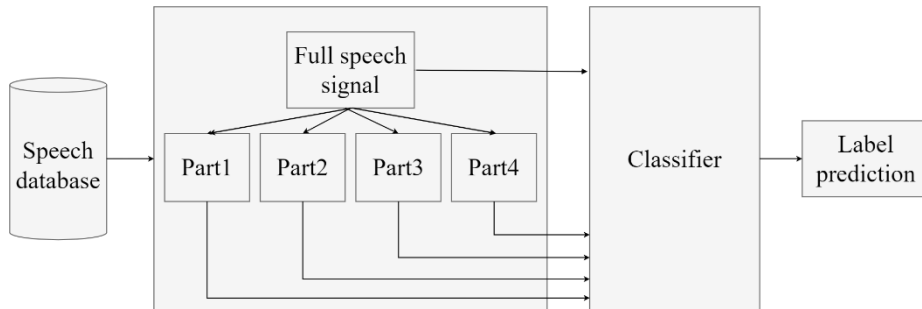
In the next study, sustained /a/ vowel, syllable repetition, words, short sentences, longer read text, and short spontaneous speech were examined separately (Sztahó et al., 2019). The research was performed using several speech descriptors involving 55 people with Parkinson's disease and 33 healthy individuals. Many classification algorithms have been tested. Of these, the linear kernel support vector machine (SVM) results are: 72.4% (sustained /a/), 77.0% (syllable triplets), 77.7% (words), 81.0% (sentences), 73.4% (read longer text) and 83.3% (short spontaneous speech) accuracy. Using radial basis function (rbf) kernel with the same SVM, the accuracy values are the following: 77.0% (sustained /a/), 78.2% (syllable triplets), 78.8% (words), 72.6% (sentences), 83.5% (read longer text) and 89.3% (short spontaneous speech).

From these results, prominent accuracy can be achieved with longer speech samples. As a form of long text is usually given as a phonetically rich tale, such as *The North Wind and Sun* or *Rainbow Passage*. However, there is no evident research in the literature that would examine the effect of different parts of these read texts for recognizing PD. Namely, if a part of the read text would provide similar recognition performance as the full-text version then only that part is sufficient to record. This would speed up the real examination procedure and would not be burdensome for the patient.

For the present study, the following hypotheses can be made according to the recent studies: Any part of the read text can achieve similar performance as the whole text.

### 3 Methodology

Fig. 1 illustrates the stepwise approach of the examination. First, speech recordings are selected from the speech database. In this case, Hungarian Parkinson Speech Database (HPSD) is used. Secondly, the full-length recordings were partitioned into four parts using segmentation files. Then, speech acoustic features were extracted from the full text and partitioned recordings. Finally, these features were the input for the classifier which conduct the label prediction after training.



**Fig. 1.** Flowchart of the examination tasks: speech database, pre-processing (partitioning the full-length recordings), classification and prediction with the trained classifier.

#### 3.1 Hungarian Parkinson Speech Database (HPSD)

79 recordings of patients with PD were selected from the HPSD. Speech acquisition was done in two health institutes in Budapest: Semmelweis University and Virányos Clinic. For the severity estimation, Hoehn & Yahr (H&Y) scale was used (Bhidayasiri & Tarsy, 2012). It defines non linearly progressive categories of motor functionality. Score 1 means unilateral involvement with minimal or no functional disability while 5 means the most severe case (bed or wheelchair bounded). Using such an estimator, 38 females (average age and standard deviation:  $65 \pm 9.4$ ) had an average of 2.8 ( $\pm 1.1$ ) and 41 males (average age and standard deviation:  $64 \pm 9.3$ ) had an average of 2.7 ( $\pm 1.1$ ) H&Y scores.

Speech of healthy population (HC) was also recorded using the same text and environmental properties. 41 females (average age and standard deviation:  $53 \pm 15.9$ ) and 38 males (average age and standard deviation:  $52 \pm 16.4$ ) were selected to balance the PD class sample size. Healthy subjects reported that they did not have any speech-related illnesses.

The text read was a tale called *The North Wind and the Sun*. External USB sound card (Terratec 6fire USB) with A/D converter was used to acquire recordings. A clip-on condenser microphone (Audio-Technica ATR3350) in a quiet office (or in a medical office) environment was applied. The recordings were stored in Pulse-code modulation (PCM) audio coding with a sampling frequency of 44.1 kHz and quantization of 16 bit. Subjects all consented to the use of their speech recordings for research purposes.

### 3.2 Speech Signal Pre-processing and Partitioning

The text of the entire tale contained 8 sentences, including also the title of the tale. This was chunked into 4 parts using Praat (Boersma & van Heuven, 2001) and annotation files available. This was done by examining the phonemes of the last word of the sentence before the place of cut. Since it was a bound text, these phonemes are in order and can be found and cuts can be performed. Using this method, four recordings were available for each person: 1) title and first sentence, 2) second and third sentences, 3) fourth longer sentence, and 4) fifth-seventh shorter sentences. Table 1 summarizes the average length and its standard deviation of recordings in seconds.

It should also be noted that no automatic chunking was made in the recording if the text was not read correctly (the phoneme set and order of the last words differed from the bound text ones). In that case, manual correction (manual chunk) was required.

After chunking the recordings, full and partitioned samples were resampled in 16 kHz sampling frequency and were normalized to the peak value in the signal. After that, the first 12 Mel Frequency Cepstral (MFC) coefficients were extracted from each sample using the Surfboard python library (Lenain et al., 2020). MFCCs were chosen according to the wide usage in the literature (Dasgupta et al., 2017; Godino-Llorente et al., 2017; Pompili et al., 2017). The mean, standard deviation, mean of the first derivative and standard deviation of the first derivative were calculated for the 12 MFCCs resulted in 48 features overall. Finally, the features were scaled between -1 and 1.

Table 1: The average length [sec.] and standard deviation of recordings separately for HC and PD classes.

	Full	Part1	Part2	Part3	Part4
HC	44.8±5.9	9.1±2.7	9.7±1.5	10.2±2.6	12.8±1.8
PD	59.1±23.4	13.3±4.0	10.6±3.6	15.6±4.9	14.1±4.5

### 3.3 Classification and model evaluation

The SVM classifier was chosen for this examination based on its widespread usage in speech-based researches (Pah et al., 2021; Sonawane & Sharma, 2021). SVM models were deployed in python (version 3.6) using *sklearn* machine learning module. Parameters were chosen as commonly used default values:  $C=1.0$ ,  $\gamma=1/(n_{features} \cdot var_X)$  where  $n_{features}$  is the number of features and  $var_X$  is the variance of the training data. For the experimentation, linear and rbf kernels were performed.

Six model training and testing scenarios were performed: (1) on full-length recordings, (2)-(5) on partitioned samples separately, (6) on partitioned samples together.

Speaker-wise leave-one-out cross-validation (LOOCV) was applied for model evaluation. With this method, samples of one speaker were grouped for testing and samples of the remaining speakers were grouped for training. This was repeated until each speaker in the dataset became test item. The performances were measured on the test items after the LOOCV. Speaker-wise separation was important for model (6), because multiple samples were available for each speaker. In this case, the majority

decision was resolved by having at least two positive samples (from four) to get a positive label for the test subject. To measure the performance of the models, sensitivity, specificity, accuracy, f1-score, and Matthews correlation were computed.

Furthermore, Mann–Whitney U non-parametric statistical test was performed to check the significance between the results of the models. With this test, we examined whether the performance of the model trained on the partitioned samples differ significantly from the performance of the model trained on the full recordings. Originally, the null hypothesis state that “*the two independent groups are homogeneous and have the same distribution*” (Nachar, 2008). So significant difference may be observed if the p-value of the statistics is lower than the significance level. So the alternative hypothesis can be stated as *the two independent groups are not homogeneous and have different distributions*. The significance level was set to 0.05, which is the most commonly used threshold in statistical studies.

## 4 Results

### 4.1 Result of Full-Length Recordings

For this experiment, SVM models with linear and rbf kernels were trained and tested with full-length recordings. The performance metrics can be seen in Table 2. The columns show the following metrics for both linear and rbf kernel classification: sensitivity (‘sens’), specificity (‘spec’), accuracy (‘acc’), f1-score (‘f1’), and Matthews correlation (‘mc’).

Table 2: Results of full-length recordings with linear and rbf kernels SVM models.

kernel	sens	spec	acc	f1	mc
linear	78.5%	79.7%	79.1%	79.0%	0.58
rbf	83.5%	84.8%	84.2%	84.1%	0.68

Based on Table 2, rbf kernel SVM achieved higher performance on all evaluation metrics. The average deviation on the metrics is 5.1%, while on the mc values it is 0.1. With an SVM with an rbf kernel, the algorithm recognized 8 speakers more correctly (true positive or true negative together) than using a linear kernel model.

### 4.2 Result of Partitioned Samples on Separate Models

For this examination, linear and rbf kernels SVM models were trained and tested with partitioned recordings separately. The performance metrics can be seen in Table 3. The columns represent the same metrics as in Table 2. *PartX* corresponds to the chunked segments (described in Section 3.2).

Based on Table 3, linear kernel models trained on part1 and part3 samples achieved the highest metric values (acc.: 86.1% and 88.0%, f1: 86.6% and 88.2%). Linear kernel SVM trained on part2 samples resulted in the lowest results (acc: 77.6%, f1: 77.7%). Using part4 samples with linear kernel SVM model, an intermediate result can be obtained (acc.: 81.6%, f1: 82.0%).

Table 3: Results of partitioned samples on separate SVM models with linear and rbf kernels.

kernel	samples	sens	spec	acc	f1	mc
linear	part1	89.9%	82.3%	86.1%	86.6%	0.72
	part2	78.1%	77.2%	77.6%	77.7%	0.55
	part3	89.9%	86.1%	88.0%	88.2%	0.76
	part4	83.5%	79.7%	81.6%	82.0%	0.63
rbf	part1	88.6%	83.5%	86.1%	86.4%	0.72
	part2	81.0%	83.5%	82.3%	82.1%	0.65
	part3	87.3%	84.8%	86.1%	86.2%	0.72
	part4	83.5%	83.5%	83.5%	83.5%	0.67

Compared to the whole recordings’ results, linear models trained on the parts classified 11 (part1), 14 (part3) and 4 (part4) more samples correctly. In the case of part2, the model predicted 2 samples less correctly than the full-length linear model.

A similar tendency was earned with rbf kernel SVM with a narrower fluctuation in the results. Using part1 and part2 the highest performances were achieved (acc.: 86.1% and 86.1%, f1: 86.4% and 86.2%). With part2 samples, the lowest metrics were obtained (acc: 82.3%, f1: 82.1%) close to the result earned with part4 samples (acc.: 83.5%, f1: 83.5%). Compared to the whole recordings’ results, the rbf models trained on parts classified 3 (part1 and part3) more samples correctly. In the case of part2 and part4, the models classified 3 (part2) and 1 (part4) fewer samples correctly than the full-length rbf model.

### 4.3 Result of Partitioned Samples Together in a Single Model

In this examination, all partitioned samples for each speaker were involved in the training and testing process. Overall, 628 samples’ features of 157 subjects were used to train SVM models and one subject with four samples was used to test the performance in each cross-validation cycle. The results can be seen in Table 4. The metrics correspond to Table 3 and Table 2. MV (majority voting) means the aggregation of predictions per individuals and no MV means the results without speaker-wise aggregation.

Table 4: Results of partitioned samples together using linear and rbf kernel SVMs.

kernel	MV	sens	spec	acc	f1	mc
linear	MV	82.1%	81.3%	81.6%	81.5%	0.63
	no MV	82.6%	81.3%	82.0%	82.1%	0.64
rbf	MV	85.0%	85.9%	85.4%	85.5%	0.71
	no MV	83.5%	84.2%	83.9%	83.8%	0.68

Based on Table 4, the results of linear and rbf kernel SVM models are similar in accuracy, f1-score and Matthews correlation. The linear kernel SVM model achieved higher performance using multiple samples and majority voting than the original full-

length recordings model (Table 2). However, majority voting did not improve the results compared to the experiment without majority voting. With rbf kernel SVM, similar results were obtained with multiple samples and majority voting as in Table 2. In this case, a slight improvement can be realized with MV (from 83.9% to 85.4% (acc), from 83.8% to 85.5% (f1)).

The difference between the two results (no MV and with MV) is one speaker for the linear model and two speakers for rbf model. Furthermore, the samples clearly defined the category (HC or PD) for the individuals. This means that three or four of four-part samples were decided with the same class label. There was no indefinite decision (2 samples HC and 2 samples PD) neither for linear and nor rbf models. It also means that if the model categorized an individual to the wrong category, it did so for the most part and not because of an indecisive situation.

### 4.3 Significance Analysis Between Different Samples Trained Models

For each model, the output labels are given for the test elements as binary variables: 0 (sample is categorized to HC) and 1 (sample is categorized to PD). For all the test elements per model, a list of binary variables has resulted. That two lists can be compared to each other with an appropriate independent two-sample test. According to this, Mann–Whitney U non-parametric test was chosen to measure whether the models perform significantly to each other.

The p-values of corresponding models can be seen in Table 5 for linear kernel models and Table 6 for rbf kernel models. Notations part1, part2, part3 and part4 correspond to the partitioned samples' models, MV represents the majority version of all partitioned samples trained model, and full means the model trained on full-length recordings. An arbitrary row and column designate the p-value resulting from a comparison of the two models. Based on this property, the main diagonal would include the p-value of the same models, which is cleared out. Furthermore, the table is symmetric to the main diagonal. The p-values smaller from the significance level are highlighted.

Table 5: p-values of linear kernel models' performances.

	part1	part2	part3	part4	MV	full
part1	-	<b>0.026</b>	0.308	0.143	0.143	0.052
part2	<b>0.026</b>	-	<b>0.008</b>	0.189	0.201	0.376
part3	0.308	<b>0.008</b>	-	0.059	0.059	<b>0.017</b>
part4	0.143	0.189	0.059	-	0.500	0.286
MV	0.143	0.201	0.059	0.500	-	0.286
full	0.052	0.376	<b>0.017</b>	0.286	0.286	-

If the model trained and tested with full-length recordings is considered as a baseline (last row or last column). The model trained and tested with the third partitioned samples is different from this baseline significantly. This deviation is positive as the model on part3 achieved higher performance (acc.: 88.0%, f1: 88.2%) than the model on full-length recordings (acc.: 79.0%, f1: 79.1%). In contrast, a model trained on part2

obtained lower result (acc.: 77.6%, f1: 77.6%) significantly compared to the part1 (acc.: 86.1%, f1: 86.6%) and part3 (acc.: 88.0%, f1: 88.2%) partitioned samples based models.

According to Table 6, no significant difference can be observed between the models' performances. This implies that an arbitrary model can be deployed to recognize PD because it will result in the same performance metrics statistically as the others.

Table 6: p-values of rbf models' performances.

	part1	part2	part3	part4	MV	full
part1	-	0.178	0.500	0.266	0.437	0.318
part2	0.178	-	0.178	0.383	0.223	0.326
part3	0.500	0.178	-	0.266	0.437	0.318
part4	0.266	0.383	0.266	-	0.321	0.440
MV	0.437	0.223	0.437	0.321	-	0.378
full	0.318	0.326	0.318	0.440	0.378	-

## 5 Discussion and conclusion

Based on the literature, many speech modalities can be used for the recognition of Parkinson's disease. Each modality has its advantages and disadvantages. Much of the research uses the longer timespan read text. However, less research is focused on the optimal or efficient length of recording. Furthermore, it may also be questionable which part of a given read text is most significant in recognizing the disease.

Therefore, we aimed the following hypotheses with the present research: Can a similar classification performance be achieved with partitioned samples based models than with full-length recordings based models.

Based on these, we partitioned the nearly one-minute original recordings into four shorter samples using annotation files. Then we created independent SVM models for the original and fragmented recordings. 12 MFCCs were extracted as features for the models. Argument in using only 12 MFCC features is that it is a desirable approach to make our model feasible for real-time implementation. The less features decrease the processing time. However, it is worth exploring results with additional speech-related features (Sztahó et al, 2017), (Sztahó & Valálik, 2019), (López et al, 2019). For evaluation, leave-one-speaker-out cross-validation (LOOCV) was performed. We conducted the following experiments: 1) using full-length recordings to train and test SVM models (baselines), 2) using partitioned samples separately to train and test SVM models, 3) using partitioned samples together to train and test SVM models. In future experiments, it would be also beneficial to explore other technologies (for example deep learning) (Faiyaz et al., 2020), (Johri & Tripathi, 2019). Finally, Mann-Whitney U non-parametric test was applied to test whether there are significant differences between the models' performances.

Using full-length recordings, the rbf kernel SVM achieved better performance (acc.: 84.2%, f1: 84.1%) to classify PD samples than the linear kernel SVM (acc: 79.1%, f1: 79.0%). The results differences were about 5% for all descriptor metrics, and 0.1 for

the Matthews correlation. That implies these features may have some nonlinear behaviour which can improve the classification performance.

Applying partitioned samples separately, similar results were achieved with both linear and rbf kernel SVM models. The models with rbf kernel produced results in a narrower deviation along with different parts, while the deviation of models with linear kernel had a wider range. For both linear and rbf kernel models, part1 and part3 samples were able to obtain even higher metric values than the baselines had. Furthermore, the models with the lowest results (part2 for both rbf and linear kernel models) are almost as high as the full-length versions. The explanation may be that if there is a part of the full-length recording that would produce a worse result, it can also degrade the results of the full recording. Conversely, it can also be said that we have seen parts that perform better and also parts that perform worse with this fragmentation technique. Some sort of weighted average may be the result of the entire recordings. With this in mind, a procedure can be developed in which just the right amount of text and the right content needs to be read aloud, and the results remain appropriate as the full text would be read aloud.

It can be mentioned that the linear SVM outperformed the rbf SVM in the part3 case. This may be explained by the lack of parameter optimization. So the linear SVM had a better default setting for this problem than the rbf SVM. Alternatively, it is conceivable that some features showed a more linear character for that part of the given recording, which the linear SVM could take advantage of. In any case, this phenomenon requires further investigation.

If all the partitioned recordings were used together, the result of the linear kernel model (acc: 82.0%, f1: 82.1%) is improved compared to the full-length model (acc: 79.1%, f1:79.0%). With rbf kernel model, the same results were obtained with all samples (acc: 83.9%, f1:83.8%) than using full-length recordings (acc: 84.2%, f1: 84.1%). The metric values experienced with the use of majority voting resulted in almost the same results as without majority voting.

From the statistical examination, it can be concluded that using a linear kernel, a significant improvement can be achieved with the part3 samples compared to the full-length study (using the Mann-Whitney U test, with a significance level of 0.05). The study also points out that the results obtained on the partitioned samples even show significant differences between them (part 2 case). Moreover, non-significant p values indicate that homogeneous results can be obtained with various speech lengths. This thus confirms that even shorter recordings may be sufficient to detect PD. Statistical analysis of rbf kernel SVM models showed no significant results. This confirms the statement stated formerly.

In summary, the results indicated that a suitable piece of longer speech recording may be sufficient to obtain the same classification performance as with full-length recordings (with a significance score of 0.05). Moreover, even a piece can be found at certain model settings that can achieve significantly better results than the full-length recordings. According to this, a shorter piece would be enough to record from the patient to examine. This is also more convenient and less burdensome for the patient.

In the future, for a more detailed analysis on result differences obtained with different parts of the speech samples, a complete investigation is advisable on exactly how these parts differ from each other (by linguistic-acoustic content).



## Acknowledgements

Project no. K128568 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K\_18 funding scheme. The research was partly funded by the CELSA (CELSA/18/027) project titled: “Models of Pathological Speech for Diagnosis and Speech Recognition.

## Bibliography

- Armstrong, M. J., Okun, M. S.: Diagnosis and Treatment of Parkinson Disease: A Review. In: *JAMA*, 323(6), pp. 548. (2020) <https://doi.org/10.1001/jama.2019.22360>
- Bhidayasiri, R., Tarsy, D.: Parkinson’s Disease: Hoehn and Yahr Scale. In: R. Bhidayasiri & D. Tarsy (Eds.), *Movement Disorders: A Video Atlas: A Video Atlas* (pp. 4–5). Humana Press. (2012) [https://doi.org/10.1007/978-1-60327-426-5\\_2](https://doi.org/10.1007/978-1-60327-426-5_2)
- Boersma, P., van Heuven, V.: Praat, a system for doing phonetics by computer. In: *Glott International* 5(9/10), pp. 341–345. (2001) *Online* [Accessed: 3 Nov. 2021] [https://www.fon.hum.uva.nl/paul/papers/speak/speakPraat\\_glot2001.pdf](https://www.fon.hum.uva.nl/paul/papers/speak/speakPraat_glot2001.pdf)
- Dasgupta, S., Harisudha, K., Masunda, S.: Voiceprint analysis for Parkinson’s disease using MFCC, GMM, and instance based learning and multilayer perceptron. In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 1679–1682. (2017) <https://doi.org/10.1109/ICPCSI.2017.8391999>
- Denisova, I. A., Chubarova, T. V., Bogatova, I. E., Vartanov, S. A., Kucheryanu, V. G., Polterovich, V. M., Turdyeva, N. A., Shakleina, M. V.: Estimating economic efficiency of preclinical diagnostics of Parkinson’s disease with cost-utility approach. In: *Population and Economics*, 4(3), pp. 111–127. (2020) <https://doi.org/10.3897/popecon.4.e59949>
- Faiyaz, A., Danish, R., Iqra, N., Mumtaz, A.: An LSTM based Deep learning model for voice-based detection of Parkinson’s disease. In: *International Journal of Advanced Science and Technology*, 29(5s), pp. 337 - 343. (2020) Retrieved from <http://serse.org/journals/index.php/IJAST/article/view/7166>
- Frid, A., Safra, E., Hazan, H., Lokey, L., Hilu, D., Manevitz, L., Ramig, L., Sapir, S.: Computational Diagnosis of Parkinson’s Disease Directly from Natural Speech Using Machine Learning Techniques. In: *2014 IEEE International Conference on Software Science, Technology and Engineering*, pp. 50–53. (2014) <https://doi.org/10.1109/SWSTE.2014.17>
- Godino-Llorente, J. I., Shattuck-Hufnagel, S., Choi, J. Y., Moro-Velázquez, L., Gómez-García, J. A.: Towards the identification of Idiopathic Parkinson’s Disease from the speech. New articulatory kinetic biomarkers. In: *PLOS ONE*, 12(12), pp. e0189583. (2017) <https://doi.org/10.1371/journal.pone.0189583>
- Hemmerling, D., Sztahó, D.: Parkinson’s Disease Classification Based on Vowel Sound. In: *11th International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA)*, pp. 29-32, Firenze, Italy, (2019) <https://doi.org/10.36253/978-88-6453-961-4>
- Johri, A., Tripathi, A., K.: Parkinson Disease Detection Using Deep Neural Networks. In: *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1-4. (2019) <https://doi.org/10.1109/IC3.2019.8844941>
- Kiss, G., Takács, A., B., Sztahó, D., Vicsi, K.: Detection Possibilities of Depression and Parkinson’s disease Based on the Ratio of Transient Parts of the Speech. In: *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 165–168. (2018) <https://doi.org/10.1109/CogInfoCom.2018.8639901>
- Lenain, R., Weston, J., Shivkumar, A., Fristed, E.: Surfboard: Audio Feature Extraction for

- Modern Machine Learning. In: *Interspeech 2020*, pp. 2917–2921. (2020) <https://doi.org/10.21437/Interspeech.2020-2879>
- Lindgren, P., von Campenhausen, S., Spottke, E., Siebert, U., Dodel, R.: Cost of Parkinson's disease in Europe. In: *European Journal of Neurology*, 12(s1), pp. 68–73. (2005) <https://doi.org/10.1111/j.1468-1331.2005.01197.x>
- López, J., V., E., Orozco-Arroyave, J., R., Gosztolya, G.: Assessing Parkinson's Disease from Speech Using Fisher Vectors. Proc. In: *Interspeech 2019*, pp. 3063–3067, (2019) <https://doi.org/10.21437/Interspeech.2019-2217>
- Nachar, N.: The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. In: *Tutorials in Quantitative Methods for Psychology*, 4(1), pp. 13–20. (2008) <https://doi.org/10.20982/tqmp.04.1.p013>
- Novotný, M., Rusz, J., Čmejla, R., Růžička, E.: Automatic evaluation of articulatory disorders in Parkinson's disease. In: *IEEE Transactions on Audio, Speech and Language Processing*, 22(9), pp. 1366–1378. (2014) <https://doi.org/10.1109/TASLP.2014.2329734>
- Pah, N. D., Motin, M. A., Kumar, D. K.: Voice Analysis for Diagnosis and Monitoring Parkinson's Disease. In: S. P. Arjunan & D. K. Kumar (Eds.), *Techniques for Assessment of Parkinsonism for Diagnosis and Rehabilitation* (pp. 119–133). Springer Singapore. (2021) [https://doi.org/10.1007/978-981-16-3056-9\\_8](https://doi.org/10.1007/978-981-16-3056-9_8)
- Parkinson, J.: An essay on the shaking palsy. 1817 In: *The Journal of Neuropsychiatry and Clinical Neurosciences*, 14(2). (2002) <https://doi.org/10.1176/jnp.14.2.223>
- Pompili, A., Abad, A., Romano, P., Martins, I. P., Cardoso, R., Santos, H., Carvalho, J., Guimarães, I., Ferreira, J. J.: Automatic Detection of Parkinson's Disease: An Experimental Analysis of Common Speech Production Tasks Used for Diagnosis. In: K. Ekštejn & V. Matoušek (Eds.), *Text, Speech, and Dialogue* (Vol. 10415, pp. 411–419). Springer International Publishing. (2017) [https://doi.org/10.1007/978-3-319-64206-2\\_46](https://doi.org/10.1007/978-3-319-64206-2_46)
- Reichmann, H.: Clinical Criteria for the Diagnosis of Parkinson's Disease. In: *Neurodegenerative Diseases*, 7(5), pp. 284–290. (2010) <https://doi.org/10.1159/000314478>
- Rusz, J., Čmejla, R., Ruzickova, H., Růžička, E.: Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated Parkinson's disease. In: *The Journal of the Acoustical Society of America*, 129, pp. 350–367. (2011) <https://doi.org/10.1121/1.3514381>
- Schulz, G. M., Grant, M. K.: Effects of speech therapy and pharmacologic and surgical treatments on voice and speech in Parkinson's disease: A review of the literature. In: *Journal of Communication Disorders*, 33(1), pp. 59–88. (2000) [https://doi.org/10.1016/S0021-9924\(99\)00025-8](https://doi.org/10.1016/S0021-9924(99)00025-8)
- Simon, D. K., Tanner, C. M., Brundin, P.: Parkinson Disease Epidemiology, Pathology, Genetics, and Pathophysiology. In: *Clinics in Geriatric Medicine*, 36(1), pp. 1–12. (2020) <https://doi.org/10.1016/j.cger.2019.08.002>
- Sonawane, B., Sharma, P.: Speech-based solution to Parkinson's disease management. In: *Multimedia Tools and Applications*, 80(19), pp. 29437–29451. (2021) <https://doi.org/10.1007/s11042-021-11061-1>
- Sztahó, D., Valalik, I., Vicsi, K.: Parkinson's Disease Severity Estimation on Hungarian Speech Using Various Speech Tasks. In: *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pp. 1–6. (2019) <https://doi.org/10.1109/SPED.2019.8906277>
- Sztahó, D., Tulics, M. G., Vicsi, K., Valalik, I.: Automatic estimation of severity of Parkinson's disease based on speech rhythm related features. In: *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 11–16. (2017) <https://doi.org/10.1109/CogInfoCom.2017.8268208>
- Sztahó, D., Valalik, I.: Speech Fluency Measurement of Patients with Parkinson's Disease by Forward-Backward Divergence Segmentation. In: *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 295–300. (2019)

- <https://doi.org/10.1109/CogInfoCom47531.2019.9090001>
- Tolosa, E., Wenning, G., Poewe, W. (2006). The diagnosis of Parkinson's disease. *The Lancet Neurology*, 5(1), pp. 75–86. [https://doi.org/10.1016/S1474-4422\(05\)70285-4](https://doi.org/10.1016/S1474-4422(05)70285-4)
- Tsanas, A., Little, M., A., McSharry, P., E., Spielman, J., Ramig, L., O.: Novel Speech Signal Processing Algorithms for High-Accuracy Classification of Parkinson's Disease. In: *IEEE Transactions on Biomedical Engineering*, 59(5), pp. 1264–1271. (2012) <https://doi.org/10.1109/TBME.2012.2183367>
- Tysnes, O.-B., Storstein, A. (2017). Epidemiology of Parkinson's disease. *Journal of Neural Transmission*, 124(8), pp. 901–905. <https://doi.org/10.1007/s00702-017-1686-y>
- Vadovsky, M., & Paralic, J.: Parkinson's disease patients classification based on the speech signals. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, pp. 321–326. (2017) <https://doi.org/10.1109/SAMI.2017.7880326>
- Vaiciukynas, E., Verikas, A., Gelzinis, A., & Bacauskiene, M.: Detecting Parkinson's disease from sustained phonation and speech signals. In: *PLOS ONE*, 12(10), pp. e0185613. (2017) <https://doi.org/10.1371/journal.pone.0185613>
- Wise, R. A.: Dopamine, learning and motivation. In: *Nature Reviews Neuroscience*, 5(6), pp. 483–494. (2004) <https://doi.org/10.1038/nrn1406>



# Sclerosis Multiplex hangalapú felismerése akusztikai alapú beágyazások használatával

Gosztolya Gábor<sup>1,2</sup>, Tóth László<sup>1</sup>, Svindt Veronika<sup>3</sup>,  
Bóna Judit<sup>4</sup>, Hoffmann Ildikó<sup>3,5</sup>

<sup>1</sup>Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup>ELKH-SZTE Mesterséges Intelligencia Kutatócsoport, Szeged

<sup>3</sup>ELKH Nyelvtudományi Kutatóközpont, Budapest

<sup>4</sup>Eötvös Loránd Tudományegyetem,

Alkalmazott Nyelvészeti és Fonetikai Tanszék, Budapest

<sup>5</sup>Szegedi Tudományegyetem, Magyar Nyelvészeti Tanszék

ggabor @ inf.u-szeged.hu

**Kivonat** A sclerosis multiplex (SM) a központi idegrendszer krónikus gyulladással megbetegedése. Mivel az SM nyelvi zavarokkal is együtt járhat, az automatikus beszédelemzés hasznosnak bizonyulhat akár az alig észrevehető beszédprodukciós változások detektálására is. Egy lényeges technikai kérdés ugyanakkor, hogy milyen jellemzőket érdemes kinyerni az alanyok beszédéből, melyeket aztán a gépi tanulási osztályozási lépés során használhatunk. Jelen cikkünkben HMM/DNN hibrid modellek mély neurális hálós akusztikus modelljeiből nyerünk ki aktivációkat, majd ezeket a teljes hangfelvételen többféleképpen összegezve (pl. átlag, szórás) használjuk jellemzőként. Kísérleteinket 23 SM alany és 22 egészséges kontroll személy négy-négy hangfelvételén végezzük. Eredményeink alapján a javasolt eljárás hatékonyabb azonosítást tesz lehetővé, mint az ugyanazon beszédadaton tanított x-vektor technika. Az elért AUC értékek tapasztalataink alapján nagyban függenek a beágyazás forrásrétegétől és a felvételszintű összegzés módjától is. A legjobb értékek az egyes beszédfeladatokon 0,824 és 0,911 közé estek.

**Kulcsszavak:** sclerosis multiplex, beszédelemzés, mély neurális hálók, beágyazások

## 1. Bevezetés

A sclerosis multiplex (SM) a központi idegrendszer krónikus gyulladással megbetegedése. A betegségnek három fő típusa különböztethető meg: relapszáló-remittáló SM (tünetes- és tünetmentes állapotok váltakozása), másodlagos progresszív SM (tünetek állandósulása és folyamatos romlás) és elsődleges SM (folyamatos romlás a tünetek első fellépésétől). Változatos idegrendszeri háttere miatt az SM tünetei jellegükben és súlyosságukban sokfélék lehetnek (Szirmai, 2006). Az SM a motoros és a kognitív funkciók érintettsége mellett nyelvi zavarokkal is együtt járhat.

A sclerosis multiplex-szel együtt járó nyelvi- és beszédzavaroknak a kognitív képességek változásaival való összefüggésének kutatása a hazai és a nemzetközi szakirodalomban is újszerű, korábban alig vizsgált terület. Ugyanakkor az SM-mel élő személyek több, mint egyharmada számol be ilyen jellegű zavarokról, emellett a betegség 60-70%-ban átmeneti vagy tartós mentális és/vagy kognitív korlátozottsággal jár együtt (pl. munkamemória és végrehajtó funkciók zavara, téri tájékozódás zavara, krónikus fáradtság). A nyelv- és beszéd folyamatok a kognitív, a szenzoros és a motoros működés dinamikus összehangolását kívánják meg. A meglassult információfeldolgozási sebesség hatással van a nyelvi- és beszéd folyamatokra, változások jelenhetnek meg a beszédpercepció és/vagy a beszédprodukciónak a folyamatában, illetve valamennyi nyelvi szintet érintheti (Bóna és mtsai, 2020; Svindt és mtsai, 2020; Renaud és mtsai, 2016). Habár csak a betegek egyharmadánál diagnosztizálnak dizartriát, az automatikus beszédelemzés így is hasznosnak bizonyulhat a megelőző, alig észrevehető beszédprodukciónak a változások detektálására (Mulfari és mtsai, 2021), mellyel lehetővé válhat akár a betegség korai észlelése vagy súlyosbodásának detektálása is.

Egy ilyen automatikus beszédelemző eljárás kulcskérdése általában az, hogy az alanyok beszédéből milyen gépi tanulási jellemzőket nyerünk ki. Az orvosi beszédfeldolgozásban bevett megközelítésnek számít, hogy jellemzőink nem elsődlegesen az adott betegségre specifikusak, hanem (ilyen értelemben) általánosnak számítanak (García és mtsai, 2018; Jenei és Kiss, 2020). A jellemzőkinyerési technika kiválasztásánál figyelembe kell venni a területen tipikus adatszűkösséget is: mivel a hangfelvételek gyűjtése nehézkes és nagyon munkaigényes (a felvételek rögzítése adott protokollt követve történik, az alanyokat diagnosztizálni kell, stb.), az adatbázisok terjedelme jellemzően a beszéd felismerésben megszokottak töredéke, legföljebb néhány órát tesz ki.

Ezt figyelembe véve az általános jellegű és statisztikai jellemzőkinyerési eljárások általában egy általános, beszéd felismerési feladatra szánt korpuszt is használnak. Például az *i*-vektorok (Dehak és mtsai, 2009) esetén, melyek Univerzális háttérmodellje (Universal Background Model, UBM) gyakorlatilag egy keretszinten tanított Gaussi keverékmodell (Gaussian Mixture Model, GMM), ez a GMM főnnakadás nélkül tanítható egy ilyen általános jellegű adatbázison. Hasonló a helyzet a *d*-vektorokkal (Variani és mtsai, 2014) és az *x*-vektorokkal (Snyder és mtsai, 2018) is: ezek a mély neurális hálón (Deep Neural Network, DNN) alapuló eljárások modelljeinek tanítása is jellemzően egy külső, jelentősen nagyobb korpuszon történik. (Bár az *i*-vektorok, *d*-vektorok és *x*-vektorok eredetileg a beszélőazonosítási feladat megoldására lettek kifejlesztve, számos tanulmányban használják ezeket az eljárásokat nemverbális, vagy kimondottan orvosi jellegű problémákon, jellemzőkinyerő eszközként (Grzybowska és Kacprzak, 2016; Huckvale és mtsai, 2020; Egas-López és mtsai, 2021).)

Ezek az alkalmazások tekinthetők úgy is, hogy valamiképpen fölépítjük a „standard beszéd” egy modelljét, és a jellemzőkinyerési lépés során azt próbáljuk kifejezni, hogy az adott hangfelvételen hallható beszédjel miben és mennyiben különbözik ettől a „standard beszéd-től”. Természetesen az alkalmazott modellben (akár gyökeresen is) különböznek a felsorolt módszerek, mint ahogy a tanítás

módjában és az ahhoz szükséges annotáció jellegében is. Az  $i$ -vektorok háttérmodelljének tanításához (mivel keretszintű jellemzőkre illesztett GMM-ről van szó) semmilyen annotáció nem szükséges. Az  $x$ -vektorok esetén a tanítási cél az aktuális beszélő azonosítója (sorszama); ennek rendelkezésre állása ugyanakkor a gyakorlatban nem igazán szigorú követelmény, ez szinte minden korpusznál megtalálható. Az  $x$ -vektorok modellje egy speciális struktúrájú neurális háló, melynek alsóbb rétegei keretszinten, felsőbb rétegei az egész felvétel szintjén működnek, a kettő között pedig egy speciális összegző réteg található. Így a háló gond nélkül tanítható csupán keretszintű jellemzőkre (pl. MFCC-kre) és felvételszintű célértékekre. Maguk az  $x$ -vektor jellemzők valamely szegmensszintű réteg aktivációiként állnak elő. A  $d$ -vektorok bizonyos szempontból az  $x$ -vektorok előzményeinek tekinthetők: esetükben a neurális hálókat hagyományos módon, keretszinten tanítjuk, bár a keretszintű címkéket (az aktuális beszélő azonosítóját) a felvételszintű címkézésből vesszük.

Vegyük észre, hogy a fősorolt jellemzőkinyerő eljárások mindegyikénél szükség van valamilyen speciális lépésre, például egy GMM vagy egy (akár egyedi struktúrájú) neurális háló tanítására. Ugyanakkor keretszintű DNN akusztikus modellek (a beszédfelismerési terület hagyományos HMM/DNN hibrid modelljéből) elég könnyen elérhetőek, illetve elég nagy tapasztalat halmozódott már föl abban is, hogy azokat hogyan érdemes tanítani. Egy ilyen akusztikus modell jellemzőkinyerési használata számos gyakorlati előnyt nyújtana (természetesen amennyiben az elért pontosságértékek versenyképesek). Jelen cikkünkben egy ilyen megközelítést mutatunk be. Az akusztikus modell tanítását egy általános beszédadatbázis (a BEA korpusz (Neuberger és mtsai, 2014)) 60 órányi részhalmozásán végezzük, míg az orvosi feladat sclerosis multiplex beszédből történő fölismerése négy különböző (beszélői) feladatból. Kísérleteinkben a javasolt eljárás hatékonyabbnak bizonyult, mint az azonos adaton tanított, szintén jellemzőkinyerésre használt  $x$ -vektor technika.

## 2. Hangfelvételek

A vizsgálatokra a budapesti Uzsoki Utcai Kórház Neurológiai Osztályán és az Eötvös Loránd Kutatóhálózat Nyelvtudományi Kutatóközpontjában került sor. A vizsgálatot az Uzsoki Utcai Kórház etikai bizottsága hagyta jóvá, és a Helsinkii Nyilatkozatnak megfelelően végeztük el. Kísérleteinket 23 SM alany (18 nő és 5 férfi) és 22 kontroll személy (16 nő és 6 férfi) felvételein végeztük. Az SM alanyok mindegyike a relapszáló-remittáló (relapsing-remitting, RRMS) altípusba tartozott. Az alanyok demográfiai adatait az 1. táblázat tartalmazza. A két csoport tagjainak jellemzőit az életkor és az iskolázottság esetében ANOVÁ-val, a beszélők nemének eloszlását  $\chi^2$ -próbával vizsgáltuk; látható, hogy a két csoport tagjai nem térnek el statisztikailag szignifikánsan egyik vizsgált jellemzőjükben sem.

A felvételi protokoll során sokféle feladatot rögzítettünk az alanyokkal; jelen tanulmányunkban (részben terjedelmi, részben technikai okok miatt) ezekből négyet használunk. Ezek a következők:

		Beszélőcsoportok		Stat.
		SM	Kontroll	$p$
<b>Életkor</b>	átlag $\pm$ szórás terjedelem	39,00 $\pm$ 8,11 [24, 56]	39,95 $\pm$ 7,22 [28, 56]	$p = 0,685$
<b>Nem</b>	férfi / nő	5 / 18	6 / 16	$p = 0,536$
<b>Iskolázottság (év)</b>	átlag $\pm$ szórás terjedelem	15,05 $\pm$ 2,17 [12, 19]	16,09 $\pm$ 1,26 [12, 19]	$p = 0,100$

1. táblázat. A vizsgált csoportok demográfiai adatai.

- az SM alanyokat a betegségükről, a kontrollokat a munkájukról kérdeztük (**betegség / munka**),
- az alanyokat megkértük, hogy meséljék el részletesen az előző napjukat (**tegnapi nap**),
- egy kétperces, számukra korábban ismeretlen tudományos ismeretterjesztő szöveg meghallgatása után az alanyoknak minél pontosabban el kellett azt mesélniük (**szövegösszefoglalás**),
- végül föl kellett olvasniuk olyan mondatokat, amelyben CVCV hangkapcsolatú álszavak voltak (**fonetika**). Az első CV hangkapcsolat egy felpattanó zárhangból ([p, t, k]) és az [i:, a:, u:] magánhangzók egyikéből állt.

A felvételeket egy Sony PCM-A10 digitális diktafonnal, csíptetős mikrofonnal rögzítettük. Az eredetileg sztereó, 48 kHz mintavételű felvételeket a feldolgozás előtt 16 kHz mintavételezésű, monó formátumra konvertáltuk.

### 3. Akusztikus beágyazások

A bevezetőben felsorolt okokból egy hagyományos előrecsatolt mély neurális hálót fogunk jellemzőkinyerésre alkalmazni. Ebből adódóan a javasolt eljárás első lépése egy ilyen modell tanításából áll, már amennyiben erre szükség van egyáltalán (ugyanis a javasolt megközelítés egyik előnye, hogy az ilyen neurális hálók elterjedtsége miatt egy ilyen jó eséllyel már eleve rendelkezésre áll). Természetesen ehhez a lépéshez szükséges, hogy rendelkezünk valamely nagyobb méretű beszédadatbázissal, valamint hozzá tartozó annotált és időzített fonetikai címkékkel (vagy legalább szöveges átirattal). Véleményünk szerint azonban ez a gyakorlatban nem egy szigorú megkötés, az ilyen adatbázisok nagy száma és elterjedtsége miatt. Ezen lépés eredménye egy (keretszinten működő) DNN akusztikus modell.

A jellemzőkinyerés második lépése során ezt az akusztikus modellt ki kell értékelnünk az alanyoktól rögzített hangfelvételeken. Ahelyett azonban, hogy (bevett módon) a kimeneteket rögzítenénk, valamely rejtett réteg aktivációit mentjük el. Mivel ezek az aktiváció-vektorok továbbra is keretszintűek, a harmadik lépésben a teljes hangfelvételen összegezzük azokat. Jelen cikkünkben négyféle



ilyen aggregációs lépést vizsgálunk meg: átlagot (mean), szórást (standard deviation), ferdeséget (skewness) és csúcosságot (kurtosis) számítunk. Amellett, hogy külön-külön is teszteljük ezeket az összegző stratégiákat, a kapott (immár felvételszintű) beágyazásvektorokat össze is fűzhetjük. Az így kapott vektorokat jellemzőkként használjuk az osztályozási lépés során; ezek mérete így az adott rejtett réteg neuronszámának egy- és négyszerese között alakul.

## 4. A kísérletek technikai jellemzői

### 4.1. A DNN akusztikus modell

Mély neurális háló akusztikus modellünket a (magyar nyelvű) BEA Spontánbeszéd-adatbázis egy részhalmazán tanítottuk (Neuberger és mtsai, 2014). 165 beszélőt választottunk ki; a felvételekből automatikusan kivágtuk azokat a részeket, melyekben a felvételvezető hangja is hallható, így 10636 hangfelvételt kaptunk, összesen 60 órányi terjedelemben. Az eredeti sztereó, 44,1 kHz-en mintavételezett bemondásokat monó, 16 kHz-es formátumra konvertáltuk.

Mély neurális hálónk 5 rejtett rétegből állt, mindegyikben 1024 ReLU neuronnal, a kimeneti rétegben pedig a softmax aktivációs függvényt alkalmaztuk. Bemenetként az ún. FBANK jellemzőkészletet használtuk, amely 40 mel szűrősor energiáiból, illetve azok első- és másodrendű deriváltjaiból állt. Tanítás és kiértékelés során 15 keret széles mozgóablakot használtunk, így a háló bemeneteinek száma 1845 volt, míg a kimeneten 911 kontextusfüggő állapotot modelleztünk.

### 4.2. Jellemzőkinyerés

A beágyazásokat az akusztikus modell mindegyik rejtett rétegeből (1...5) kimentettük; a keretszintű beágyazásvektorok mérete megfelelt a rejtett rétegek neuronszámának, így minden esetben 1024 méretű vektorokat kaptunk. Felvételszintű aggregálásra mind a négy korábban felsorolt módszert (átlag, szórás, ferdeség és csúcosság) kipróbáltuk külön-külön; emellett kísérleteztünk az átlag és szórás együttes használatával (2048 jellemző), valamint mind a négy összegző megközelítés alkalmazásával (4096 jellemző). Az összegzett értékeket (azaz a felvételszintű jellemzővektorokat) minden esetben standardizáltunk (azaz minden jellemzőt lineárisan nulla átlagra és egységnyi szórásra transzformáltunk).

### 4.3. Beszélőosztályozás

A jellemzőkinyerési lépés után a beszélőket Support Vector Machine (SVM, Schölkopf és mtsai, 2001) alkalmazásával osztályoztuk, a libSVM csomagot (Chang és Lin, 2011) használva. A túltanulás elkerülése érdekében lineáris kernelt használtunk, így egyetlen hiperparaméterünk az SVM  $C$  (complexity) értéke volt; ezt a  $10^{-5}$ ,  $10^{-4}$ , ...,  $10^1$  értékek közül választottuk ki. A tanítás beágyazott keresztvalidációval történt; minden csoportban (foldban) egy-egy SM beteg és egy kontroll alany volt (egy fold kivételével, amely egyetlen SM betegből állt), így

23 csoportot kaptunk. A  $C$  hiperparamétert minden tanítás esetén egy további (belső, 22-szeres) keresztvalidációs lépés segítségével választottuk ki, a legjobb ROC görbe alatti terület (AUC) érték alapján.

Összehasonlító kísérleteinkben az AUC értéken kívül további kiértékelési metrikákat is kiszámítottunk: osztályozási pontosságot (classification accuracy, *Pont.*), pontosságot (precision, *Prec.*), fedést (recall) és  $F_1$ -értéket (F-measure). (Pontosság (precision), fedés és  $F_1$  esetén az SM beszélőkatagóriát tekintettük pozitív osztálynak; mivel csak két beszélőkatagóriánk (SM és kontroll) volt, a két osztályra kapott AUC-értékek megegyeztek.) A két osztály közötti döntési küszöböt az irodalomban megszokott módon az egyenlő hibaértéknél (Equal Error Rate, EER) húztuk meg.

#### 4.4. x-vektor DNN-ek

Összehasonlítási alapnak x-vektor neurális hálókat tanítottunk, a BEA adatbázis azonos (hatvanórányi) részhalmazán. Ehhez a Kaldi rendszert használtuk (Povey és mtsai, 2011) mind a hálók tanítása, mind az x-vektor jellemzők kinyerése során. Keretszintű jellemzőként mindhárom variációt kipróbáltuk, amit a Kaldi támogat: 23 MFCC-vel, 40 FBANK-kal, illetve spektrogramokkal is kísérleteztünk. A tanítás során szokásos eljárás a tanító adat méretét mesterségesen megnövelni úgy, hogy az eredeti hangfelvételekhez zajt adnak és/vagy visszhangosítják azokat (Snyder és mtsai, 2018). Emiatt minden keretszintű jellemzőtípusra két DNN modellt tanítottunk: egyet augmentációval, egyet pedig ennek a lépésnek a kihagyásával. (Az augmentáció 52636 felvételre (293 órányira) növelte a tanítóanyag méretét.) Meglepő módon mind a négy beszélői feladaton az MFCC-ket használó, zajjal augmentált modellek teljesítettek a legjobban.

## 5. Eredmények

Az elért AUC értékeket a 2. táblázat foglalja össze; minden beszédfeladatra és rétegre a legjobb értékeket **félkövérrel** jelöltük. Általánosságban elmondható, hogy az elért értékek elég magasak: bár néhány esetben (főleg a *tegnapi nap* beszédfeladat esetében) kimondottan alacsony (akár 0,565) AUC értékeket is mértünk, a legtöbb esetben 0,800 fölötti pontszámokat kaptunk. A négy összegző eljárás közül egyértelműen a szórás bizonyult a leghasznosabbnak: az összesen 20 esetből 15 alkalommal vezetett a legjobb (vagy közel a legjobb) eredményhez. Az átlag és a csúcosság 5-5, a ferdeség pedig 3 esetben adta a legmagasabb (vagy ahhoz nagyon közeli) AUC értéket.

A 2. táblázat utolsó két sora a kombinált jellemzőkészletekkel elért eredményeket mutatja (itt a **félkövér** szám azt jelzi, hogy az eredmény magasabb, mint a kombinált módszerek önálló használatával kapott értékek közül a legmagasabb, tehát a kombináció javuláshoz vezetett). Ezen eredmények alapján ez a kombinációs megközelítés nem volt különösebben hatékony: még ha mértünk is javulást, az a legtöbb esetben minimális volt. Összesen három olyan esetet találhatunk,

Feladat	Jellemzők	Forrás rejtett réteg				
		1.	2.	3.	4.	5.
Spontán beszéd (munka / betegség)	Átlag	<b>0,887</b>	<b>0,875</b>	0,870	0,788	0,751
	Szórás	<b>0,885</b>	0,856	<b>0,909</b>	<b>0,911</b>	0,850
	Ferdeség	<b>0,877</b>	0,864	0,802	0,816	0,822
	Csúcsosság	<b>0,879</b>	0,832	0,796	0,824	<b>0,879</b>
	Átlag + szórás	0,836	0,870	0,907	0,836	0,781
	Összes	0,844	0,840	0,796	0,806	0,836
Tegnap nap	Átlag	0,634	0,640	0,717	0,710	0,704
	Szórás	<b>0,678</b>	<b>0,751</b>	<b>0,790</b>	<b>0,824</b>	<b>0,715</b>
	Ferdeség	0,630	0,565	0,636	0,761	0,646
	Csúcsosság	0,575	0,601	0,660	0,767	0,642
	Átlag + szórás	0,593	0,704	0,769	0,769	<b>0,723</b>
	Összes	0,615	0,626	0,672	0,745	0,642
Szöveg- összefoglalás	Átlag	0,757	0,767	0,781	0,753	<b>0,846</b>
	Szórás	<b>0,872</b>	0,824	0,816	<b>0,824</b>	<b>0,854</b>
	Ferdeség	<b>0,868</b>	0,836	<b>0,852</b>	0,808	0,812
	Csúcsosság	<b>0,866</b>	<b>0,848</b>	<b>0,850</b>	0,781	0,802
	Átlag + szórás	0,836	<b>0,850</b>	<b>0,854</b>	0,792	0,842
	Összes	0,814	0,846	0,844	<b>0,826</b>	0,834
Fonetika	Átlag	<b>0,737</b>	0,719	0,830	<b>0,846</b>	0,838
	Szórás	0,713	<b>0,767</b>	<b>0,850</b>	<b>0,854</b>	<b>0,864</b>
	Ferdeség	0,652	0,717	0,721	0,810	0,810
	Csúcsosság	0,688	0,702	0,787	0,802	0,810
	Átlag + szórás	0,731	<b>0,806</b>	0,826	0,818	0,834
	Összes	<b>0,739</b>	0,700	0,759	0,783	0,816

2. táblázat. A beágyazás-alapú jellemzőkkel elért AUC értékek a vizsgált beszédfeladatokon.

ahol a keretszintű aktivációk átlagának és szórásának együttes használata lényegesen jobb volt, mint vagy csak az átlagokat, vagy csak a szórásokat használni jellemzőként (a *szövegösszefoglalás* feladat esetén a DNN 2. és 3. rétegéből, a *fonetika* feladat esetén pedig a DNN 2. rejtett rétegéből számítva a beágyazásokat), ezek 0,026-0,038 abszolút javuláshoz vezettek. Az összes jellemző használatának mérlege még rosszabb: mindkét esetben, ahol ez a megközelítés javított az AUC értékeken, a növekedés csupán (abszolút) 0,002 volt.

Az egyes beszédfeladatok eltérő mértékben voltak hasznosak. A legmagasabb értékeket (0,751...0,911, átlag: 0,849) a *munka / betegség* feladatra kaptuk; ezt követte a *szövegösszefoglalás* (0,753...0,866, átlag: 0,821) és a *fonetika* (0,652...0,864, átlag: 0,776). A legalacsonyabb osztályozási értékekhez a *tegnapi nap* feladat vezetett (0,565...0,824, átlag: 0,682).

Feladat	Jellemző- kinyerési módszer	Pontosságértékek				
		Pont.	Prec.	Fedés	$F_1$	AUC
Munka / betegség	4. réteg	82,2%	82,6%	82,6%	82,6	0,911
	x-vektorok	73,3%	73,9%	73,9%	73,9	0,775
Tegnap nap	4. réteg	68,9%	69,6%	69,6%	69,6	0,824
	x-vektorok	60,0%	60,9%	60,9%	60,9	0,725
Szövegösszefoglalás	1. réteg	86,7%	87,0%	87,0%	87,0	0,872
	4. réteg	77,8%	78,3%	78,3%	78,3	0,824
	x-vektorok	77,8%	78,3%	78,3%	78,3	0,850
Fonetika	5. réteg	82,2%	82,6%	82,6%	82,6	0,864
	4. réteg	82,2%	82,6%	82,6%	82,6	0,854
	x-vektorok	77,8%	78,3%	78,3%	78,3	0,775

3. táblázat. A legjobb és a 4. rejtett réteg szórás függvénnel összegzett aktivációinak használatával, valamint a viszonyítási alapként megvizsgált x-vektorok használatával kapott kiértékelési metrikák. (Pont.: osztályozási pontosság; Prec.: pontosság (precision).)

Érdekes azt is megvizsgálni, hogy melyik rejtett rétegből kinyert beágyazások vezettek a legjobb osztályozási eredményekhez. Azt találjuk, hogy ez egyértelműen függ a beszélő feladatától: míg a *munka / betegség* és a *tegnapi nap* feladatok esetén a 3-4. rejtett réteggel kaptuk a legjobb eredményeket, és a többi réteg használatával kapott AUC értékek lényegesen alacsonyabbnak adódtak, a szövegösszefoglalásnál az 1. és az 5. (tehát a legalacsonyabban és a legmagasabban fekvő) réteg bizonyult a legjobbnak. A *fonetika* feladat esetén a felső (3-5.) rétegek adták a legjobb eredményt, az alsóbb rétegek ennél lényegesen rosszabb AUC értékekhez vezettek. Véleményünk szerint ez azt tükrözi, hogy az adott feladatban jellemzően miben különbözik az SM és a kontroll alanyok beszédprodukcója. Ismert, hogy egy neurális háló alsóbb rétegei egyszerűbb, alacsonyabb szintű jellemzőkinyerést végeznek, míg a legfelső rétegek már kimondottan magas szintű információkat számítanak ki. Egy DNN akusztikus modell esetében már az alacsonyabb rétegeknél fölismerhető lehet pl. a csend, míg a legfelső réteg már fonetikai szintű információkat tárol. Véleményünk szerint ennek tudható be, hogy a *fonetika* feladat esetén a felsőbb rétegek (elsősorban a 4-5. rejtett rétegek) aktivációi bizonyultak hasznosabbnak, hiszen itt maga a feladat is bizonyos fonetikai kombinációk kiejtésére koncentrál. Ugyanakkor, mivel ez egy olvasási feladat volt, az alsóbb rétegek (elsősorban az 1-2. rejtett réteg) lényegesen alacsonyabb értékekhez vezettek. Ezzel szemben a *munka / betegség* vagy a *szövegösszefoglalás* feladatok spontánbeszéd-feladatok voltak, ahol az alany memóriájának is fontos szerep jutott. Itt a legalsó rejtett rétegekből kinyert aktivációk jóval pontosabb osztályozást tettek lehetővé (0,757 és 0,887 közé eső AUC értékek), mint a *fonetika* feladatban (0,652...0,737).

A 3. táblázatban néhány kiválasztott megközelítés több kiértékelési metrikája is látható. Eddigi eredményeink alapján minden feladathoz kiválasztottuk a legjobb AUC értékhez vezető rejtett réteget, valamint, amennyiben nem az bizonyult a legjobbnak, a 4. rejtett réteget is (mely minden feladatnál stabilan jól teljesített). Felvételszintű összegzésre a szórás függvényt használtuk. Viszonyításként föltüntettük továbbá a négy beszédfeladatra az  $x$ -vektorokkal kapott pontosságértékeket is. Összességében elmondható, hogy az osztályozási eredmények is kimondottan magasak: a *tegnapi nap* feladat kivételével (melyre 69% körüli értékeket kaptunk) 77,8% és 87% közé estek. Az  $x$ -vektor jellemzőkkel kapott értékek alapvetően rosszabbak voltak: egyedül a *szövegösszefoglalás* feladat esetében érték el a 4. rejtett rétegből kinyert aktivációk számait (illetve az AUC érték magasabb is volt az  $x$ -vektorok esetében). Azonban a legjobbnak bizonyult, 1. rejtett rétegből számított beágyazásokkal kapott pontosságértékek még ennél a feladatnál is meghaladták az  $x$ -vektorok használatával elérteteket; ezek alapján a bemutatott technika hatékonynak és versenyképesnek minősíthető.

## 6. Összegzés

Jelen cikkünkben relapszáló-remittáló sclerosis multiplex (SM) betegek és kontroll alanyok beszédfelvételeit vizsgáltuk. Minden alanytól három spontánbeszédfelvételt és egy speciális, fonetikai feladat hanganyagát használtuk. Jellemzőkinyerési technikánkat egy standard HMM/DNN hibrid modell DNN akusztikus modelljére alapoztuk: a rejtett rétegek keretszintű aktivációit négy különféle módon (átlag, szórás, ferdeség és csúcosság) összegeztük. Kísérleteink során ez a jellemzőkinyerési megközelítés hatékonynak bizonyult: az elért pontosságértékek minden esetben meghaladták az  $x$ -vektorokkal elért értékeket. Az összegző eljárások közül a szórás használata bizonyult a leghatékonyabbnak, a neurális háló jellemzőkinyerésre használt rejtett rétegének kiválasztásánál azonban figyelembe kellett vennünk az alanyok aktuális beszédfeladatát is: az olvasási, fonetikai feladat esetében az alsó rétegekkel lényegesen alacsonyabb értékeket kaptunk, mint a legfelső réteggel, míg a háromból két spontánbeszéd-feladatnál a legmélyebben fekvő rétegek aktivációinak használata is versenyképes eredményhez vezetett. A legjobb AUC értékek a beszédfeladattól függően 0,824 és 0,911 közé estek, demonstrálva az alkalmazott jellemzőkinyerési technika potenciálját.

## Köszönetnyilvánítás

A kutatást részben támogatta a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal – NKFIH, K-132460, NKFIH-1279-2/2020. Gosztolya Gábor kutatásait az MTA Bolyai János ösztöndíja és az Új Nemzeti Kiválóság Program Bolyai+ pályázata (azonosító: ÚNKP-21-5-SZTE) is támogatta. A publikációban szereplő kutatást (amelyet a Szegedi Tudományegyetem valósított meg) az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal is támogatta a Mesterséges Intelligencia Nemzeti Laboratórium (MILAB) keretében.

## Hivatkozások

- Bóna, J., Svindt, V., Hoffmann, I.: Voice onset time of Hungarian voiceless plosives in Multiple Sclerosis. In: ISSP. pp. 202–205 (Dec 2020)
- Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 1–27 (2011)
- Dehak, N., Kenny, P., Dehak, R., Glembek, O., Dumouchel, P., Burget, L., Hu-beika, V., Castaldo, F.: Support Vector Machines and Joint Factor Analysis for speaker verification. In: ICASSP. pp. 4237–4240 (2009)
- Egas-López, J., Vetráb, M., Tóth, L., Gosztolya, G.: Identifying conflict escalation and primates by using ensemble x-vectors and Fisher vector features. In: Interspeech. pp. 476–480 (2021)
- García, N., Vásquez-Correa, J.C., Orozco-Arroyave, J.R., Nöth, E.: Multimodal i-vectors to detect and evaluate Parkinson’s Disease. In: Interspeech. pp. 2349–2353. Hyderabad, India (2018)
- Grzybowska, J., Kacprzak, S.: Speaker age classification and regression using i-vectors. In: Interspeech. pp. 1402–1406 (2016)
- Huckvale, M., Beke, A., Ikushima, M.: Prediction of sleepiness ratings from voice by man and machine. In: Interspeech. pp. 4571–4575 (2020)
- Jenei, A.Z., Kiss, G.: Depresszió detektálása korrelációs struktúrán alkalmazott konvolúciós hálók segítségével. In: MSZNY. pp. 59–71. Szeged (Jan 2020)
- Mulfari, D., Meoni, G., Marini, M., Fanucci, L.: Machine learning assistive application for users with speech disorders. *Applied Soft Computing* 103(May), 107147 (2021)
- Neuberger, T., Gyarmathy, D., Grácsi, T., Horváth, V., Gósy, M., Beke, A.: Development of a large spontaneous speech database of agglutinative Hungarian language. In: TSD. pp. 424–431 (2014)
- Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., Veselý, K.: The Kaldi speech recognition toolkit. In: Proceedings of ASRU (2011)
- Renauld, S., Mohamed-Said, L., Macoir, J.: Language disorders in multiple sclerosis: A systematic review. *Multiple Sclerosis and Related Disorders* 10, 103–111 (2016)
- Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, R.: Estimating the support of a high-dimensional distribution. *Neural Computation* 13(7), 1443–1471 (2001)
- Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust DNN embeddings for speaker verification. In: ICASSP. pp. 5329–5333. Calgary, Alberta, Canada (2018)
- Svindt, V., Bóna, J., Hoffmann, I.: Changes in temporal features of speech in secondary progressive multiple sclerosis (SPMS) – case studies. *Clinical Linguistics & Phonetics* 34(4), 339–356 (2020)
- Szirmai, I.: *Neurológia*. Medicina, Budapest (2006)
- Variani, E., Lei, X., McDermott, E., Moreno, I., G-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. In: ICASSP. pp. 4080–4084 (2014)

# Hangkonverzió alkalmazása dysarthriás betegek beszédminőségének javítására

Terbe Dániel<sup>1</sup>, Tóth László<sup>1</sup>, Ivaskó Lívია<sup>2</sup>

<sup>1</sup>Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup>Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup>Szegedi Tudományegyetem, Általános Nyelvészeti Tanszék  
tothl@inf.u-szeged.hu

**Kivonat** A dysarthria egy gyűjtőfogalom az artikulációs nehezítettség-ből eredő beszédzavarra, amelynek hátterében számos betegség állhat. A dysarthriás személyek beszédének minősége, érthetősége leromlik, ami az érintettek szociális kapcsolataira és így életminőségére is rossz hatással lehet. A hangkonverziós technológia fejlődésével felvetődik az ötlet, hogy vajon lehetséges-e ezen betegek hangfelvételeinek minőségét, érthetőségét gépi eszközökkel feljavítani, és beszédkommunikációjukat egy ilyen elven működő eszközzel támogatni. Cikkünkben áttekintjük a (neuronhálós) hangkonverziós algoritmusok fő változatait, majd bemutatjuk a dysarthriás betegek felvételein végzett kísérleteink tapasztalatait, és ezek alapján megvitatjuk az egyes módszerek legfontosabb előnyeit és hátrányait.

**Kulcsszavak:** dysarthria, hangkonverzió, mély neuronhálók

## 1. Bevezetés

A beszéd tervezésének és kivitelezésének zavarai komoly kommunikációs akadályt jelentenek azon személyek számára, akik szerzett vagy fejlődési eredetű beszédzavarral küzdő kommunikációs partnerként szeretnének interakcióba lépni társaikkal. A különböző neurodegeneratív betegségek, illetve egyéb eredetű idegi károsodások eredményeképpen kialakuló fonációs és artikulációs zavarok úgy befolyásol(hat)ják átmenetileg vagy tartósan, egyes esetekben pedig egyre fokozottabban megjelenő (progreddiáló) formában a beszélők beszédprodukciónak, hogy az egyébként tartalmilag és nyelvtanilag jól formált, grammatikus beszéd az interakciós partnerek számára nagyon nehezen, vagy alig érthető hangokként jelenik meg (Horváth és Hirshberg, 2013). Ez a nehéz érthetőség nagy mértékben csökkenti az érintett személyek önálló életvitelre való képességét, súlyos életminőség-romlást tud előidézni. A jobb életminőség eléréséhez és a megfelelő érthetőség szempontjából is fontosnak tartjuk azon lehetőségek számbavételét, melyek az érintettek hangminőségbeli javítását célzó törekvéseket kívánnak a számítógépes nyelvészeti és a mesterséges intelligencia kutatás eszköztárával elősegíteni. Fontosnak tartjuk az érintett populáció társadalmi reintegrálhatósága

szempontjából a beszédteljesítményük javítását célzó kutatásban való aktív részvételt (Tóth és mtsai, 2018).

Már Aronson 1981-es összefoglaló munkája (Aronson, 1981) is rámutat arra, hogy dysarthria több okból is eredhet, és attól függően, hogy milyen területek sérülése, illetve diszfunkciója áll a dysarthriás beszéd hátterében, eltérő hangminőségbeli tulajdonságok mentén lesznek az egyes előfordulások csoportosíthatóak. Annak függvényében, hogy a beszédhangok és a velük együtt realizálódó szupraszegmentális elemek milyen mértékben tudnak megtartott képességek alapján képződni, el lehet különíteni az egyes dysarthria-típusokat, azonban hazánkban ez az elkülönítés jelenleg a diagnosztikus kritériumok tekintetében elsősorban az oki tényezőket, valamint az érintett traktusnak a beszédre gyakorolt hatását figyelembe vevő módon, a hallási élményt szubjektíve értékelő skálán osztályozza (Horváth és Hirshberg, 2013). Értelmezésükben a kortikális sérülésből eredő dysarthriás beszéd folyamatok például elsősorban úgy jellemezhetőek, mint a beszéd primer motoros tervezésének és kivitelezésének nem megfelelő működéséből eredő specifikus mintázatok. A stroke eredetű dysarthria a motoros funkciók érintettségéből eredő beszédzavaroknak tekinthető, mely (a stroke kiterjedésétől függően) nem kell, hogy érintse a nyelvi tervezési folyamatokat. A klinikai differenciáldiagnosztika során az ilyen területek érintettségéből eredő hangzókülönbségek auditív úton is elkülöníthetőek a hangképzőszerveket érintő más atípusos formáktól Aronson (1981) csoportosítása szerint. Akiknél más (nem stroke eredetű) neurodegeneratív megbetegedés vagy traumatikus agysérülés okoz beszédzavart, a beszéd más összetevőinek, például a monoton beszédnek, vagy hiperkinetikus beszédnek a jegyeit produkálják verbális megnyilvánulásaik során. A hangerő, a hangmagasság és a ritmus is fontos összetevői a beszédnek, melyek a sérülés, illetve betegség eredetétől és helyétől függően mutathatnak változatosságot az eltérésben. Azt mondhatjuk tehát, hogy a dysarthria egy olyan összetett klinikai kép, mely a beszéd egyes összetevőit nem azonos mértékben és minőségben érinti az egyes kórképek esetében (pl. ALS, sclerosis multiplex, stroke, traumatikus agysérülés, kisagyi érintettség, egyes idegbénulások, Parkinson-kór, illetve egyéb idegrendszeri zavarok esetében), de a személy számára jelenthetnek ezek az eltérő formák olyan akadályt, mely miatt nehezen érhető beszéde nem teszi őt képessé a megfelelő verbális kommunikációra. Az emberi beszéd bonyolult folyamatában a beszédjel a megbetegedés alapvető sajátosságaitól, a neuroanatómiai eltérés helyétől és kiterjedésétől függően eltérő módokon torzulhat. A motoros funkciók érintettségéből eredő beszédzavarok közül az artikulációs szervek vezérlésének zavara a hangképzési folyamatot befolyásolja, például hibás formánsszerkezetű magánhangzókat eredményezhet. Ha az artikulációs szervek összehangolása sérül, akkor időben elkent, torzult hangzókat kapunk. A hangok adott ideig és hangmagasságon való kitartásának nehézsége a beszéd szupraszegmentális szintjének, a prozódiaának a torzulásaként jelentkezik. Végül, a hangszalagok vezérlésének zavara a hangminőség romlását okozza, ez az ún. diszfónia (Markó és mtsai, 2007) gyakran van jelen a dysarthriával egyidejűleg (Camillo és Ortiz, 2007). Különösen igaz ez az életkor előrehaladtával kialakuló természetes változásokat figyelembe véve.



## 2. Mély neuronhálós hangkonverziós algoritmusok

A hangkonverzió (voice conversion) egy beszédtechnológiai eljárás, melynek célja egy adott (forrás) beszélő hangfelvételének átalakítása oly módon, mintha azt egy másik beszélő (a célbeszélő) mondta volna (Mohammadi és Kain, 2017). A hangkonverziót gyakran a jóval általánosabban értelmezhető hangtranszformáció (voice transformation) speciális esetének tekintik.

A hangkonverzió fő alkalmazója a szórakoztatóipar (Turk és Arslan, 2002): segítségével utólagosan javíthatóvá, manipulálhatóvá válnak a filmek hangsávjai (pl. pár szó betoldásához nincs szükség az eredeti színészre) vagy egy pontatlanul felénekelte éneksáv, de akár régen elhunyt színészek szájába is új szöveget adhatunk. A szórakoztatóipar mellett a másik fő alkalmazást a telekommunikáció beszédszintézisre épülő ágai jelentik. A hangkonverzió segíthet a személyre szabott beszédszintézisben, például egy telekonferencia-alkalmazásban, akár valós idejű gépi fordítással egybekötve. A személyre szabott beszédszintézis speciális esete az orvosi alkalmazás, amikor a beteg eredeti beszédét próbáljuk visszaállítani (voice reconstruction), például gégeműtét után, vagy esetünkben dysarthria fennállásakor.

Hangkonverzióval már régóta próbálkoznak (Moulines és Sagisaka, 1995), de az igazi fellendülést a területen a mély neuronhálós technológiák megjelenése hozta. A legkorábbi, legegyszerűbb algoritmusok párhuzamos hangfelvételeket igényelnek, azaz a forrás- és a célbeszélőnek ugyanazt a szöveget kell beolvasnia. Jóval későbbiek a párhuzamos korpuszt nem igénylő 'non-parallel' algoritmusok (Kaneko és mtsai, 2021), amelyek a bemenő adatokra nézve jóval nagyobb szabadsági fokot biztosítanak, de ez esetben a gépi tanulási feladat is jóval nehezebb. A hagyományos, egy forrás- és egy célbeszélőt feltételező módszerek mellett próbálkoznak már sokbeszélős konverzióval is ('many-to-many', 'many-to-one') (Kaneko és mtsai, 2019). Nem célunk az összes létező szempont és technológiai megoldás áttekintése, ezért csak azokat a módszereket ismertetjük kicsit részletesebben (a 3. fejezetben), amelyekkel személyes tapasztalatot szereztünk.

### 2.1. Hangkonverzió dysarthriás beszéd feljavítására

A hangkonverzió szokványos alkalmazása esetén azt várjuk, hogy a konvertált hang lehetőleg minden szempontból hasonlítson a célbeszélő hangjához. Ezt a hasonlóságot azonban nehéz objektív, tudományos módon megfogalmazni, és még nehezebb egzakt mérőszámokkal számszerűsíteni. Mindenesetre a hasonlóság két fő tényezője a hangszín és a hangmagasság, és e két tulajdonságot viszonylag egyszerű módon lokálisan, azaz a jel időbeli lefutásának bolygatása nélkül is lehet módosítani. A dysarthriás beszéd feljavítása esetén azonban kicsit más a cél, mint a szokványos hangkonverziónál. Kiindulási hangként a beteg jelenkori felvételei állnak rendelkezésünkre, amelynek minőségén, érthetőségén szeretnénk javítani. Minden más tulajdonságát azonban lehetőleg szeretnénk változatlanul hagyni, hogy megőrizzük a beteg személyiségét tükröző vonásokat. A optimális megoldáshoz célhangként hangminták lennének szükségesek ugyanazon alany betegség megelőző állapotáról, ilyen azonban legtöbbször nem áll rendelkezésre

(de nyugaton léteznek már cégek, amelyek ilyen hangarchívum készítését kínálják lassan progrediáló betegségben szenvedőknek). Ilyenkor valamilyen donor hangot kell használnunk célhangként. Itt ismét jó lenne, ha léteznének hatalmas donor hangadatbázisok, melyekben meg lehetne találni az alany hangjához leginkább illeszkedő mintát – de ilyen magyar nyelvű adatbázisról sem tudunk egyelőre. Mivel a célunk a minőség és az érthetőség javítása, a konverzió során alapvetően az artikulációs finomozgást szeretnénk átültetni a dysarthriás felvételre, nem célunk viszont a személyre jellemző hangszín átvitele. Ezzel összhangban alapvetően a hangmagasság átvételére sincs szükségünk. A helyzet azonban nem ilyen egyszerű, mivel a dysarthria gyakran együtt jár a hangkeltés zavarával is, ilyen esetekben viszont a hangszalagok működését leíró beszédtechnikai komponens, a gerjesztőjel módosítása is szükséges lehet a beszédminőség javításához. Továbbmenve, dysarthriás beszéd esetén legtöbbször a prozódia is sérül, ami egyszerűbb esetekben csak lelassulásként jelentkezik, de súlyosabb szinten jelentősen hozzájárulhat a beszéd érthetőségének romlásához. Az itt bemutatott, kezdeti kísérleteinkben olyan pácienseket választottunk, akiknél a fő problémát az artikuláció elkentsége okozza, de a fentiek érzékeltetik, hogy általános esetben a probléma milyen sokrétű lehet. Az általunk kipróbált módszereket eredetileg elsősorban egészséges beszélők hangszínének felcserélésére fejlesztettük ki (lásd előző fejezet), nem pedig leromlott minőségű beszéd feljavítására. Munkánk során részben azt vizsgáltuk, hogy ezek – az eddig egészséges beszéden alkalmazott eljárások – mennyire alkalmazhatók dysarthriás beszéd konverziójára. A szakirodalomban megoldásként javasolt algoritmusokat részletesen a 3. fejezetben tárgyaljuk.

### 3. Kísérleti konfigurációk és tapasztalatok

#### 3.1. Adatbázisok

A dysarthriás beszéd kutatását nagyban megnehezíti, hogy nemzetközi szinten is kevés a megfelelő méretű dysarthriás korpusz, mivel a felvételek publikussá tételét a szigorú betegjogi/adatvédelmi szabályozás megnehezíti. Mi a kísérleteinkhez főképp az UASpeech nevű, angol nyelvű publikus adatbázist használtuk, amely 15 dysarthriás és 13 kontroll beszélőtől tartalmaz felvételeket, izoláltan kiejtett szavak formájában (Kim és mtsai, 2008). Kísérleteinkben egy-egy típusú leképezést igyekeztünk megvalósítani, amihez kiválasztottunk egy férfi beteget és egy férfi kontrollszemélyt. A rendelkezésre álló tanítóanyag kb. 45 perc volt a beteg részéről.

Emellett magyar nyelvű adatgyűjtésbe is fogtunk, a magyar nyelvű korpusz összegyűjtése jelenleg is folyamatban van<sup>1</sup>. Tapasztalataink között a magyar nyelvű felvételeken kapott kezdeti eredmények konklúziói is megjelennek.

<sup>1</sup> Kutatásetikai engedély: ETIKAI/IV-11043-1 TUKEB Határozat (Dysarthriával élők beszédminőségének vizsgálata), kutatásvezető orvos: Dr. Sandi Dániel (SZTE Neurológiai Klinika)

### 3.2. Vokóderek

A hangkonverzió bemenete és kimenete is egy hangfelvétel. Habár léteznek már közvetlen hullámforma inputtal és outputtal dolgozó neuronhálók is (Kim és mtsai, 2020), egyelőre szokványosabb a hangfelvételeket valamilyen spektrális reprezentációra konvertálni, ez ugyanis tömörebb tárolást és könnyebb manipulálhatóságot is biztosít. A konverzió tehát alapvetően három fő lépésből áll: a hangot először analizáljuk, "szétszedjük", az analízis eredményének bizonyos paramétereit módosítjuk, végül a hangot szintetizáljuk, újra "összerakjuk". Elsőre technológiai részletkérdésnek tűnhet, de fontos döntés, hogy az algoritmus milyen belső reprezentációt használ, azaz milyen módon analizálja a hangot. A régebbi megoldások valamilyen hagyományos, jelfeldolgozáson alapuló vokódert (voice encoder-decoder) használtak erre a célra (pl. WORLD (Morise és mtsai, 2016)), melyek tipikusan szétszedik a beszédet gerjesztőjelre és spektrális burkológörbére, és ezeket erősen tömörítve reprezentálják. Ennek ára, hogy a rekonstrukció nem tökéletes, azaz a vokóder használata már magában kisebb minőségromlással járhat. Újabban azonban megjelentek a gépi tanuláson alapuló ún. neurális vokóderek, amelyek sokkal egyszerűbb reprezentációkból – pl. egyszerű mel-spektrogramból – képesek megdöbbentően jó minőségű beszédet szintetizálni (Luong és Tran, 2021; Kumar és mtsai, 2019; Prenger és mtsai, 2019). Mi a MelGAN vokódert alkalmaztuk, amely 80 sávós mel-spektrogramot használ a jel reprezentációjára és abból szintetizál hullámformát. Fontos megemlíteni, hogy tapasztalatunk szerint ezek a vokóderek nyelvfüggetlenek, azaz angol nyelven betanított vokóder képes lesz magyar nyelvű beszéd szintetizálására is.

### 3.3. Vektorszintű párosítást igénylő módszerek

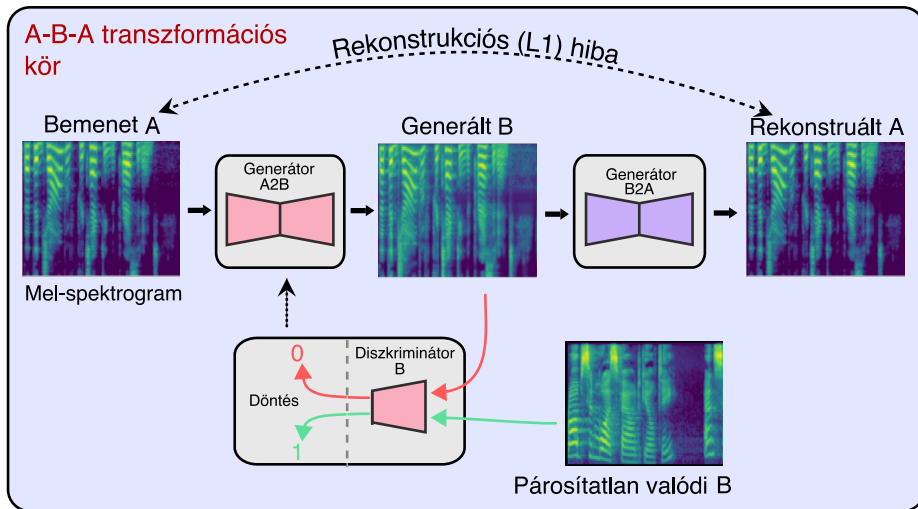
A vokóderre épülő módszerek esetében a forrás- és a célfelvételt az algoritmus egy-egy spektrális vektorsorozat formájában kapja meg. A legegyszerűbb algoritmusok a minták egymásba való transzformálását a vektorok szintjén oldják meg, azaz vektor-vektor leképezést végeznek, vagy legfeljebb rövidebb (maximum szótagnyi hosszúságú) részsorozatokon dolgoznak. Ez csak akkor megvalósítható, ha a forrás- és a célbeszélő hangfelvételei párosítottak, azaz mindketten ugyanazt a szöveget mondják. A tartalmi egyezés azonban nem garantálja a felvételek belsőjében a tökéletes szinkronitást. Ezért ezek az algoritmusok előfeldolgozásként igénylik a vektorsorozatok egymáshoz illesztését, amire olyan klasszikus algoritmusokat használhatunk, mint pl. a dinamikus idővetemítés. Cserében viszont – mivel a célértékek vektorszinten vannak definiálva – a vektor-vektor transzformáció nagyon egyszerűen, akár tradicionális neuronhálós megoldásokkal is megvalósítható. A transzformáció hatékonyságát olyan egyszerű hibafüggvényekkel mérhetjük, mint pl. az átlagos négyzetes eltérés (MSE hiba). Az egész terület egyik kulcsproblémája azonban, hogy az elvégzendő átalakítást lehetetlen egyszerű matematikai eszközökkel definiálni. Ezért a transzformációs hiba mérésére bevethetünk egy második neuronhálót, amelyet a tiszta és a dysarthriás artikuláció megkülönböztetésére tanítunk be. Ez vezet el az ún. generatív ellenséges neuronhálók (GAN) konstrukcióhoz, ami a képfeldolgozásban nagyon sikeresnek

bizonyult. Az általunk kipróbált módszer neve MMSE DiscoGAN (Purohit és mtsai, 2020). It a neurális architektúra egy egyszerű négyrétegű teljesen kapcsolt hálózat, amelynek a bemenete az eredeti cikk alapján 40 darab MFCC együtt-ható (Mel Cepstral Coefficient), de kipróbáltuk 80 sávós mel-spektrummal is. Ennél a módszernél a hibafüggvény két elemből tevődik össze: (1) a céltól való átlagos négyzetes eltérés (MSE); (2) egy összetettebb adversarial hiba. Az előbbi hibafüggvény használja ki a párosított tulajdonságot, hiszen ott a bemenethez társított célvektor elérésére törekszünk a hibatag minimalizálása során. Az utóbbi pedig a szakirodalomban DiscoGAN (Kim és mtsai, 2017) néven fellelhető GAN alapú tanítási módszer. Ez egymástól függetlenül és egyszerre jelent meg a CycleGAN technikával (Zhu és mtsai, 2017), amely később aztán jobban elterjedt. Kísérleteink során kipróbáltuk a második tag elhagyását is, tehát egyszerűen négyzetes eltérés alapján tanítottunk, ami valóban kicsit rosszabb eredményekhez vezetett – ez azt mutatja, hogy az adversarial hibatag beépítése a párosított tanításba ténylegesen hasznos lehet.

### 3.4. Párosítást nem igénylő módszerek

A képfeldolgozásban nagy sikert aratott az ún. neurális stílusztransfer, amellyel például fotóinkat Von Gogh-stílusú festményekké alakíthatjuk. Ilyenkor a képen viszonylag csekély módosítást kell végezni, hiszen maga a tartalom megőrzendő. A hangkonverzió célja is hasonló, csak hangfelvételekkel: a felvétel hangszínének adott beszélőhöz való igazítása a nyelvi tartalom megtartásával. A képfeldolgozásban erre a célra az ún. feltételes GAN-okat (conditional GAN, cGAN) alkalmazzák, ahol a 'feltétel' tulajdonképpen maga az input kép. A GAN technológia nagy előnye, hogy a tanításához nem feltétlenül kellene input-output párok. Ennek megfelelően a képi stílusztransferben bevált GAN-okat természetesen hangkonverzióra is megpróbálták alkalmazni (Yang és Chung, 2020), akár párosított felvételek nélkül, ami a tanítás során jóval nagyobb szabadsági fokot ígér. Így elméleti szinten például könnyedén megvalósíthatóvá teszi a 'many-to-many' leképezést is. Ennek a nagy szabadságnak azonban megvan a hátulütője: mivel nincsenek vektorok szintjén definiált célértékek, rendkívül nehéz tudtára adni a hálózatnak, hogy milyen jellegű módosításokat szeretnénk elérni, és mely módosítások nem kívánatosak. Technikailag különféle jellegű megszorítások megadásával szokták a hálózatot a jó irányba 'terelgetni'. A nagy szabadsági fok miatt ráadásul ezeknek a módszereknek a stabil betanításához jóval több tanítópélda szükséges és maga a tanítási folyamat is hírhedten nehéz, érzékeny a paraméterbeállításokra, ráadásul sokkal nagyobb a számításiigényük is (egy tanítás napokig is eltarthat).

A legelterjedtebb párosított adathalmazt nem igénylő tanítási módszer az ún. CycleGAN (Zhu és mtsai, 2017), amelyet az 1. ábrán szemléltetünk. A modellben két generátorhálózat van, melyek a két (forrás és cél-) tartomány közötti oda-vissza transzformációért felelősek. Emellett mindkét tartományhoz van egy-egy diszkriminátor-hálózat, melynek feladata annak eldöntése, hogy a beadott minta az adott térhez tartozik-e (a spektrum az A vagy B beszélőhöz tartozik-e). A diszkriminátor hálózatok tanítása úgy történik, hogy mutatunk neki valódi



1. ábra: A CycleGAN tanítási módszer illusztrációja. Az ábrán csak az A-B-A irányt mutatjuk, de ugyanez (párhuzamosan) a B-A-B irányban is megtörténik.

(ténylegesen a tartományból vett) és hamis (a másik tartományból átalakított) mintákat. A diszkriminátor célja ezek megkülönböztetése, míg a generátor hálózatok arra vannak tanítva, hogy képesek legyenek becsapni a diszkriminátor hálózatot (egyre jobb és jobb minőségű, élethűbb minták generálásával). Tehát a generátor és diszkriminátor hálózatok egymás ellenében vannak tanítva és versengenek egymással, innen a "generatív ellenséges neuronháló" elnevezés.

A feltételes GAN esetén kulcskérdés, hogy párosítatlan tanítópéldák mellett hogyan tudjuk garantálni az eredeti tartalom megőrzését – vegyük észre, hogy a diszkriminátor ezt nem oldja meg. Erre szolgál a modell ciklikussá tétele: a generátor hálózatoknak teljesíteniük kell azt a megszorítást, hogy az oda-vissza transzformáció után vissza kell kapnunk a kiindulási mintát (rekonstrukciós hiba minimalizálása). Ez a feltétel hozza létre a domainek közötti egy az egyhez való leképezést, illetve teszi lehetővé, hogy ne véletlenszerű mintagenerálás történjen, hanem az adott bemenethez tartozó másik térbeli párt kapjuk.

Tesztjeink során két párosítást nem igénylő eljárást próbáltunk ki. Az első cikk szerzői kifejezetten a dysarthriás beszéd javítását célozták meg, és állításuk szerint módszerük felülmúlja a CycleGAN technikát (Chen és mtsai, 2018). A módszer három neuronhálót alkalmaz: egy generátort, egy diszkriminátort és egy kontrollert. A kontrollert egy tömör kódban reprezentálja a bemenő beszédet, amelyből aztán a generátor ismét beszédet készít (ez lényegében egy autoenkóder), miközben a diszkriminátor arra sarkalja a generátort, hogy minél élethűbb mintákat gyártson. Először nagy mennyiségű (kb. egy napnyi) egészséges adaton tanítják a rendszert (ilyen adatból sokkal könnyebb nagy mennyiséget gyűjteni) és miután a generátor megtanult jó minőségű egészséges beszédet produkálni,

kisebb méretű dysarthriás adatbázison tanítják tovább már csak a kontroller részt.

A másik, MaskCycleGAN-VC elnevezésű módszert sima hangkonverzióra alkalmazták a kiindulási cikkben (Kaneko és mtsai, 2021). A modell a CycleGAN-VC legfrissebb, negyedik generációs változata, ami a CycleGAN (eredetileg képekre kifejlesztett) módszer hangkonverziós feladatra szabott változata.

### 3.5. Mondatszintű párosítással dolgozó módszerek

A párosítatlan felvételekkel dolgozó GAN-okhoz képest első pillantásra visszalépést jelentenek a sorozatból sorozatba leképező ún. sequence-to-sequence (seq2seq) neuronhálók, ezek betanításához ugyanis párosított felvételek kellene. Azonban a párosítás csak a felvételek szintjén szükséges, a vektorok illesztését már elvégzi az algoritmus. A seq2seq hálók ötlete a gépi fordítás területéről ered, de jelenleg rendkívül népszerűek például a beszéd felismerésben (Novitasari és mtsai, 2020) és a beszéd szintézisben is (Wang és mtsai, 2017). A párosított mintákon tanítás előnye az lehet, hogy konkrétan tudjuk definiálni az input-output párokat, viszont a hibalehetőséget rejtő vektorszintű illesztésre nincs szükség, azt már elvégzi a technológia. Újabban az LSTM alapú seq2seq modelleket elkezdték leváltani az ún. transzformer alapú hálózatok (Li és mtsai, 2019), melyek ugyan tipikusan nagyobb modellek, viszont a párhuzamosíthatóságuk miatt jóval gyorsabban lehet tanítani őket és teljesítményben is felülmúlják az elődjüket. Az utóbbi időben ezeket a módszereket elkezdték alkalmazni a hangkonverzióban (Zhang és mtsai, 2019; Huang és mtsai, 2019; Tanaka és mtsai, 2019; Jia és mtsai, 2019), sőt beszédjavításra is (Huang és mtsai, 2021b; Biadys és mtsai, 2019). Saját kísérleteket ezzel a módszer családdal még nem végeztünk, ez a jövőbeni terveink közt szerepel.

### 3.6. Tapasztalatok

A három kipróbált algoritmus közül tapasztalataink szerint a párosított mintákkal dolgozó DiscoGAN igényli a legkevesebb tanítóadatot, viszonylag könnyen és gyorsan tanítható. Ennek oka, hogy a vektorszintű illesztés miatt minden egyes spektrális vektorhoz jól definiált célvektor tartozik. Ugyanez a tulajdonság eredményezi azonban a módszer gyenge pontját is: azt tapasztaltuk, hogy a dinamikus idővetemítésen alapuló illesztés maga is egy hibaforrás, mivel gyakran nem sikerül megfelelően összeillesztenie a mintákat (különösen elkent artikulációjú és/vagy akadozó beszéd esetén). Konkrétan, sok esetben a vetemítés melléktermékeként fellépő minőségromlást nagyobb fokúnak éreztük, mint a dysarthria által okozottat. Összességében tehát ezt a módszert több szempontból is limitáltnak találtuk, a kimenetként kapott hangmintákat gépiesnek, nem természetes hangzásúnak éreztük. A négyzetes hibafüggvényt a GAN-hibával kiegészítve sikerült ugyan javulást elérni, de így sem értük el az általunk vágyott minőséget.

A tisztán GAN-alapú, ezért párosított tanítómintákat nem igénylő módszerek elvileg teljesen természetes hangzású beszéd előállítására is képesek, a gyakorlati

tapasztalataink azonban messze nem voltak ilyen jók. Az első kipróbált módszer szerzői azt állítják, hogy megoldásuk a nyelvi tartalmat és a dysarthriás beszélő hangszínét megtartja. Ezzel szemben nekünk az volt a benyomásunk, hogy ez már a szerzők által demonstrációként feltöltött hangmintákra sem feltételül teljesül, és az angol adatbázison való újratanítás után kapott eredményeinkben sem ezt tapasztaltuk. A másik, MaskCycleGAN-VC nevű algoritlussal meggyőzőbb eredményeket kaptunk, hosszas paraméterhangolás után sikerült minőségi javulást elérnünk a dysarthriás felvételeken. Enyhén leromlott beszéd esetén, azaz kisebb korrekcióra tehát alkalmas tűnt ez a módszer, nagyobb léptékű transzformáció esetén azonban nehézségekbe ütköztünk. Nagyobb mértékű változtatás megengedésekor ugyanis az algoritmus olyan tulajdonságokat is elkezd megváltoztatni, amelyeket mi nem szeretnénk: a célbeszélő hangmagasságát is átveszi, rosszabb esetben pedig a nyelvi tartalmat is megváltoztatja - gyakorlatilag véletlenszerű halandzsát generál. Úgy véljük, hogy ezek a nehézségek a jelenlegi GAN-alapú módszertan alapvető gyengeségére tapintanak rá: nem tudjuk pontosan definiálni (és a modell tudomására hozni), hogy az inputnak milyen fokú módosítását engedjük meg, és az a módosítás pontosan mit is módosítson (és mit nem). A párosított módszer esetén vannak ugyan célvektorok, de az olyan egyszerű függvények mint a spektrogram pixeleinek négyzetes eltérése nem képesek szétválasztani a különböző érzékelési dimenziók (hangszín, hangmagaság, artikulációs tisztaság, stb.) mentén fennálló eltéréseket. A párosítatlan módszerek GAN-jainak pedig konkrét célértékek híján még kevésbé tudjuk átadni, hogy mit is várunk tőlük. Ezért mindenképpen szükségesnek látjuk a szokványos GAN-célfüggvények kiegészítését különféle ügyes megszorításokkal, és az irodalomban látunk is ilyen irányú próbálkozásokat (Tanaka és mtsai, 2019). Úgy gondoljuk továbbá, hogy a cél pontosabb megfogalmazásában az is segítene, ha a szimpla mel-sektrogram reprezentáció helyett olyan vokódereket használnák, amelyek a beszéd részletesebb elemzését adják – elvégzik például a korábban sztendernek számító gerjesztőjel-burkológörbe szétválasztást (Ferro és mtsai, 2020).

Akár a párosított, akár a párosítatlan módszercsaládot nézzük, az időbeli transzformációra, azaz a kiindulási felvétel ritmusának és sebességének módosítására egyik sem képes, és nem is explicit céljuk. A tökéletes megoldáshoz azonban a szupraszegmentális, prozódiai jegyek átvitele is szükséges lenne (a hangmagasság mellett a hangsúly és az időzítési sajátosságok, a beszéd ritmusa is ide tartoznak). A megoldást a párosított mondatokkal dolgozó seq2seq módszerektől remélhetjük, mivel ezek a bemenetet egy időbeli sorozatként kezelik, és akár időbeli transzformációkat is képesek végezni. A hátrányuk az, hogy rengeteg (több napnyi) adaton kell őket tanítani és nagy méretű, bonyolult (nehezen implementálható) rendszerek. Ezért ezeket a modelleket tipikusan előtanítják beszédfelismerési vagy TTS (text-to-speech) feladaton, mert ilyen célra jóval nagyobb mennyiségű adat érhető el, majd ezután tanítják tovább hangkonverzióra (Huang és mtsai, 2021a).

### 3.7. A hangkonverzió objektív kiértékelése

Mint láthattuk, a hangkonverzió céljának formalizálása sem könnyű, épp ezért az eredményként kapott beszédjel kiértékelése sem triviális. A standard kiértékelési szempont, hogy a kapott beszédjel mennyire hasonlít a célbeszélő beszédére, esetünkben azonban ennél fontosabb, hogy javult-e a minősége, azaz mennyire tűnik természetes, valódi beszédnek, rosszabb kiindulási felvétel esetén pedig, hogy egyáltalán a beszéd érthetőségén sikerült-e javítani. Egyik szempontra sem könnyű objektív mérőszámot adni, a legmegbízhatóbb kiértékelési mód ezért még mindig a szubjektív, lehallgatásos tesztelés. Emellett sokan próbálkoznak olyan klasszikus, objektív metrikák használatával mint az MCD (mel-cepstral distortion), PESQ (perceptual evaluation of speech quality) vagy a STOI (short-term objective intelligibility). Ezek eredményét azonban fenntartásokkal kell kezelni, mert közismerten gyengén korrelálnak az emberi minősítéssel és preferenciával (Sündermann, 2005). Mi a STOI metrikával kísérleteztünk, de azt tapasztaltuk, hogy csak tökéletesen illesztett minták összehasonlítására alkalmas, így a két párosíthatatlan módszer eredményét nem tudtuk vele kiértékelni. Az érthetőség mérésére felvetődik a gépi beszéd felismerők használata is (Purohit és mtsai, 2020), azonban ezzel a megoldással kapcsolatban is felmerülnek aggályok (a felismerési hiba nem csak a hangminőségtől függhet). A magyar nyelvű, hasonló jellegű kiértékelések reprodukálhatóságához pedig szükség lenne egy de facto standardként kezelhető, jó hatásfokú és publikusan elérhető magyar nyelvű felismerőre. Végezetül, történtek már kísérletek a konverzió minőségének neuronháló általi megbecslésére az emberi pontozás alapján, speciálisan hangkonverziós feladatra (Lo és mtsai, 2019). Ez az irány ígéretesen hangzott, ezért megpróbálkoztunk a fenti cikkben ismertetett, betanított neuronhálózattal is. Kiderült azonban, hogy normál beszédre a hálózat gyakran rosszabb pontszámokat ad, mint dysarthriás mintákra(!), mivel csakis hangkonverzióval átesett mintákon tanították be. Azt kell mondanunk tehát, hogy semmiképp beszélhetünk kiforrott technológiáról, amely alkalmas lenne a beszédminőség és -érthetőség objektív mérésére.

## 4. Összegzés

A cikkben összefoglaltuk első tapasztalatainkat a hangkonverziós algoritmusok alkalmazhatóságára nézve a dysarthriás beszéd minőségjavításában. Ezek az első tapasztalatok nem túl kedvezőek: úgy látjuk, hogy a jelenlegi, főként GAN neuronhálókra épülő technológia nagyon érzékeny a paraméterbeállításokra, valamint nem teszi lehetővé annak finomhangolását, hogy pontosan milyen mértékű és jellegű transzformáció történjen. Úgy gondoljuk, hogy mind az alkalmazott reprezentáció (vokóderen), mind a tanítási megszorítások megfogalmazásán változtatni, bővíteni kell a továbblépéshez. Fontos lenne továbbá megbízható objektív módszereket találni a beszédminőség javulásának kiértékelésére.



## Köszönetnyilvánítás

Az SZTE Informatikai Intézetének munkáját az Innovációs és Technológiai Minisztérium és a NKFIH támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében. A kutatást az EFOP-3.6.1-16-2016-00008 Intelligens élettudományi technológiák, módszertanok, alkalmazások fejlesztése és innovatív folyamatok, szolgáltatások kialakítása a szegedi tudásbázisra építve c. pályázatának "Nyelvhasználati és jelhasználati jellegzetességek stroke-on átesett személyeknél, különös tekintettel a diagnosztikus és rehabilitációs lehetőségekre" c. pilot programja, valamint a 2019-1.2.1-EGYETEMI-ÖKO-2019-00018 pályázat keretében megvalósuló Proof of Concept Alap Pályázat „Dysarthriával élők támogatása beszédminőségük eszközös javítása által” című projektje támogatta. A TKP2021-NVA-09 számú projekt az Innovációs és Technológiai Minisztérium Nemzeti Kutatási Fejlesztési és Innovációs Alapból nyújtott támogatásával, a TKP2021-NVA pályázati program finanszírozásában valósult meg.

## Hivatkozások

- Aronson, A.: Motor speech signs of neurologic disease. In: Darby, J.K. (szerk.) *Speech Evaluation in Medicine*, pp. 159–180. Grune and Stratton (1981)
- Biadsy, F., Weiss, R.J., Moreno, P.J., Kanevsky, D., Jia, Y.: Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. *arXiv preprint arXiv:1904.04169* (2019)
- Camillo, L., Ortiz, K.: Vocal analysis (auditory-perceptual and acoustic) in dysarthrias. *Pro-Fono Revista de Atualizacao Cientifica* 19(4), 381–386 (2007)
- Chen, L.W., Lee, H.Y., Tsao, Y.: Generative adversarial networks for unpaired voice transformation on impaired speech. *arXiv preprint arXiv:1810.12656* (2018)
- Ferro, R., Onin, N., Roebel, A.: CycleGAN voice conversion of spectral envelopes using adversarial weights. In: *EUSIPCO*. pp. 406–410 (2020)
- Horváth, S., Hirshberg, J.: Diszartria/diszartrofónia. In: Hirschberg J., Hacki T., M.K. (szerk.) *Foniátria és társtudományok II.*, pp. 80–86. Eötvös Kiadó (2013)
- Huang, W.C., Hayashi, T., Wu, Y.C., Kameoka, H., Toda, T.: Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining. *arXiv preprint arXiv:1912.06813* (2019)
- Huang, W.C., Hayashi, T., Wu, Y.C., Kameoka, H., Toda, T.: Pretraining techniques for sequence-to-sequence voice conversion. *IEEE/ACM Transactions on Audio, Speech and Language Processing* pp. 745–755 (2021a)
- Huang, W.C., Kobayashi, K., Peng, Y.H., Liu, C.F., és mtsai: A preliminary study of a two-stage paradigm for preserving speaker identity in dysarthric voice conversion. *arXiv preprint arXiv:2106.01415* (2021b)
- Jia, Y., Weiss, R.J., Biadsy, F., Macherey, W., Johnson, M., Chen, Z., Wu, Y.: Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv preprint arXiv:1904.06037* (2019)

- Kaneko, T., Kameoka, H., Tanaka, K., Hojo, N.: StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion. arXiv preprint arXiv:1907.12279 (2019)
- Kaneko, T., Kameoka, H., Tanaka, K., Hojo, N.: MaskCycleGAN-VC: Learning non-parallel voice conversion with filling in frames. In: Proc. ICASSP. pp. 5919–5923 (2021)
- Kim, H., Hasegawa-Johnson, M., Perlman, A., Gunderson, J., Huang, T.S., Watkins, K., Frame, S.: Dysarthric speech database for universal access research. In: Ninth Annual Conference of the International Speech Communication Association (2008)
- Kim, J.W., Jung, H.Y., Lee, M.: Vocoder-free end-to-end voice conversion with transformer network. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2020)
- Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: International Conference on Machine Learning. pp. 1857–1865. PMLR (2017)
- Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W.Z., Sotelo, J., de Brébisson, A., Bengio, Y., Courville, A.: Melgan: Generative adversarial networks for conditional waveform synthesis. arXiv preprint arXiv:1910.06711 (2019)
- Li, N., Liu, S., Liu, Y., Zhao, S., Liu, M.: Neural speech synthesis with transformer network. In: Proc. AAAI. pp. 6706–6713 (2019)
- Lo, C.C., Fu, S.W., Huang, W.C., Wang, X., Yamagishi, J., Tsao, Y., Wang, H.M.: Mosnet: Deep learning based objective assessment for voice conversion. arXiv preprint arXiv:1904.08352 (2019)
- Luong, M., Tran, V.A.: Flowvocoder: A small footprint neural vocoder based normalizing flow for speech synthesis. arXiv preprint arXiv:2109.13675 (2021)
- Markó, A., Grácsi, T., K., B.S.: A diszfónia terápiájának hatékonysága a beteg beszédtechnikai képzettségének függvényében. *Alkalmazott nyelvtudomány* 12(1–2), 83–103 (2007)
- Mohammadi, S., Kain, A.: An overview of voice conversion systems. *Speech Communication* 88, 65–82 (2017)
- Morise, M., Yokomori, F., Ozawa, K.: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. on Information and Systems* 99(7), 1877–1884 (2016)
- Moulines, E., Sagisaka, Y.e.: Voice conversion: state of the art and perspectives. *Speech Communication (speciális különszám)* 16(2) (1995)
- Novitasari, S., Tjandra, A., Sakti, S., Nakamura, S.: Sequence-to-sequence learning via attention transfer for incremental speech recognition. arXiv preprint arXiv:2011.02127 (2020)
- Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A flow-based generative network for speech synthesis. In: Proc. ICASSP. pp. 3617–3621 (2019)
- Purohit, M., Patel, M., Malaviya, H., Patil, A., Parmar, M., Shah, N., Doshi, S., Patil, H.A.: Intelligibility improvement of dysarthric speech using mmse discogan. In: Proc. SPCOM. pp. 1–5. IEEE (2020)

- Sündermann, D.: Voice conversion: State-of-the-art and future work. In: *Fortschritte der Akustik*. pp. 735–736 (2005)
- Tanaka, K., Kameoka, H., Kaneko, T., Hojo, N.: AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms. In: *Proc. ICASSP*. pp. 6805–6809 (2019)
- Turk, O., Arslan, L.: Subband based voice conversion. In: *Proc. ICSLP* (2002)
- Tóth, L., Kovács, G., Ivaskó, L., Tóth, A., Jakab, K., Vécsei, L.: Stroke-on átesett dysarthriás betegek beszédének gépi elemzése - kezdeti eredmények. In: *Orvosi Informatika. A XXXI. Neumann Kollokvium kiadványa*. pp. 43–49 (2018)
- Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R.J., és mtsai: Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135* (2017)
- Yang, S.H., Chung, M.: Improving dysarthric speech intelligibility using cycle-consistent adversarial training. *arXiv preprint arXiv:2001.04260* (2020)
- Zhang, J.X., Ling, Z.H., Liu, L.J., Jiang, Y., Dai, L.R.: Sequence-to-sequence acoustic modeling for voice conversion. *IEEE/ACM Trans. ASLP* 27(3), 631–644 (2019)
- Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2223–2232 (2017)



## Enyhe kognitív zavar automatikus felismerése szekvenciális autoenkóder használatával

Vetráb Mercedes<sup>1</sup>, José Vicente Egas-López<sup>1</sup>,  
Balogh Réka<sup>2</sup>, Imre Nóra<sup>2</sup>, Hoffmann Ildikó<sup>3,4</sup>, Tóth László<sup>1</sup>,  
Pákási Magdolna<sup>2</sup>, Kálmán János<sup>2</sup>, Gosztolya Gábor<sup>1,5</sup>

<sup>1</sup> Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup> Szegedi Tudományegyetem, Pszichiátriai Klinika

<sup>3</sup> ELKH Nyelvtudományi Kutatóközpont, Budapest

<sup>4</sup> Szegedi Tudományegyetem, Magyar Nyelvészeti Tanszék

<sup>5</sup> ELKH-SZTE Mesterséges Intelligencia Kutatócsoport, Szeged  
{ vetrabm, ggabor } @ inf.u-szeged.hu

**Kivonat** Az enyhe kognitív zavar (EKZ) heterogén klinikai szindróma. Főbb tünetei közé tartozik a memória, a gondolkodás, az érvelés és a nyelvi képességek romlása, amely azonban nem okoz jelentős zavart a páciensek mindennapi életvitelében. A hanyatlás enyhe foka és a lappangó tünetek miatt azonban az EKZ diagnosztizálása nagyon gyakran ütközik nehézségekbe. Ebben a tanulmányban szekvenciális autoenkódert használunk a jellemzőknyeréshez, hogy robusztus és hatékony attribútumokat extraktálhassunk. A felhasznált adattbázis 25 EKZ-s alany és 25 egészséges kontrollszemély hanganyagait tartalmazza. Eredményeink alapján ez a megközelítés versenyképes teljesítményt nyújt: egy nagyobb adattbázison tanított x-vektor hálóval szemben is képes jobb eredményeket nyújtani. További kísérleteinkben enyhe Alzheimer-kórban (eAK) szenvedő alanyokat is megpróbáltuk elkülöníteni.

**Kulcsszavak:** paralingvisztika, demencia, szekvenciális autoenkóder

### 1. Bevezetés

Az enyhe kognitív zavar (EKZ) heterogén tünetegyüttes, melyet főként a memória, a gondolkodás, az érvelés és a nyelvi képességek hanyatlása jellemez. Az EKZ-t gyakran tekintik átmeneti állapotnak a normál öregedés és a demencia között; a kognitív tünetek ebben a stádiumban ugyanakkor még nem befolyásolják jelentősen a páciensek mindennapi életvitelét (Petersen és mtsai, 2014; Alzheimer’s Association, 2020). Az EKZ megjelenése akár 15 évvel is megelőzheti a demencia klinikai manifesztálódását (Laske és mtsai, 2015), amely periódus lehetőséget nyújthat az EKZ korai felismerésére és ezáltal a kognitív hanyatlás progressziójának mérséklésére is (Hahn és Andel, 2011). Ennek ellenére nem csak az EKZ, hanem a demencia is globálisan aluldiagnosztizált kórkép (Lang és mtsai, 2017). Az alacsony felismerési arány rávilágít új, hatékony módszerek szükségességére, amelyek segíthetik a betegség korai szűrését.

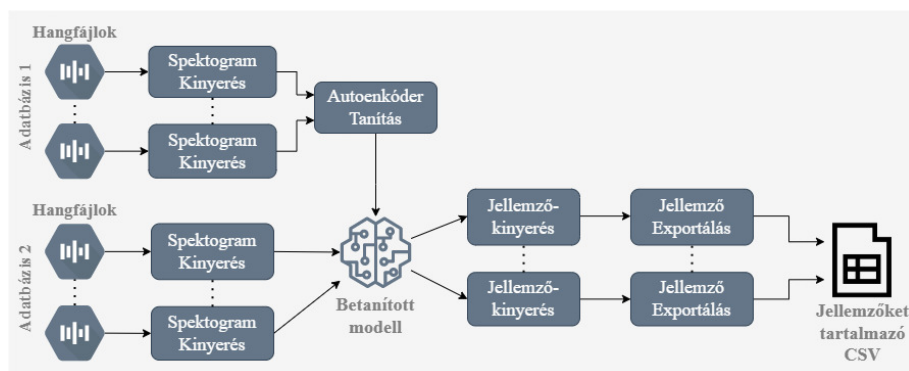
A nyelvi képességek változása az EKZ korai indikátoraként szolgálhat, mivel ezek a változások jóval az egyéb, jellegzetes kognitív tünetek megjelenése előtt is megmutatkozhatnak (McCullough és mtsai, 2019). Bizonyított, hogy a nyelvi teljesítményben bekövetkező változások összefüggést mutatnak a memória szubklinikai hanyatlásával, így például a spontán beszéd fluenssége romlik az EKZ-ban szenvedő páciensek esetében (Mueller és mtsai, 2018). Beszédük a betegség előrehaladtával egyre több szünetet és diszfluenciát tartalmaz (López-de Ipiña és mtsai, 2018), ami valószínűleg a betegek szótalálási nehézségeinek tulajdonítható (Szatlóczki és mtsai, 2015). Ezen beszédjellemzők elemzése ezért lehetővé teszi az EKZ közvetett vizsgálatát. Az automatikus beszédelemzés költség-hatékony, könnyen alkalmazható, non-invazív, és akár telemedicinás eszközként is használható lehet az EKZ tüneteinek kimutatására. Ugyan a közelmúltban számos tanulmány jelent meg az EKZ és a demencia egyéb formáinak detektálásáról (Haulcy és Glass, 2020; Pérez-Toro és mtsai, 2021), továbbra is vizsgálatok tárgyát képezi, hogy pontosan mely jellemzőtípusokat érdemes kinyerni az alanyok beszédéből. Egy lehetséges választás az általános (azaz a tényleges betegségre nem specifikus) jellemzőreprezentációs technikák alkalmazása, mint például az *i*-vektorok (Dehak és mtsai, 2009) és az *x*-vektorok (Snyder és mtsai, 2018). (Ezt a két technikát első sorban beszélőfelismeréshez dolgozták ki, de később más feladatokban is hasznosnak bizonyultak a jellemzők kinyerése során (Grzybowska és Kacprzak, 2016; Huckvale és mtsai, 2020)). Ezeknek a módszereknek a fő előnye, hogy a tanításhoz nincs szükség nagy méretű EKZ adatbázisokra, hanem általános célú, nagy beszédkorpuszokon előtaníthatók.

A mélytanulás-alapú technikák egyre növekvő népszerűségnek örvendenek. Ebben a tanulmányban szekvenciális autoenkódert használtunk a jellemzők kinyerésére. A célunk az volt, hogy megkülönböztessük az EKZ-s és az egészséges alanyok beszédét. Ezeket a technikákat korábban már sikeresen alkalmazták olyan feladatokra, mint például az akusztikus eseményfelismerés (Amiriparian és mtsai, 2017) vagy a főemlősök hangjainak kategorizálása (Schuller és mtsai, 2021). Meglátásunk szerint a végső jellemzőkinyerés során előnyt jelenthet, hogy közvetlenül a nyers hangfájlokat dolgozzuk fel, anélkül, hogy manuálisan konstruált jellemzőkészleteket használnánk (például MFCC-t).

Tanulmányunk újdonsága abban rejlik, hogy szekvenciális autoenkódereket használunk az enyhe kognitív zavar és az enyhe Alzheimer-kór kimutatására. Kísérleti eredményeinkben ez a technika felülmúlta az *x*-vektorok használatával kapott eredményeket, még akkor is, ha csak az adatok egy töredékén tanítottuk autoenkódereinket. Továbbá a peeking elkerülése és a robusztusság növelése érdekében, a kiértékelő halmaztól eltérő adatbázison tanítottuk a hálókat. Tudomásunk szerint ez az első olyan tanulmány amely keresztkorpusz-technikát alkalmaz szekvenciális autoenkódereknél.

## 2. Szekvenciális Autoenkóder

Az autoenkóder neurális hálók régóta jelen vannak a gépi tanulás területén (Hecht-Nielsen, 1995). A tanulási folyamatuk lényege, hogy a háló a bemenetét (ami nem



1. ábra: szekvenciális autoenkóder folyamatábrája.

feltétlenül hang) képes legyen rekonstruálni, miközben a bemenetből kinyert információt egy kis méretű rétegben tömöríti (*bottleneck layer*). Ez a tömörített reprezentáció felhasználható egy esetleges következő osztályozási lépés bemeneti jellemzőjeként. Hangfeldolgozás esetében a változó hosszú hangfájlok feldolgozására rekurrens neurális hálózatok vagy szekvenciális autoenkóderek is alkalmazhatók. Ezeket a technikákat korábban már sikeresen használták különböző feladatoknál (pl.: gépi fordításnál (Luong és mtsai, 2016), eseménydetektálásnál (Amiriparian és mtsai, 2017)).

A mélytanulási módszerek számára jellemzően jobbnak számítanak a nyersebb bemeneti jellemzők (pl. Mel-skála szerinti sávszűrők), mint a jobban előfeldolgozottak (pl. MFCC vagy PLP) (Mohamed és mtsai, 2011). Az autoenkóderek használatához első lépésben ki kell számítanunk a Mel-skála spektrogramokat a nyers hangokból. Amiriparian és munkatársai tanulmányát követve a Mel-spektrumokat a  $[-1, 1]$  intervallumba normalizáljuk, ami jobban illeszkedik a neurális háló bemeneti tartományához (Amiriparian és mtsai, 2017). Ez a spektrum lesz a rekurrens neurális háló enkóderének bemenete. Az enkóder állhat akár Long Short-Term Memory (LSTM) vagy Gated Recurrent Unit (GRU) egységekből is.

Az enkóder háló utolsó rejtett rétegében lévő neuronok adják a bemenetből képzett tömörített (enkódolt) reprezentációt. Ezt a réteget a dekódoló háló követi, ami szintén LSTM vagy GRU neuronokból áll. A dekódoló feladata, hogy a tömörített reprezentációból rekonstruálja a bemenetet. A dekódoló szintén lehet egyirányú vagy kétirányú hálózat. A tanítás során a teljes hálót betanítjuk a bemenetek rekonstrukciójára. A háló súlyainak optimalizálása RMSE (Root Mean Square Error) hiba használatával történik, amelyet a bemenet és a kimenet eltéréséből kapunk. A betanítás után a háló középső, bottleneck rétege felhasználható jellemzőkinyeréshez. Ehhez elég megtartani az enkódoló hálót és annak megtanult súlyait.

Általában az autoenkóder hálót ugyanazon a korpuszon tanítjuk, mint amelyet majd a végső osztályozási feladathoz is használunk. Ennek egyik hátránya,

hogy a háló képes lehet megtanulni az adott adatbázisra jellemző tulajdonságokat. A másik hátrány, hogy az orvosi beszédfeldolgozás területén az adatbázisok mérete általában rendkívül kicsi, mivel az adott betegségben szenvedő alanyokat nehéz nagy létszámban elérni, valamint a pontos diagnosztizáláshoz szakszerűen képzett személyzetre (pl. orvosokra) van szükség. Ezeknek a hátrányoknak a kiküszöbölésére az enkóder hálót egy másik, általános célú adatbázison tanítottuk, hogy a jellemzőkinyerés minél robosztusabb maradjon. Az alkalmazott eljárás általános menetét az 1. ábrán szemléltettük.

### 3. Adatbáziskinyerés

A hangfelvételek a Szegedi Tudományegyetem Pszichiátriai Klinikáján kerültek rögzítésre. Ezekből a felvételekből összesen 50 beszélő anyagát választottuk ki. A felhasznált adatbázisrész 25 enyhe kognitív zavarral rendelkező beteg (EKZ) és 25 egészséges kontroll (EK) felvételeiből áll. A beszélőket az alapján választottuk ki, hogy a két vizsgálati csoport között ne legyen szignifikáns különbség a következők tekintetében: nem ( $p = 0,734$ ), életkor ( $p = 0,150$ ) és az oktatásban eltöltött évek száma ( $p = 0,214$ ). Minden alany anyanyelvi magyar beszélő volt. Kizáró kritérium volt a kábítószer- vagy alkoholfogyasztás, a kognitív funkciókat befolyásoló gyógyszeres kezelésen való részvétel, a depresszió, a fejsérülés vagy pszichózis a kórtörténetben és a látás- vagy halláskárosodás. Az EKZ-betegeket orvosi diagnózist követően választották ki, amelyet neuropszichológiai vizsgálatokkal és CT vagy MRI vizsgálatokkal támasztottak alá. A spontán beszédre összpontosítottunk: a protokollunkban az alanyokat arra kértük meg, hogy meséljék el az előző napjuk eseményeit. A válaszokat diktafonnal vettük fel és csiptetős mikrofonnal rögzítettük. A válaszok időtartama 25...325 másodperc között volt, míg az átlagos időtartam 89,8 másodperc volt.

## 4. Kísérleti körülmények

### 4.1. Szekvenciális autoenkóderek

A kísérletek során az autoenkóder a BEA korpusz egy részhalmazán tanítottuk (Neuberger és mtsai, 2014), amely magyar spontán beszédet tartalmaz. A teljes BEA adatbázisból először leválogattuk azokat a felvételeket, amelyek a válaszadó alanyokhoz tartoztak. Ezután ezeket ezresével megkevertük a véletlenszerű sorrend érdekében és megtartottuk az első ezer felvételt. Ebben a részhalmazban 16 alany beszéde szerepel, amelynek teljes időtartama 3 óra 59 perc. Jellemzőkinyerésre és a neuronháló megvalósítására az AuDeep szoftvert használtuk (Freitag és mtsai, 2018), amely Python nyelven íródott. Előzetes tesztek eredményei alapján 128 log-skálájú Mel-spektrogram szűrőket alkalmaztunk 0,08ms széles ablakokkal és 0,04ms átfedéssel. Optimalizálóként az Adam algoritmust használtuk 0,001-es tanulási rátával, és 0,2 valószínűségű dropout-tal. A háló enkóder és dekóder része 2 – 2 rekurrens rétegből állt, amelyek mindegyike



128 GRU RNN neuront tartalmazott. A dekóder háló kétirányú volt. A modelleket 64-es mini-batch mérettel, valamint 32 epoch-kal tanítottuk. Ahogy korábbi szakirodalmak javasolják (lásd pl. (Amiriparian és mtsai, 2017; Schuller és mtsai, 2021)), mi is próbálkoztunk a háttérzaj eltávolításával. Az AuDeep szoftver az elkészített spektogramokat minden esetben 0 dB-hez normalizálja. A normalizálás után a programba beépített kapcsoló segítségével  $-30/ -45/ -60/ -75$  dB-es küszöbérték alatt levágtuk a spektogramokat. A végső osztályozást elvégeztük a küszöbérték használata nélkül, valamint a 4 küszöbértékkel, és a négy küszöbértékkel kinyert jellemzővektorok egyesítésével (konkatenálásával) is.

## 4.2. Előfeldolgozás

Bár a szekvenciális autoenkóderek elméletileg képesek bármilyen időtartamú felvételt kezelni, az Audeep Tensorflow alapú megvalósításának korlátai miatt a gyakorlatban csak korlátozott hosszúságú hangfelvételeket tudtunk feldolgozni. Emiatt a tanítás előtt a BEA korpusz összes felvételét felosztottuk 5 másodperces darabokra. Demencia-adatbázisunkat a jellemzőkinyerés előtt szintén 5 másodperces darabokra kellett vágni. Az 50 felvételből összesen 1371 darabot kaptunk. Mivel az alanyok válaszait tartalmazó felvételek hossza változó volt ezért a felvételenkénti darabok száma 5-től 60-ig terjedt, valamint egy felvételhez átlag 27, 42 darab készült.

## 4.3. Osztályozás

Lineáris SVM osztályozó módszert használtunk (libSVM implementáció (Chang és Lin, 2011)); a  $C$  komplexitás paramétert a  $10^{-5}, 10^{-4}, \dots, 10^1$  tartományban vizsgáltuk. A tanítás során 25-szörös keresztvalidációt (CV) alkalmaztunk, úgy, hogy minden egyes csoport (fold) egy egészséges és egy EKZ alany adataiból állt. A teljesítményt osztályozási pontossággal (egyenlő hiba (Equal Error Rate, EER) mellett vett döntési küszöbérték esetén) és görbe alatti terület (AUC) számításával mértük. Tanítás előtt a jellemzőket minden esetben standardizáltuk.

Az osztályozáshoz minden, azonos felvételekhez tartozó darabot, valamilyen módon együtt kellett kezelnünk a végső felvételnév meghatározásához. Ehhez két módszert is vizsgáltunk.

Az elsőben az osztályozást az 5 másodperces darabok szintjén végeztük el. Ezután minden felvétel (alany) osztályozásához vettük az adott felvételhez tartozó összes darab egyenkénti predikcióinak (poszteriorvektorának) számtani átlagát. Így mindkét osztályunkra (EKZ és EK) előállt egy-egy predikciós érték, amik közül a nagyobb alapján választottuk ki a predikált beszélő kategóriát.

Egy másik osztályozási módszerként, ahhoz, hogy megkapjuk az egyes felvételek osztályát, vettük az adott felvételhez tartozó összes darab jellemzővektorát majd ezeket a jellemzőket átlagoltuk és ezen a végső jellemzővektoron futtattuk le az osztályozást. (Ez bevett eljárás más beágyazás-alapú jellemzők, például  $x$ -vektorok esetén is.)

1. táblázat. A pontosság (Acc.) és az AUC értékek különböző tesztek során. Jellemzőkinyerés szekvenciálisautoenkódnél vágott és vágás nélküli adatokkal, és a vágások utáni jellemzők összefűzésével. A felvételek hosszát használva jellemzőnek és az X-vektor jellemzőkinyerésével.

Jellemzőkinyerési technika		Különálló		Átlagolt	
		Acc.	AUC	Acc.	AUC
Szekvenciális autoenkóder	-30 dB	64%	0,694	44%	0,482
	-45 dB	60%	0,706	56%	0,523
	-60 dB	68%	0,734	52%	0,478
	-75 dB	<b>72%</b>	<b>0,763</b>	64%	0,650
	Összefűzött	68%	0,643	48%	0,440
	Nincs vágás	68%	0,715	64%	0,603
Felvétel hossza		—	—	60%	0,615
x-vektor		—	—	60%	0,680

Viszonyításként egy másik modellt is tanítottunk, amelynek a felvételek hossza alapján kellett osztályoznia a mintákat, ahol a felvételek hosszát egyszerűen az 5 másodperces darabok számával közelítettük.

## 5. Eredmények

Az eredményeket megvizsgálva (1. táblázat) láthatjuk, hogy a frekvenciák egy bizonyos dB küszöbérték alatti levágása egyértelműen befolyásolja az EKZ osztályozásának sikerességét.

A tesztjeink során, a legalacsonyabb küszöbérték (−75 dB) eredményezte a legjobb pontosságot és AUC pontszámokat a felvételdarabok jellemzővektorainak különálló kezelésekor (72% és 0,763) és a vektorok átlagolásának esetében is (64% és 0,650). Ezekről a −60 dB-es vágáshoz tartozó eredmények sem maradtak el sokkal. Meglepő módon a négy jellemzővariáció összefűzése nem javított, hanem rontott az osztályozás pontosságán minden esetben. Habár a különálló darabok osztályozásával kapott 68%-os pontosság nem sokkal kevesebb, mint a legjobb 72%-os érték, a 0,643-as AUC-érték mégis itt a legalacsonyabb. A darabok jellemzővektorainak átlagolásakor az összefűzés jelentősen rontotta az osztályozást. A küszöbérték szerinti vágást nem tartalmazó hangfájlokból való jellemzőkinyerés eredményei érdekes eredményt hoztak. Nézzük először azt az esetet, mikor nem volt vágás és a hangfájlok darabjait először osztályoztuk és annak eredményét átlagoltuk. Ekkor szinte megegyező eredményeket kaptunk a korábbi, vágások jellemzőit összefűző módszerhez (68% és 0,715). Másodjára mikor a darabok jellemzőit átlagoltuk, a vágás nélküli eredményünk szinte megegyezett az ottani legjobb −75 dB-es eredményekkel (64% és 0,603).

Baselineként tanítottunk egy x-vektoros jellemzőkinyerő neuronhálót is a korábban említett BEA adatbázis 60 óra 14 másodpercnyi anyagával, 165 beszélővel és 40 Mel-frekvenciájú szűrősort alkalmazva („FBANK”). A szekvenciális

autoenkóderek legjobb pontosságai egyértelműen jobbnak adódtak, mintha az osztályozást csak a felvételek hossza alapján, vagy  $x$ -vektorral végeztük volna. Meglepő módon mind a pontosság, mind az AUC érték jelentősen alacsonyabb a szekvenciális autoenkóderekhez képest. Ez jól mutatja, hogy a tanulmányunkban alkalmazott módszertan valóban versenyképes.

### 5.1. Enyhe Alzheimer-kór vizsgálata

A következő kísérletben azt vizsgáltuk, hogy lehet-e a korábban használt jellemzőket három beszélő kategória megkülönböztetésére használni. Ehhez a 25 EKZ és a 25 kontroll alany (EK) mellett 25 enyhe Alzheimer-kóros (eAK) beteg beszédfelvételeit is felhasználtuk. Természetesen, az új csoport is megfelelt a korábbi csoportok statisztikai jellemzőinek az életkor, a nemi hovatartozás és az iskolai végzettségi szint tekintetében.

Az eAK betegek felvételeiből történő jellemzőkinyerésre a korábban betanított szekvenciális autoenkódereket használtuk. Az SVM modelljeinket a három osztály megkülönböztetésére tanítottuk, a korábban is használt keresztvalidációs módszerrel, tehát minden foldba egy-egy felvétel tartozott a három beszélőcsoportból.

A 2. táblázatban szemléltettük eredményeinket. Az AUC értékek tekintetében a kétszélyes esethez hasonlóan a  $-75$  dB-es küszöbértéket használva kaptuk a legjobb eredményt. Azt is megfigyelhetjük, hogy a legnehezebb feladat az eAK betegek megkülönböztetése volt a többi beszélőtől. (Olyannyira, hogy a  $-30$  dB-es vágás esetén egyáltalán nem sikerült őket azonosítani.) Ez meglepő, mivel a markásabb tünetek miatt általában könnyebb feladatnak számít az eAK alanyok megkülönböztetése az egészséges kontrolloktól, mint az EKZ enyhe tüneteinek kimutatása. Itt is megvizsgáltuk azt az esetet, mikor a hangfájlokat nem vágjuk le semmilyen küszöbérték alatt. Az EKZ alanyok megkülönböztetésében hasonlóan jól teljesített, mint a legjobb  $-75$  dB-es értékünk (0,703 és 0,710). Az eAK betegek felismerésében pedig ez a módszer bizonyult a legjobbnak (0,629).

## 6. Összegzés

Kutatásunkban az enyhe kognitív zavar (EKZ) detektálása volt a cél, amely az alany spontán beszédét is befolyásolja. A hangfelvételekből történő jellemzőkinyeréshez mély tanulást, azon belül szekvenciális autoenkóderet használtunk. A hálót olyan általános adatbázison tanítottuk, amely spontán beszédet tartalmaz. Ezáltal a háló tanítása és kiértékelése során keresztkorpusz technikát alkalmaztunk. A tömörített jellemzőreprezentációt az autoenkóderben lévő enkóder háló utolsó rétege adta. Kísérleti eredményeink azt mutatják, hogy ezzel az eljárással még jobb eredményeket értünk el, mint  $x$ -vektorok alkalmazásával, ráadásul az  $x$ -vektorok tanításához képest most a tanítóadat töredékét használtuk csak fel. Ezáltal kijelenthetjük, hogy a keresztkorpuszon tanított autoenkóder versenyképes a EKZ detektálásának területén. Eredményeink hozzájárulhatnak egy olyan

2. táblázat. Az AUC értékek a 3 osztályos kiértékelésnél. Jellemzőkinyerés szekvenciális autoenkódernél vágott és vágás nélküli adatokkal, és a vágások utáni jellemzők összefűzésével. A felvételek hosszát használva jellemzőnek és az  $X$ -vektor jellemzőkinyerésével.

Jellemzőkinyerési technika		AUC		
		EK	EKZ	eAK
Szekvenciális autoenkóder	-30 dB	0,706	0,618	0,503
	-45 dB	0,714	0,633	0,569
	-60 dB	0,732	<b>0,706</b>	0,606
	-75 dB	<b>0,771</b>	<b>0,710</b>	0,589
	Összefűzött	0,701	0,622	0,598
	Nincs vágás	0,682	<b>0,703</b>	<b>0,629</b>
Felvétel hossza		0,637	0,641	0,417
x-vektor		0,753	0,546	<b>0,606</b>

automatikus, nem invazív és költséghatékony EKZ-szűrésre alkalmas rendszer kialakításához, amely még csak nem is igényel személyes találkozást az alannal (ahol a kontaktnélküliség egy, a Covid-19 világjárványhelyezethez hasonló időszakban kifejezetten hasznos).

## Köszönetnyilvánítás

A tanulmányt a Nemzeti Kutatási Hivatal Fejlesztési és Innovációs Hivatala az NKFIH-FK-124413 szerződés keretein belül támogatta. Továbbá a Magyar Innovációs és Technológiai Minisztérium az NKFIH-1279-2/2020 számú pályázat és az Innovációs és Technológiai Minisztérium NKFIH Iroda Mesterséges Intelligencia Nemzeti Laboratóriumi Program (MILAB) keretében nyújtott támogatás. Gosztolya G. támogatásban részesült továbbá a Magyar Tudományos Akadémia által kiadott Bolyai János -ösztöndíjban és az Innovációs és Technológiai Minisztérium Új Nemzeti Kutatási Kiválósági Program ÚNKP-21-5-SZTE keretén belül.

## Hivatkozások

- Alzheimer's Association: 2020 Alzheimer's disease facts and figures. *Alzheimer's & Dementia* 16(3), 391–460 (2020)
- Amiriparian, S., Freitag, M., Cummins, N., Schuller, B.: Sequence-to-sequence autoencoders for unsupervised representation learning from audio. In: *Proceedings of DCASE*. pp. 17–21 (2017)
- Chang, C.C., Lin, C.J.: LIBSVM: A library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* 2, 1–27 (2011)
- Dehak, N., Kenny, P., Dehak, R., Glembek, O., Dumouchel, P., Burget, L., Hu-beika, V., Castaldo, F.: Support Vector Machines and Joint Factor Analysis for speaker verification. In: *Proceedings of ICASSP*. pp. 4237–4240 (2009)

- Freitag, M., Amiriparian, S., Pugachevskiy, S., Cummins, N., Schuller, B.: auDeep: Unsupervised learning of representations from audio with Deep Recurrent Neural Networks. *Journal of Machine Learning Research* 18(173), 1–5 (2018)
- Grzybowska, J., Kacprzak, S.: Speaker age classification and regression using i-vectors. In: *Proceedings of Interspeech*. pp. 1402–1406 (2016)
- Hahn, E., Andel, R.: Nonpharmacological therapies for behavioral and cognitive symptoms of mild cognitive impairment. *Journal of Aging and Health* 23(8), 1223–1245 (2011)
- Haulcy, R., Glass, J.: Classifying alzheimer’s disease using audio and text-based representations of speech. *Frontiers in Psychology* 11, 624137 (2020)
- Hecht-Nielsen, R.: Replicator Neural Networks for universal optimal source coding. *Science* 269, 1860–1863 (1995)
- Huckvale, M., Beke, A., Ikushima, M.: Prediction of sleepiness ratings from voice by man and machine. In: *Proceedings of Interspeech*. pp. 4571–4575 (2020)
- López-de Ipiña, K., Martínez-de Lizarduy, U., Calvo, P., Beitia, B., García-Melero, J., Fernández, E., Ecay-Torres, M., Faundez-Zanuy, M., Sanz, P.: On the analysis of speech and disfluencies for automatic detection of Mild Cognitive Impairment. *Neural Computing and Applications* pp. 1–9 (2018)
- Lang, L., Clifford, A., Wei, L., Zhang, D., Leung, D., Augustine, G., Danat, I.M., Zhou, W., Copeland, J.R., Anstey, K.J., Chen, R.: Prevalence and determinants of undetected dementia in the community: A systematic literature review and meta-analysis. *BMJ Open* 7(2), e011146 (2017)
- Laske, C., Sohrabi, H., Frost, S., López-de Ipiña, K., Garrard, P., Buscema, M., Dauwels, J., Soekadar, S., Mueller, S., Linnemann, C., Bridenbaugh, S., Kanagasigam, Y., Martins, R., O’Bryant, S.: Innovative diagnostic tools for early detection of Alzheimer’s disease. *Alzheimer’s & Dementia* 11(5), 561–578 (2015)
- Luong, M.T., Le, Q., Sutskever, I., Vinyals, O., Kaiser, L.: Multi-task sequence to sequence learning. In: *Proceedings of ICLR* (2016)
- McCullough, K., Bayles, K., Bouldin, E.: Language performance of individuals at risk for Mild Cognitive Impairment. *Journal of Speech, Language, and Hearing Research* 62(3), 706–722 (2019)
- Mohamed, A.R., Dahl, G., Hinton, G.: Acoustic modeling using Deep Belief Networks. *IEEE Transactions on Audio, Speech and Language Processing* 20(1), 14–22 (2011)
- Mueller, K., Kosciak, R., Hermann, B., Johnson, S., Turkstra, L.: Declines in connected language are associated with very early mild cognitive impairment: Results from the Wisconsin registry for Alzheimer’s prevention. *Frontiers in Aging Neuroscience* 9, 437 (2018)
- Neuberger, T., Gyarmathy, D., Gráci, T., Horváth, V., Gósy, M., Beke, A.: Development of a large spontaneous speech database of agglutinative Hungarian language. In: *Proceedings of TSD*. pp. 424–431 (2014)
- Pérez-Toro, P., Bayerl, S., Arias-Vergara, T., Vásquez-Correa, J., Klumpp, P., Schuster, M., Nöth, E., Orozco-Aroyave, J., Riedhammer, K.: Influence of the interviewer on the automatic assessment of Alzheimer’s disease in the context

- of the ADReSSo challenge. In: Proceedings of Interspeech. pp. 3785–3789 (2021)
- Petersen, R., Caracciolo, B., Brayne, C., Gauthier, S., Jelic, V., Fratiglioni, L.: Mild Cognitive Impairment: A concept in evolution. *Journal of Internal Medicine* 275(3), 214–228 (2014)
- Schuller, B., Batliner, A., Bergler, C., Mascolo, C., Han, J., Lefter, I., Kaya, H., Amiriparian, S., Baird, A., Stappen, L., Otth, S., Gerczuk, M., Tzirakis, P., Brown, C., Chauhan, J., Grammenos, A., Hasthanasombat, A., Spathis, D., Xia, T., P.Cicuta, Rothkrantz, L., Zwerts, J., Treep, J., Kaandorp, C.: The INTERSPEECH 2021 computational paralinguistics challenge: COVID-19 cough, COVID-19 speech, escalation & primates. In: Proceedings of Interspeech. pp. 431–435 (2021)
- Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust DNN embeddings for speaker verification. In: Proceedings of ICASSP. pp. 5329–5333 (2018)
- Szatlóczki, G., Hoffmann, I., Vincze, V., Kálmán, J., Pákási, M.: Speaking in Alzheimer’s disease, is that an early sign? Importance of changes in language abilities in Alzheimer’s disease. *Frontiers in Aging Neuroscience* 7, 195 (2015)

# Deep learning alapú agyi jel feldolgozás és beszéd-szintézis előkészítő munkálatai

Arthur Frigyes Viktor, Csapó Tamás Gábor

Budapesti Műszaki és Gazdaságtudományi Egyetem,  
Távközlési és Médiainformatikai Tanszék

{arthur,csapot}@tmit.bme.hu

**Kivonat** Ebben a cikkben bemutatjuk a nem invazív, elektorenkefalografán (EEG) alapuló kommunikációs agy-gép interfész (BCI) rendszerekkel kapcsolatos kutatásunk kezdeti eredményeit. Az agyi jelből akusztikus jelbe átalakító konverziós módszer célja, hogy az idegsejtek elektromos aktivitása alapján szintetizáljunk beszédet. A kutatás során a "Kara One" adatbázis beszélőivel végeztünk kísérleteket, melyből párhuzamos EEG és beszédfelvételeket használtunk fel. A cikkben részletesen bemutatjuk az agyi jel előfeldolgozásának lépéseit (szűrés, csatorna választás, zajok és műtermékek (artifact) eltávolítása, független komponens analízis, és szegmentálás). Az előfeldolgozott EEG jelből mély neuronhálóval becsüljük meg a beszéd mel-spektrális paramétereit. Végül egy neurális vokóderrel állítjuk elő a szintetizált beszédet. Az így szintetizált beszéd ugyan nem érthető, de emlékeztet az eredeti jelre, így a kezdeti eredményeket biztatóknak tartjuk. A kutatás hosszú távú jelentősége, hogy a beszéd valós idejű szintézise közvetlenül a mért idegi aktivitásból lehetővé tenné a természetes beszédet, és jelentősen javítaná az életminőségét, különösen a kommunikációban súlyosan korlátozott személyek számára.

**Kulcsszavak:** beszéd-szintézis, agy-gép interfész (BCI), elektorenkefalografia (EEG), némabeszéd-interfész (SSI)

## 1. Bevezetés

A beszéd az emberi kommunikáció elsődleges és legfontosabb eszköze. Sokan azonban elvesztették ezt a képességüket betegség vagy egészségkárosodás okán. Nincs sok olyan tanulmány, amely konkrét adatokat szolgáltatna ennek a fogyatékoságnak a gyakoriságáról. Dupré és Karjalainen (2003) arra a következtetésre jutnak, hogy az európai lakosság 0,4%-a szenved beszédkárosodásban. Egy későbbi, 2011-es felmérésben (Eurostat, 2011) arról számolnak be, hogy Európában az emberek 0,5%-a küzd kommunikációs nehézségekkel. Az általunk alkalmazott megközelítésben az agyi jelből akusztikus jelbe átalakító konverziós módszer célja, hogy az idegsejtek elektromos aktivitása alapján szintetizáljunk beszédet.

### 1.1. Agyi jelből akusztikus jelbe átalakítás elektroencefalogram alapján

A kommunikációs agy-gép interfész (BCI) célja, hogy természetes vagy ahhoz közeli kommunikációs csatornát biztosítsanak olyan személyek számára, akik fizikai vagy neurológiai károsodás miatt nem tudnak beszélni. A beszéd valós idejű szintézise közvetlenül a mért idegi aktivitásból lehetővé tenné a természetes beszédet, és jelentősen javítaná az életminőségét, különösen a kommunikációban súlyosan korlátozott személyek számára. A jelen cikkben leírt megközelítésünkben non-invazív módon mintavételezett elektroencefalográf (EEG) segítségével regisztrációra kerül a beszélő agyi elektromos aktivitása, mely jelből közvetlenül, a megfelelő átalakítási lépések után beszédet szintetizálunk. A konverziónak több lehetséges megoldása létezik.

Angrick és mtsai (2021) minimál invazív (sEEG) módon mért idegi aktivitásból dekódolási megközelítéssel a beszéd jó minőségű rekonstrukcióját tudták megvalósítani a közelmúltban. Sztereotaktikus mélységi elektródákkal beültetett résztvevővel megbízhatóan tudtak érthető beszédet generálni valós időben. Egy súlyos epilepsziás beteg (20 éves nő) orvosi kezelése során 11 sEEG-elektrodaszárat ültettek be a bal féltékébe, 8–18 érintkezővel. Az elektródákat az epilepsziás fókuszok meghatározása és az agykérgi funkciók feltérképezése céljából ültették be, hogy azonosítsák azokat a kritikus területeket, amelyek reszekciója hosszú távú funkcionális hiányosságokat eredményezhet (Angrick és mtsai, 2021). A bemutatott eredmény imponáló, a beszéd-neuroprotézis széleskörű alkalmazhatóságának azonban korlátot szabhat, hogy ez a megoldás invazív műtéti beavatkozáshoz kötött.

Egy másik megközelítésben Herff és mtsai (2015) intrakraniális elektroencefalográfiás (ECoG) felvételekből állítottak elő szöveget (agy jel – szöveg konverzió). Az eredmények szerint a szóhibaarány 50%, míg a fonéma hibaarány 25%-os. A szövegből egy TTS és neurális vokóder (Tacotron, Glow-TTS, SpeedySpeech, GAN-TTS) segítségével beszéd konvertálható. A viszonylag magas hibaarányok ugyan a valós életbeli felhasználást még nem teszik lehetővé, de mindenképp előremutatóak.

### 1.2. A jelen kutatás célja

A kutatás célja egy megvalósíthatósági tanulmány: azt teszteltük, hogy közvetlen agyi jelből, az idegsejtek elektromos aktivitásának regisztrátumából, érthető akusztikum közvetlenül előállítható-e mély tanulással.

## 2. Kísérlet

Először bemutatjuk röviden a mögöttes adatbázist, illetve áttekintjük a kísérlet alapjait.



## 2.1. Az adatbázis

Kutatásunkban a meglévő "Kara One Dataset" elnevezésű adatbázis került alkalmazásra, mely 12 résztvevő multimodális fiziológiai jeleiről készült felvételeit tartalmazza kiejtett beszéd, elképzelt beszéd és beszéd hallgatás közben (Zhao és Rudzicz, 2015). Minden résztvevő jobb kezes, 10 résztvevő angol anyanyelvű, míg 2 felsőfokon beszéli az angol nyelvet. Minden résztvevő legalább középfokú végzettséggel rendelkezik, valamint nem ismert, hogy lenne bármilyen neurológiai betegségük vagy álltak volna korábban kábítószer használat alatt. A kísérletben nyolc férfi valamint négy női beszélő vett részt, átlagéletkoruk 27,4 év ( $\sigma = 5$ ) (Zhao és Rudzicz, 2015).

Az adatrögzítés helyszíne a Toronto Rehabilitációs Intézet egyik irodája volt. A videó és a beszéd rögzítésére a Microsoft Kinect kamera rendszert illetve szoftvert alkalmazták, míg az EEG jel rögzítésére 64 csatornás NeuroScan "Quickcap" sapkát használtak 10–20 csatorna kiosztási módban (Sharbrough és mtsai, 1991). Négy további elektródát helyeztek el az alanyokon: a bal szem alatt és felett, valamint egyet-egyet a jobb és bal szem mellett laterálisan, hogy később a horizontális és vertikális szemmozgás által okozott műtermékek (artifact) könnyen szűrhetőek legyenek az EEG jelből. A mintavételezési frekvencia 1000 Hz volt.

A kísérletet három elkülönített fázisra bontották, melyben az alanyokat arra kérték, hogy 7 fonémával és 4 szótaggal különböző módon járjanak el. Az első fázisban az alanyokat arra kérték, hogy hallgassák azokat, a másodikban gondoljanak rá, míg a harmadikban, hogy hangosan mondják ki őket. A jelen kutatásban a fenti fázisokból csak azokat a részeket használtuk fel, amikor a kísérleti alany kimondta a fonémát vagy szótagot.

Fonémák	Szótagok
/iy/	pat
/uw/	pot
/piy/	knew
/tiy/	gnaw
/diy/	
/m/	
/n/	

1. táblázat. Rögzített promptok.

A 4 szótag a Kent-féle, fonetikailag hasonló párok listájából származott (Kent és mtsai, 1989). A promptok (stimulusok) kiválasztásakor figyelembe vették, hogy viszonylag egyenlő számú nazális, zárhang és magánhangzó, valamint zöngés és zöngétlen fonéma legyen jelen.

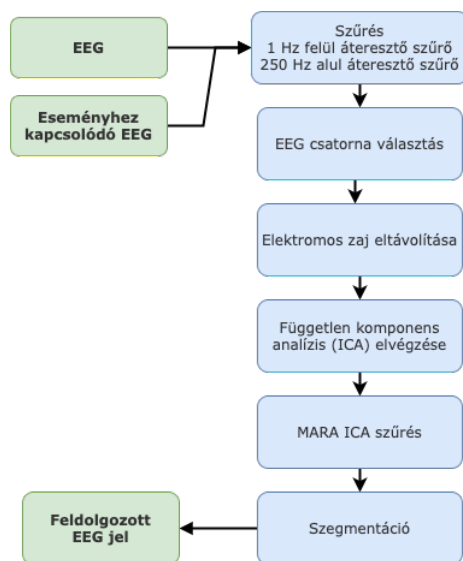
## 2.2. Az adatrögzítés módja

A résztvevők a következő feladatot hajtották végre.

1. Nyugalmi állapot: (5 másodperc) A résztvevőknek azt az utasítást adták, hogy relaxáljanak és tisztítsák ki az elméjüket.
2. Stimulációs állapot: A képernyőn megjelent a fonémát vagy szótagot megjelenítő felirat, és a számítógép hangszóróin keresztül is szólt. Ezután a résztvevőket arra utasították, hogy vegyék fel az adott fonéma vagy szótag kiejtéséhez szükséges kiindulási pozíciót.
3. Elképzelt állapot: (5 másodperc) A résztvevők elképzelték, hogy kimondják az adott fonémát vagy szótagot, mindezt mozdulatlanul.
4. Beszélő állapot: A résztvevők hangosan kimondták a fonémát vagy szótagot. A Kinect kamera rendszer ebben a szakaszban mind a hangot, mind az arcmozgást rögzítette.

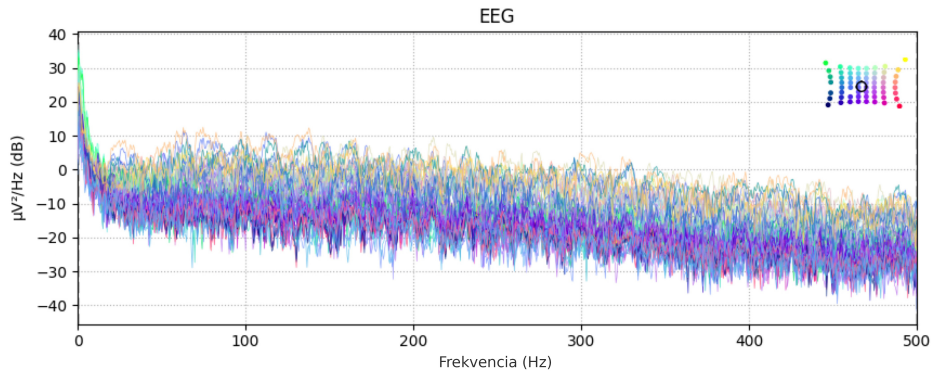
### 2.3. Az agyi jel előfeldolgozása

Saját kísérleteinkben az agyi jelek előfeldolgozása lazán követi a HAPPE (Harvard Automated Processing Pipeline for Electroencephalography) (Gabard-Durnam és mtsai, 2018) munkafolyamatát, lényegében a folytonos wavelet transzformációs szakaszig, mely helyett közvetlen szegmentációs szakasz következik (ld. 1. ábra).



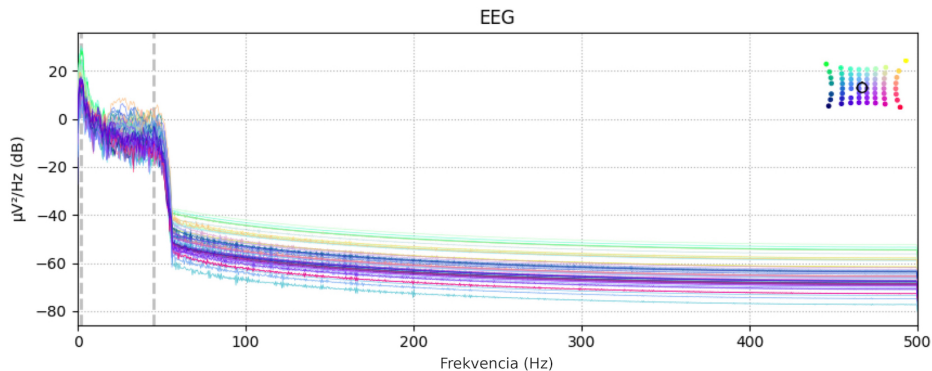
1. ábra: EEG jelfeldolgozási folyamat.

**Szűrés** Az elektroencefalogram feldolgozásának első lépése a szűrés. Egy 1 Hz-es felül-áteresztő szűrő eltávolítja nem stacionárius jelsodródást a felvételen. Ezt követően egy 250 Hz-es alul-áteresztő szűrőt alkalmaztunk.



2. ábra: A szűrést megelőző elektroencefalogram teljesítménysűrűség spektruma - [MM05 beszélő].

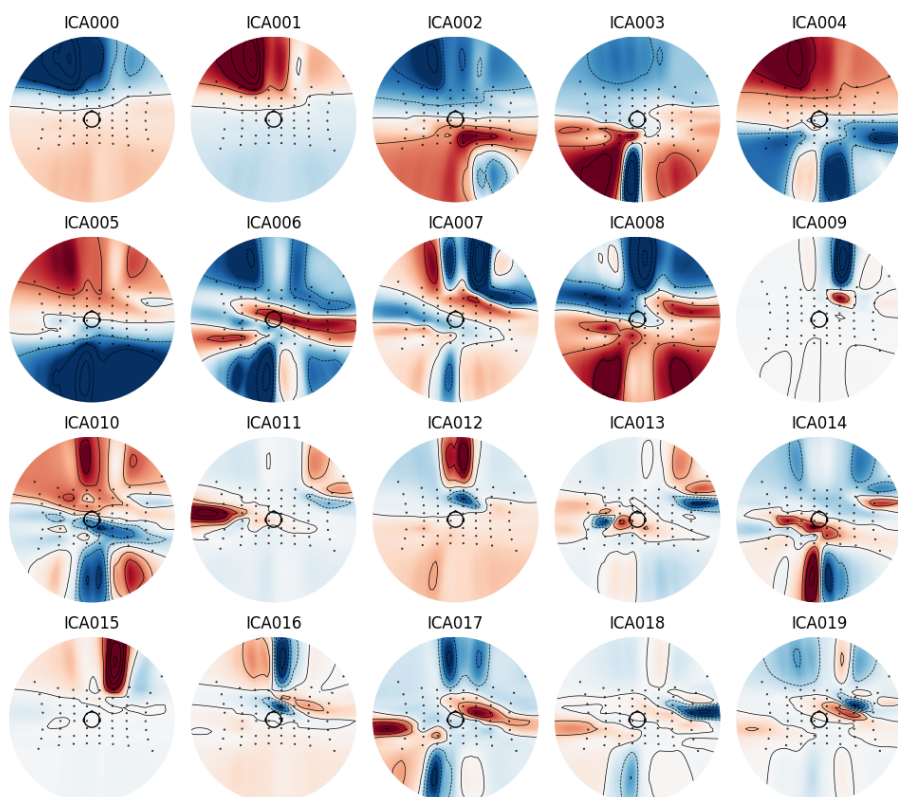
**Tápvezetési zaj eltávolítása** Az elektromos hálózat által keltett zaj a tápvezetési zaj. Éles csúcsokból áll 50 Hz-en vagy 60 Hz-en, a földrajzi helytől függően. Tekintve, hogy a Kara One adatbázist Kanadában rögzítették, ahol az elektromos hálózat harmonikus frekvenciája 60 Hz, így ennek a frekvenciának a szűrését végeztük el. Ez egy egyszerű alul-áterező segítségével történt, melynek használata egyéb okból is indokolt, hiszen, elsősorban béta (12–40 Hz) agyhullámokkal dolgozunk. Az eredményre a 3. ábra mutat egy példát.



3. ábra: Tápvezetési zaj eltávolítása után az elektroencefalogram teljesítménysűrűség-spektruma [MM05 beszélő].

**EEG csatorna választás** A csatorna választás során első lépésben azokat a csatornákat szűrjük ki, melyek esetleg egyértelmű elektróda hibával rendelkeznek. Majd azokat, melyek regisztrátumai nem korrelálnak a megfigyelt agyi tevékeny-





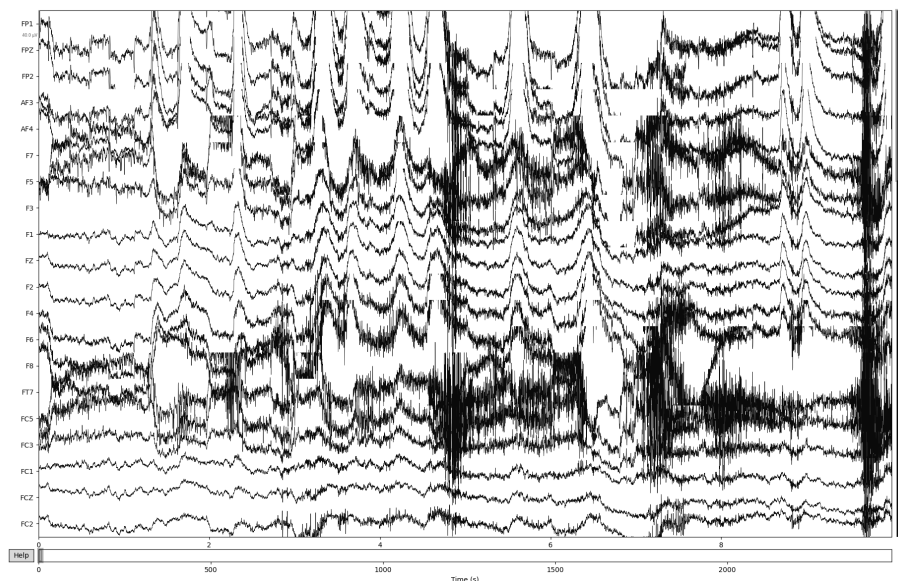
5. ábra: 20 független komponenst megjelenítő skalp topográfiai térkép (12 óra pozícióban az orr, 3 óra pozícióban a jobb míg 9 óra pozícióban a bal fül található) - [MM05 beszélő].

**MARA ICA szűrés** A Multiple Artifact Rejection Algorithm (MARA) egy gépi tanulási algoritmus, amely kiértékeli az ICA-ból származó összetevőket (Winkler és mtsai, 2014). Bár más algoritmusok is léteznek a műtermék bizonyos kategóriáinak automatikus észlelésére (pl. szemmozgási műtermékek és jelek folytonossági zavarai) (Mognon és mtsai, 2011), a MARA-t manuálisan kialakított komponens kategóriák felismerésére tanították, így viszonylag alacsony domain tudás nélkül is automatikusan képes a műtermékek széles skáláját szűrni.

**Szegmentáció** A szegmentációs lépés lényegében annotálja és epochokhoz sorolja a folytonos elektroencefalogramot, így kialakítva fragmentumokat. Beszélőnként 131 illetve 132 próbát különböztetünk meg. A szegmentáció HAPPE megközelítésben csak egy opcionális lépés, viszont számunkra elengedhetetlen volt, tekintve, hogy minden egyes epoch egy fonémához vagy szótaghoz tartozó

elektroenkefalogram szakaszt különböztet meg, melyből becsülni szándékoztuk az akusztikumot, pontosabban annak mel-spektrális paramétereit.

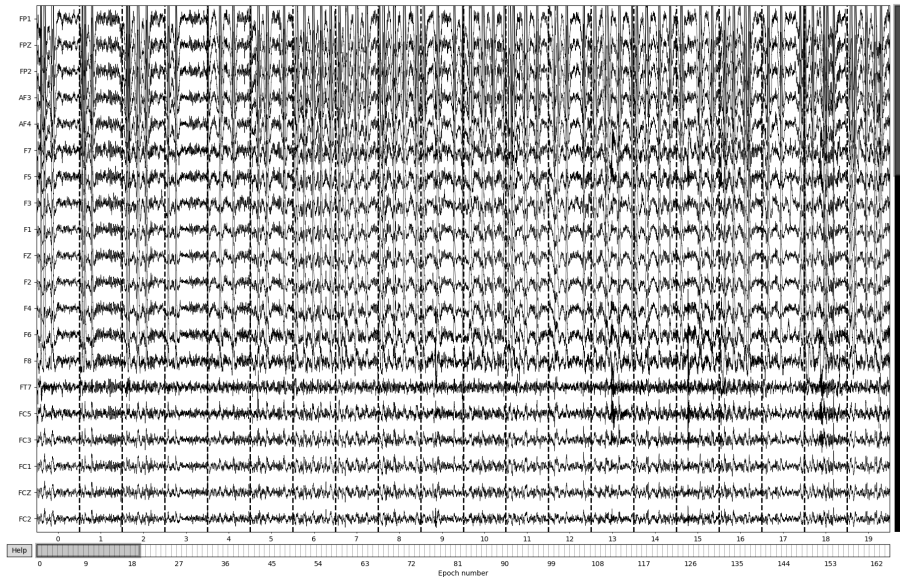
Az 6. ábrán látható egy példa nyers elektroenkefalogramra, majd a feldolgozási lépések elvégzése után a 7. ábrán látható feldolgozott elektroenkefalogramot kapjuk az MM05 azonosítójú beszélőn.



6. ábra: Nyers elektroenkefalogram [MM05 beszélő].

**Neurális vokóder** A WaveNet 2016-os bevezetése óta, melyet van den Oord és mtsai (2016) mutattak be, a neurális vokóderek egy izgalmas új módja a beszéd nyers mintáinak generálására a szöveg-beszéd szintézis (Text-To-Speech, TTS) során. A korai WaveNet-szerű modellek egyik hátránya, hogy azok rendkívül számításgényesek voltak. Jelenleg a legmodernebb TTS modellek parametrikus neurális hálózatokon alapulnak, amelyek WaveNet-szerű neurális vokóderek továbbfejlesztett változatait használják. A TTS-szintézis jellemzően két lépésben történik: az első lépésben a szövegben lévő karakterekhez rendelünk hangidőtartamokat majd átalakítjuk a beszédet leíró jellemzőkké, például mel-spektrogrammá; a második lépés ezeket a spektrális jellemzőket alakítja át a beszéddé. Azaz a második lépés során, a becsült spektrális paramétereket felhasználva beszédet generálhatunk.

A neurális vokóderek egyik legújabb típusa, a WaveGlow (Prenger és mtsai, 2018) egy áramlásalapú hálózat (flow-based network), amely képes mel-spektrogrammból jó minőségű beszédet generálni. A WaveGlow modell előnye,

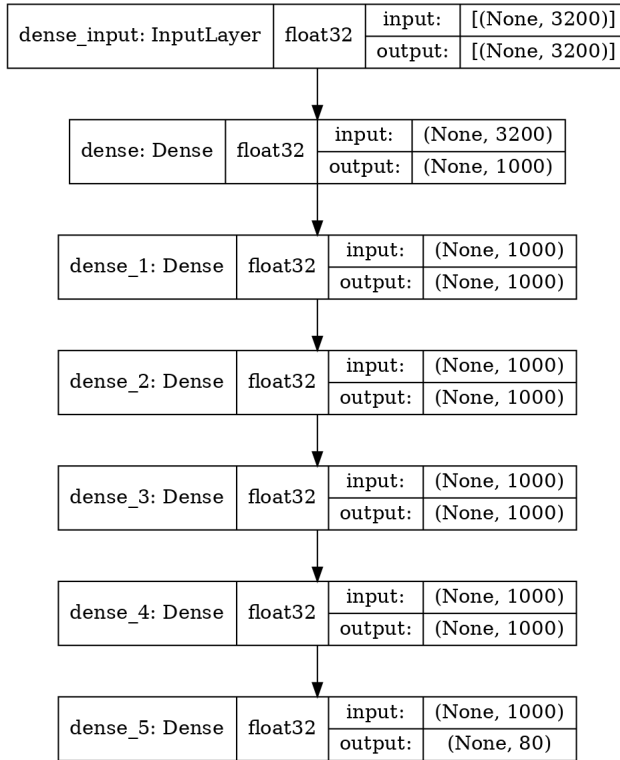


7. ábra: Feldolgozott elektroencefalogram [MM05 beszélő].

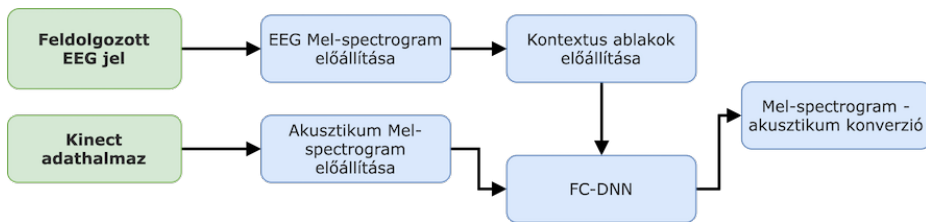
hogy viszonylag egyszerű, a szintézis valós időben elvégezhető, mégis jó természetességet eredményez. Képes beszédet generálni mintavételezéssel egy mel-spektrummal kondicionált eloszlásból.

## 2.4. Neurális hálózat

Teljesen kapcsolt mély neurális hálózati architektúrát (Fully Connected Deep Neural Network, FC-DNN) használtunk, mely 5 rejtett rétegből áll, amelyek mindegyike 1000 neuront tartalmaz. ReLU aktivációs függvényt alkalmaztunk. A bemeneti réteg 3200 neuronból áll (a bemeneti dimenzió megegyezik az EEG jelből képzett mel-spektrum, valamint a visszatekintési ablak tizedének a szorzatával). Visszatekintési ablakot azért alkalmaztunk, hogy a hálózat tudja kezelni az EEG és beszédjel közötti időben távolabbi összefüggéseket is. Az EEG jel feldolgozása során 400 ms hosszú ablakot alkalmaztunk. A beszéd mintavételezése 22 kHz-en történt, amiből 80-dimenziós mel-spektrumot számítottunk. A kimeneti réteg 80, a mel-spektrális együtthatók darabszámával megegyező neuront tartalmaz. Az FC-DNN hálózat felépítését a 8. ábra mutatja.



8. ábra: FC-DNN hálózati architektúra.



9. ábra: A beszédszintézis folyamata.

## 2.5. A beszédszintézis folyamata

Az EEG jel előfeldolgozását követően előállítjuk annak mel-spektrumját melyet a 'librosa' könyvtár segítségével teszünk meg. Ezzel párhuzamosan feldolgozzuk a 132 Microsoft Kinect rendszer által rögzített audio állományt. Az EEG



hullámhoz hasonlóan ennek is előállítjuk a mel-spektrogramját. Mielőtt a FC-DNN hálózat tanítását megkezdénénk az EEG jelet ablakozzuk, vagyis egy 400 ms hosszú hullámszakasszal bővítjük az előzőleg már szegmentált jelet. Erre azért van szükség, mert megfigyeléseink szerint az agykéregben mért idegi aktivitás és a rögzített beszéd között késleltetés is megjelenhet. Ez következhet az észlelés és a cselekvés közötti effektusfüggő motoros adaptációjából. Ez az érték beszélőnként eltérhet, de a kutatás ezen szakaszában ezzel az általános értékkel számoltunk. A tanítóhalmaz 72%-át teszi ki a teljes adathalmaznak, a validációs halmaz 8%-át, míg a teszt halmaz 20%-át. Ez beszélőnként 95, 11 illetve 26 regisztrátumot jelent. Következő lépésben betanítjuk a neurális hálózatot, mely bemenetként ablakozott EEG mel-spektrogram fragmentumokat kap, kimenetként pedig 80 mel-spektrális együtthatót kapunk, melyből a WaveGlow modellt használva inverz STFT transzformációval megkapjuk a becsült beszédet.

### 3. Eredmények

**Objektív mérések** Annak ellenőrzése céljából, hogy a tervezett modell reprodukálni tudja-e az eredeti beszéd jellemzőit, kiértékeljük a spektrális torzulásokat a természetes beszéd és a szintetizált beszéd között. Erre a "Mel-Cepstral Distortion" (MCD) mérőszámot alkalmaztuk:

$$MCD = \frac{10\sqrt{2}}{\ln 10} \frac{1}{T} \sum_{t=1}^T \sqrt{\sum_i (C_{ti} - \hat{C}_{ti})^2}, \quad (1)$$

ahol  $C_{ti}$  és  $\hat{C}_{ti}$  a mel-kepsztrum az eredeti valamint a szintetizált beszéd esetén.

A 2. táblázat a validációs adatokon számolt MCD értékeket valamint az eredeti és becsült mel-spektrális együtthatókon mért átlagos négyzetes hibát (MSE) jeleníti meg beszélőnként. A beszélők azonosítására a Kara One adatbázisban is használt azonosítók kerültek alkalmazásra. Látható, hogy az eredmények erősen függenek a beszélőtől – pl. MM12 férfi beszélővel és MM19 női beszélővel lényegesen jobb MCD eredményeket sikerült elérnünk, mint a többiekkel.

Szubjektív meghallgatásos teszt elvégzését ebben a korai szakaszban még nem eszközöltünk. A teszthalmazon becsült mel-spektrális paramétereiből visszaalakított beszéd ugyan emlékeztet az eredeti akusztikumra, de zajos és nehezen érthető.

Az MCD értékek abban a tartományban vannak (3–5 dB), ahol a beszédszintetizátor rendszerek általában már jól érthető beszédet generálnak, de meghallgatva a mintákat itt ez nem teljesül. Valószínűleg a felvételekben lévő sok szünet szakasz eredményezte a viszonylag alacsony hiba értékeket, ami viszont itt nem korrelál a szubjektív eredménnyel. A jelen kutatásban nem vizsgáltuk, hogy a szintetizálni kívánt beszédhangok sorozatától mennyire függött az MCD értéke.

Beszélő	MCD ( <i>dB</i> )	MSE
MM05	5,54612	1,25103
MM08	5,73413	1,27839
MM09	5,69279	1,07313
MM10	7,49676	1,23934
MM12	3,83136	1,00725
MM14	4,22095	1,29970
MM18	4,61194	1,29009
MM21	3,75126	1,26096
<b>Átlag (férfi beszélők)</b>	<b>5,11066</b>	<b>1,21248</b>
MM11	4,31737	1,59456
MM15	5,09933	1,20212
MM16	4,63597	1,36154
MM19	3,80825	1,53233
MM20	4,62845	1,48949
<b>Átlag (női beszélők)</b>	<b>4,49787</b>	<b>1,43600</b>
<b>Átlag</b>	<b>4,87497</b>	<b>1,29845</b>

2. táblázat. A különböző beszélőkkel tanított neuronhálókkal elért átlagos hibák.

#### 4. Összefoglalás

Ebben a cikkben részletesen bemutatjuk az agyi jel előfeldolgozásának lépéseit (szűrés, csatorna választás, zajok és műtermékek eltávolítása, független komponens analízis, és szegmentálás). Egy kezdetleges FC-DNN hálózat tanítását és predikcióját végeztük el az agyi jelből beszédbe szintetizálás feladat során. A kezdeti értékelésünk azt mutatja, hogy az eljárás megvalósítható, ugyanakkor az eredmény még nem kielégítő.

Az eljárást a továbbiakban a neurális hálózati architektúra finomításával szándékozzuk javítani. Megfigyelhető továbbá, hogy a tanító halmaz méretét növelve javítható az eredmény még a jelenlegi architektúra mellett is, így a jövőben szándékozzuk saját adatbázist rögzíteni, mely fonémák számában, mind pedig elemszámban bővebb.

#### Köszönetnyilvánítás

A kutatást részben a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal OTKA programja támogatta (FK 124584 és PD 127915 projektek). Csapó Tamás Gábor kutatásait az MTA Bolyai János kutatói ösztöndíja, valamint az Új Nemzeti Kiválóság Program Bolyai+ (ÚNKP-21-5-BME-2060) pályázata támogatta.

## Hivatkozások

- Angrick, M., Ottenhoff, M.C., Diener, L., Ivucic, D., Ivucic, G., Goulis, S., Saal, J., Colon, A.J., Wagner, L., Krusienski, D.J., Kubben, P.L., Schultz, T., Herff, C.: Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity. *Communications Biology* 4(1), 1055 (2021), <https://doi.org/10.1038/s42003-021-02578-0>
- Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: *Noise reduction in speech processing*, pp. 37–40. Springer (2009)
- Dupré, D., Karjalainen, A.: Employment of disabled people in europe in 2002. *Statistics in focus* pp. 3–26 (2003)
- Eurostat: Eurostat - data explorer - population by type of disability, sex, age and labour status (2011), [https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=hlth\\_dlm040&lang=en](https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=hlth_dlm040&lang=en)
- Gabard-Durnam, L.J., Mendez Leal, A.S., Wilkinson, C.L., Levin, A.R.: The Harvard Automated Processing Pipeline for Electroencephalography (HAPPE): Standardized processing software for developmental and high-artifact data. *Frontiers in Neuroscience* 12, 97 (2018), <https://www.frontiersin.org/article/10.3389/fnins.2018.00097>
- Herff, C., Heger, D., de Pestors, A., Telaar, D., Brunner, P., Schalk, G., Schultz, T.: Brain-to-text: decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience* 9, 217 (2015), <https://www.frontiersin.org/article/10.3389/fnins.2015.00217>
- Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. *Neural Networks* 13(4), 411–430 (2000), <https://www.sciencedirect.com/science/article/pii/S0893608000000265>
- Kent, R.D., Weismer, G., Kent, J.F., Rosenbek, J.C.: Toward phonetic intelligibility testing in dysarthria. *Journal of Speech and Hearing Disorders* 54(4), 482–499 (1989), <https://pubs.asha.org/doi/abs/10.1044/jshd.5404.482>
- Mognon, A., Jovicich, J., Bruzzone, L., Buiatti, M.: ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology* 48(2), 229–240 (2011), <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.2010.01061.x>
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A generative model for raw audio. *CoRR* abs/1609.03499 (2016), <http://arxiv.org/abs/1609.03499>
- Prenger, R., Valle, R., Catanzaro, B.: WaveGlow: A flow-based generative network for speech synthesis. *CoRR* abs/1811.00002 (2018), <http://arxiv.org/abs/1811.00002>
- Rácz, F.S.: A nyugalmi agyi konnektivitás multifraktális dinamikája (2019), <http://repo.lib.semmelweis.hu/handle/123456789/8389>
- Sharbrough, F., Chatrian, G., Lesser, R., Luders, H., Nuwer, M., Picton, T.: American electroencephalographic society guidelines for standard electrode position nomenclature. *Clinical Neurophysiology* 8, 200–202 (01 1991)

- Winkler, I., Brandl, S., Horn, F., Waldburger, E., Allefeld, C., Tangermann, M.:  
Robust artifactual independent component classification for BCI practitioners.  
*Journal of Neural Engineering* 11(3), 035013 (2014)
- Zhao, S., Rudzicz, F.: Classifying phonological categories in imagined and articulated speech. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 992–996 (2015)

## „Releasing the BEAST” – A BEA gépi beszédleiratozási feladat, megközelítések és eredmények

Mihajlik Péter<sup>1,2</sup>, Balog András<sup>1,3</sup>, Grácz Tekla Etelka<sup>1</sup>,  
Kohári Anna<sup>1</sup>, Fegyó Tibor<sup>2,3</sup>, Mády Katalin<sup>1</sup>

<sup>1</sup> Nyelvtudományi Kutatóközpont,  
Budapest, H-1394, VI. Benczúr u. 33.  
{mihajlik.peter, balog.andras, graczi.tekla.etelka,  
kohari.anna, mady}@nytud.hu

<sup>2</sup> Budapesti Műszaki és Gazdaságtudományi Egyetem,  
Vill. mérnöki és Inf. Kar, Távközlési és Médiainformaticai Tanszék,  
H-1117, Budapest, Magyar tudósok krt. 2.

<sup>3</sup> SpeechTex Kft., 1181 Budapest, Madách Imre utca 47.  
tfegyo@speechtex.com

**Kivonat:** Bemutatjuk a BEA-Base adatbázist, mely az általános BEA (Beszélt nyelvi Adatbázis) 140 beszélős részhalmazának egyszerűsített és gépi beszédfelismerési kísérletekre előkészített kivonata. Definiáljuk a beszédfelismerési feladatot, melyben csak a BEA-Base adatbázis használható felügyelt adatként. Ismertetjük az alkalmazott, legkorszerűbb beszédfelismerési technikákra épülő beszédfelismerés megközelítéseket, úgymint hibrid rejtett Markov-modell/mély neuronhálós fonéma és graféma alapú rendszerek, konvolúciós és rekurrens hálókra épülő enkóder-dekóder end-to-end rendszerek figyelmi mechanizmussal vagy anélkül. Végül eljutunk a kontrasztív, önfelügyelt módon előtanított súlyokkal inicializált transzformer alapú megközelítésig, amit BEAST-nek (BEA Speech Transcriber) neveztünk el. Utóbbi rendszer – köszönhetően az 53 ezer órás hanganyagon történt előtanításnak – a klasszikus megoldáshoz képest 40%-kal csökkenti a legrelevánsabb szó- és betűhibaarányt, valamint a magyar Common Voice tesztadatokon is versenyképes leiratot szolgáltat.

**Kulcsszavak:** mesterséges intelligencia, mélytanulás, gépi beszédfelismerés, spontán beszédatadabázisok, end-to-end beszédfelismerés, önfelügyelt tanulás.

### 1. Bevezetés

A magyar nyelvű beszédfelismerési kutatások régóta fájó pontja, hogy nem áll rendelkezésre olyan *könnyen hozzáférhető* – és legalább kutatási célra ingyenes – beszédatadabázis, amely kifejezetten beszédfelismerő modellek/motorok betanítására és kiértékelésére alkalmas. Így az egyes műhelyek – részben kényelmi, részben anyagi okokból – sokszor csak a saját adataikon dolgoznak, melyet másokkal jellemzően nem osztanak (vagy oszthatnak) meg. A közölt beszédfelismerési eredmények tehát tipikusan önmagukban állnak, a kísérletek más csoportok által nem könnyen ismételtetők meg, az elért eredmények egymással nem vehetők össze. A cikk írásának pillanatában ezen az

áldatlan helyzetben a sokak által felkarolt és üdvözölt Mozilla Common Voice projekt<sup>1</sup> sem segített még, mivel alig 19 óra ellenőrzött leiratozott hanganyag állt elő. Noha várjuk és reméljük ezen adatbázis bővülését, a majdan előálló adatok csak olvasott beszédet fognak tartalmazni. Azonban az olvasott beszéd visszaalakítása szöveggé kissé mondvacsinált problémának tűnik, hiszen az emberi kommunikáció – sőt az ember-gép interakció is – alapvetően spontán jellegű. Ha a normál társalgási (vagy előadási, esetleg számítógép-vezérlési) beszédben időnként előre meg is tervezzük, mit mondunk, tipikusan nem olvassuk fel a mondandónkat. Vagyis, az olvasott beszédre épülő korpuszok ugyan relatíve olcsón előállíthatók, illeszkedésük a természetes beszélt nyelvhez vitatható.

Munkánk érdemi változásokat kíván hozni az előbb vázolt helyzetbe. Egyrészt közreadjuk a BEA adatbázis (Gósy és mtsai, 2012) nyilvánosan elérhető, beszédfelismerésre előkészített, az akadémiai szféra számára ingyenes, 140 beszélő hanganyagát tartalmazó 'Base' részhalmozát, melyre „BEA-Base”-ként, ill. rövidítve „BEA-B” néven hivatkozunk és amit e tanulmányban röviden ismertetünk<sup>2</sup>. Az adatbázis túlnyomórészt spontán, illetve félszponán beszédből áll. A kiértékeléshez a spontán és olvasott/ismételt beszédet tartalmazó részeket külön bontjuk mind a fejlesztési (validáló) és a kiértékelési (teszt-) halmazok esetén. Így a jövőben adott tanítóhalmaz és teszt-halmazok alapján egzakt összehasonlítások tehetők beszédfelismerési megközelítések között.

A továbbiakban megmutatjuk, hogy nyílt forráskódú eszközökkel milyen beszédfelismerési pontosságok érhetők el a BEA-B adatbázison, összehasonlítjuk a hibrid HMM-ANN és az „end-to-end” mélyneuronháló alapú technikákat, valamint a legkorszerűbb, önfelügyelt kontrasztív előtanulás után az adott feladatra hangolt Transformer alapú rendszert. Utóbbi neurális modelljét „BEAST” (BEA Speech Transcriber) néven akadémiai kutatási felhasználásra ingyenesen elérhetővé is tesszük.

## 2. Kapcsolódó munkák

### 2.1. Magyar nyelvű leiratozott beszédatadtbázisok

Magyar nyelven számos fizetős, olvasott beszédet tartalmazó adatbázis érhető el, pl. (Vicsi és Víg, 1998), (Siemund és mtsai, 2000), azonban ezek megfelelősége spontán beszéd felismeréséhez kérdéses. A magyar MALACH adatbázis (Mihajlik és mtsai, 2007) spontánnak tekinthető (interjúk idős emberekkel), azonban sem a koreloszlás, sem a méret nem ideális és sajnos a leirat nem érhető el publikusan. A BEA-n kívül talán legjelentősebb magyar nyelvű társalgási jellegű beszédet (és videót) is tartalmazó adatbázis a HuComTech (Hunyadi és mtsai, 2012). Itt a használhatóságot a beszélők alacsonyabb száma és koruknak kisebb átfogása (döntően egyetemi hallgatók) befolyásolja negatívan. Az utóbbi években újabb spontán beszédet gyűjtő projektek indultak, pl. (Szabó és Galántai, 2017), (Szabó és mtsai, 2021), de utóbbi akusztikailag problémás (a jel-zaj viszony alacsony) előbbi pedig kevés beszélős és egyik sem elérhető még

<sup>1</sup> <https://commonvoice.mozilla.org/en/datasets>

<sup>2</sup> <https://phon.nytud.hu/bea>

a tanulmány megírásának időpontjában. Elmondhatjuk, hogy jelenleg a gyakorlatban nemigen érhető el beszédfelismerési kutatási célra ingyenesen hozzáférhető releváns méretű, korban, nemben reprezentatív, lejegyzett beszédadatbázis magyar nyelven, különösen spontán jellegű beszéd tartalommal.

## 2.2. Spontán magyar nyelvű beszéd felismerési eredmények

A viszonylag kis számú releváns publikációk között említhetők a magyar nyelvű spontán nyelvű beszédet is tartalmazó adatbázisokon elért, MALACH (Mihajlik és mtsai, 2010), HuComTech (Szaszák és mtsai, 2011), BEA (Beke és Szaszák, 2016), eredmények. Látható, hogy a spontán nyelvű adatokon a szófelismerési hibák meglehetősen magasak: 42% – 55%, és egyetlen olyan publikációt sem ismerünk, ahol az adott adatbázison más csoport is közölt volna összehasonlítható beszéd felismerési eredményeket.

## 3. A BEA-Base adatbázis

A BEA adatbázis (Gósy és mtsai, 2012) eredetileg nyelvészeti – főként fonetikai – kutatások céljára lett létrehozva. Fontos szempont volt, hogy nagy számú beszélőtől jó akusztikai körülmények között, tervezett módon gyűjtsenek adatot. Az adatrögzítés több éven keresztül történt, eközben a felvételek készítésnek körülményei, módszerei nem, azonban a lejegyzési módszerek lényegesen, meghatározott etapok szerint változtak (Gyarmathy–Neuberger 2011, Gyarmathy és mtsai, 2014). Az első etapban a lejegyzés Word dokumentumokba történt, megadott szabályrendszer szerint. A másodikban – részben a felmerült beszédtechnológiai felhasználás miatt – a lejegyzés a Transcriber<sup>3</sup> nevű programmal, megújított szabályrendszer szerint készült. A következő körben áttértek a Praat<sup>4</sup> használatára, ami maga után vonta a TextGrid formátumú, többszintű annotáció alkalmazását és következésképpen a szabályok adaptálását. Végül, bizonyos felvételekhez egyáltalán nem készült megfelelő leirat, így ezek számítógéppel segített utólagos lejegyzése a rövid távú tervek között szerepel.

	train-114	dev-repet	dev-spont	eval-repet	eval-spont
hossz [óra]	71.2	0.65	4.02	0.95	4.91
szegmens [db]	76 881	568	4 893	858	5 693
szó [db]	555 322	4 110	27 939	6229	35 178
OOV [%]	–	2	7.3	1.4	7.9
3-gram ppl	–	43.7	283	44.6	317.4

1. táblázat. A BEA-Base adatbázis főbb statisztikái.

<sup>3</sup> <http://trans.sourceforge.net/>

<sup>4</sup> <https://www.fon.hum.uva.nl/praat/>

Szerkezetét tekintve a BEA adatbázis jól tagolható az egyes adatközlők (beszélők) egységeire, ezt a könyvtárstruktúra is tükrözi. Jelenleg mintegy 470 adatközlőtől áll rendelkezésre felvétel. Ezek minden esetben hasonlóan zajlottak: jellemzően 8 felvételi egység rögzítése történt meg meghatározott sorrendben, bővebben lásd: (Gósy és mtsai, 2012). A felvételi egységek a felvett beszéd jellegét tekintve két csoportba oszthatók:

- kötött beszéd, aminek egyik felét repetitívnek fogjuk hívni, mert azonos mondatokat olvas ill. ismét el a főbeszélő (utóbbi esetén a kísérletvezető által felolvasott mondatot ismétli az adatközlő). A másik modulban egybefüggő szöveget olvas fel a főbeszélő. A rögzített adatok relatíve kis része tartozik ebbe a csoportba.
- spontán, 2–3 résztvevős társalgási beszéd és félspontán felvételek, melyek egy elolvasott szöveg összefoglalásán alapszanak. A BEA nagyját tehát a spontán és félspontán beszéd adja, ezekre a továbbiakban egyszerűsítve csak spontán felvételekként fogunk hivatkozni.

A BEA adatbázis készítése során kiemelt figyelmet fordítottak az elhangzottak pontos lejegyzésére, tehát a beszélő nyelvi és egyéb hibái nem lettek javítva, hanem lehetőség szerint minden, ami elhangzott – a nem érthető beszéd kivételével – le lett írva. A leirat azonban nem fonológiai szinten készült, hanem ortografikusan, a számok, kivételes írásmódú szavak és hasonló entitások tipikusan „verbalizáltan” azaz a kiejtésnek megfelelően lettek rögzítve, pl. „tévékettő”, „azér”, „íméleztünk” vagy „merhogy”.

A BEA tehát egy ambiciózus projekt, ami számos speciális beszédjelenséget is annotálni kívánt amellett, hogy a pontos szövegleíratot ortografikus formában rögzíti, mindezt nagy számú beszélő hanganyagán (és véges költségvetéssel). Ezen követelmények azonban némileg ellentmondanak egymásnak, így a konkrét lejegyzők ízlésének, illetve a különböző lejegyzési technikáknak megfelelően összességében egy heterogén annotációs anyag állt elő, mely nyelvészeti kutatásokra sem feltétlenül ideális, de gépi feldolgozásra még kevésbé optimális. Mindezek miatt jelenleg az annotációk közös nevezőre hozása, egységesítése zajlik, ami a várakozásoknál is nehezebb feladatnak bizonyult. Ugyanakkor, felismerve a gépi beszédfelismerés korábban vázolt hiányát, a legalkalmasabb, homogén adatbázis-részhalmoz kiadása mellett született döntés.

A **BEA-Base** nem más, mint a BEA Transcriberrel lejegyzett 140 beszélős részhalmozának egyszerűsített (sima txt formátumú) v.0.1-es annotációjú verziója. 10 beszélői adatot validálásra, illetve 16 beszélői adatot kiértékelésre szántunk, így a tanítóhalmazba 114 adatközlő felvételei kerültek (1. táblázat). A nemek eloszlása mindegyik teszhalmazban 50–50%-os, a tanítóhalmazban pedig 40%–60% (férfi-nő). A koreloszlást nézve mindegyik halmaz széles merítésű, többlettel a huszonévesek tekintetében.

Mind a validációs, mind a teszhalmazt felbontottuk spontán és kötött halmazokra. A kihívást az jelentette, hogy a kötött beszéd nagyját adó repetitív részben ugyanazokat a mondatokat olvassa a főbeszélő és a kísérletvezető, illetve még el is ismétli őket az adatközlő. Emiatt, ha repetitív rész bármelyik részhalmozát benn hagyjuk a tanítóadatban, a beszédfelismerési eredmények irreális képet adtak volna a beszédfelismerő modellek pontosságáról. Ennek kivédésére teljesen kihagytuk a tanítóhalmazból a repetíciós részeket (mind az adatközlő, mind a kísérletvezető felvételeit), míg a tesztelő anyagokból az egybefüggő olvasott szöveget. Ily módon a kötött beszéddel történő kiértékelés és validálás korrekt, hiszen az adott tesztszöveget nem „látta” korábban a beszédfelismerő modell. Szintén a kiegyensúlyozott kiértékelés és a beszélőfüggetlenség érdekében a tesztelő halmazokban a kísérletvezető (ill. 3. beszélő) hangját tartalmazó



szegmentumokat kivágtuk. Végeredményben tehát a kötött beszédet tartalmazó teszt-halmazokban csak az adatközlő felolvasott és visszaismételt mondatai maradtak (ezért szerepel a halmazok nevében a „repet”, mint repetitív tag), a spontán jelzővel illetett halmazokban pedig szintén csak az adatközlő hangját hagytuk meg a különböző spontán és félszponán szituációkból.

A BEA-Base adatbázis statisztikái az 1. táblázatban találhatóak. A beszédfelismerési feladat (írott) nyelvi nehézségét kifejező perplexitási mutatókat (ppl) módosított Kneser–Ney simítással (Chen és Goodman, 1999) számoltuk szó alapon. A tanítóhalmaz szavaiból képzett szótárral le nem fedhető tesztszavak arányát is megadtuk (OOV: Out Of Vocabulary rate).

A következőkben bemutatandó gépi beszédfelismerési kísérletek mindegyikében a BEA-Base adatbázis train-114 halmazát használtuk felügyelt (leiratozott) tanítóadatként, a hiperparamétereket a dev-spont validációs halmazra optimalizáltuk és az eval-spont halmazon értékeltük ki. Mérete és jellege miatt az utóbbi eredményeit tekintjük relevánsnak, a többi (validációs és nem spontán) halmazon mérteket inkább érdekességképpen közöljük.

#### 4. Hibrid HMM/DNN megközelítés

A legtöbb, gyakorlatban használt beszédleiratozó rendszer ma HMM, azaz Hidden Markov-Modell alapú. Evidens, hogy az akusztikus modellezésre mély neuronháló (Deep Neural Network) érdemes alkalmazni HMM alapú rendszerek esetén is, így vizsgálatainkat mi is erre korlátoztuk. Szintén közismert, hogy a legelterjedtebb HMM/DNN beszédfelismerési keretrendszer a Kaldi (Povey és mtsai, 2011), így ezzel a nyílt forrású eszközkészlettel kezdtük a kísérleteinket.

A hagyományos megközelítésben az akusztikai modellek **fonéma** alapúak, ami azt jelenti, hogy a szavakat a kiejtésüknek megfelelő fonémasorozatokra képezzük le, több alternatív kiejtés esetén pedig az alternatívákhoz valószínűségeket rendelünk a tanítóadatbázisbeli kiejtési variánsok alapján (Povey és mtsai, 2011). Ugyanakkor (Mihajlik és mtsai, 2007) már a korszerű end-to-end rendszerek megjelenése előtt megmutatta, hogy magyarra a graféma, azaz írott (alfabetikus) **karakter** alapú akusztikus modellezésnek is van létjogosultsága, így ezt a megközelítést is vizsgáltuk.

Az utóbbi években az egyik legnépszerűbb neurális akusztikus modell a Kaldiban az ún. **TDNN-F** megközelítés, azaz faktorált Time Delay Neural Network (Povey és mtsai, 2018), valamint ennek egy továbbfejlesztett változata, a **CNN-TDNN-F**, ahol konvolúciós neurális blokkok előzik meg a TDNN-F struktúrát (Zorila és mtsai, 2019). Kísérleteinkben mindkét neurális akusztikus modellt megvizsgáltuk. A modellek tanítására a hasonló méretű WSJ (Wall Street Journal) adatbázis S5-ös receptúráját alkalmaztuk (magyarra adaptálva), de bemeneti vektorokként csak nagy felbontású MFCC-t (Mel-Frequency Cepstral Coefficients) használtunk, i-vektorokat nem. Környezetfüggetlenség, bal-difón akusztikus modelleket tanítottunk be, az audio adatok augmentálására a felvételek sebességét  $\pm 10\%$ -kal perturbáltuk. Szó 3-gram Kneser–Ney simítású nyelvmodellt alkalmaztunk, melyet a BEA-Base train-114 szöveges leirataink tanítottunk be.

A beszédfelismerési eredmények a 2. táblázatban láthatók, a szó- és betűhibaarányt egyaránt feltüntettük (WER: Word Error Rate, CER: Character Error Rate).

Struktúra / param. szám	Akusztikai egység	dev-repet	dev-spont	eval-repet	eval-spont
TDNN-F / 18M	fonéma karakter	7.06 / 1.82 7.30 / <b>1.70</b>	27.12 / 9.04 27.08 / 8.90	6.26 / 1.58 <b>6.08 / 1.51</b>	28.41 / 9.36 28.28 / 9.26
CNN-TDNN-F / 16M	fonéma karakter	7.10 / 1.81 <b>6.98 / 1.77</b>	27.01 / 8.95 <b>26.71 / 8.59</b>	6.33 / 1.62 6.28 / 1.59	28.81 / 9.52 <b>28.15 / 9.13</b>

**2. táblázat.** Kaldi alapú WER[%] / CER[%] eredmények a BEA-Base adatbázison.

Látható, hogy a spontán beszéd felismerése adott körülmények között lényegesen nagyobb kihívás, mint az elismételt beszédé, ugyanakkor az egyes modellezési megközelítések közötti különbségek marginálisak. A legjobb eredményeket vastag betűkkel emeltük ki. Összességében a CNN-TDNN-F technika szerepelt a legjobban – karakter akusztikus modellekkel társítva –, tehát adott körülmények között a karakter alapú akusztikus modellezésnek nemcsak van jogosultsága, de optimális is lehet.

## 5. End-to-end mély neuronháló alapú megközelítés

Az end-to-end, azaz teljesen mélyneuronháló alapú beszédfelismerés csak néhány éve tört be a tudományos köztudatba (Graves és Jaitly, 2014), azonban mára a leginkább kutatott irányzattá vált. Számos eszközkészlet áll rendelkezésre, pl. (Hannun és mtsai, 2014), (Watanabe és mtsai, 2018), (Kriman és mtsai, 2020), (Ravanelli és mtsai, 2021), ezek teljes körű áttekintésére és a BEA-Base adatbázison való kipróbálására és optimalizálására nem vállalkozhattunk, ehelyett az utolsó két eszközkészletet választottunk ki és ezekkel végeztünk célzott vizsgálatokat.

### 5.1. Kísérletek a QuartzNet struktúrával karakter alapon

Az előzőekben láthattuk, hogy karakter akusztikus modellek alkalmazása fonéma modellek helyett nem rontotta a beszédfelismerési eredményeket, így természetesnek éreztük karakter alapú end-to-end megközelítéssel folytatni a kísérletezést. Hasonlóan a TDNN struktúrához – amit az 1D konvolúciós neuronhálók egy fajtájának tekintenek – a szintén konvolúciós blokkokra épülő QuartzNet end-to-end modellekkel (Kriman és mtsai, 2020) folytatjuk, melyek tanítása a CTC (Connectionist Temporal Classification) költségfüggvényre (Graves és mtsai, 2006) épül. A felépítés pontos részletei megtalálhatók a (Kriman és mtsai, 2020)-ban, itt csak azt emeljük ki, hogy a struktúra BxR soros elrendezésű blokkokból épül fel, ahol R számú azonos – idő és csatorna szerint szeparált – konvolúciós modult egy reziduális kapcsolat köt át, és ez a blokk B-szer ismétlődik, blokkonként potenciálisan különböző hiperparaméterekkel.

A kísérletek tervezésénél kiindulópontnak a (Kriman és mtsai, 2020)-et tekintettük, alapesetként az itt közölt hiperparamétereket és metodikát vettük át. A WSJ feladatra

(80 órányi olvasott beszéd) az 5x3-ös struktúrát javasolják 1200 epoch-nyi tanítással, beszédsebesség perturbálással és SpecAugmenttel (Park és mtsai, 2019). A tanítás még RTX A6000-es grafikus kártyával is napokat vett igénybe, azonban a teljesítmény több körös tanulási ráta optimalizálás után is elmaradt a várakozásoktól, így a LibriSpeech recept szerinti 15xR-es struktúra (R= 1, 2, 3, 5) alkalmazásával folytattuk szintén (Kri-man és mtsai, 2020) szerint, megtartva az egyéb kísérleti körülményeket. A dekódolás során 80-as beam-szélességet alkalmaztunk és a kimeneti hipotéziseket a már korábban is használt szó alapú n-gram nyelvmodellel súlyoztuk újra. Az eredményeket a 3. táblázatban foglaltuk össze.

Struktúra / param. szám	LM	dev-repet	dev-spont	eval-repet	eval-spont
5x3 / 6.4M	–	23.82 / 5.38	36.16 / 13.24	27.85 / 6.56	38.30 / 13.64
	3-gram	8.91 / 2.93	31.63 / 14.11	10.05 / 3.71	32.91 / 14.77
15x1 / 6.5M	–	13.72 / 2.99	28.33 / 9.36	17.00 / 3.87	29.59 / 9.70
	3-gram	6.76 / 2.01	26.66 / 10.45	6.90 / 2.43	27.97 / 11.29
15x2 / 9.6M	–	10.05 / 2.42	26.02 / 8.58	12.49 / 3.12	27.22 / 9.06
	3-gram	6.57 / 1.95	25.52 / 10.22	6.71 / 2.42	27.09 / 10.93
15x3 / 12.7M	–	9.73 / 2.20	<b>25.20 / 8.33</b>	11.56 / 2.91	<b>26.70 / 8.84</b>
	3-gram	<b>6.50</b> / 1.86	25.50 / 10.0	6.86 / 2.36	26.83 / 10.76
15x5 / 18.9M	–	12.70 / 2.89	26.43 / 8.42	13.79 / 3.32	27.63 / 8.96
	3-gram	7.30 / 2.20	25.58 / 10.25	6.90 / 2.38	26.98 / 10.71

**3. táblázat.** QuartzNet WER[%] / CER[%] eredmények a BEA-Base adatbázison.

Látható, hogy itt már jelentős különbségek adódnak az egyes struktúrák eredményei között. Elmondhatjuk, hogy a BEA-Base feladaton a WSJ-re optimalizált 5x3-as struktúra nem bizonyult optimálisnak, a hasonló paraméterszámú 15x1 elrendezés mindenben szignifikánsan jobban teljesített, míg a legjobb eredmények a 15x3-as struktúrával álltak elő. A reményeknek megfelelően a mélyebb struktúrák általában jobb eredményt adtak, többnyire meghaladva a Kaldi alapú HMM/DNN teljesítményét (vastagon szedjük ezeket a számokat).

Néhány nem várt jelenség is adódott. A leginkább szembeötlő, hogy a 15x5-ös struktúra esetén a pontosságok valamelyest visszaestek. Ennek az okát még keressük, a legvalószínűbbnek azt gondoljuk, hogy ez a hálózat a legérzékenyebb a megfelelő hiperparaméterek megfelelő beállítására, ugyanekkor ez a leglassabb és legmemóriaigényesebb megoldás is, mindamelllett, hogy a konvergenciája is ennek volt a legbizonytalanabb. Így lehetséges, hogy a jövőben ezen még tudunk javítani, azonban a 15x3-as struktúrához képest jelentős javulásra nem számítunk. Szintén némileg meglepő volt, hogy a nyelvmodell alkalmazása annál kevesebbet javít, minél pontosabb az (end-to-end) akusztikus modell; mi több, a legrelevánsabb spontán kiértékelő halmazon még rontott is. Ezt annak tulajdonítjuk, hogy a spontán beszédre jellemző rövid szavakat és kapcsolatokat a CTC alapú tanítás során impliciten megtanulja az akusztikus modell, és az ugyanezen hanganyag szövegleiratából tanult explicit nyelvmodell egyre kevesebbet tud hozzáadni az akusztikus neuronháló tudásához, független tesztadaton pedig még ronthatja is az eredményeket.

## 5.2. Kísérletek CRDNN modellel BPE akusztikus token alapon

A CTC letisztult, megbízható technika, azonban hátránya, hogy az egyes kimeneti döntéseket feltételesen függetlennek tekinti. Ezt kiküszöbölendő vezették be az „Attention”, azaz „figyelmi” mechanizmus alapú kapcsolatot az enkóder és dekóder neuronháló blokkok közé (Chan és mtsai, 2016). Még jobb eredményeket értek el a két módszer kombinálásával, vegyes költségfüggvény alkalmazásával (Watanabe és mtsai, 2017). Ezért olyan eszközkészletet kerestünk, amely implementálja ezt az üzemmódot. A (Ravanelli és mtsai, 2021)-re esett választásunk, mely egyidejűleg támogatja a karakternél nagyobb akusztikai egységek használatát.

Elsőként a viszonylag egyszerű CRDNN (Convolutional, Recurrent and Deep Neural Nets) struktúrájú enkódert GRU (Gated Recurrent Unit, Chung és mtsai, 2014) dekóderrel párosítottuk a megfelelő embedding és attention blokkok segítségével. 600 szó-tárelemű BPE tokenizálót (Sennrich és mtsai, 2015) tanítottunk be a tanítóhalmaz leírátán. CTC+Attention költségfüggvény alkalmazásával 60 epochon keresztül tanítottuk be a neuronhálót. A további hiperparaméterek a (Ravanelli és mtsai, 2021) Common Voice receptjéből származnak, felsorolásuktól eltekintünk. Nyelvmodellt nem alkalmaztunk. Tesztelésnél a korábbiakhoz hasonlóan 80-as beamszélességet használtunk. A kapott eredmények a 4. táblázatban találhatók.

Paraméterszám	LM	dev-repet	dev-spont	eval-repet	eval-spont
132.8M	–	9.80 / 3.65	<b>24.68</b> / 9.01	6.61 / 1.92	<b>25.21</b> / 9.46

**4. táblázat.** CRDNN+GRU+CTC+Attention+BPE\_600 WER[%] / CER[%] eredmények a BEA-Base adatbázison.

Láthatjuk, hogy a spontán halmazok szóhibaarányait tekintve sikerült megjavítani az eddigi legjobb eredményeket (vastagon szedve). Mindezt úgy, hogy explicit nyelvmodellt egyáltalán nem alkalmazunk – természetesen az autoregresszív dekóder réteg tekinthető implicit BPE alapú nyelvmodellnek. A javulás ára részben az előzőeknél jóval nagyobb paraméterszám és lassabb tanulás, igaz, epochból jóval kevesebbre volt szükség. A tanításhoz 4 db NVIDIA RTX A6000-es grafikus gyorsítót használtunk.

## 6. Szövegleirat nélkül előtanított transzformer alapú megközelítés, a BEAST

Jelen tanulmányban azt keressük, hogy felügyelt adatbázisként csak a BEA-Base tanítóhalmazát (train-114) használva milyen beszédfelismerési pontosság érhető el a megfelelő tesztalalmazokon. Idáig a klasszikus, felügyelt gépi tanulási megközelítést alkalmaztuk, azonban a felügyelet nélküli (avagy önfelügyelt) kontrasztív előtanulás az utóbbi időkben a beszédfelismerésben is rendkívül népszerűvé vált – elsősorban a wav2vec2.0-nek (Baevski és mtsai, 2020) köszönhetően – így mi is ebben az irányban folytattuk kísérleteinket.

Technikailag a beszédfelismerő rendszer felépítése mindenben megfelel az előzőeknek, azaz az 5.2-ben bemutatottaknak, „csupán” az enkódert cseréltük ki egy wav2vec2.0 típusú transzformer modulra. Ezt a modult nem is próbáltuk az ehhez túlzottan kicsi tanítóhalmazunkon betanítani, hanem a (Conneau és mtsai, 2021)-ben publikált, 53 ezer órányi többnyelvű beszédkorpuszon előtanított XLSR-53k nevű neurális modell súlyaival inicializáltuk. Nem célunk sem a neurális modellt, sem az előtanítást részletezni itt, azonban felhívjuk a figyelmet, hogy az XLSR-53k (elő)tanításához egyáltalán nem használtak semmilyen szöveges leíratot, az a BERT modellekhez (Devlin és mtsai, 2018) hasonló módon pusztán a bemenő strukturálatlan (itt: audio) adatokból nyeri a tudását. A tanítás során a transzformer modul súlyait nem fagyasztottuk be, hanem egy kis értékű tanulási rátával engedték a BEA-B adatokra rátanulni. Ehhez 60 epoch-ot állítottunk be, minden egyébben is az előzőek, vagyis az 5.2-ben részletezettek szerint eljárva.

Paraméterszám	LM	dev-repet	dev-spont	eval-repet	eval-spont
330.8M	–	<b>5.09 / 1.1</b>	<b>16.22 / 5.17</b>	<b>5.81 / 2.09</b>	<b>16.62 / 5.53</b>

**5. táblázat.** wav2vec2.0\_XLSR-53k+GRU+CTC+Attention+BPE\_600 = BEAST WER[%] / CER[%] eredmények a BEA-Base adatbázison.

Az eredmények önmagukért beszélnek. Szinte minden mutatóban szignifikáns javulást sikerült elérni (a minden eddiginél jobbakat vastagon szedtük). A legfontosabbnak tekintett spontán kiértékelő halmazon mind szóhibaarány mind betűhibaarány tekintetében kb. 40%-kal csökkentettük a legalacsonyabb (HMM/CNN-TDNN-F) kiindulási hibaértékeket.

### 6.1. Kiegészítő tesztek a magyar CV teszhalmazon

Bár a javulásokkal elégedettek voltunk, mindig kérdéses egy modell általánosítóképessége, különösen, ha kis tanítóadatbázisra hangolják. Ezért megnéztük, hogy a magyar CV (Common Voice) adatbázis dedikált teszhalmazán milyen pontosságot ér el. Erre WER=28.65%-ot kaptunk, ami bár elsőre nem tűnik különösebben alacsonynak, tudomásunk szerint az eddigi legjobb eredmény (Grosman, 2021) (mindezt úgy, hogy a tanítás során magyar CV hanganyagot/leíratot nem használtunk). Hogy érzékeltessük, hogy a spontán BEA-B hanganyagokhoz képesti *közel kétszeres beszédfelismerési hiba* mennyivel jobb, mint aminek látszik, mutatunk néhány *átlagosnál „rosszabb”* beszédfelismerési kimenetet a magyar CV teszhalmazból a BEAST-tel átírva és a referenciához illesztve.

Format:

```
<utterance-id>, WER DETAILS
<eps> ; reference ; on ; the ; first ; line
  I ; S ; = ; = ; S ; D
and ; hypothesis ; on ; the ; third ; <eps>
```

```
=====
common_voice_hu_23387669, %WER 33.33 [ 2 / 6, 0 ins, 1 del, 1 sub ]
a ; fehér ; ruhás ; is ; meglepődött ; kissé
= ; S ; D ; = ; = ; S ; D
a ; fehérruhás ; <eps> ; is ; meglepődött ; kissé
=====
common_voice_hu_23356499, %WER 57.14 [ 4 / 7, 0 ins, 1 del, 3 sub ]
és ; felmutatott ; egy ; szürke ; tollú ; kis ; madárra
= ; S ; = ; = ; S ; S ; D
és ; felmuthatott ; egy ; szürke ; tolló ; kismadárra ; <eps>
=====
common_voice_hu_23735287, %WER 71.43 [ 5 / 7, 1 ins, 0 del, 4 sub ]
és ; nézzétek ; hogy ; lüktet ; <eps> ; a ; kicsi ; torka
S ; = ; = ; S ; I ; = ; S ; S
s ; nézzétek ; hogy ; lük ; tehát ; a ; kicsit ; orka
=====
common_voice_hu_23631054, %WER 100.00 [ 6 / 6, 1 ins, 2 del, 3 sub ]
fel ; is ; kelt ; nyomban ; és ; továbbindult ; <eps>
S ; S ; D ; D ; = ; S ; I
a ; liskátnyomban ; <eps> ; <eps> ; és ; tovább ; indult
=====
```

Látható, hogy még a reménytelenül rossznak gondolható, lokális WER=100%-os mondatban is lehetnek hasznos részek, így a BEAST (BEA Speech Transcriber) általánosabb beszéd-szöveg átalakítási célra is használható lehet. Mivel a magyar számítógépes nyelvészeti kutatóközösségben már régóta jelentős az igény egy publikus beszéd-szöveg átalakító rendszerre, a BEAST-et elérhetővé tesszük<sup>5</sup> számukra.

## 7. Összefoglalás

Az elmúlt években a „deep learning forradalom” hatására a nyelv- és beszédtechnológia is elképesztő fejlődésen ment keresztül. Ezekből a hatásokból azonban keveset profitálhatunk, ha nincs megfelelő beszédadatbázis a nyelvünkre az adott – jelen esetben beszédfelismerési, gépi leiratozási feladatra. Munkánk egyik célja, hogy alkalmassá tegyük és megosszuk az elődeink által készített BEA adatbázist a magyar nyelvű beszéd-szöveg átalakítási feladatra. Másrészt, ezt bizonyítandó, nyílt forráskódú, hatékony beszédfelismerő motort adjunk kézre a kutatóközösségnek. Azt gondoljuk, hogy projektünk kezdeti szakaszában is jelentősen előre tudunk lépni a célokat illetően. Tisztában vagyunk vele, hogy az alkalmazott megközelítések kissé ad-hoc jellegűek és még számos vizsgálat van hátra. Talán az egyik legígéretesebb irány az önfelügyelt módon tanítható autoregresszív neurális nyelvmodellek – pl. GPT-2 – alkalmazása általánosabb magyar nyelvű szövegek alapján. További, vizsgálandó lehetőség magyar nyelvű, leírással nem rendelkező nagy mennyiségű hanganyagokon előtanítani és alkalmazni egy

<sup>5</sup> <https://phon.nytud.hu/bea>

magyar wav2vec2.0 transzformert. Terveink között szerepel még a BEA-Base bővítése BEA-Mediummá, mely a nagyobb méret miatt további pontosságjavulással kecsegtet.

## Köszönetnyilvánítás

Köszönettel tartozunk a munkánkat elősegítő projekttámogatásokért: NKFI-135038 „Prozódiai szerkezet és mondattípusok vizsgálata nagy beszédatadabázisokon mély tanulási támogatással”, ELKH infrastruktúra-fejlesztés, NVIDIA Academic Hardware Grant 2021, ITM-NKFIH MILab. Külön köszönetünket fejezzük ki mindazoknak, akik áldozatos munkájukkal a BEA adatbázis létrejöttéhez hozzájárultak.

## Hivatkozások

- Baevski, A., Zhou, H., Mohamed, A., & Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv preprint arXiv:2006.11477 (2020)
- Beke, András és György Szaszák. "Automatic summarization of highly spontaneous speech." International Conference on Speech and Computer. Springer, Cham (2016)
- Chen, S. F. and Goodman, J.: "An empirical study of smoothing techniques for language modeling," Computer Speech & Language, vol. 13, no. 4, pp. 359–393 (1999)
- Chan, W., Jaitly, N., Le, Q., & Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 4960-4964 (2016)
- Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv, abs/1412.3555. (2014)
- Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M.: Unsupervised Cross-lingual Representation Learning for Speech Recognition. ArXiv, abs/2006.13979 (2021).
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- Gósy Mária, Gyarmathy Dorottya, Horváth Viktória, Grácsi Tekla Etelka, Beke András, Neuberger Tilda, Nikléczy Péter (2012), BEA: Beszélt nyelvi adatbázis. In: Gósy Mária (szerk.), Beszéd, adatbázis, kutatások. Akadémiai Kiadó, Budapest. (2012)
- Gyarmathy, D., Neuberger, T. A BEA-adatbázis alkalmazásfüggő lejegyzései BESZÉDKUTATÁS 19 pp. 109-120. (2011)
- Gyarmathy D., Neuberger T., Grácsi T. E.: Lejegyzési útmutató a BEA Spontánbeszéd-adatbázis háromszintű annotálásához. Alkalmazott Nyelvtudomány pp. 35-44. (2014)
- Graves, A.; Fernández, S.; Gomez, F., Schmidhuber, J.: "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks". ICML 2006, pp. 369–376, (2006)
- Graves, A. and Jaitly, N.: "Towards End-To-End Speech Recognition with Recurrent Neural Networks." ICML (2014)
- Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y.: Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
- Hunyadi László, Földesi András, Szekrényes István, Staudt Alexandra, Kiss Hermina, Abuczki Ágnes, Bódog Alexa, Az ember–gép kommunikáció elméleti–technológiai modellje és nyelvtechnológiai vonatkozásai. ÁNyT XXIV (2012)
- Jelinek, F.; Bahl, L.; Mercer, R.: "Design of a linguistic statistical decoder for the recognition of continuous speech". IEEE Transactions on Information Theory. 21 (3), pp. 250. (1975).

- Grosman J.: <https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-hungarian> - 2022. 01. 09.-i állapot.
- Kriman S. et al., "Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 6124-6128, (2020)
- Mihajlik, P., Fegyó, T., Tüske Z., and Ircing P.: "A Morpho-graphemic Approach for the Recognition of Spontaneous Speech in Agglutinative Languages – like Hungarian," Interspeech'07, Antwerp, Belgium, (2007)
- Mihajlik, P., Tüske, Z., Tarján, B., Németh, B., Fegyó, T.: Improved recognition of spontaneous Hungarian speech—Morphological and acoustic modeling techniques for a less resourced task, IEEE Transactions on Audio, Speech, and Language Processing 18 (6), 1588-1600, (2010)
- Pan J., Shapiro J., Wohlwend J., Han K. J., Lei T., and Ma T., "ASAPP-ASR: Multistream CNN and Self-Attentive SRU for SOTA Speech Recognition," in Proc. INTERSPEECH, pp. 16–20. (2020)
- Panayotov V., Chen G., Povey D., and Khudanpur S., "Librispeech: an ASR corpus based on public domain audio books," in ICASSP, pp. 5206–5210 (2015)
- Park, D. S. et al., "SpecAugment: A simple data augmentation method for automatic speech recognition", in Proc. Interspeech, (2019)
- Povey, Daniel & Ghoshal, Arnab & Boulianne, Gilles & Burget, Lukáš & Glembek, Ondrej & Goel, Nagendra & Hannemann, Mirko & Motlíček, Petr & Qian, Yanmin & Schwarz, Petr & Silovský, Jan & Stemmer, Georg & Vesel, Karel: The Kaldi speech recognition toolkit. IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (2011)
- Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., Khudanpur, S.: Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. Proc. Interspeech, 3743-3747. (2018)
- Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., ... & Bengio, Y.: SpeechBrain: A General-Purpose Speech Toolkit. arXiv preprint arXiv:2106.04624. (2021)
- Sennrich, R., Haddow, B., & Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
- Siemund, R., Höge, H., Kunzmann, S., & Marasek, K.: SPEECON-Speech Data for Consumer Devices. In LREC, (2000)
- Szabó, M. K., Galántai, J.: Egy magyar nyelvű spontán beszélt nyelvi korpusz (HuTongue) létrehozásának tapasztalatai. In: XXVI. MANYE Kongresszus konferenciakötete. Pécs (2017)
- Szabó, M. K., Vincze, V., Ring, O., Üveges, I., Vit, E., Samu, F., ... & Takács, K.: StaffTalk: magyar nyelvű spontán beszélgetések korpusza. XVII. Magyar Számítógépes Nyelvészeti Konferencia, (2021)
- Szaszák, G., Tündik, Á. M. and Vicsi, K.: Automatic speech to text transformation of spontaneous job interviews on the HuComTech database, 2011 2nd International Conference on Cognitive Infocommunications (CogInfoCom), pp. 1-4. (2011)
- Vicsi K., Vig A.: Az első magyar nyelvű beszédadatbázis. Beszédkutatás'98. 163–178. (1998)
- Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi T.: Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240-1253, (2017)
- Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., ... & Ochiai, T. Espnet: End-to-end speech processing toolkit. arXiv preprint arXiv:1804.00015. (2018)
- Zorila, C., Boeddeker, C., Doddipatla, R., & Haeb-Umbach, R.: An investigation into the effectiveness of enhancement in ASR training and test for CHiME-5 dinner party transcription. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 47-53). (2019)



# ALKALMAZÁSOK



# Nyelvtanulást elősegítő feladatok automatikus előállítása finn és magyar nyelvekre

Ferenczi Zsanett

Pázmány Péter Katolikus Egyetem  
ferenczi.zsanett@hallgato.ppke.hu

**Kivonat** Nyelvtanulók számára gyakran nehézséget okoz agglutináló jegekkel rendelkező nyelvek esetén az egyes lemmák különböző esetragokkal ellátott alakjainak előállítása, valamint annak elsajátítása, hogy egy adott környezetben pontosan mely esetragot szükséges használni. Ezen kívül a nyelvek egyedi jellegzetességei is kisebb-nagyobb akadályokat gördítenek a nyelvtanulók elé. A nyelvtanulás elősegítése érdekében meghatároztunk néhány feladattípust, majd automatikus módszereket használva és bizonyos nyelvi mintázatokat kiaknázva példamondatok ezreiből egy nyelvtanuló alkalmazást hoztunk létre finn és magyar nyelvet tanulók számára. Ezen folyamatot jelen cikkben mutatjuk be.

**Kulcsszavak:** számítógéppel támogatott nyelvtanulás, nyelvtanulás, szókincsfejlesztés, digitális szókárttyák, finnugor nyelvek

## 1. Bevezetés

Az idegen nyelvek tanulásának hátterében több ok is állhat. Ha valaki külföldön szeretne elhelyezkedni, szinte elengedhetetlen az adott ország nyelvének ismerete, a diploma megszerzéséhez ma már szükség van egy államilag elismert nyelvvizsga megszerzésére, valamint pusztán az idegen nyelvek iránti érdeklődés is elengedő ok lehet arra, hogy egy nyelvet elsajátítsunk.

A digitális eszközök az oktatás és nyelvoktatás területén is egyre gyakrabban vannak jelen. A kontakt oktatási környezetet már sok esetben felváltja az online keretek között zajló oktatás. Az elmúlt években a járványügyi helyzet miatt sok nyelviskola is arra kényszerült, hogy teljesen online, esetleg hibrid (azaz félig online, félig tantermi) tanfolyamokat indítson.

Egy hagyományos nyelvkurzuson, esetleg iskolai nyelvoktatás során a nyelvoktató feladata az, hogy a diákokat elegendő gyakorlattal, nyelvi feladattal lássa el, amelynek segítségével a tanulók a nyelvórán újonnan elsajátított tananyagokat önállóan is begyakorolhatják. Elegendő mennyiségű gyakorlatot manuálisan előállítani időigényes feladat lenne, ezt az oktatók sokszor úgy próbálják meg elkerülni, hogy több tankönyvből, munkafüzetből, esetenként digitális erőforrásokból dolgoznak a nyelvórák során. Ezen túlmenően azonban a feladatokra adott válaszokat is nagy munka egyetlen tanárnak kézzel ellenőrizni, amennyiben az adott gyakorlófeladatokhoz nem tartozik olyan megoldókulcs, melyet a tanulók használhatnának a saját maguk által megírt feladatok ellenőrzéséhez.

Az uráli nyelvcsaládhoz tartozó finn és magyar nyelv tipológiailag agglutináló nyelvnek számítanak, gazdag morfológiájuk miatt viszonylag nehézkes az elsajátításuk. A finn azonban nem nevezhető prototipikusan agglutináló nyelvnek. Ez olyan morfofonológiai változásoknak (pl. fokváltakozás) köszönhető, amelyek a nyelvtanulók dolgát még inkább megnehezítik, és még több gyakorlásra adnak okot (Simon, 2015). Ezen nyelvek tanulásakor nagy erőfeszítést igényel a magánhangzó-harmónia és bizonyos morfofonológiai változások megértése, helyes használata. Nem csak a paradigmák között fennálló változatosság, eltérés jelent problémát tanulásakor, hanem az egyes szavak paradigmáinak mérete, az esetrendszer kiterjedtsége is. A magyar esetek száma (irodalomtól függően) 17 és 28 közé tehető (vö. Tompa (1961), Antal (1961), Kiefer (1987, 2006, 2018)), míg a finnben általában 14 vagy 15 nyelvtani esetet szokás megemlíteni (Hakulinen és mtsai, 2004). Ezek teljes elsajátítása tehát jóval több időt vesz igénybe ezen agglutináló nyelvek esetében, mint például az indoeurópai nyelvek közé tartozó angol 2-3 esete, a német 4 esete vagy az orosz 6 esete.

Máté (1999) felmérése alapján a magyar nyelvet tanuló finn anyanyelvűek számára nehézséget jelent még a határozott és határozatlan igeragozás elsajátítása, az igekötők, valamint a magyar birtokos szerkezetek helyes használata is.

Ugyanakkor a finnül tanulók is különböző nehézségekbe ütköznek nyelvtanulásakor. Karlsson és Chesterman (2008) szerint a legtöbb tanuló számára megdöbbentő, hogy a finn szókincs szinte semmilyen másik nyelv szókincséhez nem hasonlít, és nagyon bonyolult szabályok határozzák meg, hogy a mondat tárgya a lehetséges esetek közül éppen melyikben áll egy adott mondatban. Az is újdonságot jelent számukra, hogy a finn három múlt időt használ a magyar sztenderdben használt egy múlt idővel szemben.

## 2. Kapcsolódó irodalom

Ezen kihívást jelentő feladatok egyszerűbbé tételére, automatizálására már különféle megoldásokat kínáltak különböző nyelvek esetén. A számítógéppel támogatott nyelvtanulás (angolul *Computer-assisted language learning*, röviden CALL) célja, hogy fellendítse azon digitális eszközök, alkalmazások kiépítését, amelyek segítségével a tanulók a nyelvtudásukat, nyelvi készségeiket fejleszteni tudják. A CALL egyik tipikus megnyilvánulási formája az úgynevezett *fill in the blank*, azaz behelyettesítési feladatok, amelyeket különböző nyelvtechnológiai eszközök segítségével akár automatikusan is létrehozhatunk és kiértékelhetünk. Ilyen feladatra láthatunk példát az 1. ábrán.

The priest was innocent \_\_\_\_ he could not prove it.

1. ábra: Behelyettesítési feladat.

A tanulók önállóan gyakorolhatnak számos idegen nyelvet, illetve főleg azok szókincsét például olyan platformokon, mint a Memrise<sup>1</sup>, Busuu<sup>2</sup> vagy a Duolingo<sup>3</sup>. Ezen applikációknak csak egy része ingyenes, általában az összes általuk kínált funkció csak előfizetéssel vehető igénybe. Ezen kívül a fenti platformok csak a legtöbb beszélővel rendelkező nyelveket ölelik fel, mint például az angol, francia, orosz és kínai. A három applikáció közül csak a Duolingo tartalmaz finnugor nyelveket, a finnt és a magyart.

Vannak kifejezetten a kisebb uráli nyelveket és azok tanulását fellendíteni kívánó kutatások, melyek automatikus módszerekkel hoznak létre ingyenes tananyagokat a tanulók számára (Uibo és mtsai (2015), Antonsen és mtsai (2009)). A Revita projekt (Katinskaia és mtsai, 2017) keretein belül olyan alkalmazást fejlesztenek, amely segítségével szintén néhány kisebb finnugor nyelvet vagy akár a finnt is gyakorolhatjuk sok nagyobb nyelv (pl. orosz, svéd) mellett. A Revita online felületén a legalább középfokú nyelvtudással rendelkező felhasználó behelyettesítési feladatokat oldhat meg. Az oldalon megjelenik egy egybekezdésnyi célnyelvi szöveg, amelyből bizonyos szavakat automatikus módszerekkel eltávolítottak. A szöveget a nyelvtanulónak kell rekonstruálnia úgy, hogy beírja a megadott lemma alapján a mondatból hiányzó alakot, vagy kiválasztja a legördülő menüben feltűnő elemek közül a szövegbe leginkább illeszkedő szót. Egyes elemek esetén a tanuló hallás utáni szövegértését méri: ez esetben egy kis hangfájlt kell meghallgatni, majd a hallott szót megadni a szövegbeviteli mezőben. Ezen feladatok esetén azonban teljesen tetszőleges módon távolítják el a szavakat a szövegből. Egy nyelv tanulásakor a tanuló egyre több nyelvtani szabállyal bővíti tudását, és ezen nyelvtani egységek elsajátítását az arra kiélezett feladatokkal tudja legkönnyebben begyakorolni. Fontos, hogy a Revita felületét használók már valamilyen szinten ismerjék, beszéljék az adott nyelvet, mivel ezen projektben nem osztják fel nyelvtani típusfeladatokra a szövegeket, és nem a tanuló nyelvtudásának aktuális szintje határozza meg a feladatok nehézségét.

Jelen cikkben bemutatjuk azt az online nyelvtanuló alkalmazást, amely segítségével a finn, illetve magyar nyelvet tanulók elmélyíthetik tudásukat, bővíthetik szókincsüket, és begyakorolhatják az ezen nyelvek esetén leggyakrabban nehézséget okozó nyelvtani szabályosságokat.

### 3. Számítógéppel támogatott nyelvtanulás

#### 3.1. Szókincs

Különböző stratégiák léteznek az idegen nyelvű szókincs minél hatékonyabb bővítésére. Ezek egyike a papíralapú vagy virtuális szókérték használata.

A szókérték célja a nyelvtanuló szókincsének bővítése, új szavak, kifejezések elsajátítása. Ilyen kértéket több módon is létrehozhatunk: hagyományosan a célnyelvi (L2) szót, kifejezést felírjuk egy kérték egyik oldalára, míg a hátoldalra

<sup>1</sup> <https://www.memrise.com/>

<sup>2</sup> <https://www.busuu.com/>

<sup>3</sup> <https://www.duolingo.com/>

vagy a forrásnyelvi (L1) vagy a célnyelvi (L2) definíció kerül. Elgort (2013) kutatása azt mutatta ki, hogy kezdő nyelvtanulók szignifikánsan jobb eredményt érnek el a forrásnyelven való megfelelők segítségével, míg a haladóbb tanulók esetében a különbség L1, valamint L2 definíciók használata között nem ennyire látványos. Jo (2018) felmérése is azt mutatja, hogy L1 definíció segítségével magasabb pontszámot értek el a tanulók, mint L2 definíciókkal.

Több kutatás is létezik, amely a papíralapú szókétyákat veti össze a virtuális szókétyák használatával, és ezek közül például Kilickaya és Krajka (2010), valamint Basoglu és Akdemir (2010) is kimutatta, hogy a virtuális szókétyákat használó csoportok teljesítménye felülmúlta a papíralapú kétyákat használókat.

Ezen kutatások alapján alkalmazásunkban virtuális szókétyákat hoztunk létre mind kezdő, mind haladóbb szinteken lévő tanulók számára. A szókétyák használata előtt a tanuló döntheti el, hogy a kétyák hátoldalán a kérdéses elem fordítását vagy annak célnyelvi definícióját szeretné-e látni. Az új szóanyag elsajátítása utáni tesztfázis a produktív előhívási készségeket (Laufer és mtsai, 2004) fejleszti. Ez azt jelenti, hogy a tanulónak emlékeznie kell a korábban látott célnyelvi szóra, és meg kell adnia a megfelelő szövegbeviteli mezőben, mindezt úgy, hogy a kérdéses szónak vagy a forrásnyelvi megfelelőjét vagy a célnyelvi definícióját jelenítjük meg számára.

### 3.2. Nyelvtan

A különböző nyelvtani szerkezeteket a tanuló a korábban is bemutatott behelyettesítési feladatok segítségével gyakorolhatja, melyet finn és magyar számítógépes nyelvfeldolgozó eszközökkel megtámogatva állítottunk elő.

Alkalmazásunk létrehozásakor a nyelvtan azon részeire fókuszáltunk, amelyek a legnagyobb nehézséget okozzák magyarul tanuló finn, illetve finnül tanuló magyar anyanyelvűek számára Máté (1999) és Karlsson és Chesterman (2008) megfigyelései alapján (lásd 1. fejezet). Összesen három finn és három magyar nyelvtani egységre dolgoztunk ki olyan mintázatokat, melyek segítségével a tanulók példák ezrein keresztül tudják gyakorolni a különféle nyelvi jelenségeket. A finn esetében az egyik feladattípus a tárgy esetének kiválasztását gyakoroltatja, egy másik a három múlt idő közötti különbségtételt segíti elő, míg a harmadik a finn passzívum képzését fejleszti. A magyar nyelvtant érintő feladattípusok az igekötők gyakorlása, a határozott és határozatlan igeragozás közötti különbségek megértése, valamint a birtokos szerkezetek elsajátítása köré szerveződnek.

Egy feladatsor összeállításakor az első lépés az, hogy a korábban összegyűjtött példamondatok közül kiszűrjük a feladat típusának megfelelő mondatokat, amelyben a vizsgált nyelvi jelenség megjelenik. Ezután a kérdéses szót, szavakat eltávolítjuk a mondatból. Az eltávolított szónak bizonyos feladatok esetén megjelenítjük a lemmáját, hogy a tanulónak csak a megfelelő esetben, időben, személyben és/vagy számban kelljen elragoznia a szót. Az egyik feladattípusban (egészen pontosan a magyar igekötők esetén) a tanulónak egy zárt halmazból kell kiválasztania a megfelelő elemet, ilyenkor értelemszerűen nem adjuk meg a hiányzó szó lemmáját.

A szövegbeviteli mezők kitöltését és az űrlap elküldését követően a kiértékelés automatikusan történik, a rendszer összeveti a felhasználó választát a mondatban eredetileg szereplő kifejezéssel, és megadja a helyes válaszokat ott, ahol azok nem egyeztek meg.

## 4. Finnugor nyelvtanuló applikáció

### 4.1. A felhasznált adatok

A virtuális szókétyák és a nyelvtani gyakorlatok alapjául egy MySQL adatbázis szolgál. A finn és magyar szavakat, többszavas kifejezéseket és mondatokat tartalmazó tábla neve `Entity`. Ebben automatikus módszerekkel kigyűjtött finn és magyar lemmák, definíciók, példamondatok szerepelnek olyan erőforrásokból, mint a magyar<sup>4</sup>, finn<sup>5</sup> és angol<sup>6</sup> Wiktionary, a magyar (Miháltz és mtsai, 2008) és finn WordNet (Lindén és Carlson, 2010) és az OPUS korpusz (Tiedemann és Nygaard, 2004).

A fent említett erőforrásokat különböző nyelvtechnológiai eszközök és eljárások segítségével használtuk fel a kétnyelvű szópárok kinyeréséhez, valamint az erőforrásokban megtalálható szinonimák, példamondatok és definíciók kigyűjtéséhez.

A Wiktionary különböző verzióiból az Ács és mtsai (2013) által létrehozott `wikt2dict` eszközt használtuk fel, amellyel kétféle módon juthatunk kétnyelvű szólistákhoz. Egyrészt a Wiktionary szócikkekben megtalálható fordítási táblákból gyűjt ki fordításokat azon szócikkek esetén, amelyeknél a címszó nyelve megegyezik a Wiktionary nyelvvel, másrészt egy harmadik nyelvet felhasználva ugyanezen táblák segítségével úgynevezett háromszögelési módszerrel újabb szópárokkal bővíti a szólistákat. Ez a módszer azon az elképzelésen alapul, hogy ha egy nyelv egy bizonyos szavát lefordítjuk két másik nyelvre, akkor vélhetően ezek a fordítások egymás fordításainak is tekinthetők. Harmadik nyelvként jelen esetben az angolt használtuk, mivel ez a Wiktionary rendelkezik a legtöbb szócikkkel.

A Wiktionaryben nem csak a fordítási táblákban találhatunk finn–magyar fordításokat, hanem akkor is, amikor a finn Wiktionaryben magyar, illetve a magyar Wiktionaryben finn szavakra keresünk rá. Ezen párokat a `wikt2dict` eszköz nem gyűjti össze, így saját algoritmust írtunk ennek kiaknázására. Megoldásunk a finn és magyar nyelvű Wiktionaryk dumpjait járja be, és a kétnyelvű szólisták mellett a Wiktionary nyelvvel megegyező nyelvű szócikkekből kinyert példamondatokat és definíciókat is eltárolja. A saját fejlesztésű eszköz kódja Creative Commons Attribution-ShareAlike 4.0 licenc alatt szabadon elérhető<sup>7</sup>.

A finn és magyar WordNetekben szereplő synsetek (szinonimahalmazok) azonosítója lehetővé teszi, hogy kapcsolatot teremtsünk ezen egynyelvű erőforrások

<sup>4</sup> <https://hu.wiktionary.org>

<sup>5</sup> <https://fi.wiktionary.org>

<sup>6</sup> <https://en.wiktionary.org>

<sup>7</sup> [https://github.com/ferenczizsani/wiktionary\\_parser](https://github.com/ferenczizsani/wiktionary_parser)

elemei között. Az egymásnak megfeleltetett synseteket feloldva és az egyes lemmákat összekapcsolva szópárok ezreihez jutunk. Ezen túl a magyar WordNetben jelen lévő példamondatokat is eltárolja algoritmusunk, melynek kódja Creative Commons Attribution-ShareAlike 4.0 licenc alatt szabadon hozzáférhető<sup>8</sup>.

Az OPUS-ban található finn és magyar, szavak szintjén párhuzamosított szótárak számos fordítási párt tartalmaznak. Ezen listák elemeit szükséges volt lemmatizálásnak alávetni, mivel ez az erőforrás sok esetben különböző esetragokkal ellátott, de lemmájukat tekintve ismétlődő szópárokat tartalmazott. A finn és magyar szópárhuzamosításokat kigyűjtő algoritmus kódja szabadon elérhető CC BY-SA 4.0 licenc alatt<sup>9</sup>.

Az **Entity** tábla tartalmazza a fent bemutatott módszerekkel kinyert entitásokhoz az entitás nyelvét, szófaját és annak típusát (amely lehet lemma, többszavas kifejezés vagy akár mondat is). A tábla felépítését az 1. táblázat foglalja össze (**id** = azonosító, **text** = szöveg, **lang** = nyelv, **upos** = szófaji címke, **type** = típus).

id	text	lang	upos	type
145	gyermek	2	NOUN	lemma
436	A szülők közvetlen leszármazottja.	2	NONE	sentence
733	A bál tánccal zárult.	2	NONE	sentence
918	Tiikeri on iso kissa.	1	NONE	sentence

1. táblázat. Az **Entity** tábla felépítése.

Az entitások között bizonyos kapcsolatok jöhetnek létre, amelyeket a **Relation** tábla tárol. Ez rögzíti a reláció típusát (pl. célnyelvi megfelelője, definíciója, példamondata, szinonimája) és a relációban részt vevő két entitás azonosítóját.

A nyelvtani gyakorlatok automatikus létrehozásához a különböző erőforrások magyar és finn példamondatait használtuk fel. Az adatbázisban szereplő finn nyelvű példamondatok száma 29.087, a magyar példamondatok száma 20.158. Ezen mondatok összegyűjtése után az alábbi kritériumok mindegyikének megfelelő adatokat további elemzésnek vetettük alá:

- legalább három szóból álló mondatok;
- nagybetűvel kezdődő mondatok;
- mondatvégi írásjelet (., ! és ?) csak a mondat végén tartalmazó mondatok;
- matematikai jeleket (+, =) nem tartalmazó mondatok.

Az így megmaradt egyedi mondatok száma a finn nyelv esetén 18.043, míg a magyar mondatok száma 17.450. Ezen mondatok ezt követően tokenizáláson, lemmatizáláson, valamint morfológiai és függőségi elemzésen estek át.

<sup>8</sup> [https://github.com/ferenczizsani/connect\\_wordnets](https://github.com/ferenczizsani/connect_wordnets)

<sup>9</sup> [https://github.com/ferenczizsani/opus\\_extractor](https://github.com/ferenczizsani/opus_extractor)



A magyar nyelvre az `emtsv` eszközt (Indig és mtsai, 2019), míg finn nyelvre az `omorfi` (Pirinen, 2015) és az `uralicNLP` (Hämäläinen, 2019) eszközt használtuk fel. Ezek kimenete egységesen a Universal Dependencies CoNLL-U<sup>10</sup> formátumát követi, így az adatok átjárhatósága adott.

A kinyert adatokat szintén a fenti adatbázisban tároljuk el. A `TokenAnalysis` tábla az egyes tokeneket és azok elemzését (lemma, `lang` = nyelv, `upos` = szó-faj, `feats` = egyéb morfológiai, morfoszintaktikai jegyek) foglalja magában (lásd 2. táblázat). Az `Analysis2Sentence` tábla a mondatok és a bennük előforduló tokenek közötti kapcsolatokat tárolja, együtt a token mondatban elfoglalt helyével (= `token_position`), valamint a dependenciaelemzés kimenetével, azaz a szintaktikai címkével (= `deprel`) és a szülőcsomópont pozíciójának számával (= `head`), lásd 3. táblázat.

token_id	token	lemma	lang	upos	feats
13	.	.	2	PUNCT	_
56	kissa	kissa	1	NOUN	Case=Nom Number=Sing
102	A	a	2	DET	Definite=Def PronType=Art
479	bál	bál	2	NOUN	Case=Nom Number=Sing
480	tánccal	tánc	2	NOUN	Case=Ins Number=Sing
481	zárult	zárul	2	VERB	Definite=Ind Mood=Ind Number=Sing Person=3 Tense=Past VerbForm=Fin Voice=Act

2. táblázat. A `TokenAnalysis` tábla felépítése.

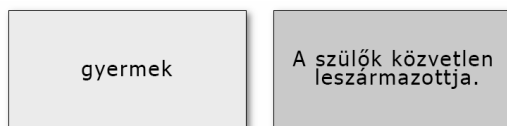
sentence_id	token_id	token_position	deprel	head
733	102	1	DET	2
733	479	2	SUBJ	4
733	480	3	OBL	4
733	481	4	ROOT	0
733	13	5	PUNCT	0

3. táblázat. Az `Analysis2Sentence` tábla felépítése.

<sup>10</sup> <https://universaldependencies.org/format.html>

## 4.2. Szókártyák

Mivel az adatbázisban a fordítási párok közötti, valamint a lemmák és azok definíciói közötti relációkat is eltároltuk, ezek felhasználhatók egy- vagy kétnyelvű virtuális szókártyák létrehozásához. Az alkalmazásunkban létrehozott virtuális szókártyák egyik oldalán tehát a célnyelvi szó, lemma található, míg a másik oldalon a tanuló kiválaszthatja, hogy a szó definíciója a célnyelven vagy a forrásnyelven szerepeljen-e (lásd 2. és 3. ábra).



2. ábra: Egy virtuális magyar nyelvű szókártya célnyelvi definícióval.



3. ábra: Egy virtuális magyar nyelvű szókártya forrásnyelvi definícióval.

A szókártyákkal való tanulás folyamata két részre osztható. A gyakorló fázisban a tanuló először begyakorolhatja, áttekintheti a számára eddig ismeretlen szavakat. Ebben a fázisban a tanuló a virtuális szókártya mindkét oldalát megtekintheti, és amikor úgy érzi, sikerült megjegyeznie az adott elemet, továbbléphet a következő kártyára. A tesztfázisban csak a kártyák definíció oldala jelenik meg véletlenszerű sorrendben, és a nyelvtanulónak meg kell adnia a definíció által leírt fogalmat.

Ezen kártyákat automatikusan generáljuk az adatbázis megfelelő tábláit felhasználva, és a válaszok is automatikusan kerülnek kiértékelésre. Fontos megjegyezni, hogy egy választ akkor fogad el helyesnek a rendszer, ha az teljes egészében megegyezik az első fázisban elsajátított fogalommal. Szinonimák megadására még nincs lehetőség, ezt a folyamatot a projekt egy későbbi szakaszában tervezzük megvalósítani.

## 4.3. Nyelvtani gyakorlatok

A finn és magyar nyelvtan három-három aspektusára hoztunk létre nyelvtani gyakorlófeladatokat, melyet ebben a fejezetben mutatunk be részletesen.

**Finn feladatok** A finn mondat tárgya négyféle esetben állhat a morfológiai és függőségi elemző kimenetét megfigyelve: nominatívusban, partitívusban, akkuzatívusban és genitívusban. Komplex szabályok határozzák meg, hogy ezen négy eset közül éppen melyiket veszi fel a tárgy. Az adatbázisban szereplő mondatok közül bizonyos nyelvi mintázatokat felhasználva automatikusan kiválaszthatjuk azokat, amelyek tartalmazznak legalább egy tárgyat, amely a fenti esetek egyikében áll. Az erre vonatkozó szabályt a következőképp fogalmazhatjuk meg: válasszuk ki azon finn nyelvű mondatokat, amelyekben található olyan névszó, amely a függőségek tekintetében a DOBJ jegyet kapta, és morfológiai jegyei között megtalálható az alábbiak egyike: *Case=Nom*, *Case=Par*, *Case=Acc* vagy *Case=Gen*. Összesen 7.088 mondat felel meg ennek a kitételnek, és ezen mondatokból a megfelelő tokent eltávolítva, azt egy szövegbeviteli mezővel helyettesítve, valamint a hiányzó token lemmáját és számát (egyes vagy többes szám) megadva előállítható a finn tárgyra vonatkozó feladatok gyűjteménye.

A finn nyelvben háromféle múlt időt különböztethetünk meg: imperfektumot, perfektumot és pluszkvamperfektumot. Az igék múlt idejű alakjainak előállítását a példamondatokból eltávolított igék megfelelő alakjának behelyettesítésével gyakorolhatja a tanuló. Az összetett múlt idők (a perfektum és a pluszkvamperfektum) az *olla* segédigéből és egy múlt idejű participiumból tevődnek össze. Ehhez a feladathoz olyan példákat gyűjtünk ki az adatbázisból, amelyekben vagy imperfektumban áll az ige, vagy amelyek tartalmazzák az *olla* segédigét és egy múlt idejű melléknévi igenevet.

Ezen kitétel után megfigyeltük, hogy a segédige és a participium nem feltétlenül állnak egymás mellett, így az imperfektum alakot azonnal ki lehet zárni a lehetséges múlt idők közül, mivel két törölt szó is megjelenik a mondatban (lásd 4. ábra). A feladat célja éppen az, hogy a tanuló a három igeidő közül kiválassza a legmegfelelőbbet az adott kontextusban, ezért ezen mondatokat jelen esetben ki kell zárnunk a példák közül. Ezt pillanatnyilag egy olyan további feltétel bevezetésével oldottuk meg, amely szerint a participiumnak közvetlenül az *olla* ragozott alakja után kell következnie.

Tämä hypoteesi  toistaiseksi  kiistämättömänä (säilyä - E/3).

4. ábra: Példafeladat összetett múlt idő esetén.

A fennmaradó példák száma a szűrést követően 5.133. Ezekből az igealakokat (és esetenként a participiumot) eltávolítottuk, és a szövegbeviteli mező után zárójelben feltüntettük az ige első infinitívuszi alakját. Mivel — a magyarhoz hasonlóan — a finn is pro-drop nyelv, a helyes alak előállításához szükség van a mondat alanyának számára és személyére vonatkozó információkra is.

A passzívum fontos szerepet tölt be a finn nyelvben. Ezt mutatja az is, hogy az adatbázisban minden tizennegyedik finn példamondat passzív alakú igét tartalmaz. Ez a jelenség a beszélt nyelvben még gyakoribb, ugyanis a többes szám első személyű igealakokat a passzív jelen idejű igealakok váltották fel. Ebbe a

feladatba azok a példamondatok kerülhetnek be, amelyekben megtalálható legalább egy passzív alakban álló finit ige, azaz a morfoszintaktikai jegyek között mind a **Voice=Pass**, mind a **VerbForm=Fin** megtalálható. Ezt követően a mondatból eltávolítjuk a passzív alakot, és annak csak a lemmáját jelenítjük meg, hogy a tanuló ezt felhasználva be tudja helyettesíteni a megfelelő alakot. Ezen nyelvtani jelenséget összesen 2.092 finn mondaton lehet gyakorolni.

**Magyar feladatok** A magyar nyelvre szintén három feladattípus implementálása történt meg.

Az első feladatot az igezőknek szenteltük. Az, hogy pontosan mely lexikai elemek számítanak az igezők kategóriájába, vitatott téma (Kalivoda, 2021). Jelen alkalmazás 13 szót kezel igezőként, de ezek listája bármikor bővíthető, illetve szűkíthető. A magyar igezők mind igemódosító pozícióban (a finit ige előtt), mind az igezőt követően, posztverbálisan is megjelenhetnek. Az általunk használt **emtsv** eszköz dependenciaelemző modulja ugyan **PREVERB** címkével látja el az igezőket, azonban erre csak akkor kerül sor, ha azok elválnak az igezőtől. A mi esetünkben a közvetlenül az ige előtt, preverbális helyzetben megjelenő igezőkre is szükség van, így a mondatban előforduló karakterláncokra hagyatkozunk. Azon mondatokra lesz szükség ezen feladat megvalósításához, amelyek vagy valamely ige elején vagy önálló szóként tartalmazzák a következő szavak egyikét: **be, ki, le, fel, meg, el, át, bele, ide, oda, szét, össze, vissza**. Az igezőt tartalmazó elemek közül 300 mondatot alaposan megvizsgálva azt állapíthattuk meg, hogy vannak bizonyos igék, amelyek ugyan valamely igezővel kezdődnek, ezek mégis az igező részét képezik. Ilyen igékre példa a *beszél*, a *becsül*, a *felejt*, a *felel* vagy a *kiabál*. Ezeket az igéket összegyűjtöttük, és kizártuk az olyan mondatokat, amelyek ezek közül bármelyiket is tartalmazzák. Ezen szűréseket követően 5.227 példamondat maradt, ezek képezik részét az igezőket feldolgozó feladatnak. A mondatokból eltávolítottuk az igezőket, és a tanuló feladata az, hogy a 13 lehetséges szó közül kiválassza a mondatba leginkább illeszkedő elemet.

A határozott és határozatlan igeragozás közötti különbségtétel sok magyarul tanulóknak okoz gondot. A tanuló olyan mondatok segítségével tudja ezen nyelvi jelenséget gyakorolni, amelyekben egy tranzitív ige található akár alanyi, akár tárgyas ragozásban. Ezt a morfoszintaktikai jegyek között megtalálható **Definite=Def** jegy-érték pár adja meg. Az lényegtelen, hogy egy tranzitív ige éppen milyen ragozásban fordul elő egy adott mondatban, hiszen a feladattal éppen az a célunk, hogy a tanulóknak kelljen eldöntenie, mikor használunk alanyi és mikor tárgyas ragozást. Az alkalmazás ezen alegysége 5.830 mondatot tartalmaz. Itt szükség van a mondat alanyának számára és személyére, illetve az igeidőre ahhoz, hogy a helyes válasz megadható legyen.

A magyar birtokos szerkezet felépítése meglehetősen eltér sok más nyelv birtokos szerkezetétől. A magyar nem a birtokoson, hanem a birtokon tünteti fel a birtokos személyjelet. Ez a birtokot kifejező szó morfoszintaktikai jelei között úgy jelenik meg, hogy a **Number[psor]** és **Person[psor]** jegyek értékei adják meg a birtokos számát és személyét. Az adatbázisból azon mondatokat kell kigyűjteni,

amelyekben előfordul egy olyan token, amelynek morfoszintaktikai jegyei között ezen elemek megtalálhatók. Egyelőre a rendszerben csak a nominatívuszban álló birtokok (**Case=Nom**) szerepelnek, mivel a hangsúlyt a birtokos személyjel helyes feltüntetésére szeretnénk fektetni. Az adatok közül 4.896 példamondat illeszkedik erre a mintázatra. Ezen feladattípus során a mondatokból eltávolítottuk a birtokot kifejező szavakat, és a tanuló a birtok számát, valamint a birtokos számát és személyét is megkapja a behelyettesítendő szó lemmájával együtt, hiszen a magyar mondatokban előfordulhat üres névmási birtokos is (lásd 5. ábra). A birtokos számának és személyének feltüntetése nélkül a behelyettesítendő szó akármely számú és személyű birtokos esetén grammatikus mondatot eredményezne, ugyanígy a birtok egyes és többes számban is elfogadható lenne.

Ma különösen csintalanok voltak a   (hűg, birtokos: E/1, birtok: E) kisfia.

#### 5. ábra: Üres névmási birtokos a magyarban.

A felhasználó válaszait minden feladattípus esetén egy külön táblában tároljuk, amely lehetőséget ad arra, hogy a jövőben az idegennyelv-elsajátítás során felmerülő problémákat a kutatók szabadon elemezhesék (a nyelvtanulók személyiségi jogait és a személyes adatok védelmét szem előtt tartva), illetve megvizsgálhassák az általunk összegyűjtött adatokat.

#### 4.4. Kiértékelés

Minden fordítási pár, valamint mondat és mondatelemzés kézi ellenőrzésen esik át, mielőtt szabadon elérhetővé tennénk őket. Nagyon fontos ugyanis, hogy a nyelvtanulók ne tanuljanak meg hibás célnyelvi megfelelőket, nyelvtani szerkezeteket, elemzéseket. A kézi elemzés jelenleg is folyamatban van az összegyűjtött adatok száma miatt, azonban a mondatok egy kisebb halmazát kézzel ellenőrizve kimutatható a fent bemutatott finn és magyar nyelvtechnológiai eszközök pontossága. Az eddigi eredményeket a 4. táblázatban foglaltuk össze. Jól látható, hogy a finn mondatok esetén a lemmatizálás az esetek nagy részében hibás kiemethez vezet, amelyet az **omorfi** elemzőrendszer végez. Az 50 ellenőrzött finn mondat pontossága mindössze 30%. A magyar mondatokat az **emtsv** rendszer többnyire helyesen elemezte, ennek pontossága 68%.

A fennmaradó mondatok elemzése folyamatban van, alkalmazásunkat úgy alakítottuk ki, hogy a nyelvtanulók csak a kézi ellenőrzésen már átesett mondatokon gyakorolhatnak különböző nyelvtani jelenségeket.

### 5. Összegzés

A cikkben bemutatunk egy olyan nyelvtanuló alkalmazást, amely a finn és magyar szókincs építését, valamint a nyelvtan gyakorlását teszi lehetővé különböző

	finn (db)	% finn	magyar (db)	% magyar
Nem teljes mondat	2	4%	1	2%
Hibás lemmatizálás	28	56%	5	10%
Hibás morfológiai elemzés	3	6%	3	6%
Hibás dependenciaelemzés	2	4%	7	14%
Pontosság	15	30%	34	68%
Összes kiértékelt mondat	50	100%	50	100%

4. táblázat. A mondatok kiértékelése.

feladatokon keresztül. A feladattípusok könnyen bővíthetők, egyedül a nyelvi mintázatok pontos leírására van szükség ahhoz, hogy az adatbázisból új példamondatokat gyűjtsünk ki, és behelyettesítéses feladatot generáljunk belőlük automatikusan.

Ezen alkalmazás egyik hiányossága, hogy csak a rendszer által tárolt választ fogadja el egyetlen helyes megoldásként, az egyéb alternatívákat, amelyek szintén grammatikus mondatokhoz vezetnének, nem. A szókétyák esetében egy-egy definíció meghatározhat több fogalmat is, és van, hogy a meghatározás nem szolgál elegendő információval egy adott fogalom felismeréséhez. Amennyiben a felhasználó nem a rendszer által korábban ismertetett szót vagy kifejezést adja meg, az automatikus kiértékelés során válasza hibásnak lesz feltüntetve. Erre irányuló fejlesztéseket a projekt egy következő szakaszában fogunk végezni.

További probléma, hogy a feladatokban használt példamondatokat, illetve a szókétya alkalmazásban megjelenített új kifejezéseket nem osztályozzuk a tudásszinteknek megfelelően, ezáltal lehet, hogy a nyelvtanulók nem a saját nyelvi szintjüknek megfelelő feladatokat kapnak. A jövőben ezt úgy tervezzük megoldani, hogy a mondatokat és kifejezéseket automatikus módszerekkel besoroljuk a Közös Európai Referenciakeret (KER) egyes nyelvi szintjeibe, és csak a tanuló szintjének megfelelő elemeket jelenítünk meg.

Amint az itt bemutatott feladatok és az azokban használt mondatok kézi kiértékelése megtörténik, a webes alkalmazást szabadon elérhetővé kívánjuk tenni minden nyelvtanuló számára. A tanulók segítségével és a feladatokra adott válaszaikkal még pontosabb képet kaphatunk alkalmazásunk hasznosságáról, illetve azokról a pontokról, amelyek még javításra szorulnak a rendszerünkben. Kifejezetten érdekes lehet a jövőben az adatbázisban tárolt válaszok alapján kvantitatív vizsgálatokat folytatni, és esetlegesen olyan típushibákra bukkanni, amelyekről korábbi kutatások még nem tettek említést. Feltételezhető, hogy empirikusan alátámaszthatóvá válnak majd olyan elméleti kutatások, amelyek tárgya a finnugor nyelveket idegen nyelvként tanulók nyelvsajátítással kapcsolatos nehézségeire irányulnak. Az adatok alapján kézzelfoghatóvá válik, hogy pontosan mely nyelvtani szerkezet jelenti a legnagyobb nehézséget a tanulók számára, és mely az, amelyet gond nélkül tudnak alkalmazni.

## Hivatkozások

- Ács, J., Pajkossy, K., Kornai, A.: Building basic vocabulary across 40 languages. In: Proceedings of the Sixth Workshop on Building and Using Comparable Corpora. pp. 52–58. Association for Computational Linguistics, Sofia, Bulgaria (2013)
- Antal, L.: A magyar esetrendszer. *Nyelvtudományi Értekezések* 29 (1961)
- Antonsen, L., Huhmarniemi, S., Trosterud, T.: Interactive pedagogical programs based on constraint grammar. In: Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009). pp. 10–17 (2009)
- Basoglu, E.B., Akdemir, O.: A comparison of undergraduate students' English vocabulary learning: Using mobile phones and flash cards. *Turkish Online Journal of Educational Technology-TOJET* 9(3), 1–7 (2010)
- Elgort, I.: Effects of L1 definitions and cognate status of test items on the Vocabulary Size Test. *Language Testing* 30(2), 253–272 (2013)
- Hakulinen, A., Vilkuna, M., Korhonen, R., Kovisto, V., Heinonen, T.R., Alho, I.: Iso suomen kieliooppi [Nagy finn nyelvtan]. In: SKS:n toimituksia 950. Suomalaisen Kirjallisuuden Seura, Helsinki (2004)
- Hämäläinen, M.: UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software* 4(37), 1345 (2019)
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – The emtsv pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (2019)
- Jo, G.: English Vocabulary Learning with Wordlists vs. Flashcards; L1 Definitions vs. L2 Definitions; Abstract Words vs. Concrete Words. *Culminating Projects in English* 132 (2018)
- Kalivoda, Á.: Igekötös szerkezetek a magyarban. Ph.D.-értekezés, Pázmány Péter Katolikus Egyetem (2021)
- Karlsson, F., Chesterman, A.: *Finnish: An Essential Grammar*. Routledge (2008)
- Katinskaia, A., Nouri, J., Yangarber, R.: Revita: a system for language learning and supporting endangered languages. In: Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition. pp. 27–35 (2017)
- Kiefer, F.: The Cases of Hungarian Nouns. *Acta Linguistica Academiae Scientiarum Hungaricae* 37(1/4), 93–101 (1987)
- Kiefer, F.: *Magyar nyelv*. Akadémiai Kiadó (2006)
- Kiefer, F.: *Strukturális magyar nyelvtan 3. kötet: Morfológia*. Akadémiai Kiadó (2018)
- Kilickaya, F., Krajka, J.: Comparative usefulness of online and traditional vocabulary learning. *Turkish Online Journal of Educational Technology-TOJET* 9(2), 55–63 (2010)
- Laufer, B., Elder, C., Hill, K., Congdon, P.: Size and strength: Do we need both to measure vocabulary knowledge? *Language testing* 21(2), 202–226 (2004)
- Lindén, K., Carlson, L.: FinnWordNet–Finnish WordNet by Translation. *LexicoNordica–Nordic Journal of Lexicography* 17, 119–140 (2010)

- Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., Váradi, T.: Methods and Results of the Hungarian WordNet Project. In: Proceedings of The Fourth Global WordNet Conference. pp. 311–321 (2008)
- Máté, J.: A magyar nyelv elsajátításának nehézségei a finn anyanyelvű tanulók szempontjából. *Hungarologische Beiträge* 12, 91–112 (1999)
- Pirinen, T.A.: Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development. *SKY Journal of Linguistics* 28, 381–393 (2015)
- Simon, V.: Ensiapu – Elsősegély: Módszertani segédanyag a finn nyelv oktatásához. Eötvös Loránd Tudományegyetem, Budapest (2015)
- Tiedemann, J., Nygaard, L.: The OPUS Corpus - Parallel and Free. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal (2004)
- Tompa, J.: A Mai magyar nyelv rendszere: leíró nyelvtan. Akadémiai Kiadó, Budapest (1961)
- Uibo, H., Pruulmann-Vengerfeldt, J., Rueter, J., Iva, S.: Oahpa! Õpi! Opiq! Developing free online programs for learning Estonian and Võro. In: Proceedings of the fourth workshop on NLP for computer-assisted language learning. pp. 51–64 (2015)



# Magyar nyelvű Fedőnevek-ágensek létrehozása nyers szövegek alapján épített gráf felhasználásával

Cserhádi Réka, Kolláth István, Kicsi András, Berend Gábor

Szegedi Tudományegyetem, Informatikai Intézet  
Szeged, Árpád tér 2.  
{cserhatir, kollathistvan}@gmail.com, {akicsi, berendg}@inf.u-szeged.hu

**Kivonat** A Fedőnevek társasjátékot játszó "kémfőnök" ágens feladata olyan szavak keresése, amelyeket egy emberi játékos bizonyos előre megadott szavak közül minél többhöz kapcsolódónak talál, de más megadott szavakhoz nem. Ezt a mai fejlett nyelvtechnológiai módszerekkel is kihívást jelent jól megvalósítani. Cikkünkben kísérletet teszünk ennek a feladatnak a matematikai leírására, és bevezetünk új, elméleti szempontból megalapozott módszereket, amivel egyben létrehozuk az első magyar nyelven működő Fedőnevek-ágenseket is, melyekkel a véletlenszerű tippelést jelentősen felülmúló eredményeket érünk el valós, emberekkel folytatott játékok keretében.

**Kulcsszavak:** Fedőnevek, Játék, PMI-gráf, Szóasszociációk

## 1. Bevezetés

A mesterséges intelligencia kutatásának már nagyon régóta egyik központi témája a különféle játékokat emberi szinten, vagy annál jobban játszó ágensek fejlesztése. A területen a legtöbb tanulmány egymás ellen játszott, matematikailag könnyen formalizálható játékokra fókuszál (ld. pl. Allis és mtsai, 1994). A Fedőnevek című, napjainkban igen népszerűnek számító társasjáték viszont ezektől nagyban különbözik.

Az eredeti játékban két csapat versenyez egymással. Egy 25 szókérttyéből álló táblán vannak a kék vagy a piros csapathoz tartozó, illetve semleges kártyák, valamint egy azonnali vereséget jelentő kártya (fekete). Egy csapat akkor nyer, ha az ő csapatukhoz tartozó összes kártya felfedésre kerül a másik csapat kártyáinál korábban, vagy az ellenfél felfedi a fekete kártyát. Azt viszont, hogy melyik kártya milyen színű, mindkét csapatból csak 1-1 ember (a kémfőnök) tudja. Ezért a kémfőnökök minden körben egy-egy utalást adnak a csapatnak, mely egy utalószóból és egy számból áll. A csapat többi tagja (ügynökök) egymással egyeztetve sorban felfedik a táblán azokat a kártyákat, amelyeket az utalószóhoz kapcsolódónak gondolnak, amíg rossz kártyára nem tippelnek, vagy el nem érik a kémfőnök által számként megadott limitet.

Tehát a játékhoz kétféle ágens, egy kémfőnök és egy ügynök(csapat) létrehozása is lehetséges. Mindkettő fő feladata az, hogy az emberi játékosokkal jó együttműködésre legyen képes. Ehhez azonban a játékban megfigyelhető emberi viselkedés modellezésére is szükség van, jelen esetben annak kiszámítására, hogy az emberek általánosan mely szavakat tartják egymáshoz kapcsolódónak.

Ez a feladat meglehetősen kapcsolódik a pszicholingvisztikában régóta és sokat kutatott szóasszociáció-modellezéshez (Palermo és Jenkins, 1964; McNeill, 1966), azonban semmiképp sem nevezhető vele ekvivalensnek. A szóasszociációs kísérleteknél az alanyoknak a lehető leggyorsabban kell adott szóhoz egy hozzá kapcsolódó tetszőleges szót megnevezni, de ebben az esetben a kémfőnök feladata egy olyan szó megtalálása, ami minél több szóhoz kapcsolódik a sajátjai közül, a többi szóhoz viszont nem, vagy jelentősen kevésbé szorosan. A feladatra szánható ideje is legfeljebb nagyon lazán (a többi játékos türelmében) van korlátozva, és – személyes tapasztalatok alapján – a kémfőnökök gyakran több perc gondolkodási időt is felhasználnak a megfelelő utalás kigondolására. Emiatt előfordul, hogy az összekapcsolt szavak bonyolult módon, akár csak áttételesen függenek össze. Az ügynökök feladata – a tábla szavai közül az utalószóhoz kapcsolódók megkeresése – ennél jobban hasonlít egyszerű asszociációkhoz, de az idő itt sem lényeges, és az asszociációknak fontos korlátját képezi a táblán szereplő szavak száma. Egy személyes játékban a játékosok közötti viszony és közös ismeretek is szerepet játszhatnak, egy robottal való játék esetén viszont ez nem befolyásol.

## 2. A játék matematikai leírása

Tételezzük fel, hogy  $V$  szótárra létezik egy  $D \in \mathbb{R}^{|V| \times |V|}$  távolság- vagy hasonlóságmátrix, amelyben  $D_{ij} = d(v_i, v_j)$  a kapcsolat egzakt mérőszáma két tetszőleges  $v_i, v_j$  szó között, vagyis minden ember tudatában pontosan ilyen erősek a kapcsolatok. Ekkor az ügynök-agens megvalósítása egyszerű: a táblán szereplő szavak közül mindig azt kell választani, amelyik a legszorosabban kapcsolódik az utalószóhoz, vagy a legközelebb van hozzá. Így egy mohó kémfőnök-agens sem túl bonyolult: legyen  $v_i$  a szótár  $i$ -edik szava, és minden  $i$ -re  $[w_{i1}, w_{i2}, \dots, w_{in}]$  az adott körben a táblán szereplő felfedetlen szavak, sorba rendezve a  $v_i$ -hez legszorosabban kapcsolódótól a legkevésbé kapcsolódóig. Ekkor azt az  $i$ -t keressük, amelyre a legnagyobb olyan  $k$  szám létezik, hogy  $w_{i1}, w_{i2}, \dots, w_{ik}$  mind az ágens csapatához tartozó szavak.

Valójában azonban ilyen feltételek mellett az ügynökök viselkedése determinisztikus, tehát tulajdonképpen a két kémfőnök játszik egymás ellen. A szótár, vagyis a lehetséges döntéseik száma véges, és a kémfőnökök ismerik minden döntés kimenetelét, azaz egymás lehetséges stratégiáit is. Így a játék már kétszemélyes diszkrét matematikai játékká, azaz véges fával leírhatóvá válik. A mohó döntés nem feltétlenül optimális, mert egy kémfőnöknek figyelembe kell vennie, hogy a saját és a másik kémfőnök döntései függvényében milyen lehetőségeik lesznek később, és ez alapján kell optimalizálnia a döntését. Ilyen keretek között az optimális stratégia kialakítása még további kutatás tárgyát képezheti, de nem a nyelvtechnológia területén, így ebben a cikkben ezzel többet nem foglalkozunk.

A fenti feltételek persze a valóságtól nagyon távol állnak, hiszen ilyen távolságfüggvény, ami minden ember mentális reprezentációinak tökéletesen megfelel, biztosan nem létezik. Ezt már csak onnan is láthatjuk, hogy a klasszikus asszociációs teszteken, ahol tulajdonképpen a legközelebbi szomszédok keresése a feladat, a kísérleti alanyok sosem adják mindannyian ugyanazt a választ (Palermo és Jenkins, 1964; Postman és Keppel, 2014). Értelmes probléma viszont az, hogy olyan távolságfüggvényt hozzunk létre, melyben a szavak közelsége átlagosan a lehető legkisebb mértékben tér el az emberek által megítélt kapcsolat-erősségtől. Sőt, ha korlátlan mennyiségű emberi adat állna rendelkezésünkre, azt is megállapíthatnánk, hogy egy véletlenszerűen választott emberi ügynök-játékos bármely utalásra mekkora valószínűséggel választja adott tábla egyes szavait, így egy sztochasztikus, de matematikailag továbbra is megoldható játékot kapnánk.

Korlátlan mennyiségű adat hiányában azonban a célunk egy a priori távolságmátrix kialakítása, ami viszonylag jól közelíti az ideális, az emberek megítéléséhez átlagosan legközelebb álló távolságmátrixot; továbbá ezen egy olyan pontozófüggvény definiálása (szintén a priori), ami a lehetséges utalásokat valószínűségi rangsorolja aszerint, hogy arra várhatóan hány jó tippet tud majd adni az emberi ügynök-játékos. Ezzel a távolságmátrixunk és a pontozófüggvényünk együtt már egy (mohó) kémfőnök-ágenst is meghatároz.

Tanulmányunkban kémfőnök-ágenseket hozunk létre kétféle távolságfüggvény és ezeken négy pontozófüggvény alkalmazásával. Ezek tesztelésére egy saját fejlesztésű webes játékban gyűjtjük az emberi ügynökjátékosok döntéseit, így először értékelünk ki Fedőnevek-ágenseket valós játékkörnyezetben. Az eredmények szerint sikerül a véletlenszerű tippelést jelentősen meghaladó, és más korábbi (de más körülmények között tesztelt) Fedőnevek-ágensekével összemérhető eredményeket elérnünk.

### 3. Kapcsolódó irodalom

#### 3.1. Asszociációk

A szóasszociációkat régóta, számos okból vizsgálják pszichológiai, pszicholingvisztikai és nyelvészeti kutatások: a segítségükkel vizsgáltak bizonyos betegségeket, használták a memória és kognitív folyamatok működésébe való betekintésre, valamint a kognitív lexikonnak és a nyelv bizonyos folyamatainak modellezésére is (Bel-Enguix és mtsai, 2019).

Az asszociációkkal foglalkozó tanulmányok közül számunkra Spence és Owens (1990) eredményei a legfontosabbak. Ők kimutatták, hogy egy korpuszban a szavak együttlőfordulásainak mennyisége jól jelzi a köztük lévő szemantikai kapcsolat szorosságát, és asszociációk erősségének mérésére is alkalmas. Jelen munkánkban ezért a szavak kapcsolódását modellező távolságfüggvényt súlyozott konkurrenciák alapján valósítjuk meg (részletesen ld. 4.1.).

Habár létezik olyan gráf (Bel-Enguix és mtsai, 2014), és ebből olyan szóbeágyazás-modell is (Bel-Enguix és mtsai, 2019), amelyet speciálisan az asszociációk alapján hoztak létre, ezek megvalósításához nehezen megszerezhető asszo-

ciációs adatokra van szükség, ami nagy erőforrásigénynek számít a nyers korpuszokkal összevetve, és nehezzé teszi az ezt felhasználó módszerek alkalmazását újabb nyelveken.

### 3.2. Fedőnevek-ágensek

A – legjobb tudomásunk szerint – első Fedőnevek-ágensre hasonlító algoritmusokat kifejezetten azért hozták létre Shen és mtsai (2018), hogy a segítségükkel az emberi asszociációkat modellezhessék. Az ő egyszerűsített játékukban a tábla mindig három főnévből áll, ezek közül előre meghatározott kettőhöz kell megfelelő utalósót találni, ami viszont csak három melléknév közül lehet az egyik. Az alábbi öt különböző távolságszámítás alapján állították elő az utalásokat:

- Bigramok valószínűsége a szavak gyakoriságához viszonyítva,
- Skip-gram (Mikolov és mtsai, 2013) szerinti koszinusz hasonlóság,
- GloVe (Pennington és mtsai, 2014) szerinti koszinusz hasonlóság,
- A ConceptNet5 (Speer és Havasi, 2013) tudásgráf szerinti kapcsolat,
- Topikmodellezéssel számított távolság.

Úgy találták, a bigramok valószínűsége alapján modellezhető az emberi játékosok viselkedése a legjobban, ami összevág Spence és Owens (1990) eredményeivel (bár ők az együttlőfordulásokat sokkal nagyobb ablakmérettel számolták).

Kim és mtsai (2019) célja már közvetlenül a játékot jól játszó ágensek építése volt. Metrikáik háttereként a

- CBOW, Skip-gram és GloVe szóbeágyazásokat (több konfigurációban),
- valamint a WordNet gráf alapú adatbázist (Miller, 1995)

használták, utóbbit számos különböző távolságfüggvénnyel. Tanulmányukban azonban nem emberi adatokkal értékelik az ágensek teljesítményét, hanem kémfőnök- és ügynökágensek párosításával, amivel viszont csak két ágens működésének hasonlósága deríthető ki, az emberekkel való együttműködési képességüktől függetlenül.

Jaramillo és mtsai (2020) a távolságot a következő reprezentációk alapján számítják:

- Wikipédia-cikkekből és szótári definíciókból számolt TF-IDF hasonlóság,
- a szavak naiv-Bayes alapú osztályozása, illetve
- a GPT2 transzformer nyelvi modell (Radford és mtsai, 2019) első rétegéből kinyert szóbeágyazások.

Ezen módszerek közül ők a GPT2 vektorait találják legalkalmasabbnak az emberekkel való együttműködésre.

A legfrissebb a témában Koyalagunta és mtsai (2021) cikke, amelyben a korábban is használt Skip-gram és GloVe szóbeágyazások mellett távolságmátrixaik előállítására

- a FastText-et (Bojanowski és mtsai, 2017),

- a BERT modellt (Devlin és mtsai, 2018),
- valamint a BabelNet tudásgráfon (Navigli és Ponzetto, 2010) egy erre a célra kifejlesztett, szavakat speciális szabályok mentén egymáshoz rendelő keretrendszert

is felhasználnak.

A fenti munkák a szavak közti távolság számításán kívül az utalások pontozófüggvényeiben is eltérnek. Koyyalagunta és mtsai (2021) jelöléseivel igazodva, legyen  $\tilde{c}$  egy lehetséges utalószó,  $I_n$  a célzott szavak halmaza, vagyis a  $\tilde{c}$  utalószóhoz legközelebbi  $n$  db kék szó,  $R$  az összes, nem a csapathoz tartozó (rossz) szó halmaza, és  $s(\cdot, \cdot)$  a két szó hasonlóságát számító függvény. Kim és mtsai (2019) pontozófüggvénye ekkor

$$g_{kim}(\tilde{c}, n) = \begin{cases} \min_{b \in I_n} s(\tilde{c}, b), & \text{ha } \min_{b \in I_n} s(\tilde{c}, b) > \max_{r \in R} s(\tilde{c}, r) \\ 0, & \text{különben.} \end{cases} \quad (1)$$

Jaramillo és mtsai (2020) ugyanezt a függvényt veszik át, emellett kiegészítik egy veszélyességi súlyozással a kártyák színe szerint. Koyyalagunta és mtsai (2021) viszont definiálnak egy másikat is:

$$g_{koy}(\tilde{c}, n) = \lambda_B \left( \sum_{b \in I_n} s(\tilde{c}, b) \right) - \lambda_R \left( \max_{r \in R} s(\tilde{c}, r) \right), \quad (2)$$

ahol  $\lambda_B$  és  $\lambda_R$  beállítható paraméterek.

Ezenkívül pedig bevezetnek egy másik módszert is, amellyel nem kizárólag a szóhasonlóságok alapján pontozzák az utalásokat, hanem a gyakoriságuk és a szótári definícióikból létrehozott vektorok (Dict2vec, Tissier és mtsai, 2017) hasonlósága alapján is – de ez tulajdonképpen az eredeti távolságmátrix módosításának tekinthető.

Az ő eredményeik szerint a GloVe alapján számolt távolságok teljesítenek legjobban a szótári definíciókkal és gyakorisággal kombinálva, anélkül viszont a FastText vektorok hasonlósága bizonyul a legjobb metrikának.

### 3.3. Gráfos szóreprezentációk

A szavak reprezentálására a legjobban bevált módszer azok vektorokhoz rendelése, de emellett találkozhatunk gráfos szóreprezentációkkal is. Vektorok esetén minden szót egy-egy vektor reprezentál, melyek között a távolságot leggyakrabban a két vektor bezárt szöge alapján számítják ki. Gráfos szóreprezentációk esetén a szótár összes szava egy nagy gráf csúcsainak feleltethető meg, ezek között a távolságot a gráftól függően sokféleképpen lehet definiálni, egy lehetőség például a két csúcset közti legrövidebb út hossza. Ilyenek a korábbi módszerekben használt tudásgráfok (Miller, 1995; Speer és Havasi, 2013; Navigli és Ponzetto, 2010) is például. Ahogy az előbb láttuk, egy Fedőnevek-ágens távolságmátrixának előállítására mindkét struktúra használható.

Jelen munkánkban egy gráfot (is) használunk, ami valamennyire hasonlít Hope és Keller (2013) együttelőfordulások alapján létrehozott gráfjára. Ezt a

Word Sense Induction (többjelentésű szavak jelentéseinek szétválasztása háttér-információk nélkül) feladatra használják, később Pelevina és mtsai (2016) egy hasonló módszert alkalmaznak szóbeágyazás-modellek többértelműsítésére.

Másik szempontból hasonlóan tekinthető gráf a GraphGlove (Ryabinin és mtsai, 2020) is, melyben a GloVe célfüggvénye szerint optimalizálják a távolságokat egy gráf csúcsainak megfeleltetett szavak között.

## 4. Saját ágenseink

Az általunk létrehozott ágensek<sup>1</sup> nemcsak abban különböznek a korábbiaktól, hogy magyar nyelven működnek, hanem javasolunk két új távolságszámítási módszert és több pontozófüggvényt is, amelyeket egymással kombinálva valós, emberekkel játszott játékban tesztelünk.

### 4.1. Távolságszámítás

A korábbi, asszociációk és együttlőfordulások kapcsolatáról szóló eredmények (Spence és Owens, 1990; Shen és mtsai, 2018) figyelembe vételével a távolságmátrixainkat nem a nyelvtechnológia legfrissebb neurális módszereivel, hanem nyers szövegben számolt együttlőfordulások alapján hozzuk létre.

Legyen  $\mathbf{V} = \{v_1, \dots, v_n\}$  egy szótárban szereplő összes szó. Az egyszerűbb módszer, hogy két szó távolságának tekintjük a következőt:

$$d_0(v_i, v_j) = \begin{cases} 1/\text{PMI}(v_i, v_j), & \text{ha } \text{PMI}(v_i, v_j) > 1 \\ \infty, & \text{különben,} \end{cases} \quad (3)$$

ahol  $\text{PMI}(v_i, v_j) = \ln \left( \frac{p(v_i, v_j)}{p(v_i) \cdot p(v_j)} \right)$ , vagyis az együttes előfordulás valószínűsége a gyakoriságokhoz viszonyítva.

A másik javaslatunk távolságszámításra (egyben egy saját, új gráfos szóreprezentáció) ennek kiterjesztése egy  $G(V, E, w)$  súlyozott gráffá, ahol a csúcsok a  $\mathbf{V}$  szótár szavainak felelnek meg, és az élsúly két csúcs között  $w(e(v_1, v_2)) = d_0(v_1, v_2)$ . Ekkor két szó távolságát Ryabinin és mtsai (2020)-hoz hasonlóan a  $G$ -ben köztük lévő legkisebb összsúlyú út összsúlyaként definiálhatjuk, azaz

$$d_G(v_i, v_j) = \min_{\pi \in \Pi_G(v_i, v_j)} \sum_{e_k \in \pi} w(e_k), \quad (4)$$

ahol  $\Pi_G(v_i, v_j)$  a  $G$ -ben  $v_i$ -ből  $v_j$ -be vezető összes út halmaza. Ennek előnye, hogy (4)-ben olyan szópárokra is értelmeződik a távolság, ahol (3) végtelent adott.

Jelen tanulmányunkban az együttlőfordulásokat a Magyar Webkorpusz 2.0 (Nemeskey, 2020) egy részén (a Wiki alkorpuszon és a 2019-es szövegeken, összesen 1,414 milliárd token), lemmatizált alakok között számláljuk, a szavak távolsága szerint is súlyozva, a GloVe szóbeágyazások tanításához használt módszerrel<sup>2</sup>, maximum 10-es ablakmérettel. Csak a legalább 700-szor előforduló szavakat vesszük figyelembe, a szótár mérete így 12 267.

<sup>1</sup> <https://github.com/xerevity/CodeNamesAgent>

<sup>2</sup> <https://github.com/stanfordnlp/GloVe/blob/master/src/cooccur.c>

## 4.2. Pontozófüggvények

Mondjuk azt, hogy az ágens a kék csapatban játszik, azaz a kék szavakhoz kapcsolódó utalásokat szeretnénk generálni, természetesen a fenti távolságfüggvények alapján. Kim és mtsai (2019) függvényei egy lehetséges utalás pontszámát az alapján határozták meg, hogy az adott szó mennyire hasonló a kék szavakhoz. Ennek viszont hátránya, hogy lehetnek az utalószóhoz hasonlító kék (jó) szavak mellett az utalószóhoz csak egy nagyon kicsivel kevésbé hasonlító más színű (rossz) szavak. Feltehetjük, hogy ilyen esetben az ügynökök kisebb valószínűséggel választják a célzott szavakat; vagy általánosan, mennél kisebb a különbség két szó utalástól való távolsága között a mi távolságfüggvényünk szerint, annál nagyobb valószínűséggel lesz az emberi játékos szerint fordított a két szó sorrendje az utaláshoz való hasonlóságban.

A korábbi munkáktól eltérően, a gráffal történő távolságszámítás miatt nekünk nem a szavak hasonlóságáról, hanem a szavak távolságáról vannak adataink, melyek alapján a lehetséges utalásokat rangsorolni szeretnénk, ezért a pontozófüggvényeinknek ebben különbözniük kell a korábbiaktól. Mivel hasonlóságról sokféle módon át lehet térni távolságra (és fordítva), a távolságokon és hasonlóságokon alapuló pontozófüggvények között ekvivalencia csak bizonyos, ezen áttérésre vonatkozó feltételek ismeretében állapítható meg.

Pontozófüggvényeinkben a célszavak távolságának és a rossz szavak távolságának különbségét, illetve arányát vizsgáljuk. Ha előre megadott számú szót kell összekötni (ahogy például Koyyalagunta és mtsai (2021) csak 2 szóra célzott utalásokkal tesztelték az ágenseiket), akkor a mi pontozófüggvényünk a legközelebbi rossz szó és a legtávolabbi célzott szó utalástól való távolságának különbségét számítja ki:

$$g_K(\tilde{c}, n) = \min_{r \in R} d(\tilde{c}, r) - \max_{b \in I_n} d(\tilde{c}, b), \quad (5)$$

a 3.2 részben használt jelölésekkel.

Vagy a legközelebbi rossz szó és a legtávolabbi célzott szó utalástól való távolságának arányát ugyanígy:

$$g_H(\tilde{c}, n) = \min_{r \in R} d(\tilde{c}, r) / \max_{b \in I_n} d(\tilde{c}, b). \quad (6)$$

Egy teljes játékban viszont realiztikusabb, hogy az ágens maga választja ki azt is, hogy az utalással hány szót céloz meg. Ezért egy másik pontozófüggvényünk a következő:

$$g_K(\tilde{c}) = \sum_{b \in I} \left( \min_{r \in R} d(\tilde{c}, r) - d(\tilde{c}, b) \right), \quad (7)$$

ahol  $I$  az összes olyan kék szó halmaza, amelyek a lehetséges utaláshoz az összes rossz szónál közelebb vannak. Amennyiben itt is csak az  $n$  legközelebbi kék szót számítanánk, ez a függvény megfelelő hasonlóság-távolság áttérés esetén ekvivalens lenne a (2) függvénnyel, ha abban  $\lambda_R = n \cdot \lambda_B$ .

Ennek a különbség helyett hányadossal számolt változata pedig:

$$g_H(\tilde{c}) = \sum_{b \in I} \left( \min_{r \in R} d(\tilde{c}, r) / d(\tilde{c}, b_i) \right). \quad (8)$$

Az utóbbi pontozófüggvényeknek viszont két fontos hibája van. Egyrészt lehetséges, hogy egyetlen, az utaláshoz nagyon közeli szó miatt adnak egy utalásnak magas pontszámot. Ennek orvoslására a legközelebbi rossz szó távolságától való különbségnek szabunk egy felső határt,  $g_{K_S}$  esetében 0.3-at,  $g_{H_S}$ -nél 5-öt.

Másrészt pedig lehetnek olyan szavak a célzottak között, melyek utalástól való távolsága csak minimálisan tér el a legközelebbi rossz szóétól, így az ügynők túl nagy valószínűséggel választják majd azokat. Ennek kiküszöbölésére pedig csak azokat a szavakat tekintjük a pontszámításnál a legközelebbi rossz szónál közelebbinek, amelyeknek az utalástól számított távolsága legalább 0.05 értékkel kisebb, mint a legközelebbi rossz szóé. Ezek önkényesen, intuíció alapján választott értékek, melyek finomhangolása az emberi ügynőktől gyűjtött adatok alapján javíthatja az ágensek teljesítményét.



1. ábra: Egy kiértékeléshez használt tábla, a kékkel keretezett szavak felfedendők.

## 5. Kiértékelés és eredmények

Az ágensek kiértékelésében nagyrészt Koyyalagunta és mtsai (2021)-hoz igazodunk. Ők a táblákra csak olyan szavakat sorsolnak, amelyek az eredeti Fedőnevek társasjáték szavai, és mi is ezt tesszük. A játékhoz tartozó 400 szóból 245-öt használunk, mert ennyi szerepel a saját adatbázisunkban, vagyis a korpuszban legalább 700-szor (az idézett tanulmányban ez a szám 208). Szintén az ő mintájukra egy táblára 20 szót sorsolunk, melyek közül 10 felfedendő, a másik 10 között



Távolság	Pontozás	Utalószó	Célzott szavak
$d_G$	$g_K(\cdot, 2)$	tűzijáték	torta, bomba
$d_G$	$g_K(\cdot, 3)$	meghökkenő	háló, csavar, egyenes
$d_G$	$g_{K_s}(\cdot)$	kapus	háló, bomba, kesztyű
$d_G$	$g_H(\cdot, 2)$	kapus	háló, kesztyű
$d_G$	$g_H(\cdot, 3)$	sarok	háló, csavar, kesztyű
$d_G$	$g_{H_s}(\cdot)$	szeglet	csavar, bomba, kesztyű, bánya, fűszer, egyenes
$d_0$	$g_{K,H}(\cdot, 2)$	tripla	torta, csavar
$d_0$	$g_{K,H}(\cdot, 3)$	hagyományos	torta, bomba, fűszer
$d_0$	$g_K(\cdot)$	szárít	háló, csavar, kesztyű, fűszer, egyenes
$d_0$	$g_H(\cdot)$	szárít	fűszer, egyenes, kesztyű, csavar, háló

1. táblázat. Távolság- és pontozófüggvényeink kombinációjával generált utalások az 1. ábrán látható táblához

nem teszünk különbséget (a kiértékelés szempontjából ezek színe nem lényeges). 100 táblát generálunk a kiértékeléshez, közülük az egyik látható az 1. ábrán.

Koyyalagunta és mtsai (2021) csak 2 szóra irányuló utalásokkal értékeli az ágenseket; egy emberi játékostól legalább 2, legfeljebb 4, az utaláshoz kapcsolódó, táblán lévő szót kérve. Mi ezért a játékosoknak legalább  $n$ , legfeljebb  $n + 2$  szó bejelölését engedjük meg ( $n$  értelemszerűen a célzott szavak száma), a sorrendet is figyelembe véve. Az adatokat egy saját fejlesztésű webes alkalmazással<sup>3</sup> gyűjtöttük, részletesebb információk találhatóak a 3. táblázatban.

Koyyalagunta és mtsai (2021)-hoz hasonlóan, a kiértékelés mérőszámaiként kiszámítjuk a játékosok által megjelölt szavak pontosságát és fedését a kémfőnök ágens által célzott szavakon. Legyen  $A$  az ágens célzott szavainak halmaza,  $u_i$  az ügynökjátékos által  $i$ -ediknek megjelölt szó,  $U_k = \{u_1, u_2, \dots, u_k\}$ ,  $U$  pedig az ügynök által megjelölt összes szó. Ekkor a következőképpen számítjuk ki az értékelés mérőszámait:

– Pontosság:

$$P@n = \frac{|A \cap U_n|}{|A|}$$

– Fedés:

$$R@n+2 = \frac{|A \cap U|}{|A|}$$

Ezeket kívül bevezetünk egy saját, az eredeti játék szempontjából relevánsabb mérőszámot, mely egy tényleges, ágens és ember által végigjátszott játék szempontjából pontosabban mutatja meg az ágens teljesítményét. A tényleges játékban ugyanis csak addig tippelhetnek az ügynökök, amíg egy rossz kártyát el nem találnak, vagyis a bejelölés sorrendje is fontos. Az előrehaladás mérőszáma azt mutatja meg, ilyen szabályok mellett hány jó kártyát sikerült az ügynököknek a körben felfedni, ha csak a célzott szavakat tekintjük jónak, vagyis azt is

<sup>3</sup> <https://fedonevekagens.herokuapp.com/>

rossznak számítjuk, ha egy nem célzott, de jó szót fed fel véletlenül. Képlettel:

$$H = k \mid \forall u \in U_k : u \in A, u_{k+1} \notin A.$$

Ezek várható értéke véletlenszerű tippek esetén a következők:

– Pontosság:

$$\mathbb{E}(\text{P@n}) = \frac{|A|}{20}$$

– Fedés:

$$\mathbb{E}(\text{R@n+2}) = \frac{|U|}{20}$$

– Előrehaladás<sup>4</sup>:

$$\mathbb{E}(H) = \frac{21}{21 - |A|} - 1.$$

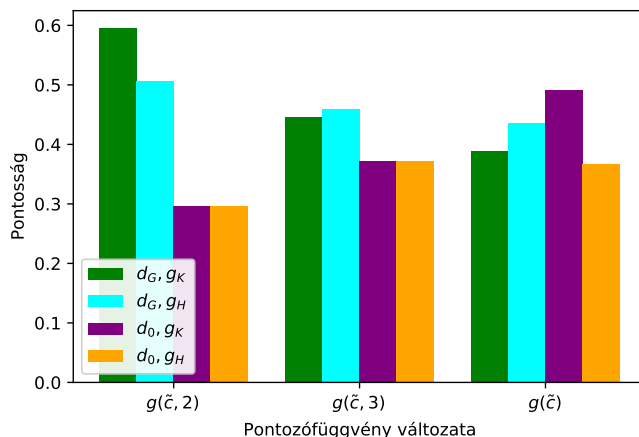
Távolság	Pontozás	Pontosság	Fedés	Előrehaladás
$d_G$	$g_K(\cdot, 2)$	0.595 (0.1)	0.611 (0.11)	1.111 (0.105)
$d_G$	$g_K(\cdot, 3)$	0.446 (0.15)	0.457 (0.165)	0.903 (0.167)
$d_G$	$g_K(\cdot)$	0.389 (0.233)	0.398 (0.243)	0.8 (0.285)
$d_G$	$g_H(\cdot, 2)$	0.507 (0.1)	0.529 (0.115)	0.865 (0.105)
$d_G$	$g_H(\cdot, 3)$	0.46 (0.15)	0.503 (0.164)	0.818 (0.167)
$d_G$	$g_H(\cdot)$	0.435 (0.273)	0.435 (0.276)	1.1 (0.351)
$d_0$	$g_{K,H}(\cdot, 2)$	0.296 (0.1)	0.304 (0.106)	0.453 (0.105)
$d_0$	$g_{K,H}(\cdot, 3)$	0.372 (0.15)	0.389 (0.166)	0.728 (0.167)
$d_0$	$g_K(\cdot)$	0.491 (0.248)	0.502 (0.256)	1.142 (0.309)
$d_0$	$g_H(\cdot)$	0.367 (0.216)	0.468 (0.227)	0.738 (0.260)

2. táblázat. Távolság- és pontozófüggvényeink eredményei, zárójelben a célzott szavak és a tippek számának átlagából számított várható értékekkel.

A kiértékelés eredményeit a 2. táblázat tartalmazza. A  $d_0$  távolságfüggvény mellett az (5) és (6) pontozófüggvények a 100 felhasznált táblán pontosan ugyanazokat az utalásokat generálták, így ezeket együtt értékeltük ki. A random játékot minden konfiguráció magasan felülmúlja, és bár a tesztelés körülményei között számos különbség van (nyelv, táblákra sorsolható szavak, emberi játékosok), a 2 szóra irányuló, gráffal történő távolságszámítással generált utalásaink jól teljesítenek Koyyalagunta és mtsai (2021) eredményeihez képest is. Az ő ágensiek pontosságai 0.3 és 0.667 közöttiek, a Dict2vec nélküliek között pedig 0.6 a legmagasabb pontosság.

Megfigyelhető ugyanakkor, hogy a gráffal történő távolságszámítás esetén a 2 szóra célzott utalások jelentősen jobban teljesítenek, mint 3 vagy nagyobb számra, pedig véletlenszerűen a nagyobb számhoz tartozó utalások eredményeinek

<sup>4</sup> A levezetést ld. pl.: Ahlgren (2014).



2. ábra: Az ágenseink pontossága

várható értéke magasabb. Sajnos az eddigi irodalomban nincs kiértékelés 2-nél több szóra célzott utalásokra, így az nyitott kérdés marad, hogy ezt a feladat nehézsége vagy a módszereink hiányosságai okozzák.

A meghatározatlan számú célzott szóra generált utalások általános problémája, ahogy az 1. táblázat példáján és a 3. táblázatban is látszik, hogy túl sok szóra célozzák az utalásokat, így azok feltehetőleg kevésbé egyértelműek. Egy ötlet ennek orvoslására a későbbiekben, ha a távolságkülönbségek harmonikus közepét vesszük a számukkal megszorozva, az összeg helyett, ami másképp a számtani közép az elemszámmal megszorozva.

Azt is láthatjuk, hogy a pontozófüggvények nem egyformán működnek a különböző távolságszámítási módszerekkel kombinálva. Sajnos eszerint a későbbiekben a legjobb konfiguráció megtalálásához szükséges lesz erre vonatkozó heurisztikákat bevezetni, vagy minden távolságfüggvényt minden pontozófüggvénnyel tesztelni kell.

## 6. Összegzés és fejlesztési irányok

Jelen munkánkban kísérletet tettünk a Fedőnevek társasjáték matematikai leírására egy kémfőnök-ágens feladatának két részre bontásával. Az emberekkel való együttműködéshez először egy olyan távolság- vagy hasonlóságmátrix megadása szükséges, amely kellően jól közelíti az emberek megítélése szerinti kapcsolatokat, majd ezen egy olyan pontozófüggvényt kell definiálni, amely a lehetséges utalásokat aszerint rangsorolja, hogy azokra hány jó tippet fog várhatóan egy emberi játékos adni.

Korábbi, asszociációkkal kapcsolatos kutatások alapján a távolságmátrixainkat egy korpusz szavai közti együttlőfordulások alapján hoztuk létre. Pontozó-

Távolság	Pontozás	Játékok	Célzott szavak	Jelölt szavak	Gondolkodási idő
$d_G$	$g_K(\cdot, 2)$	63	2	2.190	00:34 min
$d_G$	$g_K(\cdot, 3)$	62	3	3.306	00:36 min
$d_G$	$g_K(\cdot)$	50	4.66	4.860	01:09 min
$d_G$	$g_H(\cdot, 2)$	67	2	2.298	00:24 min
$d_G$	$g_H(\cdot, 3)$	55	3	3.290	00:43 min
$d_G$	$g_H(\cdot)$	50	5.46	5.520	00:47 min
$d_0$	$g_{K,H}(\cdot, 2)$	64	2	2.125	00:35 min
$d_0$	$g_{K,H}(\cdot, 3)$	59	3	3.322	00:34 min
$d_0$	$g_K(\cdot)$	49	4.96	5.122	00:55 min
$d_0$	$g_H(\cdot)$	111	4.33	4.558	00:48 min

3. táblázat. A lejátszott játékok száma, a célzott szavak és a tippek számának átlaga, és az egy táblára jutó átlagos gondolkodási idő a kiértékelésben.

függvények tekintetében is vezettünk be újításokat, egyrészt figyelembe vettük a legközelebbi rossz szó, illetve a célzott szavak utalástól való távolságának különbségét, másrészt olyan függvényt is definiáltunk, amely képes a célzott szavak számának eldöntésére is.

A továbbiakban mindenképpen célunk a saját távolság- és pontozófüggvényeink finomítása a kiértékeléssel gyűjtött adatok alapján, és az összehasonlításuk a korábbi megfelelőikkel, azonos kiértékelési körülmények között. Ezenkívül szeretnénk adatot gyűjteni emberi kémfőnök-játékosok döntéseiről is, és ezekhez ügynök-ágenseket létrehozni, ami közvetlenül a távolságfüggvény optimalizálását teszi lehetővé.

Bár számtalan nyelvtechnológiai módszer került már felhasználásra a távolságmátrixok létrehozásához, vannak még olyanok, amiket érdemes lenne kipróbálni. Ilyen például az asszociációk gráfba foglalása (Bel-Enguix és mtsai, 2014) és a GraphGlove (Ryabinin és mtsai, 2020). A távolságmátrix bevezetésének köszönhetően lehetőség nyílik több különböző módon előállított mátrix aggregálására is, létrehozva például egyszerre együttesfordulásokon, tudásgráfokon és neurális szóreprezentációkon alapuló távolságmátrixokat, ami nagyon érdekes új iránynak ígérkezik. A különböző módszerekkel előállított távolságmátrixok hasonlóságának vizsgálata pedig más területeken, például kognitív modellezésben és neurális szóreprezentációk interpretálhatóságában is fontos tanulságokat tartogathat.

## Köszönetnyilvánítás

Köszönjük a játékot tesztelő ismerőseinknek a kiértékeléshez szükséges adatokat, valamint a bírálóknak, hogy hasznos javaslataikkal hozzájárultak a cikk végső verziójához.

A cikkben bemutatott eredmények részben az Innovációs és Technológiai Minisztérium ÚNKP-21-1 kódszámú Új Nemzeti Kiválóság Programjának a Nem-

zeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készültek. A dolgozatban szereplő kutatási eredmények létrejöttét az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal is támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- Ahlgren, J.: The probability distribution for draws until first success without replacement (2014)
- Allis, L.V., és mtsai: Searching for solutions in games and artificial intelligence (1994)
- Bel-Enguix, G., Gómez-Adorno, H., Reyes-Magaña, J., Sierra, G.: Wan2vec: Embeddings learned on word association norms. *Semantic Web* 10(6), 991–1006 (2019)
- Bel-Enguix, G., Rapp, R., Zock, M.: A graph-based approach for computing free word associations. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. pp. 3027–3033 (2014)
- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146 (2017)
- Dagan, I., Lee, L., Pereira, F.C.: Similarity-based models of word cooccurrence probabilities. *Machine learning* 34(1), 43–69 (1999)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding pp. 4171–4186 (2018), <https://aclweb.org/anthology/papers/N/N19/N19-1423/>
- Hope, D., Keller, B.: Uos: A graph-based system for graded word sense induction. In: *Second Joint Conference on Lexical and Computational Semantics (\* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. pp. 689–694 (2013)
- Jaramillo, C., Charity, M., Canaan, R., Togelius, J.: Word autobots: Using transformers for word association in the game codenames. In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. vol. 16, pp. 231–237 (2020)
- Kim, A., Ruzmaykin, M., Truong, A., Summerville, A.: Cooperation and codenames: Understanding natural language processing via codenames. In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. vol. 15, pp. 160–166 (2019)
- Koyalagunta, D., Sun, A., Draelos, R.L., Rudin, C.: Playing codenames with language graphs and word embeddings. *Journal of Artificial Intelligence Research* 71, 319–346 (2021)
- McNeill, D.: A study of word association. *Journal of Verbal Learning and Verbal Behavior* 5(6), 548–557 (1966)
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)

- Miller, G.A.: Wordnet: a lexical database for english. *Communications of the ACM* 38(11), 39–41 (1995)
- Navigli, R., Ponzetto, S.P.: Babelnet: Building a very large multilingual semantic network. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. pp. 216–225 (2010)
- Nemeskey, D.M.: *Natural Language Processing Methods for Language Modeling*. Ph.D.-értékezés, Eötvös Loránd University (2020)
- Palermo, D.S., Jenkins, J.J.: *Word association norms: Grade school through college*. (1964)
- Pelevina, M., Arefiev, N., Biemann, C., Panchenko, A.: Making sense of word embeddings. In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. pp. 174–183. Association for Computational Linguistics, Berlin, Germany (Aug 2016), <https://aclanthology.org/W16-1620>
- Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543 (2014)
- Postman, L., Keppel, G.: *Norms of word association*. Academic Press (2014)
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., és mtsai: Language models are unsupervised multitask learners. *OpenAI blog* 1(8), 9 (2019)
- Ryabinin, M., Popov, S., Prokhorenkova, L., Voita, E.: Embedding words in non-vector space with unsupervised graph learning. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 7317–7331 (2020)
- Shen, J.H., Hofer, M., Felbo, B., Levy, R.: Comparing models of associative meaning: An empirical investigation of reference in simple language games. In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. pp. 292–301 (2018)
- Speer, R., Havasi, C.: Conceptnet 5: A large semantic network for relational knowledge. In: *The People’s Web Meets NLP*, pp. 161–176. Springer (2013)
- Spence, D.P., Owens, K.C.: Lexical co-occurrence and association strength. *Journal of Psycholinguistic Research* 19(5), 317–330 (1990)
- Tissier, J., Gravier, C., Habrard, A.: Dict2vec: Learning word embeddings using lexical dictionaries. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 254–263 (2017)

## Absztraktív összefoglaló PreSumm módszerrel

Agócs Ádám<sup>1</sup>, Yang Zijian Győző<sup>1,2</sup>

<sup>1</sup>MTA-PPKE Magyar Nyelvtudományi Kutatócsoport  
1083 Budapest, Práter u. 50/a.

agadam98@gmail.com, yang.zijian.gyozo@itk.ppke.hu

<sup>2</sup>Nyelvtudományi Kutatóközpont

1068 Budapest, Benczúr u. 33.

yang.zijian.gyozo@nytud.hu

**Kivonat** Kutatásunk során egy szöveges összefoglaló szoftvert készítettünk magyar nyelvre, többnyelvű és magyar BERT alapú modellek felhasználásával. Alapvetően kétfajta szöveg összefoglalási módszert különböztetünk meg egymástól, extraktív és absztraktív. Az extraktív összefoglalók csak olyan szavakat, kifejezéseket tartalmaznak, melyek megtalálhatóak az eredeti, összegezni kívánt szövegben is. Ez a módszer az eredeti szövegben található, a legfontosabb szavak kiemelésével készíti az összefoglalót. Az absztraktív összefoglalás sokkal inkább hasonlít egy ember által összefoglalt szövegre, megjelenhetnek benne olyan szavak is, melyeket az eredeti szöveg nem tartalmaz. Kutatásunk során absztraktív modelleket tanítottunk magyar nyelvre. A modellekhez többnyelvű és magyar egynyelvű BERT modelleket használtunk. Létrehoztunk egy demó alkalmazást is, amelynek segítségével, valós időben is használhatjuk az összefoglaló rendszerünket. Jelen kutatásunkban a PreSumm kódot alapul véve készítettük el az absztraktív összefoglaló demónkat.

**Kulcsszavak:** absztraktív összefoglalás, BERT, huBERT, HILBERT, mBERT

### 1. Bevezetés

Az automatikus összegző alkalmazások használata jelentős idő- és költségmegtakarítást eredményezhet, ezért egyre nagyobb az igény az automatikusan működő alkalmazások iránt. Az automatikus szövegösszegzés különösen fontos, megoldatlan probléma a magyar nyelvre. Az automatikus szövegösszefoglaláshoz egy bemeneti szövegre van szükségünk, valamint egy összefoglaló rendszerre. A rendszer a bemeneti szövegből előállít egy összefoglalót, azaz oly módon csökkenti az eredeti szöveg hosszát, hogy a tartalma közben megmarad. Az összefoglalókat generáló technológiák figyelembe vesznek olyan változókat, mint a hossz, a stílus vagy a szintaxis. A hagyományos automatikus szövegösszegző módszerek a szöveg jellemzőinek logikai számszerűsítésére támaszkodnak, ideértve a kulcsszavak súlyozását, valamint a mondatok rangsorolását. Két különböző gépi összefoglaló módszer létezik: az extraktív és az absztraktív összegzés. Az absztraktív módszerrel elkészült összefoglaló tartalmazhat teljesen új szavakat is az eredeti

szöveghez képest úgy, hogy megtartja az eredeti szöveg jelentését. Az absztraktív módszerek általában bonyolultabbak, mivel a gépnek elemeznie kell a szöveget és a legfontosabb információkat, majd meg kell tanulnia a vonatkozó fogalmakat, és összefoglalót kell készítenie. Az extraktív módszerrel történő összefoglaló készítésekor az elkészült összefoglaló nem tartalmaz új szavakat az eredeti, összegezni kívánt szöveghez képest. Az eredeti szövegben található szavakat, mondatokat rangsorolja, és innen ragadja ki az összefoglalóhoz szükséges szövegrészeket. Az extraktív összefoglaló technikák közé tartozik a mondatok és kifejezések fontossági sorrendben való rangsorolása, valamint a dokumentum legfontosabb alkotóelemeinek kiválasztása az összefoglaló elkészítéséhez. Manapság a seq2seq neurális architektúra a legkiemelkedőbb, ahol egy neurális hálózat a bemeneti szekvenciákat a kimeneti szekvenciákhoz rendeli, ezen belül a transformer modellek érik el a legjobb eredményeket, melyek teljesen új utakat nyitottak az NLP feladatokban.

Kutatásunkban a neurális absztraktív összefoglalási módszert vizsgáljuk.

## 2. Kapcsolódó irodalom

Az elmúlt évek folyamán sokféle módszerrel közelítették a problémát. A Refresh (Narayan és mtsai, 2018) egy ROUGE (Lin, 2004) metrikán alapuló módszer, amelyet a mondatok rangsorolására használnak a megerősítéses tanulási módszer segítségével.

A Latent (Zhang és mtsai, 2018) a disztribúciós szemantika egyik technikája, amely elemzi a dokumentumok és a bennük található kifejezések közötti kapcsolatokat, feltételezi, hogy a közeli jelentésű szavak hasonló szövegdarabokban fordulnak elő, célja a kulcsszavak legpontosabb követése helyett az emberi munkával készült absztraktokhoz való minél közelebbi hasonlóság elérése volt.

A Sumo (Liu és mtsai, 2019) olyan módszert alkalmaz, amely a dokumentumból kinyerhető többgyökerű függőségi fa-struktúrákra épül, és az összefoglaló lehetséges formájának előbecslésén alapszik. A NeuSum (Zhou és mtsai, 2018) a mondatok pontozásával és szelektálásával közelíti meg a problémát.

A PTgen (See és mtsai, 2017) eszköz mutatókat (pointereket) generál a szavak azonosítására a forrásszövegben, majd egy lefedettségi mechanizmus használatával tartja meg az összefoglalóban felhasznált szavakat. A Deep Communicating Agent (Celikyilmaz és mtsai, 2018) olyan ágens alapú megközelítés, ahol az ágensok együtt reprezentálják a feldolgozandó dokumentumot és ennek dekódolásához kapcsolódik egy hierarchia figyelő ágens. Ezek a kódolók egyetlen dekóderhez vannak csatlakoztatva, kiképezve a végpontokat a megerősítéses tanulás segítségével, hogy fókuszált és koherens összefoglalót készítsenek. A Deep Reinforced Modell (Paulus és mtsai, 2018) olyan belső mechanizmust használ, amely külön-külön figyeli a bemenetet és a folyamatosan generált kimenetet, valamint egy új képzési módszert, amely ötvözi a szabványos felügyelt szóbecslést és a megerősítő tanulást.

A BottomUp (Gehrmann és mtsai, 2018) megközelítés egy tartalomválasztót használ, az eredeti szövegben választja ki azokat a mondatokat, szavakat, melyeket tartalmazhatja a végleges összefoglaló.



A neurális hálóval végzett absztraktív összefoglalás a problémát egy szekvenciából egy másik szekvenciává való transzformálásként (sequence-to-sequence: seq2seq) közelíti meg. A feladathoz egy úgynevezett enkóder-dekóder architektúra szükséges. Az enkóder a változó hosszúságú forrás dokumentum tokenjeiből egy vektor reprezentációt készít, majd a dekóder az enkóder által készített vektor reprezentációjának segítségével, tokenről tokenre állít elő egy új szöveget.

A PreSumm (Liu és Lapata, 2019) eszköz számított a legmodernebbnek 2019-ben. Az extraktív és absztraktív összefoglaló modellek tanítását egy előre tanított BERT-modellre alapozza. Egy BERT-modell előtanításához rengeteg adatra és számítási kapacitásra van szükség. Szerencsére alkalmazhattuk a PreSumm eszközt, mivel az utóbbi időben több BERT modellt hoztak létre a magyar nyelv számára, valamint használhatjuk a többnyelvű BERT<sup>1</sup> modellt, amely tartalmaz magyar nyelvi információt is. Magyar nyelvre Nemeskey (2020a) hozott létre BERT modelleket, ezeket felhasználtuk kutatásainkhoz.

Az elmúlt időszakban több magyar nyelvű modellt tanítottak sikeresen<sup>2</sup>: ELECTRA, RoBERTa, ALBERT, BART és BERT large (Feldmann és mtsai, 2021).

Az elmúlt időszakban az autó regresszív módszerek érték el a legjobb eredményeket az összefoglalás területén. Ezen módszerek a Transofmer modell (Vaswani és mtsai, 2017) dekóderére támaszkodnak, és egy figyelem-maszkot használnak a teljes mondat tetején, így a modell csak az aktuális szöveg előtti tokeneket látja. Ez a módszer magasabb eredményeket ért el számos szöveggenerálási feladatnál (Yang és mtsai, 2019).

A BART (Lewis és mtsai, 2020) modell egy sequene-to-sequence modell, amely az enkóder oldalán maszkolás segítségével zajosítja a szövegeket, ebből megtanulja rekonstruálni az eredeti szöveget, ehhez kapcsolódik egy dekóder, ami a szöveggenerálásért felelős. Gyakorlatilag egy BERT Vaswani és mtsai (2017) jellegű modell összekapcsolása egy GPT (Radford és Narasimhan, 2018) jellegű modellel. Ez a modell rendkívül hatékony a szövegösszefoglaló feladatok finomhangolásához.

A T5 (Raffel és mtsai, 2020) szintén egy enkóder-dekóder modell, mely felügyelt és felügyelet nélküli feladatokon egyaránt lett tanítva, minden feladat szövegből-szöveg formátumú. Egy modellel tanul meg osztályozni, szöveget összegezni és gépi fordítani. Egy prefix segítségével különíti el a különböző feladatokat.

A PEGASUS (Zhang és mtsai, 2020) egy nyelvi modell, a fontos mondatokat eltávolítják/elfedik egy bemeneti dokumentumból, majd egy kimeneti sorozatként generálják ki őket a fennmaradó mondatokból, hasonlóan az extraktív összefoglalóhoz, a legjobb eredményeket éri el az absztraktív összefoglalás területén.

Magyar nyelven az OpinHu rendszer rendelkezik összefoglaló funkcióval (Miháltz, 2010). A rendszer kulcsszavakat és szövegkontextust használ az információkinyerésre. Lengyelne Molnár Tünde (Molnár Lengyelne, 2010) a kutatási kiadványok automatikus generálásának lehetőségeit és korlátait vizsgálta.

<sup>1</sup> <https://github.com/google-research/bert/blob/master/multilingual.md>

<sup>2</sup> <https://hilanco.github.io/>

A PreSumm (Liu és Lapata, 2019) eszköz segítségével Yang és társa megépítette az első magyar nyelvű extraktív összefoglaló eszközt (Yang és mtsai, 2020). Ebben a cikkben bemutatunk egy absztraktív összefoglaló eszközt, melyet a PreSumm rendszer felhasználásával hoztunk létre.

Jelen cikkünk első sorban a PreSumm eszközre koncentrál, ami része a TransformerSum-nak <sup>3</sup>, de ebbe sajnos nem integrálható a modell, amit mi tanítottunk az eredeti PreSumm-on. Azonban végeztünk kísérleteket a TransformerSum eszközzel is.

### 3. A BERT modell

Kísérleteink során különböző típusú BERT modelleket használtunk fel az összefoglalók elkészítéséhez.

A BERT (Bidirectional Encoder Representation from Transformer) egy több-rétegű, kétirányú Transformer enkódoló (Vaswani és mtsai, 2017). A BERT modell két nyelvi feladatra lett tanítva: maszkolás és következő mondat előrejelzés. A maszkolás során a korpuszban lévő szavak 15%-a véletlenszerűen elmaszkolásra kerül, majd a rendszer ezeket az elmaszkolt szavakat próbálja kitalálni. A következő mondat előrejelzésnél a modell kap két mondatot, az a feladat, hogy megmondja, a kapott két mondat az eredeti szövegben egymás mellett helyezkedik el, vagy csak két, a szövegből véletlenszerűen kiválasztott mondatról van szó. A szótár méretének korlátozásához, és az ismeretlen szavak problémájának kezeléséhez a WordPiece (Schuster és Nakajima, 2012) tokenizáló került felhasználásra.

A BERT egyik nagy előnye, hogy a modelleket nem csak angolul tanították. A Google létrehozott két többnyelvű modellt<sup>4</sup> is, egyik a kisbetűs másik a nem kisbetűs. A modelleket a 104 legnagyobb Wikipédiával rendelkező nyelven tanították. Ezen nyelvek Wikipédia mérete nagyban különbözik egymástól, az adatok közel 20%-át az angol Wikipédia teszi ki, ezért normalizálással kontrollálták a mintavételezést a probléma elkerülése érdekében. Ezután minden nyelvet tokenizálásnak vetettek alá, ami négy lépésből állt: kisbetűsítés, ékezetek eltávolítása, írásjelek leválasztása, whitespacek kezelése. A nem kisbetűsített modell is ezeken a lépéseken esett át, a WordPiece szótárral kezelik a nem kisbetűs és ékezetes szavakat. A WordPiece tokenizálás és szótár kezeli a kisbetűs és ismeretlen szavakat. A magyar nyelv is része ennek a modellnek.

Az első magyar BERT modellt Nemeskey (2020b) tette közzé, mely a huBERT<sup>5</sup> nevet kapta. Három huBERT modell született:

- huBERT: Magyar Webkorpusz 2.0-n<sup>6</sup> tanított BERT base modell
- huBERT Wikipedia cased: Magyar Wikipédián tanított nem kisbetűsített BERT base modell

<sup>3</sup> <https://github.com/HHousen/TransformerSum>

<sup>4</sup> <https://github.com/google-research/bert/blob/master/multilingual.md>

<sup>5</sup> <https://hlt.bme.hu/en/resources/hubert>

<sup>6</sup> <https://hlt.bme.hu/en/resources/webcorpus2>

- huBERT Wikipedia lowercased: Magyar Wikipédián tanított kisbetűsített BERT base modell

Jelenleg a huBERT modellek érik el a legjobb eredményeket a névelem felismerésében, valamint a főnévi csoportok felismerésében (Nemeskey, 2020a).

#### 4. Felhasznált korpuszok és modellek

A finomhangoláshoz használt korpuszok felépítéséhez 4 különböző forrást használtunk fel: HVG<sup>7</sup>, index.hu<sup>8</sup>, nol.hu<sup>9</sup> (NOL) és a magyar MARCELL korpusz (Váradí és mtsai, 2020). Az 1. táblázat a korpuszok fő jellemzőit mutatja be.

A HVG, az index.hu és a NOL esetében a napi online újságból vettük a cikkek törzsét, valamint az összefoglalókat reprezentáló leadeket. A HVG és az index.hu esetében két korpuszt építettünk belőlük. Az első változatban csak a HVG dokumentumokat használtuk. A második változatban (H+I korpusz) egyesítettük a HVG és az index.hu cikkeit. A MARCELL esetében jogi dokumentumokat használtunk forrásként, mely mindegyikéhez tartozik egy rövid mondatos témaleírás, amelyet az összegzéshez alkalmaztunk.

	HVG	index.hu	H+I	MARCELL	NOL
Év	2012–2020	1999–2020	-	1991–2019	1999–2016
Dokumentumok	480.660	183.942	559.162	24.747	397.343
Token	129.833.741	104.640.902	159.131.373	28.112.090	168.789.330
Type	5.133.030	3.921.893	3.053.703	450.115	2.589.211
Átlagos token # - cikk	246,27	496,27	265,17	1124,82	384,52
Átlagos token # - lead	12,43	22,33	29,97	11,22	39,71
Átlagos mondat # - cikk	23,74	35,76	11,40	49,26	17,36
Átlagos mondat # - lead	1,46	2,23	1,57	1,00	1,86

1. táblázat. A korpuszok fő jellemzői.

A BERT modellnek van egy maximum 512 szóelem (subword) hosszú megkötése (a BERT tokenizálása után), ezért kutatásunkban csak az online napi cikkeket és a hozzájuk tartozó leadeket használtuk, mert a hetilap (HVG) cikkei sokkal hosszabbak. A MARCELL esetében az átlagos mondathossz 1124,82 szóelem, ami jóval több, mint az 512, de a medián csupán 340, ami elég rövid ehhez a feladathoz.

Első feladatként (kivételesen csak HVG korpuszt használó kísérlet és MARCELL korpuszt) különböző tisztítási folyamatokat végeztünk. A tisztítási és normalizálási szempontok a következők:

<sup>7</sup> <https://hvg.hu>

<sup>8</sup> <https://index.hu>

<sup>9</sup> <http://nol.hu>

- Eltávolításra kerültek a hosszú (500<token) dokumentumok a korpuszból.
- Eltávolításra kerültek a rövid (5>token) dokumentumok a korpuszból.
- Eltávolításra kerültek azon cikkek, melyek rövidebbek voltak, mint a hozzájuk tartozó lead (Például lásd: 5. táblázat).
- Eltávolításra kerültek az irreleváns cikk részletek, mind például a: "Kövessen minket Facebook-on", "Kattintson további részletekért", "Kvíz indítása" stb.
- Eltávolításra kerültek azok a dokumentumok, amelyek szkripteket tartalmaztak.

Absztraktív összefoglaló kísérleteink során 4 különböző típusú, előre betanított BERT modellt használtunk: huBERT, huBERT Wikipedia cased, HILBERT, BERT-Base-Multilingual-Cased.

A **huBERT** (Nemeskey, 2020a) a jelenlegi „state-of-the-art” magyar (nem kisbetűs) BERT base modell, amely a Webcorpus 2.0-án<sup>10</sup> (9 milliárd token, 110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej) tanított.

A **huBERT Wikipedia cased** (Nemeskey, 2020a) egy magyar BERT base modell, amely a magyar Wikipédián tanított (170 millió token, 110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej).

A **HILBERT** (Feldmann és mtsai, 2021) egy magyar nyelvű BERT large modell, amely az NYTK v1 korpuszon (3,7 milliárd token, 340 millió paraméter, 24 réteg, 1024 rejtett réteg méret, 16 figyelmi fej) tanított.

A **BERT-Base-Multilingual-Cased**<sup>11</sup> egy többnyelvű BERT base modell, a modell tanításához kiválasztották az első 104 nyelvet, amely a legnagyobb Wikipédiával rendelkezik (110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej).

## 5. Kísérletek

Az előre betanított nyelvi modellek segítségével finomhangoltuk a magyar nyelvű absztraktív összefoglaló modelljeinket. Kutatásunk első lépése az eredeti szöveg előzetes feldolgozása volt. A cikkeket és a hozzájuk tartozó leadeket, az e-magyar tokenizálással<sup>12</sup>, a quntoken (Mittelholcz, 2017) eszközével tokenizáltuk. Ezután a tokenizált szöveget JSON formátumba konvertáltuk az összefoglaló rendszer számára. A rendszer ezután beilleszt két speciális elemet, az első a szöveg elejét jelzi, a másik pedig a mondathatárokat. Az előfeldolgozás után különböző összefoglaló modelleket tanítottunk.

Munkánk során sokat kísérleteztünk a TransformerSum-mal, ami magába foglalja a PreSumm eszközt is. A TransformerSum egy olyan könyvtár, melynek segítségével képesek vagyunk tanítani, kiértékelni valamint használni különböző szöveg összegzésre használatos Transformer modelleket. A TransformerSum támogatja mind az extraktív, mind az absztraktív, valamint a hosszú mondatos (4.096 - 16.384 token) összefoglalók elkészítését. Ezt extraktív összefoglalók

<sup>10</sup> <https://hlt.bme.hu/en/resources/webcorpus2>

<sup>11</sup> <https://github.com/google-research/bert/blob/master/multilingual.md>

<sup>12</sup> <https://e-magyar.hu>

esetében a longformer, absztraktív összefoglaló esetében pedig a LongformerEncoderDecoder -rel oldja meg. A TransformerSum olyan modelleket is tartalmaz, amelyek korlátozott erőforrás eszközökön is futnak, miközben továbbra is nagy pontosságot biztosítanak. A modellek kiértékelése automatikusan történik a ROUGE segítségével. Rengeteg problémát okozott a TransformerSum integrálása, egyes rendszereinken a mai napig nem sikerült működésképesre bírni, más rendszerünkön sikeresen el tudtuk indítani, de számos problémába ütköztünk, rengeteg kompatibilitási, valamint függőségi probléma fedte fel magát, melyek egyes rendszereinken megoldatlanok. Továbbá kódszinten ki kellett egészíteni az implementációt, mivel a párhuzamosítás során ellentmondásos beállításokat észlelt a rendszer. Ilyen például az „amp\_backend” beállítása „apex” értékre, valamint a „Trainer” függvény „strategy” paraméter beállítása. Végül, de nem utolsó sorban a saját adat használata esetén az adat struktúrájáról sem volt leírás, azt is hosszas utánajárás után sikerült kideríteni. A sok probléma ellenére az egyik rendszerünkön sikeresnek bizonyult a próbálkozásunk, a H+I korpuszunkon egy absztraktív modell tanítását sikerült elindítani. Sajnos azonban több mint 20 tanítás után sem sikerült a megfelelő hiperparaméter beállításokat megtalálni, amivel a modell konvergálna és értelmes összegzést tudna tanulni. Ezért ezen a területen további kísérletek szükségesek még.

Az absztraktív modellek tanításának másik vonala a PreSumm (Liu és Lapata, 2019) eszközzel<sup>13</sup> való kísérletek. A PreSumm módszerrel könnyebben lehet a tanítást elvégezni.

A 2. táblázatban láthatjuk a BERT base modellek és a BERT large modell módosított tanítási (finomhangolási) hiperparamétereit. Az összes többi hiperparaméter Liu és Lapata (2019) kísérleteinek alapértelmezett értékeire voltak beállítva.

	tanulási ráta (tr)	tr csökkentés	batch méret	hardver
BERT base	1e-03	0,1	20	4x GeForce RTX 2080 (12GB)
BERT large	5e-05	0,02	10	4x Tesla V100 (32GB)

2. táblázat. A BERT base és a BERT large módosított hiperparamétereit.

A kísérleteink során azt tapasztaltuk, hogy minél nagyobb a korpusz, annál több lépésre van szükség. Ennek megfelelően a következő tanítási lépéseket alkalmaztuk:

- Absztraktív összefoglalás:
  - HVG: 200.000
  - H+I: multi és huBERT: 600.000; HILBERT: 800.000
  - MARCELL: 50.000
  - NOL: 600.000

<sup>13</sup> <https://github.com/nlpyang/PreSumm>

Kísérletünk során végeztünk transzfer kísérletet is, ami azt jelenti, hogy a multi-BERT modellt először az eredeti CNN/Daily Mail korpuszon<sup>14</sup> finomhangoltuk 200.000 lépésszámmal, majd azt tovább finomhangoltuk a magyar korpuszokon, remélve, hogy az angol finomhangolásból is tanul fontos információkat.

## 6. Eredmények és kiértékelés

Az eredmények kiértékeléséhez a ROUGE (Lin, 2004) metódus került felhasználásra. A ROUGE (Recall-Oriented Under-study for Gisting Evolution) egy fedés (recall) alapú módszer, mely a gépi fordításban használt BLEU metrikán alapszik. Maga a ROUGE számos módszert tartalmaz, ezek közül a ROUGE-1, a ROUGE-2 valamint a ROUGE-L módszereket használtuk a mérésekhez. A ROUGE-1 egy uni-gram, míg a ROUGE-2 egy bigram fedést számító algoritmus. A ROUGE-L a bekezdések és mondatok szintjén vizsgálja a leghosszabb közös szószekvenciát.

A 3. és 4. táblázatban láthatóak az absztraktív modellek ROUGE eredményei. Mivel a HILBERT modell hatalmas erőforrásokat igényel, csak a H+I kísérletében használtuk, és ebben a feladatban nem a huBERT wikit használtuk, mert a huBERT tartalmazza a wikit. A MARCELL esetében, a generált kimenetnek az első mondatát vettük csak, mivel a referencia is csak egy mondatból áll.

		ROUGE-1	ROUGE-2	ROUGE-L
MARCELL (1. mondat)	multi	87,37	77,38	84,97
	huBERT wiki	89,37	79,91	86,14
	huBERT	<b>89,64</b>	<b>80,29</b>	<b>86,46</b>
HVG	multi	47,02	19,72	39,29
	huBERT wiki	49,49	21,62	41,46
	huBERT	<b>51,47</b>	<b>23,27</b>	<b>43,82</b>
H+I	multi (600k)	51,85	23,22	43,45
	multi transfer (650k)	51,61	22,25	42,85
	huBERT (450k)	<b>57,07</b>	<b>26,97</b>	<b>48,28</b>
	HILBERT (800k)	44,98	14,22	37,06
NOL	multi (600k)	43,41	17,24	35,70
	multi transfer (600k)	43,42	16,26	35,09
	huBERT (750k)	<b>51,18</b>	<b>22,61</b>	<b>43,03</b>
CNN/Daily Mail – multi		60,32	25,79	56,91

3. táblázat. ROUGE fedési eredmények.

Az eredményben az látható, hogy a magyar huBERT modellek minden esetben felülmúlják fedés értékben a multi-BERT és HILBERT eredményét. A 3. táblázatban a fedési (recall) értékeket láthatjuk, mivel a modell több mondatot

<sup>14</sup> <https://github.com/abisee/cnn-dailymail>

		ROUGE-1	ROUGE-2	ROUGE-L
MARCELL (1. mondat)	multi	72,99	65,38	71,53
	huBERT wiki	74,23	66,56	72,90
	huBERT	<b>75,85</b>	<b>68,35</b>	<b>74,61</b>
HVG	multi	<b>26,92</b>	<b>10,63</b>	<b>22,26</b>
	huBERT wiki	21,50	8,67	17,81
	huBERT	21,69	9,09	18,27
H+I	multi (600k)	<b>28,34</b>	<b>12,40</b>	<b>23,45</b>
	multi transfer (650k)	27,81	11,71	22,81
	huBERT (450k)	22,42	10,24	18,73
	HILBERT (800k)	17,36	5,41	14,14
NOL	multi (600k)	<b>30,56</b>	<b>11,57</b>	<b>24,99</b>
	multi transfer (600k)	30,34	10,83	24,42
	huBERT (750k)	26,53	11,08	22,19
CNN/Daily Mail – multi		25,76	10,91	24,37
CNN/Daily Mail – BERT		41,72	19,39	38,76

4. táblázat. ROUGE F-mérték eredmények.

is generál, és az esetek nagy részében a generált mondatok száma meghaladja a referencia mondatok számát, ezért a pontosság (precision) alacsony.

A H+I és NOL korpuszok esetében zárójelben láthatjuk a legjobb eredményt elérő lépésszámokat. A lépésszámok alapján a huBERT 450.000 lépésnél érte el a legjobb eredményeket, sokkal korábban, mint a többi modell. A HILBERT esetében nem értük el az elméleti optimumot, mert a ROUGE értékek folyamatosan nőttek. A 3. táblázatból látható, hogy a HILBERT teljesítménye sokkal alacsonyabb, mint a többi modellé, mivel a HILBERT large méretű, kétszer annyi paraméterrel rendelkezik, mint a BERT base, a modell robusztusabb és a finomhangolás nehezebb. 47 sikertelen kísérlet után találtuk meg azt a hiperparaméterek halmazát (Lásd: 2. táblázat), amelyekkel a modell konvergált. Úgy véljük, hogy a HILBERT magasabb eredményeket érhet el, de további kísérletekre van szükségünk ahhoz, hogy megtaláljuk a legjobb hiperparamétereket a legmagasabb eredmény eléréséhez.

A 4. táblázatban láthatóak az F-mérték eredmények. A nemzetközi irodalomban is az F-mérték a mérvadó. A PreSumm eszköz jellege miatt, hogy több és sokkal hosszabb mondatot is generál, mint a referencia, a fedés mértékek inkább a relevánsabbak. Példának a H+I korpuszban:

- Eredeti leadek méretei: átlag token szám: 29,97 (Lásd: 1. táblázat).
- PreSumm összefoglalók méretei: átlag token szám: 104,61.

Azokban fontos megvizsgálni az F-mértékeket is. Az érdekesség, ami látható a 4. táblázatban, hogy a multi-BERT modellek javítják a pontosságot (precision), bár fedésben gyengébbek, F-mértékben erősebbek, ami azt jelenti, hogy tömörebben tudnak releváns összefoglalókat generálni (kivételek a MARCELL kísérletek). Továbbá MARCELL esetében érdekes az F-mérték, mivel ott pontosan egy

mondatot veszünk figyelembe. Azonban itt a huBERT minden esetben a legjobb eredményt nyújtja.

A transzfer kísérletek is láthatóak az eredményekben. Az látható, hogy a transzfer tanításaink nem javították az eredményt egy esetben sem (egy esetet leszámítva, a NOL ROUGE-1 fedés esetében, de csak 1 századnyi a különbség).

Láthatunk néhány példát a 5. és a 6. táblázatban, amelyeket absztraktív összefoglaló modelljeink generáltak. A példákat elemezve észrevehetjük modelljeink néhány közös vonását. Ha a cikk hosszú (Lásd: az 5. táblázat), a modellünk kivonja a kifejezéseket az eredeti cikkből, majd egyesíti őket új mondatok létrehozásával. Hasonló az extraktív modellekhez, a különbség az, hogy az extraktív modellek teljes mondatokat választanak a cikkből, és rangsorolás után adják vissza a felhasználónak. Általában az absztraktív modellek által előállított mondatok nyelvtanilag többnyire helyesek. Minden modell több mondatot generál, de a végére "elfogy", és mondatfoszlányokat hagyhatnak (Lásd: a 5. táblázat).

Ha a cikk rövid (Lásd: az 6. táblázat), a modellek megmutatják valódi absztraktív tulajdonságukat, vagyis olyan részeket generálnak, amelyeket az eredeti cikk nem tartalmazott. Ebben az esetben túl kevés információ található az eredeti cikkben, így a kimenet többet „hallucinál” és a teljesítménye alacsonyabb.

A példákat vizsgálva láthatjuk a ROUGE metódus hátrányait, valamint a leadek felhasználásának problémáját. A ROUGE mutató csak azt mutatja, hogy a generált kimenet mennyire hasonlít a leadhez. A lead szerepe azonban gyakran az, hogy felhívja magára a figyelmet, vagy nagyon tömören csak a lényegét írja le, és nem az, hogy összegezze a cikk szövegét. A 1. példában (Lásd: az 5. táblázatot) a cikk egy teljes esetet leír, azonban a lead egy nagyon szűkszavú szöveg, míg a modellek sokkal részletesebben adják vissza a szöveget. Ez az egyik oka annak, hogy az eredményekben (Lásd a 3. táblázatot) csak körülbelül 50 %-os fedési eredményt láthatunk. A példában azt is láthatjuk, hogy a multi-BERT rövidebb összefoglalókat generál a huBERT-hez képest. A nagyon hosszú szövegek generálásának másik hibája az, hogy az összefoglaló olyan hosszú, mint a cikk maga, ami elveszíti összefoglaló jellegét. De azért az esetek többségében hosszabbak a cikkek, példának a H+I korpuszon:

- Eredeti cikkek méretei: átlag token szám: 496,27 (Lásd: 1. táblázat).
- PreSumm összefoglalók méretei: átlag token szám: 104,61.

## 7. Demó alkalmazás

A PreSumm eszközből kiindulva elkészítettünk egy demó alkalmazást, melyet a Docker program segítségével hoztunk létre. A Docker<sup>15</sup> segítségével különböző konténereket tudunk kezelni, melyek egymástól elkülönítve különböző alkalmazásokat, könyvtárakat és eszközöket kötnek össze. A konténerek imagefile-okból jönnek létre. Minden konténert egy operációs rendszer kernel működtet, így kevesebb a rendszerigénye, mint a virtuális gépeknek.

<sup>15</sup> <https://www.docker.com>



<p><b>Cikk</b></p> <p>Kedd délután a rendőrség megerősítette az Index nek , hogy reggel elfogtak a rendőrök egy férfit Budapesten , aki lőfegyverrel a kezében álldogált a Nagyvárad téren a 24-es villamos megállójában . A lapnak több szemtanú is beszámolt az esetről , a rendőrség pedig azt közölte , hogy kedd reggel 7 óra 12 perckor bejelentést kaptak arról , hogy a Nagyvárad tér 1. szám alatti villamosmegállóban egy fegyver van egy ember kezében . Lövés nem dördült , és senki nem sérült meg , a járőrök percekben belül a helyszínre érkeztek , és intézkedés alá is vontak egy 49 éves budapesti férfit . P. Lászlónál egy gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatos szabálysértés miatt előállították , és eljárást kezdeményeztek vele szemben .</p>
<p><b>Lead</b></p> <p>Lövés nem dördült , a férfit elfogták .</p>
<p><b>huBERT web</b></p> <p>Garázdaság és lőfegyverrel kapcsolatos szabálysértés miatt előállították P. Lászlót , akinél gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatban szabálysértés miatt eljárást kezdeményeztek vele szemben . Lövés nem dördült , senki nem sérült meg , de a rendőrök percekben belül elfogták a férfit . P. Lászlónál egy gáz-riasztó fegyvert is találtak . Garázdaság miatt előállították , és eljárást kezdeményeztek ellene . Garázdaság és lő fegyverrel kapcsolatos szabálysértése miatt előállították . A férfi ellen eljárást kezdeményeztek , és eljárás indult ellene . Videó</p>
<p><b>HILBERT</b></p> <p>A férfi fegyverrel a kezében sétált a 24-es villamosmegállóban , a rendőrök percekben belül elfogták , és kezdeményezik előzetes letartóztatásukat . A rendőrség eljárást indított vele szemben , és eljárást indítottak vele szemben . A férfi ellen eljárás indult . A rendőrség szerint egy gáz-riasztó fegyvert is találtak a rendőrök , és elrendelték az előzetes letartóztatását . Az MTI úgy tudja , hogy a férfi egy gáz-riasztó fegyverrel is felszerelkezett a 24-es villamos megállójában . Videó is van a helyszínről , a rendőrség a hvg.hu-nak arról számolt be , hogy kedd reggel fogtak el egy férfit , aki lőszerrel a rendőrök elől menekülve próbált meg egy embert a 24-es villamoson . hu hu</p>
<p><b>multi-BERT</b></p> <p>Kedd reggel elfogták a rendőrök azt a férfit , aki lőfegyverrel a kezében álldogált a Nagyvárad téren a 24-es villamos megállójában . Lövés nem dördült , és senki nem sérült meg , a járőrök percekben belül a helyszínre érkeztek , és intézkedés alá is vontak egy 49 éves budapesti férfit . P. Lászlónál egy gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatos szabálysértés</p>

5. táblázat. 1. példa absztraktív modellre (H+I korpusz).

Szerver-kliens architektúrára alapul a demó, azaz a szerver végzi a számítást, amelynek egy kliensen keresztül adhatjuk át az összegezni kívánt szöveget. Az alkalmazásban, a Flask keretrendszer<sup>16</sup>, valamint az uWSGI program fontos szerepet játszik. Az uWSGI<sup>17</sup> egy szoftveralkalmazás, amelynek célja full stack alkalmazások fejlesztése a tárhelyszolgáltatásokra. Az uWSGI egy natív bináris protokoll, amelyet az uWSGI használ más szerverekkel való kommunikációhoz. A Flask egy webes keretrendszer (Python modul), amely lehetővé teszi a we-

<sup>16</sup> <https://flask.palletsprojects.com>

<sup>17</sup> <https://uwsgi-docs.readthedocs.io>

<b>Cikk</b> A Daily Mail Twitter-oldalán közzétett videó szerint most épp nem történt nagy baj , de ez könnyen történhetett volna másképp is .
<b>Lead</b> Valószínűleg nem gondolták át igazán ezek a fiatalok mit tesznek .
<b>huBERT web</b> A Daily Mail Twitter-oldalára került fel egy videó arról , ahogy az egyik brit autópályán egy motorcsónaknak csapódott . A videón jól látható , hogy a sofőrnek nem sok esélye van a balesetre , de a videón jól látszik , hogy nem is kell aggódnia . És nem is ez volt az első ilyen eset . A videóban rögzítették , ahogy a britek egy hatalmasat ugranak a fejük előtt . Az alábbi videón látható , hogyan történt a dolog , és hogyan lett volna ez a videóból . A videón látható .
<b>HILBERT</b> A Magyar Kétfarkú Kutya Párt szerint most már nem is történt nagy baj , de ez így is elég lett volna . nem ez volt az első eset , hogy nem lett komoly baj . " Ez történt volna " - reagált a párt . Videón az is látszik , hogy a párt szerint nem történt baj . Igaz , ez még így is így lett volna , ha nem is lett volna baj , az eredmény nem lett volna ennyire egyszerű .
<b>multi-BERT</b> A Daily Mail videója szerint nem történt nagy baj . A videó könnyen előfordulhatott volna , hogy egy hatalmas bajba került . Nem kellett volna . Videó . Szerencsére nem történt baj , de ez könnyen történhetett volna másképp is . De nem . Videón . Mutatjuk . Szerencsén volt . Videót . És . Munkatársunk . Sőt . Mutatja !Videó !Szavazzon !! A hvg.hu hu . A

6. táblázat. 2. példa absztraktív modellre (H+I korpusz).

bes alkalmazások egyszerű fejlesztését. Kicsi és könnyen bővíthető. Az elkészült demóval képesek vagyunk valós időben is összefoglalókat készíteni. A demó és további példák a projekt oldalunkon<sup>18</sup> érhetőek el.

## 8. Összegzés

Különböző absztraktív szövegösszefoglaló eszközöket hoztunk létre a magyar nyelvre. Az összefoglaló modellek elkészítéséhez különböző BERT alapú modelleket használtunk. Az absztraktív modellekhez az előre betanított többnyelvű BERT modellt, valamint a magyar egynyelvű huBERT base, valamint a HILBERT large modelleket használtuk. Továbbá végeztünk transzfer tanítást is. A BERT alapú modellek finomhangolásához, az összefoglalók elkészítéséhez, a Pre-Summ eszközt használtuk.

Az eredmények azt mutatják, hogy az egynyelvű magyar modellek minden esetben felülmúlták a többnyelvű modellt fedés szempontjából, azonban ha az F-mértéket nézzük, a multi modellek teljesítenek jobban, ami azt jelenti, hogy tömörebben tudnak generálni, mint a magyar modellek. A transzfer tanítással azonban nem tudtunk elérni eredményjavulást.

<sup>18</sup> <http://nlp.itk.ppke.hu/projects/summarize>

## Hivatkozások

- Celikyilmaz, A., Bosselut, A., He, X., Choi, Y.: Deep communicating agents for abstractive summarization. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1662–1675. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)
- Feldmann, A., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: Hilbert, magyar nyelvű bert-large modell tanítása felhő környezetben. XVII. Magyar Számítógépes Nyelvészeti Konferencia pp. 29–36 (2021)
- Gehrmann, S., Deng, Y., Rush, A.: Bottom-up abstractive summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4098–4109. Association for Computational Linguistics, Brussels, Belgium (2018)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics (2020)
- Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
- Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (2019)
- Liu, Y., Titov, I., Lapata, M.: Single document summarization as tree induction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1745–1755. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Miháltz, M.: Opinhu: online szövegek többnyelv véleményelemzése. VII. Magyar Számítógépes Nyelvészeti Konferencia pp. 14–23 (2010)
- Mittelholcz, I.: emtoken: Unicode-képes tokenizáló magyar nyelvre. XIII. Magyar Számítógépes Nyelvészeti Konferencia pp. 61–69 (2017)
- Molnár Lengyel, T.: Automatic abstract preparation. 10th International Conference On Information: Information Technology Role in Development pp. 550–561 (2010)
- Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1747–1759. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)

- Nemeskey, D.M.: Egy embert próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged (2020a)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020b)
- Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada (2018)
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020), <http://jmlr.org/papers/v21/20-074.html>
- Schuster, M., Nakajima, K.: Japanese and korean voice search. In: ICASSP. pp. 5149–5152. IEEE (2012)
- See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. Association for Computational Linguistics, Vancouver, Canada (Jul 2017)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems* 30, pp. 5998–6008. Curran Associates, Inc. (2017)
- Váradi, T., Koeva, S., Yamalov, M., Tadić, M., Sass, B., Nitoń, B., Ogrodniczuk, M., Pezik, P., Barbu Mititelu, V., Ion, R., Irimia, E., Mitrofan, M., Păiş, V., Tufiş, D., Garabík, R., Krek, S., Repar, A., Rihtar, M., Brank, J.: The MARCELL legislative corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3761–3768. European Language Resources Association, Marseille, France (May 2020)
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems*. vol. 32, pp. 5753–5763. Curran Associates, Inc. (2019)
- Yang, Z.G., Perlaki, A., Laki, L.J.: Automatikus összefoglaló generálás magyar nyelvre bert modellel. XVI. Magyar Számítógépes Nyelvészeti Konferencia pp. 343–354 (2020)
- Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: Thirty-seventh International Conference on Machine Learning (2020)
- Zhang, X., Lapata, M., Wei, F., Zhou, M.: Neural latent extractive document summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 779–784. Association for Computational Linguistics, Brussels, Belgium (2018)

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 654–663. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)



# Twitter bot detection using deep learning

Adam Kenyeres<sup>1</sup>, György Kovács<sup>1</sup>

Luleå University of Technology  
adamkenyeres@gmail.com, gyorgy.kovacs@ltu.se

**Abstract.** Social media platforms have revolutionized how people interact with each other and how people gain information. However, social media platforms such as Twitter and Facebook quickly became the platform for public manipulation and spreading or amplifying political or ideological misinformation. Although malicious content can be shared by individuals, today millions of individual and coordinated automated accounts exist, also called bots which share hate, spread misinformation and manipulate public opinion without any human intervention. The work presented in this paper aims at designing and implementing deep learning approaches that successfully identify social media bots. Moreover we show that deep learning models can yield an accuracy of 0.9 on the PAN 2019 Bots and Gender Profiling dataset. In addition, the findings of this work also show that pre-trained models will be able to improve the accuracy of deep learning models and compete with Classical Machine Learning methods even on limited dataset.

## 1 Introduction

Social media bot detection has more than a decade of history. In the early days, most detection methods were based on supervised machine learning algorithms, based solely on the information of individual accounts. However, as time went by, bots became more sophisticated and coordinated and distinguishing bots from real accounts became almost impossible even for humans (Cresci et al., 2017). At this point, researchers turned to unsupervised machine learning algorithms which focused on groups of accounts rather than individual ones. The problem with the previous approaches is that they cannot be generalized as they are platform dependent. Recently the PAN 2019 Bots and Gender Profiling task has taken place, with the intention of using only the textual characteristics (Tweet) of social media accounts (Rangel and Rosso, 2019). This has the advantage of creating platform-independent solutions as these methods do not rely on specific social media characteristics. The work presented here will be based on the PAN 2019 Bots and Gender Profiling task. In the following sections we show that deep learning models can compete with Classical Machine Learning (CML) approaches even on a limited dataset. Moreover, this work also shows that pre-trained models improve the accuracy of bot detectors.

## 1.1 Motivation

Today, platforms such as Twitter have become tools for manipulating public opinion and spreading or amplifying political misinformation. However, large scale manipulation and sharing of politically polarized content is not only done by humans but by social media bots (Bessi and Ferrara, 2016).

Bots can even play a key role in political elections. Although at the time it was not clear, today we know that social media bots played a key role in the 2016 U.S. presidential elections (Bessi and Ferrara, 2016) by spreading divisive messages which may have contributed to Trump’s victory<sup>1</sup>. During the Brexit referendum in the United Kingdom, bots manipulated public opinion in favor of the country to leave the European Union (Howard et al., 2018). Bots also interfered with the French presidential elections in 2017 by spreading the Macron-Leaks campaign against Emmanuel Macron (Ferrara, 2017). As we can see bots actively shape public opinion, usually by negative campaigning (Howard et al., 2018) and influence individuals who may not be able to distinguish between human or bot-generated content. Therefore, countermeasures have to be taken in order to neutralize bots that exploit and spread misinformation by creating state-of-the-art bot detection methods.

## 1.2 Problem Definition

The project will be based on the PAN 2019 Bots and Gender Profiling task (Rangel and Rosso, 2019) where author profiling had to be solved by classifying Twitter feeds as bots or humans, based solely on the account’s textual form of the tweets without any additional information, such as tweet time, name, followers, accounts followed or profile picture. The dataset consisted of 6760 labeled English and 4800 Spanish users, each with 100 tweets.

This paper aims to research current state of the art approaches for bot detection and implement deep learning-based models. Moreover, we try to improve the existing solutions of the English dataset.

Furthermore, the paper aims at answering the following research questions:

1. Can deep learning-based approaches compete with classical machine learning methods on the limited dataset which was provided?
2. How would increasing the available data with data augmentation influence the performance of the models?
3. What is the effect of using pre-trained models and language representations on the results?

---

<sup>1</sup> <https://www.bloomberg.com/news/articles/2018-05-21/twitter-bots-helped-trump-and-brexit-win-economic-study-says>



## 2 Related Literature

There are different bot detection approaches. One could use the Socialbakers rule set or the Camisani-Calzolari rule set (CC) (Camisani-Calzolari, 2012), which assigns an account human and bot scores. Based on the two scores, the classification of an account can be given. Unsupervised methods learn patterns from untagged data, such as by clustering accounts. Therefore, they are effective at identifying coordinated and synchronized accounts (Cresci, 2020). Moreover, adversarial methods are also examined where researchers try to predict the evolution of bots before the evolution actually takes place. As this work focuses on supervised methods below we will discuss this type of approach in more detail.

**Supervised Methods** During our literature review, most papers we discovered were using supervised methods. Similarly, when Cresci (2020) reviewed 236 bot detectors published since 2010, they also found the majority of detectors to be based on supervised methods. Most papers cited in this paper use CML methods (which require features to be predefined) but deep learning methods (which do not require a predefined set of features (Yan et al., 2015)) are also researched (see Table 1). An example of CML model is BotoMeter which calculates a score of a given account by extracting more than 1000 features and using a random forest. Kudugunta and Ferrara (2018) developed deep learning models by creating LSTM models (Hochreiter and Schmidhuber, 1997). Moreover, the researchers also created Contextual LSTM models, which were trained on the tweet’s embedding along with user metadata. The LSTM model trained on tweets only, reached 95% accuracy, whereas the best contextual LSTM network achieved about 96%. The top 3 submissions of the PAN 2019 bots and gender profile task were also based on supervised approaches, moreover based on CML methods using feature extraction such as tweet length, number of URLs/mentions etc. Johansson (2019) combined a Logistic Regression (LR) with a Random Forest (RF), Fernquist (2019) trained a CatBoost classifier while Bacciu et al. (2019) used the output of a Support Vector Machine (SVM) and AdaBoost to train a Soft-Voting classifier. These submissions reached an accuracy of more than 0.94 and these solutions serve as a benchmark of this work.

Researcher(s)	Type	Algorithm(s)
<i>Yang et al. Yang et al. (2019)</i>	Supervised	RF
<i>Pozzana et al. Pozzana and Ferrara (2020)</i>	Supervised	RF, DT, ET, AB
<i>Varol et al. Varol et al. (2017)</i>	Supervised	RF, AB, LR, DT
<i>Beskow et al. Beskow and Carley (2018)</i>	Supervised	NB, LR, SVM, DT, RF
<i>Lee et al. Lee et al. (2011)</i>	Supervised	30 classifiers, best RF
<i>Almaatouq et al. Almaatouq et al. (2016)</i>	Supervised	ZeroR, BN, NB, LR, DT, RF
<i>Chu et al. Chu et al. (2012)</i>	Supervised	RF
<i>Kudugunta et al Kudugunta and Ferrara (2018)</i>	Supervised	LR, AB, LSTM

Table 1: Summary of the cited supervised based papers.

### 3 Data and Methods

In this section different data preprocessing steps and modelling approaches will be introduced for detecting Twitter bots.

#### 3.1 Data Preparation

The dataset of the PAN 2019 Bots and Gender Profiling task is split into three sets, namely training (2800 accounts), validation (1240 accounts) and test (2640 accounts) sets. Each account has 100 tweets and an account can belong to a human or a bot. In addition, we worked with the assumption that tweets written by bot accounts shall be classified as bot tweets. Moreover, all data sets are balanced making them ideal for machine learning algorithms.

The bot detectors which will be introduced shortly use GloVe (Pennington et al., 2014) (trained on tweets<sup>2</sup>) as a pre-trained word embedding. During the data preparation phase an important aspect is to preprocess the data in a similar manner as the authors of the word embedding. This would allow the predictive model to learn the semantic and syntactic meaning of words, which could also boost the detectors' performance.

During the data cleaning phase HTML tags, accented characters were removed. Mentions, hashtags, URLs, numbers, repeated characters and emojis were replaced by a tag, contractions were fixed, and special characters were surrounded by spaces. Moreover, the maximum tweet length was limited to 40 tokens in order to reduce the dimensionality of the corpus. Additional preprocessing steps had also been experimented such as using named entity recognition to replace words based on their entity, removing punctuations, stop words and lemmatizing the corpus. Unfortunately, these steps did not improve the accuracy; hence, they were omitted from the data preprocessing steps. After cleaning the data, the cleaned text is fed into the word embedding which creates the embedded representation of the tweets.

#### 3.2 Modeling Approaches

In this section 17 models will be introduced, which will be evaluated and discussed. As Tweets are posted in a chronological order we have decided to use Recurrent neural networks (RNNs) as they are well suited for sequence classification, thus the core of seven models are based on LSTM networks, four are based on BERT models (Devlin et al., 2018), one a combination of the two. In addition the work by Kovács et al. (2019) was also combined with the models introduced here. The detectors were written in Python and the deep learning models are built using Pytorch.

<sup>2</sup> <https://nlp.stanford.edu/data/glove.twitter.27B.zip>

### 3.3 Tweet Classification

**LSTM Tweet Classifier** The aim of this model is to classify whether a tweet was written by a human or by a bot and based on the individual predictions new models can be trained which can classify the accounts. From here on, this model will be referenced as Tweet Classifier. After preprocessing the data, the sequences of the token indexes are fed into the GloVe embedding, which creates the embedding representation of the tokens. Next, an LSTM model learns the patterns of the tweets and finally, two fully connected layers output the prediction. The model outputs the classification, the probability of a tweet being bot and the hidden states of the last LSTM layer which may hold a mix of content and non-content features. These outputs will be used by models which classify the individual accounts.

**Fine-tuning BERT** BERT has also been fine-tuned to classify tweets as bots or humans. We have decided to use the BertForSequenceClassification model from the hugging face library. The preprocessing steps described earlier are irrelevant as BERT requires special preprocessing which can be found in the paper by Devlin et al. (2018).

### 3.4 Account Classification

**Majority Vote** In order to classify an account one could simply feed the individual tweets to the Tweet Classifier, collect the predictions and, based on a majority vote, classify the account.

**Probability Based Prediction** A similar approach to the majority vote is to use tweet probabilities. In this scenario, the account's probability is calculated by averaging the sum of each of its tweets' probability. If the account probability is larger than 0.5, the account is considered a bot, otherwise a human.

**Combined Model** The output of the Tweet Classifier can also be the input of an account classifier model. For this approach, each tweet of an account is first fed into the Tweet Classifier model and the latent representation which holds information of the tweet is used as inputs for a second model to classify accounts. The individual hidden states are first collected, then used as inputs for the second model. In this model configuration, the Tweet classifier learns the features and most important characteristics of tweets corresponding to their origin (man-made or bot-generated). The hidden states represent this information; therefore the Tweet Classifier serves as an encoder where the model's output is disregarded.

The model consists of two LSTM layers and a fully connected layer which outputs the prediction for a given account (see Figure 1a). Moreover, during the implementation, 5-fold cross validation was used, where 5 independent models were trained on different partitions of the training set. The prediction of the models is based on the majority vote of the 5 predictions or based on the average of the 5 probabilities, creating an ensembled configuration.

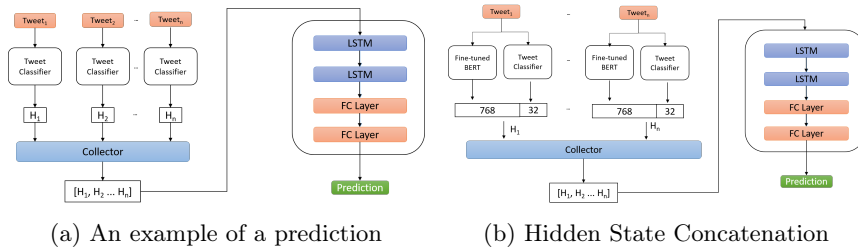


Fig. 1: Model Architectures

**Fine-tuning BERT** The same steps had been tried as described for the LSTM models that is, majority vote, probability-based account prediction and representing each tweet as the last hidden state of BERT and feeding them to the same combined LSTM model. Moreover, the LSTM based account classifier was also trained on the hidden states of a BERT model which was not fine-tuned.

The hidden states of BERT and the LSTM Tweet Classifier can also be combined. As can be seen on Figure 1b the information of the BERT model are concatenated and are fed into the LSTM Account Classifier. This approach allows the account classifier to learn from both the BERT and LSTM models which may supplement each other.

**Combining Deep Learning and CML Methods** Kovács et al. (2019) in their work extracted more than 160 features from the PAN 2019 Bots Dataset and trained an Adaboost model to classify individual accounts. They reached an accuracy of 0.89. In this work we extended this model by implementing a late fusion method which combines the classification and probabilities of the Adaboost model with the Account classifier model. In Addition the Adaboost model was also trained by adding the classification/probabilities of the LSTM Account classifier as an extra feature. The architecture can be seen on Figure 2.

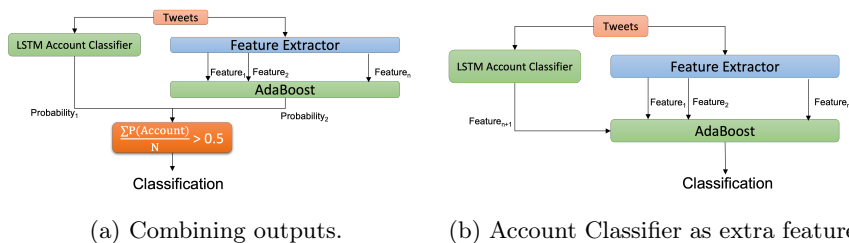


Fig. 2: Model Architectures

Model	LR	Hidden Size	Layers	Batch Size	FC Size	LSTM Dropout	FC Dropout	L2
<i>Tweet Classifier</i>	0.00100	32	1	64	128	0.400	0.100	0.1
<i>BERT</i>	0.00100	768	-	64	-	-	-	-

Table 2: Best hyper-parameters for tweet classification.

### 3.5 Data Augmentation

As the training set has fewer than 3k accounts, training the LSTM based account classifier network is quite challenging. In order to overcome this issue data augmentation was used on the hidden states of the accounts. Let  $A$  be the set of  $\{T_1, T_2 \dots T_n\}$  representing an account with  $n$  tweets and the set of the hidden states  $H = \{H_1, H_2 \dots H_n\}$ . We define an augmentation function  $f(H, l)$ , where  $l \leq n$ , which returns all  $n-l$  consecutive subsets of  $H$  of length  $l$ . As Tweets form a chronological sequence, subsets of tweets of an account should also form a valid account. This augmentation method significantly increased the availability of the training data, which should also increase the accuracy of the model.

### 3.6 Hyper-parameter Selection

For hyper-parameter optimization Ray-Tune (Liaw et al., 2018) was used using Tree-structured Parzen Estimator Approach (TPE) as the search algorithm and Asynchronous SHA (ASHA) to early terminate bad configurations. In order to find the best configuration 30 trials were executed. Table 2 and 3 shows the hyper-parameters of each model. It is also important to highlight that an additional preprocessing step lies before the execution of the LSTM account classifier which serves to normalize the hidden states of the tweets.

Model	LR	Hidden Size	Layers	Batch Size	FC Size	LSTM Dropout	FC Dropout	L2
<i>Account Classifier (LSTM, LSTM)</i>	0.00800	32	2	64	128	0.499	0.900	9.1e-08
<i>Account Classifier + Aug. (LSTM, LSTM)</i>	0.00890	64	2	64	32	0.466	0.055	1.1e-06
<i>Account Classifier (BERT, LSTM)</i>	0.00040	128	2	64	512	0.088	0.100	6.7e-06
<i>Account Classifier + Aug. (BERT, LSTM)</i>	0.00329	512	2	64	32	0.198	0.500	2.8e-06
<i>Account Classifier (BERT w/o fine-tuning, LSTM)</i>	0.00017	256	1	32	32	0.402	0.176	3.9e-07
<i>Account Classifier (BERT + LSTM, LSTM)</i>	0.00171	128	2	64	64	0.4661	0.1055	9.9e-06

Table 3: Best hyper-parameters for account classification.

Model	Accuracy	F1	Precision	Recall
<i>LSTM Tweet Classifier</i>	0.753	0.753	0.753	0.754
<b><i>BERT Tweet Classifier</i></b>	<b>0.828</b>	<b>0.800</b>	<b>0.937</b>	<b>0.700</b>

Table 4: Tweet classification results.

## 4 Results and Discussion

All 17 models designed in this work were evaluated on the test set and the results can be seen on Table 4, 5 and 6. Moreover, submissions to the competition were evaluated based on the accuracy of the models, therefore we will also report based on this measure.

The fine-tuned BERT model classified the origin of tweets with the highest accuracy, yielding an accuracy of 0.828. Following, the LSTM model reached an accuracy of 0.753 (see Table 4).

On the other hand, the single LSTM model performed much better than the fine-tuned BERT model during account classification using majority and probability based account predictions, beating the best BERT model by almost 3% (see Table 5).

As can be seen on Table 6, from the combined deep learning models the combined LSTM model yielded the highest accuracy of 0.892. On the other hand, combining the LSTM Account Classifier with the Adaboost model surpassed the best deep learning model with an accuracy of 0.9.

Augmenting the input data on average resulted in a 1% decrease in the accuracy (see Table 6). Although, data augmentation typically increases the accuracy, there can be several reasons why the models did not perform better. One reason is that during the hyper-parameter optimization, the search space was ill-defined and the optimization algorithm did not explore configurations that reach a high performance. On the other hand, maybe the augmented data was too complex and the models could not learn all the patterns. This can explain why the training accuracy was not improving while the validation accuracy was during training.

An interesting point is that although the fine-tuned BERT model predicts individual tweets better (+6%) than the LSTM based tweet classifier, the majority and probability-based account predictions are worse than the majority or probability predictions of the LSTM tweet classifier. This can be because of two

Model	Accuracy	F1	Precision	Recall
<i>Tweet Classifier - Majority Vote</i>	0.873	0.864	0.930	0.806
<b><i>Tweet Classifier - Probability based pred.</i></b>	<b>0.878</b>	<b>0.870</b>	<b>0.936</b>	<b>0.811</b>
<i>BERT - Majority Vote</i>	0.849	0.826	0.975	0.717
<i>BERT - Probability based pred.</i>	0.852	0.830	0.976	0.721

Table 5: Account classification results based on single models.

<b>Model</b>	<b>Accuracy</b>	<b>F1</b>	<b>Precision</b>	<b>Recall</b>
<i>Account Classifier - Majority Vote (LSTM, LSTM)</i>	0.891	0.890	0.895	0.886
<i>Account Classifier - Probability based pred. (LSTM, LSTM)</i>	0.892	0.893	0.881	0.906
<i>Account Classifier + Augmentation (LSTM, LSTM)</i>	0.880	0.880	0.878	0.884
<i>Account Classifier - Majority Vote (BERT, LSTM)</i>	0.836	0.828	0.872	0.787
<i>Account Classifier - Probability based pred. (BERT, LSTM)</i>	0.838	0.830	0.874	0.790
<i>Account Classifier + Augmentation (BERT, LSTM)</i>	0.835	0.821	0.903	0.753
<i>Account Classifier (BERT w/o Fine-tuning, LSTM)</i>	0.806	0.793	0.853	0.740
<i>Account Classifier (BERT + LSTM, LSTM)</i>	0.838	0.820	0.916	0.743
<i>Account Classifier + Adaboost</i>	0.878	0.880	0.890	0.870
<i>Account Classifier prob. Feature + Adaboost</i>	0.891	0.890	0.890	0.890
<b><i>Account Classifier pred. Feature + Adaboost</i></b>	<b>0.900</b>	<b>0.900</b>	<b>0.900</b>	<b>0.900</b>

Table 6: Account classification results based on multiple models.

factors. First, the BERT model has a low recall which means many human accounts are classified as bots; second, the distribution of the incorrect tweets are spanned across more accounts than the LSTM based classifier, which can have a significant impact during the account classification. Moreover, the LSTM account classifier model performed poorer with the BERT hidden states. Again, this could be because of the incorrect hyper parameter configurations of the LSTM account classifier, or the hidden representation of the tweets are too similar; that is, the LSTM model cannot differentiate humans and bots.

It is also important to highlight that the LSTM based tweet classifier classifies accounts just 1% lower than the highest performing model introduced in this work. However, it can be trained in a fraction of the training time of a complex model such as fine-tuning BERT or the combined LSTM models.

#### 4.1 Comparison of the best results of the PAN 2019 Author Profiling task

The best results and the majority of the submissions solved the task by CML algorithms and used feature extraction. On the other hand the work presented here was based on deep learning models. Therefore, it is hard to compare the results. The top 3 submissions achieved more than 0.94. In summary it can be stated that the work presented is not superior to the benchmarked solutions and further improvements should be made to improve the performance of deep learning methods. Having said that, some submissions to the competition were

based on deep learning approaches. Onose et al. (2019) created Hierarchical Attention Networks and reached an accuracy of 0.89, while some submissions were based on LSTM models (Rangel and Rosso, 2019) but the accuracy was 0.87. As we can see the models introduced in this work outperformed the deep learning approaches that were reviewed and cited in this paper. Nevertheless, the work presented in this paper should be further improved and more research shall be made in this research area.

## 4.2 Reflection on the research questions

This paper aims at addressing three research questions, which are described in the problem definition. Referring to RQ1 it can be argued that deep learning based methods, especially LSTM based models, achieved good but not superior results compared to classical approaches.

For RQ2, data augmentation did not improve the accuracy because of the already mentioned reasons, such as by incorrectly specifying the hyper-parameter search space or incorrect implementation. Nevertheless, data augmentation in the NLP domain is challenging and an area that needs to be researched.

Regarding RQ3, it is clear that pre-trained models and language representations do improve the results. The fine-tuned BERT model beat the LSTM based tweet classification model by almost 7%.

## 5 Conclusion and Future work

The bot detectors introduced here successfully solve bot detection with an accuracy of 0.90. Although it can be argued, the presented deep learning models are not better than traditional classical machine learning methods. Having said that, the data set was quite limited in size and on large scale data the outcome may have been different. In addition, to the best of our knowledge, the presented architecture of the combining LSTM/BERT models with another LSTM model for classifying accounts as well as combining the above with an Adaboost model have not been researched before. Therefore, this work can also serve as a baseline for future improvements for deep learning and hybrid (deep learning and classical machine learning) based approaches.

### 5.1 Further development

The work presented here can be further improved in several areas. In the future, text augmentation should be applied, such as the approach by Wei and Zou (2019), which combines synonym replacement, random insertion, random swap and random deletion. Architectural changes could also be made to the models such as by using different loss functions, optimizers, replacing LSTMs by GRUs or by using different word embeddings.



Another approach could be to improve the hidden representation of the models by using Siamese neural networks (Chopra et al., 2005). These networks are trained to predict whether two input samples are the same or not by calculating the similarity of the inputs. These networks have shown great results in computer vision in the area of face verification, but they could also be used to improve the bot detection models of this work. If a Siamese network were to be used, which is trained on tweets of humans and bots, the model would be forced to learn the characteristics of bots and humans in order to differentiate them. Therefore, the hidden representations of the tweets would also be more similar if they came from the same account or same type of accounts such as a bot or a human. These hidden states could then be used by the account classification model, which was introduced earlier.

## References

- Almaatouq, A., Shmueli, E., Nouh, M., Alabdulkareem, A., Singh, V.K., Alsaleh, M., Alarifi, A., Alfaris, A., et al.: If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. *International Journal of Information Security* 15(5), 475–491 (2016)
- Bacciu, A., La Morgia, M., Mei, A., Nemmi, E.N., Neri, V., Stefa, J.: Bot and gender detection of twitter accounts using distortion and lsa. In: *CLEF (Working Notes)* (2019)
- Beskow, D.M., Carley, K.M.: Bot-hunter: a tiered approach to detecting & characterizing automated activity on twitter. In: *Conference paper. SBP-BRiMS: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. vol. 3, p. 3 (2018)
- Bessi, A., Ferrara, E.: Social bots distort the 2016 us presidential election online discussion. *First Monday* 21(11-7) (2016)
- Camisani-Calzolari, M.: Analysis of twitter followers of the us presidential election candidates: Barack obama and mitt romney. (Online). <http://digitalevaluations.com> (2012)
- Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. vol. 1, pp. 539–546. IEEE (2005)
- Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on dependable and secure computing* 9(6), 811–824 (2012)
- Cresci, S.: A decade of social bot detection. *Communications of the ACM* 63(10), 72–83 (2020)
- Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: The paradigm-shift of social spambots: Evidence, theories, and tools for the arms

- race. In: Proceedings of the 26th international conference on world wide web companion. pp. 963–972 (2017)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- Fernquist, J.: A four feature types approach for detecting bot and gender of twitter users. In: CLEF (Working Notes) (2019)
- Ferrara, E.: Disinformation and social bot operations in the run up to the 2017 french presidential election. arXiv preprint arXiv:1707.00086 (2017)
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* 9(8), 1735–1780 (1997)
- Howard, P.N., Woolley, S., Calo, R.: Algorithms, bots, and political communication in the us 2016 election: The challenge of automated political communication for election law and administration. *Journal of information technology & politics* 15(2), 81–93 (2018)
- Johansson, F.: Supervised classification of twitter accounts based on textual content of tweets. In: CLEF (Working Notes) (2019)
- Kovács, G., Balogh, V., Mehta, P., Shridhar, K., Alonso, P., Liwicki, M.: Author profiling using semantic and syntactic features. In: CLEF (Working Notes) (2019)
- Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. *Information Sciences* 467, 312–322 (2018)
- Lee, K., Eoff, B., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on twitter. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 5 (2011)
- Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 (2018)
- Onose, C., Nedelcu, C.M., Cercel, D.C., Trausan-Matu, S.: A hierarchical attention network for bots and gender profiling. In: CLEF (Working Notes) (2019)
- Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), <http://www.aclweb.org/anthology/D14-1162>
- Pozzana, I., Ferrara, E.: Measuring bot and human behavioral dynamics. *Frontiers in Physics* 8, 125 (2020)
- Rangel, F., Rosso, P.: Overview of the 7th author profiling task at pan 2019: bots and gender profiling in twitter. In: Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop (2019)
- Varol, O., Ferrara, E., Davis, C., Menczer, F., Flammini, A.: Online human-bot interactions: Detection, estimation, and characterization. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 11 (2017)
- Weì, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
- Yan, L.C., Yoshua, B., Geoffrey, H.: Deep learning. *nature* 521(7553), 436–444 (2015)

XVIII. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2022. január 27–28.

Yang, K.C., Varol, O., Davis, C.A., Ferrara, E., Flammini, A., Menczer, F.:  
Arming the public with artificial intelligence to counter social bots. *Human  
Behavior and Emerging Technologies* 1(1), 48–61 (2019)



# Forensic Authorship Classification by Paragraph Vectors of Speech Transcriptions

Dávid Sztahó<sup>1</sup>, András Beke<sup>1</sup>, György Szaszák<sup>1</sup>, Attila Fejes<sup>2</sup>

<sup>1</sup> Budapest University of Technology and Economics,  
Magyar tudósok körútja 2., 1117 Budapest, Hungary  
sztaho.david@vik.bme.hu, beke.andras@lsa.tmit.bme.hu,  
szaszak.gyorgy@vik.bme.hu

<sup>2</sup> University of Public Service Doctoral School of Law Enforcement,  
Ludovika tér 2., 1083 Budapest, Hungary  
fejes.attila@nbsz.gov.hu

**Abstract:** In forensic comparison, document classification techniques are used mainly for authorship classification and author profiling. In the present study, we aim to introduce paragraph vector modelling (by Doc2Vec) into the likelihood-ratio framework paradigm of forensic evidence comparison. Transcriptions of spontaneous speech recording are used as input to paragraph vector extraction model training. Logistic regression models are trained based on cosine distances of paragraph vector pairs to predict the same and different author origin probability. Results are evaluated according to different speaking styles (transcriptions of speech tasks available in the dataset).  $C_{lr}$  and equal error rate values (lowest ones are 0.47 and 0.11, respectively) show that the method can be useful as a feature for forensic authorship comparison and may extend the voice comparison methods for speaker verification.

## 1 Introduction

In forensic comparison practice, a widely spreading automatic evaluation method is getting more and more accepted. The method is called likelihood ratio (LR) framework (Morrison, 2011; Saks and Koehler, 2005). It emerges in numerous topics, most widely known in DNA identification. The key point of the paradigm is to resolve the question: how characteristics are given measures to an individual or to a population. In the course of actual application, it considers two hypotheses: “What is the probability that the sample in question comes from the suspect?” And the so-called counter-hypothesis: “What is the probability that the sample in question comes from another person randomly selected from a given population?”. Based on these, the probability of the evidence can be written:

$$LR = \frac{p(E|H_{so})}{p(E|H_{do})} \quad (1)$$

where LR is the likelihood-ratio, E is the evidence,  $H_{so}$  is the hypothesis of same-origin subjects,  $H_{do}$  is the hypothesis of different-origin subjects.

In forensics, natural language processing (NLP) methods target mainly authorship classification (Khonji et al., 2021) and author profiling in which features of a person such as gender, age, and cultural characteristics are classified. The basic method for both methods is to extract single features (such as vocabulary size, word frequency) from a document and apply a machine learning method (such as support vector machines) for classification (Adame-Arcia et al., 2017; Estival et al., 2007; Hsieh et al., 2018). As deep learning methods emerged in NLP, two main document classification methods became more widespread: Doc2Vec and BERT language modelling. Doc2Vec (Le and Mikolov, 2014) is an extension of Word2Vec (Mikolov et al., 2013) and is used basically for document classification and document similarity scoring. It was also applied for author profiling (Markov et al., 2016) detection certain speaker characteristics (age, gender) but not the speaker ids directly. This method replaced hand-crafted feature extraction by automatic modelling trained on a large corpus. In this method the Word2Vec method is extended by a vector called document vector (Fig. 1) that is trained along with the vector representation of words (Word2Vec, (Mikolov et al., 2013)). This document vector will contain an accumulated information about the paragraphs for a single document.

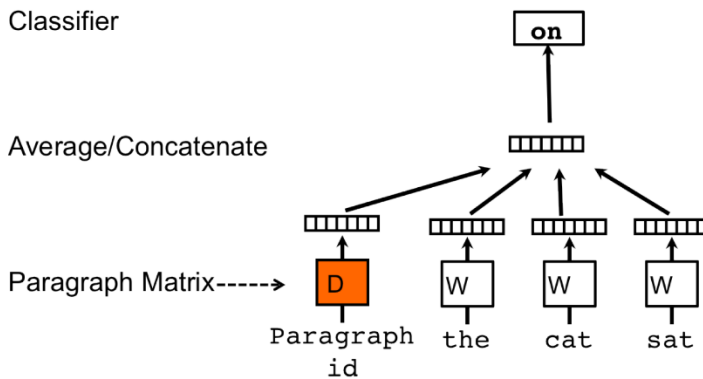


Fig. 1. Deep learning architecture for Doc2Vec (Le and Mikolov, 2014)

The present work aims to insert document classification (by Doc2Vec) into the LR framework of forensic authorship verification (as a text-based speaker recognition) based on transcriptions of spontaneous speech. To our knowledge, this type of LR authorship verification based on paragraph vectors has not been done before. The transcriptions are created based on samples of the ForVOICE project (Beke et al., 2021) containing free dialogues and monologues. Logistic regression models were created for same and different speaker probabilities (Eq. 1). The results are evaluated by  $C_{in}$  and equal error rate scores and plotted on tippet plots common in forensic evidence comparison. The method can be used to combine text-based features and voice-based features in order to improve overall speaker verification results.

## 2 Methods

### 2.1 Transcriptions

The trained models were evaluated using the ForVOICE dataset (Beke et al., 2021). It contains spontaneous speech of three different speaking styles: (1) free dialogue (~10 minutes), (2) guided dialogue (~8 minutes) and (3) monologues (~3 minutes). 80 speakers were recorded twice (with 2 weeks interval apart) and transcriptions of all speech samples were created manually. 60 speakers were randomly selected for model training and the remaining 20 speakers were used for evaluation.

Beside the 60 speakers of the ForVOICE dataset, transcriptions of spontaneous speech were used from the BEA (Gósy et al., 2012) and the HuComTech (Szekrényes, 2014) datasets. The total number of words and paragraphs used is shown in Table 1.

Training data were created in two ways from transcriptions: (1) splitting every sample transcription to word lengths of 200 and using 100 words overlap (multiple fixed length paragraphs for a speech sample) and (2) using every sample transcription as a single training paragraph (a single variable length paragraph for each speech sample). In the former case, multiple paragraphs are available for a given recording. For example, the original transcription of the first monologue recording of a speaker is split into multiple overlapping parts with 200 word lengths and is used in the experiments. In the latter case, the original total transcription is used. Same splitting is done for the BEA and the HuComTech transcriptions.

Table 1: Number of words and paragraphs in corpora used

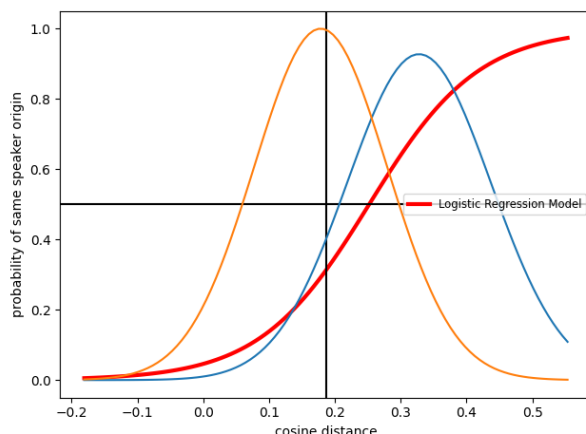
Dataset	#speakers	#words (text lengths)	#paragraphs with splitting	#paragraphs without splitting
Train: ForVOICE	60	163874	1813	360
Train: ForVOICE + BEA + HuComTech	182	352238	3865	773
Test: ForVOICE	20	127208	1407	120

### 2.2 Doc2Vec modelling

Doc2Vec (Le and Mikolov, 2014) approach is implemented by the Gensim Python package (version 4.1.2). In this method the Word2Vec method is extended by a vector called document vector that is trained along with the vector representation of words (Word2Vec, (Mikolov et al., 2013)). This document vector will contain an accumulated information about the paragraphs for a single document. Vocabulary for the Doc2Vec method is built from the training dataset. Due to the limited training corpus size available from spontaneous speech, this results that 37% and 27% of the test vocabulary

entries were not covered by train vocabulary entries in case of ForVOICE and ForVOICE + BEA + HuComTech cases, respectively.

Doc2Vec models were built using different paragraph vector length (20, 100, 200) and training epoch number (40, 100, 200) using speaker ids. The maximum distance between the current and predicted word within a sentence was set to 12 based on (Markov et al., 2016). After the model is created, paragraph vectors extracted on the evaluation speaker set were used for same speaker and different speaker origin modelling by logistic regression (LR, implemented by Python sklearn package). LR models are built using cosine distances of extracted paragraph vectors as input using the target variable if a vector pair is of same or different speaker origin. The output of the LR model is the probability of the same speaker decision. This enables the calculation of Eq. 1. Fig. 2 shows a sample of a trained LR model. Distribution of same and different origin vector pairs are shown in yellow and blue, respectively.



**Fig. 2.** A trained LR model sample. Yellow and blue lines show the distributions of paragraph vector pairs of same and different author origin.

### 2.3 Evaluation

Models (Doc2Vec and LR) were created on the training dataset (training speaker set of ForVOICE with and without augmentation by BEA and HuComTech) and evaluated using the test speaker set of ForVOICE. Evaluation metrics are equal error rate (EER) of author verification (EER is the level where false acceptance rate and false rejection rate are equal, commonly used in biometric security systems) and log-likelihood-ratio cost ( $C_{llr}$ , Eq. 2) (Van Leeuwen and Brümmer, 2007), defined as

$$C_{llr} = \frac{1}{2} \left( \frac{1}{N_{so}} \sum_{i=1}^{N_{so}} \log_2 \left( 1 + \frac{1}{LR_{so_i}} \right) + \frac{1}{N_{do}} \sum_{j=1}^{N_{do}} \log_2 \left( 1 + LR_{do_j} \right) \right)$$

where  $N_{so}$  and  $N_{do}$  are the number of same-origin and different-origin comparisons and  $LR_{so}$  and  $LR_{do}$  are the likelihood ratios derived from same-origin and different-origin comparisons.  $C_{llr}$  is a function measuring the balance of LR scores of same-origin and



different-origin comparisons measured using all possible same-origin and different-origin vector pair combinations. Ideal same-origin and different-origin comparisons have  $\log LR > 0$  and  $\log LR < 0$ , respectively. Incorrect (not as ideal as the mentioned inequalities) produce a higher  $C_{lr}$ . The better the performance of a forensic comparison system, the more correct  $LR$  values are produced, the lower  $C_{lr}$  is achieved, supplying the evidence magnitude.

### 3 Results

Doc2Vec and LR models were created for various speech task types to evaluate if different speaking styles (domains) affect authorship verification performance. Four cases were considered: using texts from all speech tasks altogether and from each single tasks individually. Doc2Vec models were trained in two variations: texts of the total ForVOICE corpus (of speakers selected for training) with and without augmentation by BEA and HuComTech corpora. Logistic regression models were always trained on texts of the given speech styles, while Doc2Vec models were created without speech task filters. Tables 2 and 3 contain  $C_{lr}$  and EER values (without and with splitting paragraphs into word lengths of 200, respectively) calculated on the test speaker set of the ForVOICE corpus for all cases, paragraph vector lengths and training epoch numbers. Best cases for each speech tasks and training corpora based on the  $C_{lr}$  and EER values are highlighted. EER and  $C_{lr}$  values should not necessarily be perfectly correlated, so there are occurrencies where a case with higher  $C_{lr}$  achieves lower EER and vice versa. A higher  $C_{lr}$  means a shift in the threshold of same-different origin speaker decision from the optimal 0 value. It may indicate a worse generalization level.

It is clear from the tables that LR models trained on single speaking styles achieve better performance than using all tasks at once. There seems to be no significant difference between splitted and non-splitted paragraphs and also training corpora augmentation shows no real difference. The best  $C_{lr}$  values (and EERs) for speaking styles are: 'all' - 0.87 (0.35); 1 - 0.64 (0.2); 2 - 0.47 (0.11); 3 - 0.65 (0.15). Tippet plots, showing the proportion of correctly identified same and different author origin (commonly used plots in forensic comparison) of the aforementioned results are shown in Fig. 3. Blue and yellow lines show the proportion of correctly identified same and different author origin vector pairs as function of  $\log LR$  score thresholds. While the blue line measures the proportion of vector pairs belonging to different origin below the given  $\log LR$  threshold, the yellow line depicts the proportion of the vector pairs belonging to same origin above the given  $\log LR$  threshold. EER is measured at the crossing of the two lines.

Table 2: Result without splitting paragraphs

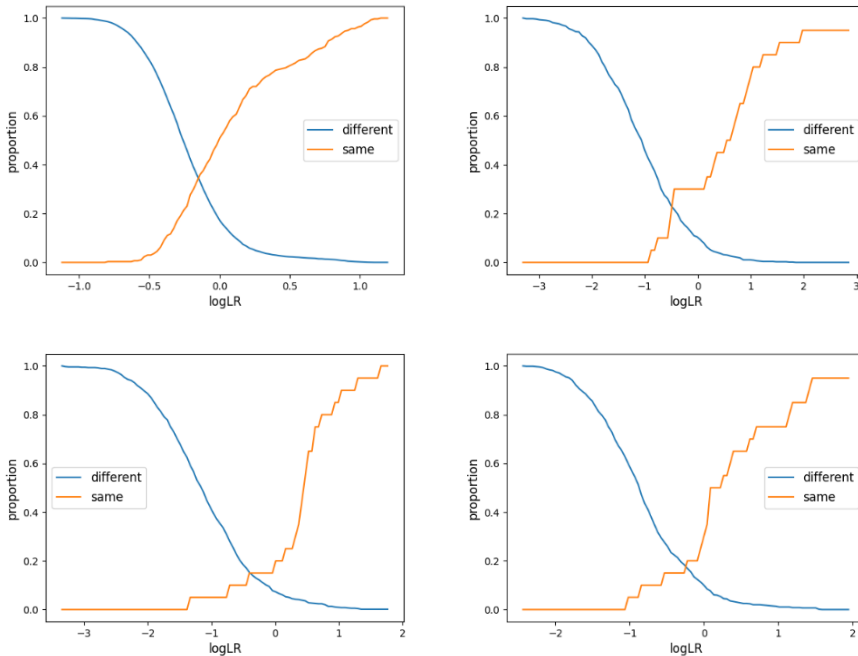
training corpus	task	vector	epoch						
			40		100		200		
			eer	C <sub>lr</sub>	eer	C <sub>lr</sub>	eer	C <sub>lr</sub>	
ForVOICE	all	20	0,4	0,94	0,44	0,97	0,44	0,97	
		100	<b>0,35</b>	<b>0,87</b>	0,36	0,94	0,41	1,03	
		200	<b>0,35</b>	<b>0,87</b>	<b>0,34</b>	<b>0,93</b>	0,39	1,09	
	1	20	0,3	1,1	0,25	0,93	0,31	1,01	
		100	0,3	0,87	0,25	0,86	0,3	0,94	
		200	0,2	0,79	<b>0,2</b>	<b>0,64</b>	0,25	1,04	
	2	20	0,25	0,79	0,2	0,71	0,17	0,59	
		100	0,19	0,72	<b>0,11</b>	<b>0,6</b>	0,15	0,62	
		200	0,2	0,8	<b>0,1</b>	<b>0,6</b>	0,14	0,86	
	3	20	0,25	0,84	0,31	0,99	0,3	0,94	
		100	0,2	0,77	0,2	0,78	0,26	0,89	
		200	<b>0,15</b>	<b>0,69</b>	0,2	0,74	0,25	1,23	
	ForVOICE + BEA + HuComTech	all	20	0,41	0,95	0,44	0,98	0,45	0,98
			100	0,34	0,89	0,39	0,94	0,39	0,96
			200	<b>0,33</b>	<b>0,88</b>	0,37	0,95	0,41	1,05
1		20	0,25	0,78	0,3	0,89	0,3	0,87	
		100	<b>0,2</b>	<b>0,78</b>	0,25	0,83	0,25	0,77	
		200	<b>0,25</b>	<b>0,69</b>	<b>0,2</b>	<b>0,95</b>	0,25	1,16	
2		20	0,23	0,69	0,3	0,77	0,31	0,76	
		100	0,2	0,71	<b>0,15</b>	<b>0,63</b>	0,2	0,76	
		200	0,25	0,7	0,15	0,7	0,15	0,93	
3		20	0,25	0,76	0,22	0,83	0,25	0,9	
		100	0,2	0,72	0,2	0,81	0,25	0,74	
		200	<b>0,15</b>	<b>0,66</b>	0,17	1,03	0,23	1,28	

Table 3: Results with splitting paragraphs

training corpus	task	vector	epoch						
			40		100		200		
			eer	$C_{lr}$	eer	$C_{lr}$	eer	$C_{lr}$	
ForVOICE	all	20	0,42	0,96	0,45	0,98	0,44	0,98	
		100	0,35	0,9	0,41	0,96	0,41	0,99	
		200	<b>0,33</b>	<b>0,89</b>	0,38	0,97	0,41	1,03	
	1	20	0,3	0,77	0,25	0,82	0,32	0,77	
		100	0,29	0,87	0,3	0,75	0,31	0,86	
		200	0,25	0,75	<b>0,2</b>	<b>0,68</b>	0,26	0,88	
	2	20	0,31	0,93	0,25	0,66	0,25	0,71	
		100	0,15	0,67	<b>0,11</b>	<b>0,47</b>	0,1	0,55	
		200	0,12	0,58	0,15	0,55	0,15	0,62	
	3	20	0,26	0,92	0,31	0,92	0,35	1,01	
		100	0,24	0,71	0,2	0,68	0,25	0,77	
		200	<b>0,15</b>	<b>0,65</b>	0,15	0,73	0,23	0,89	
	ForVOICE + BEA + HuComTech	all	20	0,44	0,98	0,46	0,99	0,45	0,99
			100	0,36	0,91	0,4	0,95	0,41	0,97
			200	<b>0,34</b>	<b>0,9</b>	0,38	0,95	0,41	0,99
1		20	0,3	0,85	0,25	0,72	0,3	0,84	
		100	0,25	0,82	0,25	0,81	<b>0,27</b>	<b>0,7</b>	
		200	<b>0,21</b>	<b>0,79</b>	0,25	0,83	0,29	0,87	
2		20	0,3	0,76	0,3	0,84	0,37	0,85	
		100	0,15	0,65	0,2	0,72	<b>0,15</b>	<b>0,57</b>	
		200	<b>0,14</b>	<b>0,59</b>	0,2	0,62	<b>0,15</b>	<b>0,57</b>	
3		20	0,3	0,93	0,3	1,03	0,29	1,02	
		100	0,2	0,71	0,2	0,7	0,25	0,81	
		200	<b>0,15</b>	<b>0,66</b>	0,2	0,76	0,2	0,82	

## 4 Discussion and Conclusion

Based on the results, it can be stated that the paragraph vector modelling can be used in a forensic authorship verification framework. However, it is not clear right now as to what vector length to choose. It seems that longer vector lengths perform better. Increasing the number of epochs also increases the  $C_{lr}$  indicating an overfitting effect. This can be overcome by using a technique such as early stopping in which a development dataset is used to measure performance during training. Thus, a desired generalization ability can be set.



**Fig. 3.** Tippet plots of best models for speech tasks. From top left to bottom right: ‘all’, 1, 2 and 3. Blue and yellow lines show the proportion of same and different author origin vector pairs as function of logLR score thresholds, respectively.

It is also clear that domain specific logistic regression models achieve lower  $C_{llr}$  and equal error rate values. It may be due to a speaking style mismatch of the speech tasks investigated in the present work.

Comparing current results to related works is hard due to the corpora and target mismatch. In (Kaur et al., 2020), social network posts are identified if they originate from the same user or not. In their work, carefully crafted textual features are used. The current method presented here may serve as an additional feature extending current feature sets, not only in text-based authorship classification/verification but also in forensic voice comparison.

Creating ForVOICE is currently in its final step. Current results and the method will be extended using the final dataset which includes samples of 120 speakers. The final goal is to combine text-based (traditional hand-crafted and deep learning based) features and voice-based features, such as x-vectors, into a final method and evaluate it on a dataset matching forensic needs. Beside Doc2Vec, BERT modelling would also be useful to investigate as a document classification method.

## Acknowledgement

The work was funded by project no. FK128615, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK\_18 funding scheme.

## References

- Adame-Arcia, Y., Castro-Castro, D., Bueno, R.O., Muñoz, R., 2017. Author profiling, instance-based similarity classification. Notebook for PAN at CLEF2.
- Beke, A., Szaszák, G., Sztahó, D., 2021. FORvoice 120+: magyar nyelv\Hu utánkövetéses adatbázis kriminalisztikai célú hangösszehasonlításra.
- Estival, D., Gaustad, T., Pham, S.B., Radford, W., Hutchinson, B., 2007. Author profiling for English emails, in: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics. p. 272.
- Gósy, M., Gyarmathy, D., Horváth, V., Grácsi, T.E., Beke, A., Neuberger, T., Nikléczy, P., 2012. BEA: Beszélt nyelvi adatbázis.
- Hsieh, F., Dias, R., Paraboni, I., 2018. Author profiling from facebook corpora, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).
- Kaur, R., Singh, S., Kumar, H., 2020. TB-CoAuth: Text based continuous authentication for detecting compromised accounts in social networks. *Applied Soft Computing* 97, 106770.
- Khonji, M., Iraqi, Y., Mekouar, L., 2021. Authorship Identification of Electronic Texts. *IEEE Access* 9, 101124–101146.
- Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: International Conference on Machine Learning. PMLR, pp. 1188–1196.
- Markov, I., Gómez-Adorno, H., Posadas-Durán, J.-P., Sidorov, G., Gelbukh, A., 2016. Author profiling with doc2vec neural network-based document embeddings, in: Mexican International Conference on Artificial Intelligence. Springer, pp. 117–131.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems. pp. 3111–3119.
- Morrison, G.S., 2011. Measuring the validity and reliability of forensic likelihood-ratio systems. *Science & Justice* 51, 91–98.
- Saks, M.J., Koehler, J.J., 2005. The coming paradigm shift in forensic identification science. *Science* 309, 892–895. <https://doi.org/10.1126/science.1111565>
- Székrenyess, I., 2014. Annotation and interpretation of prosodic data in the hucomtech corpus for multimodal user interfaces. *Journal on Multimodal User Interfaces* 8, 143–150.
- Van Leeuwen, D.A., Brümmer, N., 2007. An introduction to application-independent evaluation of speaker recognition systems, in: Speaker Classification I. Springer, pp. 330–353. [https://doi.org/10.1007/978-3-540-74200-5\\_19](https://doi.org/10.1007/978-3-540-74200-5_19)



## Absztraktív összefoglalás arab nyelvre

Kahla Mram<sup>1</sup>, Yang Zijian Győző<sup>1,2,3</sup>

<sup>1</sup>Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar

<sup>2</sup>MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport

1083 Budapest, Práter u. 50/a.

{kahla.mram, yang.zijian.gyozo}@itk.ppke.hu

<sup>3</sup>Nyelvtudományi Kutatóközpont

1068 Budapest, Benczúr u. 33.

yang.zijian.gyozo@nytud.hu

**Kivonat** Kutatásunkban arab nyelvre tanítunk különböző absztraktív összefoglaló modelleket. A jelen tanulmány a kutatásunk jelenlegi fázisát mutatja be. Arab nyelvre az absztraktív összefoglalás területén kevés kutatás történt, ezért korábbi kutatásunk során első feladatként saját adatot kellett gyűjteni. Adatgyűjtés után sikeresen finomhangoltunk különböző enkóder-dekóder architektúrájú transzformer modelleket. Kísérleteinkben kipróbáltuk a PreSumm és a többnyelvű mBART módszereket. A PreSumm módszerrel ezen a területen „state of the art” eredményt értünk el. Jelen tanulmány ezt a kutatási sorozatot folytatja. Kutatásunk során saját egynyelvű és többnyelvű BART modell tanításával kísérleteztünk, valamint az mT5 modellt próbáltuk arab összefoglaló generálásra finomhangolni. Kísérletünk során korlátozott mennyiségű adattal kísérleteztünk, célunk az volt, hogy megvizsgáljuk ezen módszerek alkalmazhatóságát. Kutatásunkkal ezért várakozásunknak megfelelően nem tudtuk felülmúlni a korábban elért legjobb eredményünket. Azonban így is versenyképes eredményeket tudtunk elérni, amelyek további kutatásoknak adnak teret, ez azonban nagyobb mennyiségű adat és infrastruktúra előfeltételt is megkövetel.

**Kulcsszavak:** arab absztraktív összefoglalás, BART, mBART, PreSumm, mT5

### 1. Bevezetés

Az összefoglaló generálás a nyelvtechnológia egyik kiemelt feladata lett. Kétféle összefoglaló módszert különböztetünk meg. Az első az extraktív, amikor a meglévő szövegből kiválasztjuk azokat a szövegrészeket, amelyek összefoglalóként funkcionálhatnak, ez gyakorlatilag egy osztályozási feladat. A másik módszer az absztraktív, amikor az emberhez hasonlóan a modell egy adott szövegből önállóan megfogalmaz egy összefoglalót. Az utóbbi módszerrel a modell olyan kifejezéseket is használhat, amelyek nem szerepeltek az eredeti szövegben. Kutatásunk elsősorban az arab nyelvre koncentrál. Az arab beszélt nyelvnek számos dialektusa van, de írás szempontjából erősen szabványosított. Ez nagyban segít az arab

írással szövegek feldolgozásában, azonban számos nem anyanyelvű felhasználó is létrehoz elektronikus tartalmakat, ami nehezíti a szövegfeldolgozást. Az arab szövegek feldolgozását tovább nehezíti az a jelenség, hogy a rövid magánhangzók nincsenek jelölve a szövegben, ezért az olvasónak mélyebb nyelvi tudással kell rendelkeznie, ha meg szeretné érteni a szöveget. Az arab nyelv a nagy nyelvek között szerepel, nagy mennyiségű szövegadatbázissal. Szövegösszegzéssel kevesen foglalkoztak eddig, ilyen jellegű korpusz nem volt elérhető. Korábbi kutatásunkban összegyűjtöttük az első összefoglaló generálásra alkalmas arab nyelvű szövegkorpuszt, majd különböző transzformer modelleket finomhangoltunk. A jelen kutatásban saját kísérleti egynyelvű és többnyelvű BART modellekkel kísérleteztünk, illetve egy mT5 modellt finomhangoltunk absztraktív összefoglalásra.

## 2. Kapcsolódó irodalom

Arab nyelvre elsősorban extraktív összefoglalás területén végeztek kutatásokat. Az első extraktív rendszer a Lakhas (Douzidia és Lapalme, 2004), ami a szövegből 10 szót vonatolt ki összegzésnek, majd gépi fordító segítségével angol nyelvre fordította, hogy össze tudja hasonlítani a rendszert más rendszerekkel. A kiértékelést a ROUGE metrikával végezték el. Al Qassem és mtsai (2019) egy fuzzy logikán alapuló megközelítéssel, főnevek kivonatolásával végezték az összegzést. A SumSat (Lakhdar és Chérageui, 2019) három módszert ötvözve, hibrid módon összegez: egy szöveg környezetének szemantikai feltárása, indikátorként használható kifejezések kiválasztása, illetve az összefoglaló generálása a reprezentatív kifejezésekkel.

Absztraktív összefoglalás szempontjából Azmi és Altmami (2018) az extraktív összefoglaló rendszerből kiindulva egy négy lépéses absztrakt összefoglaló módszert javasol. Első lépés a téma szegmentálása, második a címsor generálása, harmadik az extraktív összefoglaló generálása, végül negyedik lépés a mondatcsökkentés. Al-Maleh és Desouki (2020) kutatásukban a cikkek első bekezdéseiből generáltak címsorokat, amelyek összefoglalóként funkcionálnak. Az összefoglaló modellhez egy enkóder-dekóder architektúrájú rekurrens hálózatot alkalmaztak. Elmadani és mtsai (2020) a PreSumm (Liu és Lapata, 2019) módszerrel és a többnyelvű BERT (Devlin és mtsai, 2019) modellel finomhangoltak extraktív és absztraktív modelleket egyaránt. Kahla és mtsai (2021) szintén kísérleteztek a PreSumm módszerrel, viszont emellett az mBART modellt is sikerült finomhangolniuk. Kutatásukban többnyelvű (cross-lingual) finomhangolásokkal növelték a rendszer teljesítményét.

Kutatásunk során Kahla és mtsai (2021) kutatását tovább gondolva egynyelvű és többnyelvű BART modelleket finomhangoltunk, illetve napjaink egyre népszerűbb mT5 modelljét próbáltuk ki.

## 3. Felhasznált modellek

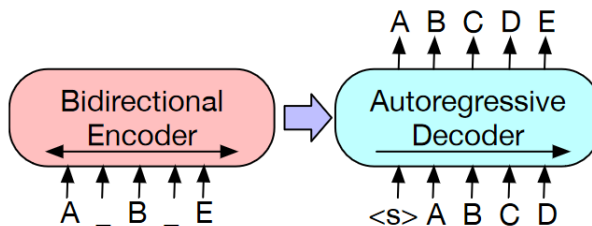
A **BART** (Lewis és mtsai, 2020) modell egy enkóder-dekóder architektúrán alapuló transzformer modell (lásd 1. ábra), amelyet a Fairseq (Facebook AI Rese-



arch Sequence-to-Sequence Toolkit) fejlesztett<sup>1</sup>. Az enkóder kétirányú (Bidirectional), a dekóder autoregresszív (Autoregressive). A korábbi kutatások alapján a csak enkóder típusú modellek (pl. BERT (Devlin és mtsai, 2019)) kiválóan alkalmasak magas minőségű szövegrepresentáció képezésére, azonban szöveggenerálás feladataira kevésbé. A csak dekóder típusú autoregresszív modellek (pl. GPT (Radford és Narasimhan, 2018)) a szöveggenerálás feladatain nyújtanak magas eredményt. A BART a két architektúra előnyeit ötvözi, ezért kiválóan alkalmas szövegösszefoglaló generálásra. Korábban kétféle BART modellt publikáltak:

- BART-base: 6 réteg enkóder és 6 réteg dekóder; 12 figyelmi fej; 768 szóbeágyazás dimenzió; bementi hossz: 512; 140 millió paraméter
- BART-large: 12 réteg enkóder és 12 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; 400 millió paraméter

Az **mBART** (Liu és mtsai, 2020) egy több nyelven előtanított BART modell. Az előtanításhoz a Common Crawl adatbázisból kivonatolt 25 nyelvet tartalmazó CC25 (Wenzek és mtsai, 2020) korpuszt használták. Az mBART modellel végzett kísérletek rávilágítottak arra, hogy abban az esetben hasznos igazán a célnyelvtől eltérő nyelveken történő előtanítás, amennyiben a célnyelven rendelkezésre álló egynyelvű adathalmaz redukált méretű. Az mBART modellel végzett munka rámutat a többnyelvű előtanításban rejlő lehetőségek transzfer tanulási (transfer learning) irányba való felhasználhatóságára.



1. ábra: BART modell architektúrája (Lewis és mtsai, 2020).

A **T5** (Text-To-Text Transfer transzformer) (Raffel és mtsai, 2020) a Google által készített enkóder-dekóder típusú modell. Az utóbbi időben a nyelvtechnológia területén kiemelt jelentőséggel bír a transzfer tanulás, amelynek során a nyelvi modellt egy adatokban gazdag feladaton tanították be, majd ezt követően került finomhangolásra egy soron következő célfeladatra. Ideális esetben a modell az előtanítás során olyan általános tudásra tesz szert, amely átvihető, és sikeresen alkalmazható a célfeladatok megoldásában. A T5 projekt alapötlete, hogy minden szövegelemzési feladatot (fordítás, kérdések megválaszolása, osztályozás stb.) szövegből szöveg (text-to-text) problémaként közelít meg, azaz

<sup>1</sup> <https://github.com/pytorch/fairseq/tree/master/examples/bart>

szöveg a bemenet és a modell ez alapján szöveget generál kimenetként (lásd 2). Itt fontos kiemelni a BERT-alapú modellekkel szemben mutatkozó alapvető különbséget a felépítésben: a T5 esetében mind a bemenet, mind pedig a kimenet szöveg formátumú, míg a BERT-alapú modellek esetében a bemenet szöveges, a kimenet azonban vagy egy osztályozó címke vagy pedig csak valamilyen bemenetből származó töredék.

A T5 projekt elsődleges célja nem az, hogy új módszerek kerüljenek kifejlesztésre, a munka mögött álló csapat elsődleges motivációja az, hogy bemutassák a terület jelenlegi állását, és összehasonlítsák az elérhető technikákat. Emellett a jelenlegi megközelítések határait is próbálják megállapítani azáltal, hogy szisztematikus módon és nagy mértékben megnövelt paraméterszámmal (modellek betanítása 11 milliárd paraméterig) kísérleteznek. A modell tanításához felhasznált korpusz a Colossal Clean Crawled Corpus (rövidítve C4), amely egy több száz gigabájtnyi világhálóról összegyűjtött és tisztított angol nyelvű szöveget tartalmaz. Az T5 esetében a paraméterek száma alapján 5 különböző méretű modell került betanításra:

- Small (300 millió paraméter), Base (580 millió paraméter), Large (1,2 milliárd paraméter), XL (3,7 milliárd paraméter), XXL (13 milliárd)

Az **mT5** (Xue és mtsai, 2021) a T5 több nyelvre kiterjesztett verziója. Az mT5 létrehozása során a szerzők törekedtek arra, hogy minél inkább megőrizték a T5 strukturális jegyeit, ezért az mT5 örökölte a szövegből szöveg (text-to-text) tulajdonságot és az általános előtanítás menetét is, amelyhez szintén rendkívül nagy méretű korpuszt használtak.

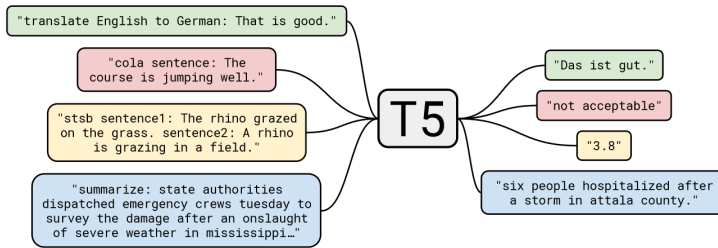
Az mT5 betanításához az mC4 korpuszt használták, amely a T5 tanítására alkalmazott C4 többnyelvű változata. Az mC4 101 különböző nyelvű szövegeket tartalmaz. A T5 modellel összehasonlítva az mT5 nagyobb paraméterszámokkal rendelkezik, ez a nagyobb szótárméret következménye. Fontos megjegyezni, hogy T5-alapú modellek az enkóder-dekóder struktúrát követik, ezért paraméterszámuk általában kétszer akkora, mint egy hasonló méretű csak enkóder struktúrájú modell.

Az mT5 kutatás megmutatta, hogy a T5 modell kiválóan alkalmazható többnyelvű kontextusban is, továbbá rendkívül magas eredményeket tud elérni különböző referenciateladatokban.

#### 4. Felhasznált korpuszok

Az egy nyelvű BART modellünk tanításához erőforrás hiányában jelen kutatáshoz egy csökkentett méretű arab Wikipédia szöveget használtunk. A korpusz előállításához első körben 150.000 szegmensnyi szöveget töltöttünk le. A BART tanításához olyan bekezdések kellenek, amelyek legalább két pont írásjellel rendelkező mondatot tartalmaznak. Ezért első lépésként ez alapján szűrtük a szöveget, a szűrés után 9.773 bekezdésünk maradt.

A többnyelvű BART modellünk tanításához angol, magyar és arab Wikipédiából vett bekezdéseket vettünk, amelyek minimum kettő pont írásjellel rendelkeztek (ez az elvárás a BART esetében). A kiegyensúlyozottság végett 10.000



2. ábra: T5 modell (Raffel és mtsai, 2020).

szegmenst vettünk mind az angol mind a magyar korpuszból. Az 1. táblázatban láthatóak az előtanításhoz használt korpuszokra jellemző kvantitatív tulajdonságok.

	Arab	Angol	Magyar
Szegmens	9.773	10.000	10.000
Token	761.371	1.357.875	818.420
Type	108.982	60.248	139.996
Átlagos mondatszám	1,73	5,06	4,36
Átlagos tokenszám	77,897	135,78	81,84

1. táblázat. Előtanításhoz felhasznált korpuszok tulajdonságai.

A finomhangoláshoz ugyanazt a korpuszt (Arab-Szum) használtuk, mint az előző kutatásunkban (Kahla és mtsai, 2021). Továbbá végeztünk transzfer tanulási kísérletet is, amihez vegyesen válogattunk angol és magyar szegmenseket (Multi-Szum). A kiegyensúlyozottság végett 20.000 angol és 20.000 magyar szegmens került kiválasztásra a finomhangolási korpuszba. Az angol szegmenseket a CNN/Daily Mail korpuszból (Nallapati és mtsai, 2016) vettük, míg a magyar szegmenseket a HVG korpuszból (Yang és mtsai, 2021). A jelen kutatáshoz kiválasztott finomhangolási korpusz részkorpusza az előző kutatáshoz használt angol és magyar finomhangolási korpuszoknak. A 2. táblázatban láthatóak a korpuszokra jellemző kvantitatív tulajdonságok.

## 5. Kísérletek

BART kutatásunk során előtanítottunk egy egynyelvű és egy többnyelvű (angol, magyar, arab) BART base modellt. A Facebook nem tette közzé az előtanítás implementációját, ezért a Hugging Face transformers<sup>2</sup> könyvtárai által biztosított előtanítási függvényeket használtuk. A BART előtanításához a BartFor-

<sup>2</sup> [https://huggingface.co/transzformers/model\\_doc/bart.html](https://huggingface.co/transzformers/model_doc/bart.html)

	Arab		Angol		Magyar	
	Cikk	Lead	Cikk	Lead	Cikk	Lead
Szegmens	21.508		20.000		20.000	
Token	6.929.974	2.867.754	15.795.098	1.050.273	5.387.638	602.136
Type	290.138	178.614	169.709	56.902	397.628	99.166
Átlagos mondatszám	14,42	1,47	28,69	1	11,19	1,56
Átlagos tokenszám	412,05	35,131	789,76	52,51	269,38	30,10

2. táblázat. Finomhangoláshoz felhasznált korpuszok tulajdonságai.

CausalLM<sup>3</sup> függvényt használtuk. A BartForCausalLM a BART modell dekóder önálló része, melynek a tetején egy nyelvmodell réteg található. Ez alkalmas a következő szó prediktálására (causal language modeling). A modell tovább finomhangolható BART finomhangolási feladatokra. A kutatásunk során egy kísérleti arab egynyelvű és egy kísérleti háromnyelvű BART base modellt tanítottunk elő:

- **Arab BART**: Egynyelvű arab BART base modell, 512 bemeneti szöveg-hossz, közel 19.808 bekezdésnyi arab Wikipédia szövegen tanítva. A szótárméret: 40.000.
- **Multi BART**: háromnyelvű BART base modell, 512 bemeneti szöveghossz, bekezdés alapú Wikipédia szövegeken tanítva: 19.808 arab, 20.000 angol és 20.000 magyar. A szótárméret: 50.000.

Az egynyelvű BART modell előtanításához az alábbi hiperparamétereket használtuk: 512 bemeneti szöveghossz; batch méret: 8/GPU (4 db GeForce GTX 1080 + 4 db GeForce RTX 2080); epoch szám: 50; tanulási ráta: 2e-6; fp16.

A többnyelvű BART modell előtanításához az alábbi hiperparamétereket használtuk: 512 bemeneti szöveghossz; batch méret: 6/GPU (4 db GeForce GTX 1080 + 4 db GeForce RTX 2080); epoch szám: 50; tanulási ráta: 8e-7; fp16.

Finomhangolós kísérleteinkben kettő BART modellt tanítottunk:

- **BART arab szum**: Arab BART finomhangolva Arab-Szum korpuszon.
- **BART multi transz**: Multi BART finomhangolva a Multi-Szum korpuszon, majd azt tovább finomhangoltuk az Arab-Szum korpuszon.

Az egynyelvű BART finomhangolásához az alábbi hiperparamétereket használtuk: 512 maximum bemeneti és 256 maximum kimeneti szöveghossz, batch méret: 8/GPU (4 db GeForce GTX 1080 - 12GB) méret, epoch szám: 120, tanulási ráta: 2e-5, warmup lépés: 5000; fp16.

A többnyelvű kísérlet során először finomhangoltuk a Multi BART modelünket a háromnyelvű Multi-Szum korpuszon az alábbi hiperparaméterekkel: 512 maximum bemeneti és 256 maximum kimeneti szöveghossz; batch méret 5/GPU (4 db GeForce GTX 1080 + 4 db GeForce RTX 2080); epoch szám: 40; tanulási ráta: 5e-5, warmup lépés: 5000; fp16.

Majd a többnyelvű finomhangolás után tovább finomhangoltuk az Arab-Sum korpuszon, az alábbi hiperparaméterekkel: 512 maximum bemeneti és 256 maximum kimeneti szöveghossz; batch méret 5/GPU (4 db GeForce GTX 1080 +

<sup>3</sup> [https://huggingface.co/transzformers/model\\_doc/bart.html#bartforcausalml](https://huggingface.co/transzformers/model_doc/bart.html#bartforcausalml)

4 db GeForce RTX 2080); epoch szám: 80; tanulási ráta: 5e-5, warmup lépés: 5000; fp16.

A többnyelvű kísérletek összeségében szintén 120 (40+80) epoch szám mellett tanultak. Azt tapasztaltuk, hogy a magas epoch szám nem okoz túltanulást, inkább egyre finomabb dolgokat tanult meg.

Végül kísérleteztünk az mT5 modellel:

- **mT5 arab szum**: mT5 small modell finomhangolása az Arab-Szum korpuszon.

Az mT5 finomhangolása alábbi hiperparaméterekkel történt: 512 maximum bemeneti és 256 maximum kimeneti szöveghossz; batch méret 2/GPU (4 db GeForce GTX 1080); epoch szám: 40; tanulási ráta: 2e-5, warmup lépés: 5000; prefix: "summarize: ". Az fp16 paramétert nem használtuk, mivel a T5 típusú modellek esetében az fp16 használatával nem konvergál a tanítás. Az erőforrásaink korlátai miatt nem tudtuk az mT5 nagyobb modelljeit kipróbálni.

## 6. Eredmények

A 3. és a 4. táblázatban láthatóak a modellek tulajdonságai, illetve a méréseink eredményei. A dupla vonal alatti modelleket tanítottuk a jelen kutatásunkban. A „+” jel jelöli azokat a modelleket, amelyekhez a mostani kutatásunk során előtanítást is végeztünk. Korábbi kutatásunkban a PreSumm sajátosságai miatt a fedés mértékeket publikáltuk, azonban a nemzetközi sztenderd szerint az F-mérték a mérvadó, ezért a modelleket újra teszteltük az F-mérték alapján. Röviden a modellekről:

- *AraBERT*: AraBERT (Antoun és mtsai, 2020) finomhangolása arab korpuszon PreSumm eszközzel
- *mBERT*: többnyelvű BERT (Devlin és mtsai, 2019) finomhangolása arab korpuszon PreSumm eszközzel
- *mBERT + hun*: mBERT finomhangolása magyar HVG korpuszon (Yang és mtsai, 2021) PreSumm eszközzel, majd tovább finomhangolása arab korpuszon
- *mBERT + eng*: mBERT finomhangolása angol CNN/Daily Mail korpuszon PreSumm eszközzel, majd tovább finomhangolása arab korpuszon
- *mBART-50*: mBART-50 (Tang és mtsai, 2020) finomhangolása arab korpuszon
- *mBART-50-rus*: Gazeta korpuszon (Gusev, 2020) (52.400 szegmens) finomhangolt mBART-50, majd tovább finomhangolva arab korpuszon

Az összevethetőség végett a 3. táblázatban feltüntettük a paraméterszámokat, az általunk felhasznált előtanításhoz (Elő) és finomhangoláshoz (Finom) használt korpuszok méreteit és azt, hogy melyik modell milyen nyelvi tudással rendelkezik.

Korábbi kutatásunkból az látható, hogy az elő-finomhangolt (más nyelven transzfer tanulással) modellekkel tudtunk növelni a rendszer minőségén.

Modell	Paraméter #	Elő (token)	Finom (szegmens)	Nyelv
AraBERT	136 millió	-	19.808	arab
mBERT	110 millió	-	19.808	104 nyelv
mBERT+hun	110 millió	-	442.739+19.808	104 nyelv
mBERT+eng	110 millió	-	286,817+19.808	104 nyelv
mBART-50	610 millió	-	19.808	50 nyelv
mBART-50-rus	610 millió	-	19.808	50 nyelv
+ BART arab szum	140 millió	761.371	19.808	arab
+ BART multi transz	140 millió	2.937.666	59.808+19.808	arab, angol, magyar
mt5 arab szum	300 millió	-	19.808	101 nyelv

3. táblázat. Modellek tulajdonságai.

A mostani eredményekből az látható, hogy az általunk tanított kísérleti BART modellek nem tudják felülmúlni a korábbi kutatásunkban elért eredményeket. Ez várakozásunknak megfelel, hiszen azokat a modelleket sokkal nagyobb adathalmazon tanították elő. Mind a többnyelvű BERT, mind a huBERT, vagy az mBART óriási mennyiségű adaton tanult szemben a körülbelül 30.000 szegmensű Wikipédia korpuszunkkal, de csak kevés értékkel marad le. Azonban AraBERT modellt így is szignifikánsan felülmúlja. Továbbá a korábbi kutatásunk tapasztalata, miszerint többnyelvű transzfer tanulással tovább növelhető a rendszer teljesítménye, újra bebizonyosodott. Az angol-magyar adatokkal hozzáadott korpuszon való elő-finomhangolás javított a rendszer minőségén.

Manuálisan vizsgálva az összefoglalókat, azt figyeltük meg, hogy az összefoglalók nyelvtanilag helyesek, a BART modellek kevés hibát vétenek és a témakör szintjén relevánsak. Az egyetlen típushiba, hogy gyakran belekever olyan elemeket, amelyek szemantikailag nem helytállóak.

Ezzel a kutatással bebizonyosodott, hogy képesek vagyunk saját BART modellt tanítani, valamint nagyobb erőforrás és tanítóanyag mellett tovább növelhető a modellek minősége.

Modell	ROUGE-1	ROUGE-2	ROUGE-L
AraBERT	0,772	0,008	0,772
mBERT	4,264	0,164	4,264
mBERT+hun	4,909	0,178	4,903
mBERT+eng	12,610	2,107	12,610
mBART-50	5,952	0,312	5,921
mBART-50-rus	7,145	0,766	7,101
+ BART arab szum	3,066	0,023	3,007
+ BART multi transz	3,895	0,114	3,877
mt5 arab szum	6,851	0,294	6,840

4. táblázat. Arab összefoglaló generálás F-mérték eredmények.

Végül, de nem utolsósorban azt láthatjuk az mT5 small modell finomhangolásának eredményében (lásd 4. táblázat), hogy minőségében felülmúlja az mBERT,

mBERT + hun és az mBART-50 modelleket is. Elő-finomhangolás nélkül közel olyan magas eredményt ér el, mint az mBART-50-rus. Fontos megjegyezni, hogy ez egy small modell, ami paramétereit tekintve sokkal kisebb mint az mBART. Ezzel a méréssel azt láthatjuk, hogy nagyobb erőforrás mellett és esetleg nagyobb epoch szám mellett további eredményjavulást tudunk elérni.

További példák és modellek a projekt oldalunkon<sup>4</sup> érhetőek el.

## 7. Összegzés

Kutatásunk során arab nyelvre tanítottunk különböző transzformer modelleket absztraktív összefoglaló generálás feladatában. A jelen tanulmány egy pillanatképet mutat a kutatásunk jelenlegi fázisáról, amelyben saját egynyelvű és többnyelvű BART előtanításával és finomhangolásával kísérletezünk. Továbbá a napjaink egyik népszerű T5 többnyelvű modelljét is finomhangoltuk. Kutatásunkat ezen fázisában kevés erőforrással és tanítóanyaggal végeztük el, ezért várakozásunknak megfelelően eredményeinkkel nem tudtuk felülmúlni a korábbi „state of the art” eredményét. Azonban bemutattuk, hogy így is versenyképes teljesítményt tudtak nyújtani, ami kiváló alapot képez a nagyobb erőforrással való kísérletek számára.

A jövőben a jelen tanulmányban bemutatott kísérleteket fogjuk elvégezni nagy teljesítményű szuperszámítógépeken és nagy mennyiségű adatokon.

## Hivatkozások

- Al-Maleh, M., Desouki, S.: Arabic text summarization using deep learning approach. *Journal of Big Data* 7, 1–17 (2020)
- Al Qassem, L., Wang, D., Barada, H., Al-Rubaie, A., Almoosa, N.: Automatic Arabic text summarization based on fuzzy logic. In: *Proceedings of the 3rd International Conference on Natural Language and Speech Processing*. pp. 42–48 (2019)
- Antoun, W., Baly, F., Hajj, H.: AraBERT: Transformer-based model for Arabic language understanding. In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. pp. 9–15. European Language Resource Association, Marseille, France (May 2020)
- Azmi, A.M., Altmami, N.I.: An abstractive arabic text summarizer with user controlled granularity. *Information Processing and Management* 54(6), 903–921 (2018)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

<sup>4</sup> <http://nlp.itk.ppke.hu/projects/summarize>

- Douzidia, F.S., Lapalme, G.: Lakhas, an Arabic summarization system. *Proceedings of DUC2004* (2004)
- Elmadani, K.N., Elgezouli, M., Showk, A.: BERT fine-tuning for Arabic text summarization. *ArXiv abs/2004.14135* (2020)
- Gusev, I.: Dataset for automatic summarization of russian news. In: *Artificial Intelligence and Natural Language*. pp. 122–134. Springer International Publishing, Cham (2020)
- Kahla, M., Yang, Z.G., Novák, A.: Cross-lingual fine-tuning for abstractive Arabic text summarization. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. pp. 655–663. INCOMA Ltd., Held Online (Sep 2021)
- Lakhdar, S.M., Chérageui, M.A.: Building an extractive Arabic text summarization using a hybrid approach. In: *International Conference on Arabic Language Processing*. pp. 135–148. Springer (2019)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (2019)
- Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., Zettlemoyer, L.: Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics* 8, 726–742 (11 2020)
- Nallapati, R., Zhou, B., dos Santos, C., Caglar, G., Xiang, B.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. pp. 280–290. Association for Computational Linguistics, Berlin, Germany (Aug 2016)
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020)
- Tang, Y., Tran, C., Li, X., Chen, P.J., Goyal, N., Chaudhary, V., Gu, J., Fan, A.: Multilingual translation with extensible multilingual pretraining and finetuning (2020)
- Wenzek, G., Lachaux, M.A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., Grave, E.: CCNet: Extracting high quality monolingual datasets from web



- crawl data. In: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France (May 2020)
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021)
- Yang, Z.G., Agócs, Á., Kúspér, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (2021)



# SZEMANTIKA, GÉPI FORDÍTÁS



## Mondatszintű szentiment analízis teljesítményének javítása adatkiterjesztéses eljárásokkal

Laki László János, Yang Zijian Győző

Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.

{laki.laszlo, yang.zijian.gyozo}@nytud.hu

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport

1083 Budapest, Práter u. 50/a.

{laki.laszlo, yang.zijian.gyozo}@itk.ppke.hu

**Kivonat** A szentiment analízis egy olyan eljárás, amelynek segítségével információkat nyerhetünk az írott tartalmak emocionális töltetét illetően. Számítógépes algoritmusok révén olyan újgenerációs modellek kifejlesztése válik lehetségessé, amelyek korábban nem tapasztalt mennyiségű és minőségű adatot képesek feldolgozni. Ugyanakkor, ezen modellek gyakran rendkívül nagy mennyiségű erőforrást igényelnek, hogy a kívánt teljesítményt elérjék. Ennek megfelelően kiemelt szerep jut azon kutatásoknak, amelyek a modellek struktúrájának és funkciójának javításával magas minőségű eredményeket tudnak generálni amellet, hogy a működésükhöz szükséges erőforrásigényt csökkenteni képesek. A kognitív tudományok szemszögéből fontos célkitűzés, hogy tanulmányozzuk és mélyebb megértésére jussunk egy adott személy mentális állapotának, illetve annak valamilyen aktivitás formájában történő kivetülésének, és ezek potenciális felhasználhatóságát a mögöttes érzések és érzelmek meghatározására. A digitális eszközök egyre elterjedtebb használatával nagy mértékben növekszik a szociális médiában és az egyéb internetes platformokon megjelenő írott tartalmak mennyisége, amely fontos forrásként használható a tartalmakat generáló személyek érzelmeinek meghatározására. Ebből kifolyólag ezek az internetes tartalmak kifejezetten alkalmas lehetőséget nyújtanak szentiment analízis elvégzésére. Az elmúlt években megfelelően finomhangolt nyelvi modellek egyre nagyobb mértékben lettek alkalmazva olyan természetes nyelvelemzési feladatokra, mint a szentiment analízis. Számos hátráltató tényező nehezíti azonban a finomhangolás folyamatát, ilyen például a betanításhoz használható megfelelő méretű korpusz hiánya, vagy az ilyen irányú felhasználásra alkalmas korpuszok teljes hiánya. Jelen kísérletes megközelítésünk során olyan adatnövelő módszereket alkalmazunk, mint a gépi fordítás és a nyelvek közötti transzfer tanítás, és ezek segítségével növeljük a betanító korpuszok méretét. 9 különböző nyelvi modellel végzett kísérleteink eredményét mutatjuk be, melyeket a Hungarian Twitter Sentiment Corpus-on tanítottunk be. Kutatásunk rávilágít arra, hogy a betanított modellek teljesítménye növelhető, ha géppel lefordított szöveget adunk a betanító korpuszhoz.

Továbbá, több általunk használt modell is képes volt jobb teljesítményre a jelenlegi magyar state-of-the-art modelleknél.

**Kulcsszavak:** szentiment analízis, mondatszintű osztályozás, adatkiterjesztés, gépi fordítás, transzfer tanítás, zeroshot

## 1. Bevezetés

A szentiment analízis az érzelmek automatizált azonosítása egy adott szövegben és ezek osztályozása olyan kategóriákba, mint negatív, semleges vagy pozitív. A szociális médiát használók köre egyre növekszik, ezáltal óriási mennyiségű szöveges információ keletkezik és áll rendelkezésre az interneten, amely rendkívül hasznosnak bizonyul a tartalmakat szerző személyek érzelmeinek meghatározásában. Mivel egyedülálló lehetőség nyílik erre számos potenciális alkalmazhatósággal karöltve, mind az akadémia mind az ipar képviselői komoly és egyre növekvő érdeklődést mutatnak a szövegekből történő szentimentális információ kinyerése iránt (Hoang és mtsai, 2019).

A neurális háló alapú nyelvi modellezés áttörést eredményezett a legtöbb természetes nyelvfeldolgozási feladatban. Szinte egyik napról a másikra jelennek meg újabb és újabb modellek, amelyek túlszárnyalják a korábbi modellek teljesítményét. A nyelvi modellek nemcsak a betanításhoz használt adat tekintetében különböznek, hanem a használt neurális hálók belső struktúrájában és a tanítási módszerekben is. Következésképpen, egy adott természetes nyelvfeldolgozási feladat megoldható egy megfelelően választott nyelvi modellel. Továbbá, érdekes felvetés, hogy egy nyelvspecifikus modell túlteljesíthet-e egy többnyelvű modellt, amely egy kulcskérdés a magyar nyelvű szentiment analízis esetében is. A jelenlegi legjobb eredményt hozó megoldás egy adott természetes nyelvfeldolgozási feladat megoldására egy már előtanított nyelvi modell további finomhangolása egy alkalmazási területre specifikus feladatra. Az ilyen rendszerek minősége nemcsak az előtanított modellektől függ, hanem a finomhangoló adathalmaz méretétől is. Az olyan természetes nyelvfeldolgozási feladatok, mint a magyar nyelvre kifejlesztett szentiment analízis komoly érdeklődésre tett szert az ipari szegmens részéről, ellenben limitálva van a szabadon elérhető adatok szempontjából. Jelen állás szerint nem találtunk olyan korábban publikált megoldást a magyar nyelvű szentiment analízisre, ami azt jelenti, hogy az általunk kínált alkalmazás a legelső ilyen megjelenés a témában.

Kutatásunk során egy gépi fordító rendszert alkalmaztunk angol nyelvű szentiment analízis adathalmaz magyar nyelvre történő lefordításához. A gépi fordító rendszer betanítását magunk végeztük. A lefordított korpuszt integráltuk a rendszerünkbe.

A 2. Rész bemutatja az eddig leírt megoldásokat a témában, a 3. Rész a gépi fordító rendszert, a 4. Részben bemutatásra kerülnek a korpuszok és a modellek, majd az 5. Részben ismertetjük az elvégzett kísérleteket. Az eredmények az 6. Részben találhatóak.

Modelljeink és szkriptjeink megtalálhatóak a Github<sup>1</sup> és Hugging Face<sup>2</sup> oldalainkon.

## 2. Kapcsolódó irodalom

A szentiment analízis egy rendkívül komplex természetes nyelvfeldolgozási feladat és számos területen alkalmazható, így például szociális média monitorozása során (Neri és mtsai, 2012), befektetők döntéshozatali folyamatának támogatására a gazdasági hírek szemantikus kontextusának elemzésével (Lutz és mtsai, 2018; Saura és mtsai, 2019), digitális marketing esetében (Kinholkar és Waghmare, 2016), pszichés állapot vizsgálatánál (Jo és mtsai, 2018), valamint számos további területen, ahol az ilyen magas szintű szövegbányászati eljárások nagyban megnövelhetik a hatékonyságot. Jelenleg többféle irányban zajlanak a szentiment analízis célú fejlesztések. Kezdetben a cél az volt, hogy osztályozni lehessen dokumentumokat és szövegeket az általános polaritásuk alapján (negatív, pozitív vagy semleges) (Pang és mtsai, 2002). Egy másik fő irány az aspektus-alapú módszer, amely kifinomultabb és célja, hogy azonosítsa egy tárgy vagy bármilyen entitás olyan aspektusait, melyek az érzelmek kiváltásáért felelősek (Pontiki és mtsai, 2014). Egy alternatív stratégia pedig a mondatszintű szentiment analízis, mely a mondatot állítja vizsgálódása középpontjába, tehát feladata egy szövegben szereplő mondat érzelmi polarizáltságát (Feldman, 2013; Lutz és mtsai, 2018).

A Kahla és mtsai (2021) által publikált nyelvek közötti transzfer kísérletek (Kahla és mtsai, 2021) bebizonyították, hogy egy kétlépcsős finomhangolási megközelítés jelentősen javíthatja a természetes nyelvfeldolgozási feladat kimenetelét. Az első finomhangolás angol nyelven történt, melyet a második arab nyelvű finomhangolási lépés követett.

Magyar nyelvre elsősorban aspektus-alapú szentiment analízis korpuszok és eszközök léteznek. OpinHuBank (Miháltz, 2013) egy manuálisan annotált korpusz, mely a véleménykutatást és a szentiment analízist támogatja. 10000 mondatból áll és személyneveket tartalmaz a főbb magyarországi honlapokról és blogokról. Minden egyes entitás értékelve volt 5 különböző emberi annotátor által a szentiment polarításra nézve (semleges, pozitív vagy negatív). A HuSent (Szabó és mtsai, 2016) egy manuálisan annotált szentiment korpusz, mely a Dívány honlapról<sup>3</sup> vett magyar nyelvű véleményeket tartalmaz a különböző termékekkel kapcsolatban. A korpusz 154 véleményt tartalmaz, mintegy 17.000 mondatból és 251.000 tokenből áll. Steinberger és mtsai (Steinberger és mtsai, 2011) kutatásai során előállított egy aspektus-alapú szentiment korpuszt többnyelvű párhuzamos korpuszokkal, amely tartalmaz egy magyar nyelvű alkorpuszt is.

Jelen kutatás keretében a mondatszintű szentiment analízisre fókuszáltunk, emellett aspektus-alapú szentiment analízis kísérleteket is tervezünk a jövőben.

<sup>1</sup> <https://github.com/nytud/sentiment-analysis>

<sup>2</sup> <https://huggingface.co/NYTK>

<sup>3</sup> <http://divany.hu>

### 3. Adatkiterjesztés gépi fordítással és nyelvek közötti transzferrel

A bevezetésben említettek alapján egyértelműen kirajzolódik, hogy a betanításhoz szükséges adathalmaz mérete kiemelkedő fontosságú a neurális háló alapú modellek estében. Sajnos jelenleg nem áll rendelkezésre megfelelő minőségű adat, a manuális módon történő adatgenerálás pedig rendkívül költséges. Kutatásunk során ezen hiány áthidalására gépi fordítást és nyelvek közötti transzfert alkalmaztunk az adathalmazunk méretének növelése érdekében (Fadaee és mtsai, 2017). Azzal az ötlettel álltunk elő, hogy már meglévő angol nyelvű korpuszokat használunk és lefordítva azokat kiegészítő betanító adatként alkalmazzuk. Az ötlet a gépi fordítás területéről származik, amikor visszafordított korpuszokat használnak annak érdekében, hogy javítsák a fordítás minőségét egy alacsony ellátottságú nyelvpár esetén (Poncelas és mtsai, 2018).

A lefordított korpuszok felhasználására két lehetőség adódik. Az első esetben a nyelvek közötti adattranszfer során az angol nyelvű korpusz úgy van alkalmazva, mint egy első körös finomhangolós adathalmaz mielőtt felhasználásra kerülne, mint doménon belüli magas minőségű változat (későbbiekben fordított+finomhangolás megjelöléssel utalunk rá). Másodsorban, a segédkorpusz konkatenálható a doménon belüli (erre mix megjelöléssel utalunk). Az első finomhangolási lépés a konkatenált adattal (mix) történik, majd a második lépésben a doménon belüli (mix+finomhangolás).

Kísérletes munkánk során a MarianNMT (Junczys-Dowmunt és mtsai, 2018) szoftvercsomagot alkalmaztuk, amely egy szabad forráskódú C++ programnyelven írott alkalmazás. Ez egy könnyen installálható, alaposan ledokumentált, memória- és erőforráskímélő implementáció, amely gyakori használatnak örvend az akadémiai és a fejlesztői körökben egyaránt (Barrault és mtsai, 2019). Egy transzformerrel alapuló enkóder-dekóder struktúra lett alkalmazva SentencePiece tokenizáláshoz. A tokenizáló algoritmus egy általános szöveget használt mindkét nyelv irányába 32000-es szótárméret mellett. Az alapbeállítások mellett használtuk az alkalmazást a rejtett rétegek méretét és az optimalizációs metrikákat illetően. A betanító adathalmazként a ParaCraw16 és az OpenSubtitles (Lison és Tiedemann, 2016) platformokról származó angol-magyar nyelvpárok kerültek felhasználásra. A teljes betanító adathalmaz 45,5 millió szegmenst és 573 millió angol nyelvű tokent tartalmaz. A rendszer 36,873 százalékos BLEU (Papineni és mtsai, 2002) értéket ért el a tesztelés során (3000 teszt adathalmazból random módon kiválasztott szegmensen).

### 4. Korpuszok és Modellek

A mondatszintű szentiment analízis betanításhoz a Precognox Kft.<sup>4</sup> Által készített Hungarian Twitter Sentiment<sup>5</sup> (MTS) korpuszt használtuk. Jelenleg ez az

<sup>4</sup> <https://www.precognox.hu>

<sup>5</sup> <http://opendata.hu/dataset/hungarian-twitter-sentiment-corpus>



egyetlen szabadon elérhető korpusz, amely magyar nyelvű szentiment analízisre használható. Az 1. táblázat foglalja össze a MTS korpusz főbb jellegzetességeit. A skálázás 1-től 5-ig a következő módon történik: 1 – nagyon negatív, 2 – negatív, 3 – semleges, 4 – pozitív, 5 – nagyon pozitív. Erre a korpuszra a MTS5 megjelöléssel utalunk. Egy másik esetben a 0 és 1 értékeket negatívnak, a 4 és 5 értékeket pedig pozitívnak értékeltük, a 3-as értékeket pedig kihagytuk. Ez utóbbi korpuszra MTS2 néven utalunk.

Adataink kiterjesztéséhez az SST2 és SST5 korpuszokat választottuk (Wang és mtsai, 2018). Mindkét korpusz angol nyelvű mondatokat tartalmaz. Gépi fordítás segítségével ezeket a korpuszokat felhasználtuk hozzáadott adatként (SST2\_hu és SST5\_hu megjelöléssel utalunk ezekre a magyarra fordított korpuszokra). Az 1. táblázat alapján látható, hogy az SST korpuszok lényegesen nagyobbak, mint a MTS korpuszok.

	SST2	SST5	MTS2	MTS5
Segments	70.045	11.855	2.737	4.000
Token	652.594	227.245	33.279	46.683
Type	17.516	21.699	15.900	21.689
Tanító anyag	67.350	8.544	2.193	3.200
Validációs anyag	-	-	273	400
Teszt anyag	873	1.101	273	400
Osztályok	0;1	1;2;3;4;5	0;1	1;2;3;4;5

1. táblázat. A korpuszok tulajdonságai.

Kutatásunk során a MTS5 korpuszt felosztottuk 90-10 százalékos arányban betanító és tesztelő korpuszokra. Az első 400 tweet képezi a tesztelő korpuszunkat. A MTS2 esetében a 3-as értéket tartalmazó dokumentumok kihagyásra kerültek. A 2. táblázat mutatja korpuszok felépítését és az adatok megoszlását. Kísérleteink során 6 különböző egynyelvű (magyar) kontextuális nyelvi modellt, 2 többnyelvű modellt és egy klasszikus szóbeágyazáson alapuló modellt használtunk.

	train	test	train	test
label	SST2	MTS2	SST5	MTS5
0	29.755	428	1.021	108
1	37.539	444	1.448	162
1	1.089	139	93	12
2	2.200	289	936	88
3	1.594	229	1.111	150
4	2.259	279	1.349	141
5	1.266	165	111	9

2. táblázat. Címkék eloszlása a korpuszokban.

A fejezet második részét e modellek rövid bemutatására szánjuk.

**huBERT** (Nemeskey, 2021): egy magyar BERT (Devlin és mtsai, 2019) base modell, mely a Webcorpus 2.0 korpuszon (Nemeskey, 2020) lett betanítva, ez utóbbi a Common Crwal webarchívumból és a magyar nyelvű Wikipédiából tevődik össze. Kisbetűsített és kisbetűsítés nélküli verzió is készült a huBERT-hez. Kiemelendő, hogy a huBERT túlszárnyalja a többnyelvű BERT modellt számos feladatban, így például a maszkolt nyelvi modellezésben, névelemfelismerésben vagy névcsoport azonosításban. Jelenleg a state-of-the-art megoldásnak számít a névelemfelismerés azonosítás területén.

**HILBERT** (Feldmann és mtsai, 2021): egy BERT large modell magyar nyelvre, amely kiemelkedő teljesítményt nyújt nyelvfeldolgozási feladatokban. A HILBERT a NYTK-BERT (Feldmann és mtsai, 2021) korpuszon lett betanítva. A modell számos feladat esetén figyelemreméltó eredményeket ér el, így például névelemfelismerésben és összefoglaló generálásban (Yang és mtsai, 2021). Az egyik előnye ennek a modellnek a huBERT-tel szemben, hogy több paraméteret tartalmaz, ellenben kevesebb betanító adattal.

**HIL-RoBERTa**<sup>6</sup>: az egyik legfőbb kihívás a nyelvi modellek optimalizálása során az előtanításban mutatkozik. Mivel az előtanítás egy rendkívül erőforrásigényes folyamat, ezért kiemelten fontos az új módszerek kutatása és kifejlesztése, amelyek szignifikáns javulást tudnak indukálni ezen a területen. A RoBERTa (Liu és mtsai, 2019) a Robustly optimized BERT pre-training approach angol nyelvű szakirodalmi megjelölés rövidítéséből származik. A RoBERTa kiemelkedő eredményeket ér el olyan sztenderd feladatokban, mint a GLUE (Wang és mtsai, 2018), a RACE (Lai és mtsai, 2017) vagy a SQuAD (Rajpurkar és mtsai, 2016), amelyet, hogy lényegesen kevesebb erőforrást használ optimalizált előtanítási paradigmájának köszönhetően. A HIL-RoBERTa egy RoBERTa small modell, amely a magyar nyelvű Wikipédián lett betanítva. HIL-RoBERTa kiváló eredményeket ér el névelemfelismerésben és névcsoport azonosításban, megközelítve a huBERT teljesítményét.

**HIL-ALBERT**<sup>7</sup>: számos erőfeszítés irányul arra, hogy emeljük a nyelvi modellek teljesítményét a célfeladatra történő előtanítás erőforrásigényének csökkentése mellett. Az ALBERT (A Lite BERT rövidítése) egy olyan modell, amely paraméter csökkentő technikákat foglal magába (Lan és mtsai, 2020). A magyar nyelvre történő implementáció során két előtanított, kisbetűsítés nélküli ALBERT modell készült el: az egyik a magyar nyelvű Wikipédián lett betanítva (a Webcorpus 2.0 korpusz része), a másik pedig a NYTK-BERT korpusz egy részén. Kutatásunk során a HIL-ALBERT NYTK modellt használtuk.

**HIL-ELECTRA**<sup>8</sup>: az ELECTRA (Efficiently Learning an Encoder that classifies Token Representation Accurately) (Clark és mtsai, 2020) modellek egy sikeres alternatív megoldást nyújtanak a maszkolt nyelvi modellezés (MLM) mellett azáltal, hogy felcserélt token detektálást alkalmaznak, amely egy önfelügyelő előtanítási feladat, melynek során a modell megtanulja megkülönböztetni az ere-

<sup>6</sup> <https://hilarco.github.io/models/roberta.html>

<sup>7</sup> <https://hilarco.github.io/models/albert.html>

<sup>8</sup> <https://hilarco.github.io/models/electra.html>

deti bemenetet a mesterségesen generált behelyettesítésektől. Az ELECTRA modellek a GAN (Generative Adversarial Network) módszeren alapulnak. Kísérletes eredmények azt mutatják, hogy ez a módszer hatékony és nagy teljesítményű más módszerekkel összehasonlítva. Az ELECTRA magyar nyelvű implementációja során két verzió született, az ELECTRA wiki és az ELECTRA NYTK-BERT. Az előbbi a magyar Wikipédián lett betanítva, míg az utóbbi az NYTK-BERT korpuszon. Kutatásunk során a HIL-ELECTRA NYTK modellt alkalmaztuk.

**HILBART**<sup>9</sup>: a BART (a Bidirectional and Auto-regressive Transformers) (Lewis és mtsai, 2020) alapú megközelítések komoly potenciállal rendelkező eszközök a seq2seq (sequence to sequence) előtanítás tekintetében. A BART gyakorlatilag ötvöz egy BERT (Devlin és mtsai, 2019) és egy GPT (Radford és Narasimhan, 2018) típusú modellt. A BART a szöveggenerálási feladatokban teljesít a legjobban de kiemelkedő eredményeket ér el diszkriminatív és összefoglaló feladatokban is. A BART magyar nyelvű implementációjának eredményeként jöttek létre a HILBART modellek. Ezek a HILBART large web (Webcorpus 2.0 1 százalékan tanítva), a HILBART base web (Webcorpus 2.0 10 százalékan tanítva) és a HILBART base wiki (magyar nyelvű Wikipédián tanítva). Kutatásunk során a HILBART base web modellt használtuk.

**mBERT** (Devlin és mtsai, 2019): az mBERT (multilingual BERT) struktúráját tekintve a BERT-en alapul, ugyanazon betanítási paradigmát is használja azzal a fontos különbséggel, hogy az előtanítás során 104 különböző nyelv Wikipédia cikkek szövegeit használták. Az mBERT modell alkalmazása különösen előnyös akkor, ha alacsonyan ellátott nyelvekről van szó, vagyis amikor kevés annotált mondat áll rendelkezésre az adott nyelven. Nyelvek közötti előtanító modelleket (mBERT-et is beleértve) alkalmaztak például egy névelemfelismerési feladat során magyar és ujjur nyelvekre (Chen és mtsai, 2021). Kutatásaink során a kisbetűsítés nélküli mBERT base modellt használtunk.

**XLM-RoBERTa** (Conneau és mtsai, 2020): a nyelvek közötti érthetőség (angolul Cross-Language Understanding, rövidítve XLU) elérése egy komoly kihívás és egy innovációs gyorsítóként szolgál a többnyelvű modellek esetében. 2020-ban a Facebook mesterséges intelligenciával foglalkozó csapata előállt az XLM-RoBERTa (XLM-R-ként is rövidítve) modellel, amely egy transzformer alapuló többnyelvű maszkolt nyelvi modell. A modell előtanítása során a CC-100 korpuszt használták, amely 100 különböző nyelv szövegeit tartalmazza, köztük magyar nyelvű szövegeket is (magyar nyelvű tokenek száma: 7807 millió, magyar nyelvű korpusz mérete: 58,4 GiB). A szerzők publikációja alapján az XLM-RoBERTa versenyképes eredményeket ért el számos sztenderd feladat elvégzése során olyan egynyelvű modellekkel összevetve, mint például a RoBERTa. Továbbá, XLM-R képes volt túlteljesíteni az mBERT-et nyelvek közötti osztályozásban olyan nyelvek esetében, ahol relatíve kevés a rendelkezésre álló forrásanyag. Figyelemre méltó, hogy state-of-the-art eredményeket ért el az XLM-RoBERTa XNLI, NER és nyelvek közötti válaszadási feladatokban. Kutatásunk során az XLM-RoBERTa base modellt használtunk.

<sup>9</sup> <https://hilaico.github.io/models/hilbart.html>

**fastText** (Joulin és mtsai, 2016b,a): a fastText szintén a Facebook mesterséges intelligenciával foglalkozó csapatának fejlesztése, melynek célja a szövegosztályozás és a reprezentációs tanulás elősegítése. A létrehozott paradigma azon alapul, hogy karakter  $n$ -grammokot foglal az úgynevezett skipgram modellbe, amely egy gyors és hatékony megoldást kínál anélkül, hogy előfeldolgozás vagy felügyelet szükséges lenne (Bojanowski és mtsai, 2017). Szövegosztályozás szempontjából más deep learning-alapú megoldásokkal összevethető a teljesítménye a pontosság tekintetében, és egy lényegesen gyorsabb lehetőség tanítás és kiértékelés szempontjából (Joulin és mtsai, 2017). A platformon elérhetőek szövektorok angolra és 157 másik nyelvre, ezáltal egy nagyon kézenfekvő és lehetőségekkel teli eszköznek számít a többnyelvű nyelvfeldolgozás terén.

## 5. Kísérletek

Kutatásunk során 7 különböző kísérletet végeztünk el:

- **eredeti**: minden előtanított modell finomhangolásra került az eredeti MTS korpuszon. Ezt tekintjük az alap eljárásnak.
- **zeroshot**: többnyelvű modellek képesek magyar nyelvű NLP feladatok predikciójára. Ez esetben angol nyelvű korpuszokat használtunk finomhangolásra és a rendszernek magyar nyelvű mondatokra kellett prediktálnia.
- **transzfer**: többnyelvű modellek finomhangolva lettek az SST korpuszon, majd további finomhangolásra kerültek a MTS tanító korpuszon.
- **fordított**: minden előtanított modell finomhangolva lett a magyarra fordított SST korpuszon (SST\_hu).
- **fordított+finomhangolás**: minden előtanított modell finomhangolva lett az SST\_hu korpuszon, majd ismét finomhangolva lett a MTS tanító korpuszon.
- **mix**: minden előtanított modell finomhangolva lett a konkatenált és összekevert SST\_hu és MTS tanító korpuszon, majd le lett tesztelve a MTS tesztelő korpuszon.
- **mix+finomhangolás**: minden előtanított modell finomhangolva lett a konkatenált és összekevert SST\_hu és MTS tanító korpuszon, majd újabb finomhangolás következett a MTS tanító korpuszon

Az elvégzett kísérletek eredményei a MTS tesztelő korpuszon lettek kiértékelve.

A jobb összehasonlíthatóság érdekében ugyanazokat a hiperparaméter beállításokat alkalmaztuk gyakorlatilag az összes modellre. A hiperparaméterek a következők: learning rate:  $2e-5$ , batch méret: 32 eszközönként (4 x GPU), epoch érték: 4, maximális szekvenciahossz: 128. A HILBERT modell esetében a CUDA memóriatúllépés elkerülése végett a batch méretet eszközönkénti 8-as értékre módosítottuk. Az ELECTRA modellek egyetlen GPU felhasználása mellett futottak. Végül, fastText esetében GPU nem került felhasználásra, csak CPU, a batch méret pedig 1-es értékre lett állítva. Minden kísérletnél 4 darab GeForce RTX 2080 Ti típusú videokártyát és 40 darab Intel(R) Xeon(R) Silver 4114 típusú CPU-t használtunk.

A transzformer modellek (kivével az ELECTRA) finomhangoláshoz a Hugging Face által rendelkezésre bocsátott „transformers text classification library”-t<sup>10</sup>, az ELECTRA finomhangolásához a Google által implementált kódot<sup>11</sup>, míg a fastText esetében a Facebook által közzétett eszközt<sup>12</sup> használtuk. Minden általunk használt és e publikáció keretében bemutatott modell és implementáció megtalálható a projekt honlapunkon<sup>13</sup>.

## 6. Eredmények

A 3. táblázat foglalja össze kísérleteink eredményeit. Általánosan elmondható, hogy lefordított szöveg hozzáadása a tanító korpuszhoz javítja a mondatszentiment analízis osztályozás teljesítményét. Minden egyes általunk tesztelt modell esetében valamely fordítási metódus jobb eredményeket hozott, mint az alap eljárás.

Három minőségi sáv definiálható az általunk használt modellek teljesítménye alapján. A leggyengébb rendszerek a HILBART és a fastText révén keletkeztek. Ezek a várakozásunknak megfelelő eredmények, hiszen a HILBART elsősorban szöveggenerálásra használatos, a fastText pedig egy bizonyos tekintetben elavult, statikus, nem kontextuális szórepresentációs eljárás, mely rosszabbul teljesít a kontextuális nyelvi modellekhez képest. Ugyanakkor hozzá kell tennünk, hogy a fastText alkalmazás lényegesen kevesebb erőforrást igényel a rendszer betanításához és online predikció esetén csak CPU-t használ.

A második minőségi sáv tartalmazza azokat a rendszereket, amelyek 77-80 százalékos pontossági értéket mutattak a bináris osztályozásban és 58-63 százalékos pontosságot az 5-osztályos besorolás alkalmával.

Végül három rendszer található a legfelső minőségi sávban (huBERT, XLM-RoBERTa és HILBERT modellek) 85,5 százalékos pontossággal a bináris osztályozásban és 66-69 százalékos pontossággal az 5-osztályos besorolásban. Érdekes eredményként könyvelhető el, hogy az XLM-RoBERTa többnyelvű modell jobban teljesített a MTS2 feladatban, mint a magyar nyelvre specifikus huBERT modell, amely a state-of-the-art nyelvi modell a legtöbb természetes nyelvfeldolgozási feladatban. Továbbá, a HILBERT modell szintén túlszárnyalta a huBERT modellt a MTS2 feladatban, amely szintén várt eredmény, hiszen a kevesebb betanító adatot ellensúlyozni tudta a modell nagyobb mérete, a több paraméter és a hozzáadott finomhangolási adat.

Az 1. ábrán összehasonlítottunk 5 különböző modell teljesítményét az F-mértékek tekintetében. A magyar nyelvű state-of-the-art huBERT modell, a HILBERT large modell, a nem-kontextuális fastText és két többnyelvű modell került összehasonlításra. Az egyetlen szignifikáns eredmény abban mutatkozott, hogy a huBERT és a HILBERT modellek több 1-es és 5-ös értéket prediktáltak, mint a

<sup>10</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification>

<sup>11</sup> <https://github.com/google-research/electra>

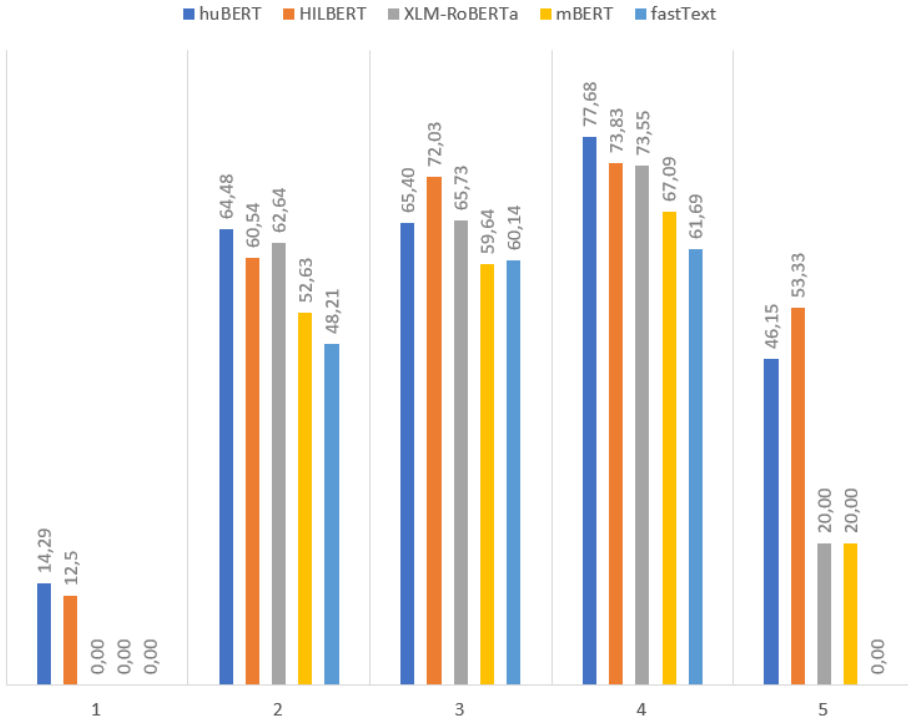
<sup>12</sup> <https://fasttext.cc>

<sup>13</sup> <https://github.com/nytud/sentiment-analysis>

		MTS2	MTS5
huBERT	eredeti	84.07	66.00
	fordított	73.33	29.25
	fordított+finomhangolás	85.55	66.50
	mix	<b>85.55</b>	<b>68.99</b>
	mix+finomhangolás	84.81	68.00
HILBERT	eredeti	83.33	68.00
	fordított	74.07	34.75
	fordított+finomhangolás	82.59	67.75
	mix	82.22	<b>68.50</b>
	mix+finomhangolás	<b>85.56</b>	68.00
HIL-RoBERTa	eredeti	75.92	59.15
	fordított	48.89	29.75
	fordított+finomhangolás	<b>79.63</b>	56.75
	mix	76.66	<b>59.25</b>
	mix+finomhangolás	77.78	57.99
HIL-ALBERT	eredeti	75.56	55.49
	fordított	52.59	28.75
	fordított+finomhangolás	77.03	56.75
	mix	72.22	60.50
	mix+finomhangolás	<b>77.41</b>	<b>60.75</b>
HIL-ELECTRA	eredeti	78.89	59.11
	fordított	55.02	37.34
	fordított+finomhangolás	<b>79.93</b>	61.15
	mix	76.58	60.90
	mix+finomhangolás	79.18	<b>62.66</b>
HILBART	eredeti	71.11	51.25
	fordított	47.77	31.00
	fordított+finomhangolás	74.07	53.25
	mix	71.48	52.50
	mix+finomhangolás	<b>76.66</b>	<b>54.75</b>
mBERT	eredeti	78.51	57.74
	zeroshot	47.41	30.50
	transzfer	78.52	57.99
	fordított	48.88	28.75
	fordított+finomhangolás	<b>79.25</b>	56.75
	mix	77.77	56.99
	mix+finomhangolás	78.89	<b>59.75</b>
XLM-RoBERTa	eredeti	83.33	63.49
	zeroshot	68.88	40.99
	transzfer	84.81	66.25
	fordított	68.51	35.25
	fordított+finomhangolás	85.18	66.00
	mix	85.18	66.25
	mix+finomhangolás	<b>85.56</b>	<b>66.50</b>
fastText	eredeti	71.9	53.2
	fordított	62.2	32.0
	fordított+finomhangolás	73.3	<b>56.2</b>
	mix	74.1	51.7
	mix+finomhangolás	<b>75.6</b>	53.5

3. táblázat. Mondatszintű szentiment analízis eredmények.

többnyelvű modellek vagy a fastText. A fastText nem prediktált sem 1-es, sem pedig 5-ös értékeket. Ez azt jelenti, hogy az 1-es és 5-ös értékek ritkán fordulnak elő a betanító korpuszban (lásd 2. táblázatban). A huBERT és a HILBERT magyar nyelvű modellek, az előbbi egy 9 milliárd tokent tartalmazó korpuszon lett betanítva, míg az utóbbi egy large modell 340 millió paraméterrel, aminek a segítségével még kifinomultabb részletek megtanulására is képesek.



1. ábra: MTS5 F-mértékek összehasonlítása az osztálycímkék függvényében.

A legjobban teljesítő magyar (huBERT<sup>14,15</sup>) és többnyelvű (XLM-RoBERTa<sup>16,17</sup>) modellek megtalálhatóak a Hugging Face oldalunkon.

<sup>14</sup> <https://huggingface.co/NYTK/sentiment-hts2-hubert-hungarian>

<sup>15</sup> <https://huggingface.co/NYTK/sentiment-hts5-hubert-hungarian>

<sup>16</sup> <https://huggingface.co/NYTK/sentiment-hts2-xlm-roberta-hungarian>

<sup>17</sup> <https://huggingface.co/NYTK/sentiment-hts5-xlm-roberta-hungarian>

## 7. Összegzés

Jelen tanulmányunk új megközelítéseket alkalmaz a Twitter közösségi média platformról származó magyar nyelvű szövegek mondatszintű szentiment analízisében. Kísérletes eredményeink alapján az a következtetés vonható le, hogy magyarra fordított szövegek hozzáadása a korpuszokhoz képes javítani a modellek teljesítményét, amely egy rendkívül fontos előrelépés a szentiment analízis folyamatának optimalizálásában. Kiemelkedő jelentőségű, hogy az általunk használt modellek jobban teljesítettek több feladatban is a jelenlegi state-of-the-art-nak számító huBERT modellnél, amely ígéretes új utakat nyit meg a jelen publikáció keretében bemutatott eredmények szélesebb körű alkalmazhatóságát illetően, elősegítve ezzel a szentiment analízishez kapcsolódó területek haladását. Mindamellet, kutatási eredményeink kifejezetten relevánsak az olyan új stratégiák kialakításában, ahol a rendelkezésre álló tanítóanyag mennyisége nem elegendő egy osztályozási modell tanítására.

## Hivatkozások

- Barrault, L., Bojar, O., Costa-jussà, M.R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Mázler, M., Pal, S., Post, M., Zampieri, M.: Findings of the 2019 conference on machine translation (wmt19). In: Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1). pp. 1–61. Association for Computational Linguistics, Florence, Italy (August 2019), <http://www.aclweb.org/anthology/W19-5301>
- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146 (2017)
- Chen, S., Pei, Y., Ke, Z., Silamu, W.: Low-resource named entity recognition via the pre-training model. *Symmetry* 13(5) (2021)
- Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training text encoders as discriminators rather than generators. In: *ICLR (2020)*
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale (2020)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 567–573. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), <https://aclanthology.org/P17-2090>



- Feldman, R.: Techniques and applications for sentiment analysis. *Commun. ACM* 56(4), 82–89 (Apr 2013)
- Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: Hilbert, magyar nyelvű bert-large modell tanítása felhő környezetben. XVII. Magyar Számítógépes Nyelvészeti Konferencia pp. 29–36 (2021)
- Hoang, M., Bihorac, O.A., Rouces, J.: Aspect-based sentiment analysis using BERT. In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. pp. 187–196. Linköping University Electronic Press, Turku, Finland (Sep–Oct 2019)
- Jo, H., Kim, S.M., Ryu, J.: What we really want to find by sentiment analysis: The relationship between computational models and psychological state (2018)
- Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016a)
- Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016b)
- Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. pp. 427–431. Association for Computational Linguistics, Valencia, Spain (Apr 2017)
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Hermann, U., Aji, A.F., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: *Proceedings of ACL 2018, System Demonstrations*. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)
- Kahla, M., Yang, Z.G., Novák, A.: Cross-lingual fine-tuning for abstractive arabic text summarization. In: *Proceedings of International Conference Recent Advances In Natural Language Processing (RANLP 2021)*. pp. 660–668. INCOMA Ltd., Shoumen, Bulgaria (2021)
- Kinholkar, S.A., Waghmare, P.K.C.: Enhance digital marketing using sentiment analysis and end user behavior. In: *International Research Journal of Engineering and Technology (IRJET)*. vol. 3 (2016)
- Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: RACE: Large-scale ReAiding comprehension dataset from examinations. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 785–794. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017)
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricic, R.: Albert: A lite bert for self-supervised learning of language representations. In: *Proceedings of the Eighth International Conference on Learning Representations* (2020)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *Proceedings of the 58th Annual Meeting of the Association for Computational*

- Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Lison, P., Tiedemann, J.: OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 923–929. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), <https://www.aclweb.org/anthology/L16-1147>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
- Lutz, B., Pröllochs, N., Neumann, D.: Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning (2018)
- Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2013)
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)
- Neri, F., Aliprandi, C., Capeci, F., Cuadros, M., By, T.: Sentiment analysis on social media. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 919–926 (2012)
- Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). pp. 79–86. Association for Computational Linguistics (Jul 2002), <https://www.aclweb.org/anthology/W02-1011>
- Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002), <https://www.aclweb.org/anthology/P02-1040>
- Poncelas, A., Shterionov, D.S., Way, A., de Buy Wenniger, G.M., Passban, P.: Investigating backtranslation in neural machine translation. In: Proceedings of the 21st Annual Conference of the European Association for Machine Translation. pp. 249–258. Alacant, Spain (2018)
- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 27–35. Association for Computational Linguistics, Dublin, Ireland (Aug 2014)
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)

- Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016)
- Saura, J.R., Palos-Sanchez, P., Grilo, A.: Detecting indicators for startup business success: Sentiment analysis using text data mining. *Sustainability* 11(3), 917 (Feb 2019), <http://dx.doi.org/10.3390/su11030917>
- Steinberger, J., Lenkova, P., Kabadjov, M., Steinberger, R., van der Goot, E.: Multilingual entity-centered sentiment analysis evaluated by parallel corpora. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2011. pp. 770–775. Association for Computational Linguistics, Hissar, Bulgaria (Sep 2011), <https://www.aclweb.org/anthology/R11-1113>
- Szabó, M.K., Vincze, V., Simkó, K.I., Varga, V., Hangya, V.: A Hungarian sentiment corpus manually annotated at aspect level. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 2873–2878. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016)
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018)
- Yang, Z.G., Agócs, Á., Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (2021)



## Magyar páciensek narratív tapasztalatainak elemzése BERT témamodellezéssel és szentimentelemzéssel

Osváth Mátyás<sup>1</sup>, Yang Zijian Győző<sup>2</sup>, Kósa Karolina<sup>1</sup>

<sup>1</sup> Magatartástudományi Intézet, Általános Orvostudományi Kar, Debreceni Egyetem,  
osvath.matyas@med.unideb.hu

<sup>2</sup> Nyelvtudományi Kutatóközpont, Sightspot Network Kft.,  
yang.zijian.gyozo@nytud.hu

<sup>1</sup> Magatartástudományi Intézet, Általános Orvostudományi Kar, Debreceni Egyetem,  
kosa.karolina@med.unideb.hu

**Kivonat:** A pácienseknek a betegellátás során szerzett tapasztalatai fontos információkkal szolgálnak az ellátás problémáiról, amelyek elemzésével és felhasználásával az egészségügyi ellátás minősége javítható. A páciens-élményeket vizsgáló kutatások száma folyamatosan növekszik, azonban hazánkban kevés kutatás használ erre természetesnyelv-feldolgozáson alapuló módszereket. A jelen vizsgálat célja a hazai betegellátásban szerzett tapasztalatokról szóló írásokban megjelenő témák azonosítása és az írások érzelmi polaritásának meghatározása volt. Egy hazai online fórumon 2009 és 2020 közt megjelent, Web Scaper-rel letöltött 1663 blogbejegyzést elemeztünk témamodellezéssel és szentimentelemzéssel, előtanított huBERT és egy finomhangolt HIL-SBERT transzformer modell segítségével. A modell 326 és 200 témát azonosított, amelyeket manuális elemzéssel kategóriákká vontuk össze. A modell az írások 94,4%-át negatívan osztályozta, hasonló tendenciát mutatva az éves bontásra is. A szóbeágyazási BERT modellen alapuló témamodellezéssel eredményesen kinyerhetők a betegellátásban szerzett tapasztalatokat leíró szövegekben megjelenő mintázatok. Az eredmények minőségét azonban befolyásolja a korpusz nagysága, azaz a blogbejegyzések száma és terjedelme.

### 1. Bevezetés

A betegellátás teljesítményének egyik indikátora az ellátás minősége. A beteget közép-pontba állító, magas minőségű ellátás az egészségpolitikai ágazat és a betegellátást végző intézmények számára is fontos prioritás, melynek mérésére több, különféle indikátort tartalmazó elméleti keretrendszert fejlesztettek ki. Avedis Donabedian 1966-ban megjelent úttörő közleménye az egészségügyi ellátás minőségét a struktúra, folyamat és kimenet dimenziókba sorolható indikátorokkal javasolta jellemezni (Donabedian, 2005). A Gazdasági Együttműködési és Fejlesztési Szervezet (OECD) 2006-ban tette közzé az egészségügyi rendszer teljesítményének mérésére javasolt keretrendszerét többdimenziós mátrix formájában, amely az ellátás Donabedian-i minőségét továbbfejlesztve tartalmazza (Carinci és mtsai., 2015; Mattke és mtsai., 2006). A mátrix egyik fontos dimenzióját a betegtapasztalatok alkotják, amelyek a páciensekkel az ellátás során egyénekenként történő események eredményeként alakulnak ki, jelentősen

befolyásolva a pácienseknek az ellátásról alkotott véleményét. E tapasztalatok közé tartozik egyebek közt a személyzet és beteg közötti kommunikáció, az ellátáshoz való hozzáférés és várakozási idő. A páciensek tapasztalataitól el kell különíteni a szakirodalomban sokszor szinonimaként megjelenő, de nem azonos indikátort, a betegelégedettséget, amely a beteg előzetes elvárásai és a ténylegesen kapott ellátás közötti diszkrépancia mértékét mutatja (Jenkinson és mtsai., 2002).

A betegtapasztalatok nemzetközi felmérése és összehasonlítása érdekében az OECD összesen 11 konkrét, úgynevezett „Patient-reported experience” mutatót (PREM) határozott meg. Ezek olyan indikátorok, amelyek kiemelkedő jelentőségűek a páciens tapasztalata és az ellátás kimenete szempontjából egyaránt, mint például az orvosnak a pácienssel töltött elegendő ideje, a beteg megfelelő tájékoztatása az orvos által, a beteg bevonása a döntéshozásba, vagy a felírt, de anyagi okok miatt ki nem váltott gyógyszerek aránya (Brito Fernandes és mtsai., 2019). A konkrétan meghatározott indikátorok révén a betegellátás minőségi dimenziói nemcsak országokon belül, hanem országok közt is összehasonlíthatóvá válnak.

Az indikátorokhoz kvantitatív, például zárt kérdéseket tartalmazó kérdőívekkel, és kvalitatív módon, például szövegelemzéssel is lehet adatokat gyűjteni. A kutatások és rendszeres felmérések legnagyobb része kvantitatív módszereket alkalmaz (pl. az USA-ban a Hospital Consumer Assessment of Healthcare Providers and Systems, HCAHPS, (Giordano és mtsai., 2010), azonban az utóbbi időben egyre nagyobb az érdeklődés a páciensek narratív visszajelzései iránt is. Az USA-ban több szervezet működtet olyan honlapokat (RateMD, Healthgrades, Yelp), amelyeken páciensek szöveges visszajelzést adhatnak egy meghatározott intézményben kapott ellátásról, kvantitatív és kvalitatív formában. A brit Nemzeti Egészségügyi Szolgálat (NHS) online portálján is lehetőség van szöveges vélemény írására (Lagu és mtsai., 2013).

Nemzetközi kutatások szerint a betegeknek az orvosokról online írt értékelései és az azokra vonatkozó hozzászólások befolyással vannak az orvosválasztásra. A páciensek 59%-a tartja fontosnak az orvosokról írt értékeléseket (Burkle & Keegan, 2015; Hanauer és mtsai., 2014). Az online térben elérhető vélemények, értékelések és hozzászólások egyre nagyobb jelentőséggel bírnak, mivel számuk növekszik, és egyre többen használják ezeket az internetes forrásokat és értékeléseket a döntéshozáshoz.

Hazánkban több kutatás irányult páciens tapasztalatok felmérésére kérdőíves módszerekkel (Brito Fernandes és mtsai., 2019; Fernandes és mtsai., 2020; Lucevic és mtsai., 2019). Hazai és nemzetközi eredmények szerint is összefüggés van a páciensek pozitív tapasztalatai és az alap- és járóbeteg ellátás használata között (pl. kórházi felvételek száma és kórházba való visszatérések száma) (Doyle és mtsai., 2013; Price és mtsai., 2014). A magasabb betegelégedettséggel jellemzett kórházakban nagyobb mértékű a betegek adherenciája, kisebb a halálozás, és rövidebb időt töltenek az intézményben a betegek (Bidmon és mtsai., 2020; Nawab és mtsai., 2020).

A kvantitatív módszerek alkalmazása dominánsnak tekinthető a betegek tapasztalatainak és élményeinek értékelésére. Ezek a módszerek azonban az információvesztésből fakadóan szükségszerűen korlátozottak, és érthetőségük is alacsonyabb mértékű a dolgozók számára. A szöveges (narratív) visszajelzések jobban érthetőek az érintetteknek, de ezek elemzése sokkal bonyolultabb, az intézmény(ek) egészére vonatkozó következtetéshez pedig nagy számú narratívára van szükség. Ugyanakkor az ellátottaktól érkező szöveges visszajelzések részletes és alapos elemzése iránt egyre nagyobb az igény ezek információgazdagsága miatt (Khanbhai és mtsai., 2021) (Emmert és mtsai.,

2014; López és mtsai., 2012; Zhang és mtsai., 2018). A nagy mennyiségű, strukturálatlan szövegek azonban megnehezítik, sőt egy bizonyos volumenen túl lehetetlenné teszi a manuális elemzést. A probléma megoldását a természetesnyelv-feldolgozás mint a mesterséges intelligencia egyik részterületének robbanásszerű fejlődése hozta el, amely lehetővé tette nagy mennyiségű szövegek gépi elemzését.

### 1.1. Kapcsolódó irodalom

Greaves és munkatársai a páciensek hozzászólásait szentimentelemzéssel vizsgálták, valamint gépi tanuláson alapuló modelleket alkalmaztak különböző kategóriák előrejelzésére, mint például a higiénia, tisztaság és a beteggel szembeni tisztelet (Greaves és mtsai., 2013). Doing-Harris és munkatársai gépi tanuláson alapuló megközelítést használtak betegeknek az egészségügyre vonatkozó hozzászólásaiban rejlő témák azonosítására. Az elemzés során a kórházi időpont elérhetősége, az empátia és az orvos által nyújtott érthető magyarázat bukkantak fel, mint a leggyakoribb kategóriák (Doing-Harris és mtsai., 2016). Li és munkatársai több mint 122 ezer, kínai orvosokat véleményező szöveget kívántak rendszerezni irodalmi áttekintést, manuális annotációt és természetesnyelv-feldolgozást alkalmazva. Az elemzés eredményeként olyan rendszert hoztak létre, amely három fő (orvossal, beteggel, rendszerrel kapcsolatos témák) és 9 alkategóriát tartalmazott (Li és mtsai., 2018). 2021-ben jelent meg az első olyan rendszeres áttekintés az orvosi szakirodalomban, amely betegek által írott szabad szövegeket elemző, az elmúlt 20 évben megjelent, természetesnyelv-feldolgozást és gépi tanulást alkalmazó, összesen 19 tudományos közleményt foglalt össze (Khanbhai és mtsai., 2021). Ezek között hat olyan közlemény volt, amelyben nem felügyelt gépi tanuláson alapuló modelleket, specifikusan témamodellezést használtak a szövegben és dokumentumokban megjelenő témák és kategóriák automatikus azonosítására. Tudomásunk szerint egy kutatás alkalmazta a legkorszerűbb word2vec és kontextuális BERT modelleken alapuló módszereket a páciensek szövegeinek feldolgozására és az információk kinyerésére, mint például a betegség súlyossága, időbeli hossza és fennállása (Saha és mtsai., 2020).

Hazánkban tudomásunk szerint eddig nem történt olyan vizsgálat, amelynek fókuszában szöveges páciens-élmények elemzése állt volna természetesnyelv-feldolgozással.

### 1.2. Célkitűzés

A jelen tanulmány célja az volt, hogy a hazai betegellátásban szerzett tapasztalatokat bemutató spontán, nem strukturált írásokban rejlő jellemző információkat természetesnyelv-feldolgozás segítségével kinyerjük. A betegtapasztalatok alapján meghatározható mintázatok és témák automatizált kinyerésével a magyar betegellátás minőségének fejlesztéséhez szeretnénk hozzájárulni. Két kontextuális BERT modell került felhasználásra a szövegek enkódolására, azután nem felügyelt gépi tanulási algoritmussal klaszterelemzést végeztünk. További célunk volt finomhangolt BERT modellek segítségével szentimentelemzést végezni az írások érzelmi polaritásának (pozitív, negatív, semleges) meghatározására.

## 2. Módszerek

### 2.1. Adatgyűjtés

A betegek vagy hozzátartozóik által spontán írott szövegek egy magyar online egészség-fórumról kerültek letöltésre (<https://praxis.blog.hu/>). A magánkézben lévő platform 2009 óta működik, napi nézettsége 5000-7000 látogató között van.

A hazai betegellátásban szerzett tapasztalatokat leíró, spontán írott szövegeket a páciensek vagy hozzátartozóik a platform üzemeltetőinek küldik meg emailben, melyek cenzúrázás és szerkesztés nélkül kerülnek feltöltésre, év és hét szerint rendezve. A posztokhoz bárki fűzhet kommentárt. Az írások terjedelme nincs korlátozva. Az adatok nyilvánosan elérhetőek és szabadon hozzáférhetőek, ezért nem szükséges beleegyező nyilatkozat és etikai engedély a felhasználásukhoz. A jelenlegi kutatás keretében a portálon 2009 és 2020 közt feltöltött, páciensek tapasztalatait narratívan ismertető blogbejegyzéseket elemeztük. A blogbejegyzéseket egy keresőrobot segítségével töltöttük le. Minden, személyes azonosításra szolgáló szentitív információt eltávolítottunk. A megtisztított nyers szöveg 1660 blogbejegyzésből, 54,843 mondatból és 873,442 szóból állt, blogbejegyzésenként átlagosan 532,6 szóval.

### 2.2. Adatelemzés

Az írások numerikus adatokká való transzformálására szóbeágyazási modelleket alkalmaztunk. Ezek a kontextuális hierarchia megtartásával a szavakat vektorrepresentációkká konvertálják, amelyekből távolsági mutatókat számíthatók (pl. koszinusz távolság). Korpusz elemzése érdekében gyakran kerülnek felhasználásra mélytanulási modellek - mint például a word2vec - a jellemzők kinyerésére és kontextus figyelembevételére, azonban a nagyon specifikus kontextusban problémák merülhetnek fel. Ennek elkerülésére a „Bidirectional Encoder Representations from Transformer” (BERT) modellt használtuk, ami kétirányú nyelvmoddellen alapuló, ún. kontextualizált szóbeágyazás, az egyes szavakhoz kontextusfüggő jellemzővektort rendelve (Devlin és mtsai., 2019). A BERT modell előtanítása és finomhangolása nagy méretű korpuszt igényel, azért, hogy különböző kontextusokban is érzékeny legyen.

A jelen tanulmány Angelov (2020) korábbi munkájára épít, amely a legkorszerűbb transzformer modellen alapuló szóbeágyazásokat alkalmazza a témamodellezéshez. A szövegekből kinyert mondatok kódolására kétféle BERT modellt használtunk. Az első, huBERT modellhez – a tokenizálást leszámítva – nincs szükség a bemeneti szövegek előfeldolgozására, mivel a modellt a 9 milliárdos tokenizált Magyar Webkorpusz 2.0 változatán tanították elő (Nemeskey, 2020), teljesítményben felülmúlva más magyar nyelvi tudással rendelkező BERT modelleket több nyelvtechnológiai feladatban (Nemeskey, 2019). A huBERT modell architektúrája ugyanaz, mint a BERT-base modell. 12 kódoló réteggel és 12 figyelmi fejjel rendelkezik, és 768 rejtett feldolgozó elemmel rendelkezik (Ács és mtsai., 2021). HIL-SBERT modellt is alkalmaztunk a



nyelvi jellemzők kinyerésére<sup>1</sup>, aminek célja a magas minőségű mondatbeágyazások elérése (Reimers & Gurevych, 2019).

A témamodellezésre használt, nem felügyelt klaszterelemzésre alkalmas algoritmus eredményeinek javítása érdekében elsőként dimenzióredukcióra került sor, amihez az „Uniform Manifold Approximation and Projection for Dimension Reduction” (UMAP) algoritmust használtuk (McInnes és mtsai., 2018). Következő lépésben egy sűrűség alapú térbeli klaszterezési algoritmust (HDBSCAN) alkalmaztunk az alacsony dimenziójú vektorreprezentációkra, hogy az adatstruktúrát felhasználva klasztereket, majd azokból témákat alkossunk. A HDBSCAN egy sűrűség alapú, hierarchikus klaszterelemzésre alkalmas nem parametrikus algoritmus, amely optimális megoldást kínál a futásidőre, valamint a klaszterek számát nem szükséges előre meghatározni (McInnes és mtsai., 2017). A modellt 15 szó per klaszterre állítottuk. A klaszterekben lévő szavakat lemmatizáltuk (spaCy 0.3.1 magyar modell<sup>2</sup> segítségével) és az NLTK<sup>3</sup> magyar stopwords-listában szereplő szavakat töröltük, valamint a duplikált témákat és a null súllyal rendelkező témákat eltávolítottuk a könnyebb értelmezhetőség érdekében. A klaszter eredményeinek értelmezéséhez, illetve a témák azonosításához első körben c-TF-IDF<sup>4</sup> (class-based TF-IDF) automatikus módszerrel végeztünk témakiválasztást, majd ezt követően humán interpretáció volt szükséges. Egyes hasonló jelentésű témákat kategóriákká vontuk össze.

A blogbejegyzésekből kinyert mondatok emocionális töltetének (pozitív, negatív és semleges) meghatározására szentimentelemzést alkalmaztunk. Az előtanított huBERT modell finomhangolására a Magyar Twitter Szentiment Korpuszt használtuk<sup>5</sup>. A nemzetközi irányelveknek megfelelően két részkorpuszt készítettünk (Wang és mtsai., 2018), amelyek a szövegek emocionális tartalmának prediktálására szolgáltak.

- Bináris (kétosztályos) korpusz: Az ötfokozatú skálából az 1 és 2 értékeket negatív, a 4 és 5 értékeket pozitív kategóriába konvertáltuk, a közepső (3) értéket figyelmen kívül hagytuk. Tanító anyag: 2,468 szegmens. Teszt anyag: 269 szegmens.
- Háromosztályos korpusz: Az ötfokozatú skálából az 1 és 2 értékeket negatív, a 3-as értéket semleges, a 4 és 5 értékeket pedig pozitív kategóriába soroltuk. Tanító anyag: 3,600 szegmens. Teszt anyag: 400 szegmens.
- Ötosztályos korpusz: az eredeti ötfokozatú skálán jelölt korpusz, ahol az 1 – nagy negatív, 2: negatív, 3: semleges, 4: pozitív, 5: nagyon pozitív. Tanító anyag: 3,600 szegmens. Teszt anyag: 400 szegmens.

A finomhangolást az alábbi módosított hiperparaméterekkel végeztük: 8-as batch méret / GPU (4 darab GeForce GTX 1080Ti); tanulási ráta:  $2e-5$ ; maximális szekvenciahossz: 128; epoch 15 (legjobb eredményt elérő modelleket alkalmaztuk). A modellek pontossága 85,92% -os volt a bináris osztályozásnál, 72,18% a háromosztályos és 68,50% az ötosztályos korpusznál.

<sup>1</sup> <https://hilanco.github.io/models/sbert.html>

<sup>2</sup> <https://github.com/spacy-hu/spacy-hungarian-models>

<sup>3</sup> <https://www.nltk.org/>

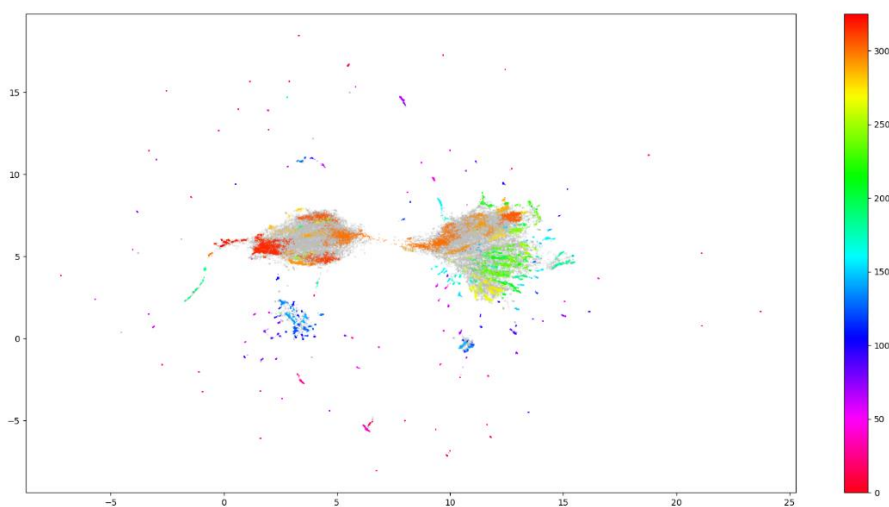
<sup>4</sup> <https://github.com/MaartenGr/cTFIDF>

<sup>5</sup> Magyar Twitter Szentiment Korpusz. Letöltve: 2021. november 3. <http://opendata.hu/dataset/hungarian-twitter-sentiment-corpus>.

### 3. Eredmények

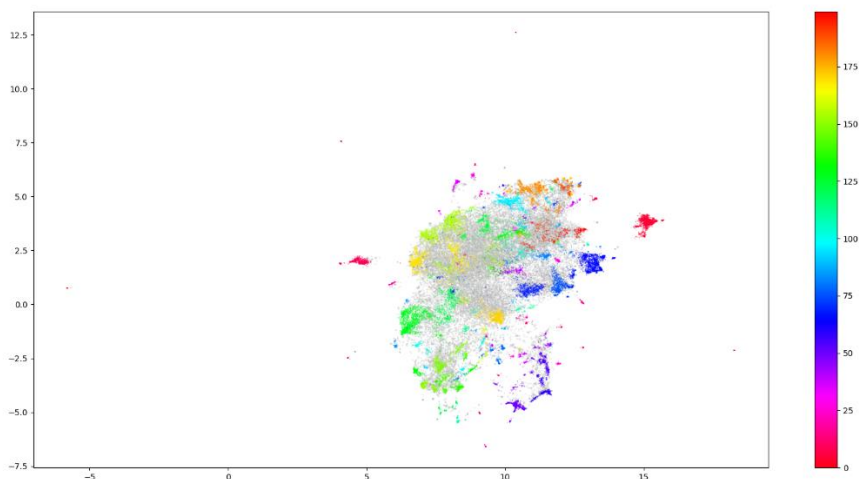
#### 3.1. Témamodellezés

A klaszterelemzés során a finomhangolt BERT modell szemantikailag értelmezhető témákat eredményezett. A huBERT és HIL-SBERT által feldolgozott mondatok alapján 326 és 200 téma került meghatározásra. Érdekes módon a huBERT modell csak 20,503 mondatot, míg a HIL-SBERT modell 20,502 mondatot használt fel az elemzés során, a maradékot zajként (huBERT: 33,744, HIL-SBERT: 34,745) értelmezve. A klaszterek vizualizációjára dimenziócsökkentést alkalmaztunk, a mondatrepresentációkat vektortérben ábrázolva. A színes területek a HDBSCAN algoritmus által azonosított mondatok sűrűségét mutatják, míg a zaj szürke színeként van ábrázolva (1. és 2. ábra).



1. ábra: UMAP-redukált mondatvektorok a huBERT modellre.

Kvantitatív metrikát nem találtunk a két modell kiértékeléséhez, ezért manuálisan vizsgáltuk meg a két modell által kiadott eredményeket. Azt tapasztaltuk, hogy a huBERT modell kimenetét könnyebb volt értelmezni, ezért a további elemzéshez a huBERT modell által azonosított témákat használtuk fel. A modell eredményeként meghatározott 326 téma közül kizártuk azokat, amelyek az elemzés szempontjából nem relevánsak. A megmaradt témák nagy száma miatt humán interpretáció volt szükséges ahhoz, hogy az összetartozó témákat kategóriákká csoportosítsuk. A kategóriák a szerzők közötti iteratív megbeszélések révén kerültek kiválasztásra a torzítás csökkentése érdekében. A mondatok nagymértékben függenek a betegellátásban megélt konkrét helyzettől, valamint a blogbejegyzésekben található további témáktól, mivel egy blogbejegyzés számos témát említhet. A témák nagy száma miatt a továbbiakban csak az öt legjelentősebb kategóriát mutatjuk be (1. táblázat), illetve a modell által fontosnak jelölt, 2. táblázatban látható 18 témát.



2. ábra: UMAP-redukált mondatvektorok a HIL-SBERT modellre.

## Diagnózis, tünetek és betegségek

Ez a kategória a páciensek vagy hozzátartozóik által észlelt tüneteket, a betegségeket és a kapott diagnózisokat tartalmazza. A tünetek akut és krónikus betegségekhez és diagnózishoz kapcsolódtak, az influenzától a gyomordaganatig terjedve. A kategóriához számos téma tartozott, mivel a modell több betegségekategóriát ismert fel és különített el. Ebbe a kategóriába tartoztak például a fogászati, gerinc- és szemészeti problémák, nemi betegségek, valamint a plasztikai beavatkozások. A témák között többször megjelenő fájdalom kifejezés a betegséggel való nehéz állapotra és megküzdésre utalhat. Például, az egyik páciens így érzett: „..., *leírhatatlan volt a fájdalom, hisz nem érzéstelenítettek.*”, illetve egy másik az alábbi módon fogalmazott: „*A diagnózis már itt gennyes agyhártyagyulladás gyanúja volt, valószínűleg helyesen...*”.

## Nőgyógyászat, szülés és születés

A nőgyógyászattal, szüléssel és születéssel kapcsolatos élményeket és eseményeket a BERT modell markánsan differenciálta. A témák visszatérő előfordulása indokoltá tette a témák kategóriává való összevonását. Ez a kategória az alábbi témákat foglalta magában: *nőgyógyászati vizsgálatok, szülés, terhesség, abortusz, kisbaba, újszülött, ol-tás és védettség.* A modell olyan témát is azonosított, mint a szülés utáni lehangoltság és postpartum depresszió. Olyan mondatok tartoztak ide, mint például „*A szülés utáni időszak az anyák életébe paradox módon nem hoz feltétlenül boldogságot.*” és „*A szülés maga simán ment, az orvos korrekt volt.*”

## Család és gyermek

A kategória a családok életében nagy valószínűséggel megjelenő témákat foglalta magában, úgymint a *gyerek hospitalizációja, aggodás a gyermekért, családi és szülői szerepek, gyermek védőoltása, és egyéb hozzátartozók állapota*. Megemlíthető, hogy több blogbejegyzés számolt be az iskolában és iskolaorvossal kapcsolatos eseményekről is. A kategória előbbtől való szétválasztását indokolja, hogy míg az előbbi az anyára fókuszál, addig az utóbbi témákban többször jelennek meg a gyermekkel és más hozzátartozókkal kapcsolatos szavak. Egy hozzátartozó így fogalmazott: „*Egy gyermekkel, főleg, ha beteg, türelmesen és odafigyeléssel kell bánni.*”

## Struktúra

A páciensek több olyan témát tárgyaltak, amelyek besorolhatók az ellátás szervezeti alapját képező struktúra kategóriába, összhangban a Donabedian által korábban megalkotott dimenzióval. A modell által azonosított témákban megjelennek szakmai problémák, a humán erőforrás hiánya, a betegellátás és a politika kapcsolata, valamint az ellátórendszer nem megfelelő működése. Több utalás történt a kórházakban és osztályokon tapasztalt nem megfelelő higiéniai körülményekre is. Egy páciens következőképp fogalmazott: „*Azért gondoltam ezt megosztani veletek, mert most tényleg mindenhol káosz, zűrzavar, csőd, orvoshiány van.*”

## Ellátás kimenete

Ebbe a kategóriába soroltuk a kedvező és a kedvezőtlen kimeneteket egyaránt. A kedvezőtlen kimenethez olyan témák kapcsolódtak, mint a haldoklás, halál, gyász, a gyógyulás bizonytalansága. A modell több kedvezőtlen kimenetű témát azonosított. Például egy hozzátartozó ezt írta: „*A család úgy véli, a kórházban mulasztás történt, és gyógyszer-tuladagolás vezetett szerettük halálához.*” A kedvező kimenet a *gyógyulás és hála* témákban megjelenő kifejezéseket foglalta magában, valamint utalásokat a személyzet által nyújtott minőségi ellátásra. Egy beteg az alábbi módon fejezte ki: „*A többieknek, akik minden nehézség ellenére emberségesek, ezer hála.*”

#	Kategória	Gyakori szavak
1	Diagnózis, tünetek és betegségek	gerinc, derék, fogfájás, herpesz, fájdalom, porckorongsérv, infúzió, panasz, szklerózis, zsibbadás, kettős látás, izomfájdalom
2	Struktúra	nővérhiány, szervezetlenség, orvoshiány, megbánt, ellátórendszer, igazságtalan, szakmai, fejetlenség, nagyfokú,
3	Ellátás kimenete	munka, osztály, hálás, szeret, nővér, dolgozó, kötelék, szívvel, gyász, ember, gyászmunka, halál, halott, folyamat
4	Nőgyógyászat és szülés	anyuka, nőgyógyász, gyermekágy, szülőszoba, szülés, abortusz, ribanc, oltás, védettség
5	Család és gyerek	kisgyermek, ágy, intelligencia, igény, fejlődés, gyermek, apa, szerep, anya, nyelv, szülő, hordozó

**1. táblázat.** Páciens tapasztalatok jelentőségteljes kategóriái.

#	Téma	Kulcsszavak
1	Orvosi eskü	Eskü, hippokratészi, ha, orvosi, méltó, belátás, rész, elismerés, teljesít
2	Gyógyszerek	Frontin, hormon, escitilt, hatóanyag, acetil, torokfájás, ibuprof, tableta, paracetamol, szer, szopogatótableta
3	Védőoltás	Oltás, oltópont, elleni, influenza, védettség, védőoltás, patika, beadás
4	Vér	Vér, véradás, véradó, hív, potenciális, vérnyomás
5	Alternatív gyógymód	Magnester, hosszú, hatás, kezdeti, alternatív, romlás, gyógymód, különféle, váratlan, javulás
6	Isten és tudomány	Isten, ezo, tudomány, szélsőséges, orvostudomány, ezotéria, piederstál,
7	Plasztikai beavatkozások	Plasztikai, borzalmas, műtét, mellő, részletez, hét, rend, cső, cm, pár, sebész, mellnagyobbító
8	Egyéb egészségügyi ellátás	Megnyugtató, diszpécser, mentős, mosoly, türelmet, beteg, kocsi, nyugtat, ápol, szakszerű
9	Ünnepek és vallás	Koszorú, adventi, gyertya, tűzhely, keresztény, hangulat, mitológia, díszít, fenyőfa
10	Egészséges életmód	Étrend, étel, diétás, változatos, hétféle, táplálkozástudományi, magnézium, stressz, munkahelyi, szervezet, magnéziumhiány, b6, stresszes
11	Edzés	Edzés, mozgás, amplifitness, megfelelő, edzésterv, fitnessterem, izomcsoport, testébresztő, készülék
12	Mellékhatás(ok)	Mellékhatás, truxal, generalizált, rulid, pszichotikus, paranoid, szorongó
13	Dohányzás	Dohányzás, jár, tapasztalat, bárki, orvos, sokan, visszaszokik, szenvedélybetegség
14	Állapotközlés	Orvos, állapot, közöl, ct, testvér, hány, nap
15	Beutaló	Időpont, vizsgálat, kér, lap, beutaló, ad, tud, doktornő
16	Tapasztalattmegosztás	Történet, szeret, megoszt, tapasztalat, mesél, blog, kórház, ír, leír
17	Pénz és anyagiak	Tb, fizet, ft, adó, rokkantjáraadék, segély, fizetés, támogat, pénz
18	Várakozási idő és időpont	Három, hét, nap, múlva, óra, időpont, kap, év, műtét, vizsgálat, is, kerül, vár, sor, múlva, fél, év, várakozás

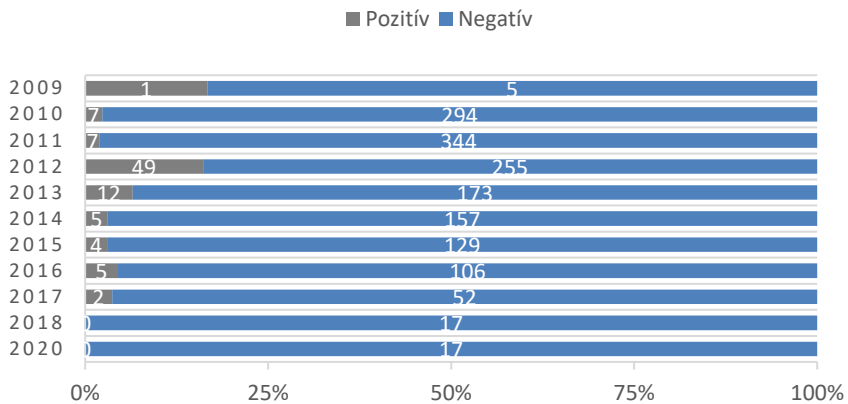
**2. táblázat.** Azonosított témák és kulcsszavak.

A további témák a 2. táblázatban láthatóak. A páciensek és hozzátartozóik számos egyéb témáról számoltak be a betegellátással kapcsolatban, mint vér, védőoltás, gyógyszerek és mellékhatásaik, alternatív gyógymódok, vallás és tudomány, dohányzás, életmód és edzés. A megjelenő témák egyedisége miatt további kérdések merülnek fel,

amelyek további vizsgálatokkal válaszolhatók meg, amelyek például az alternatív gyógymódokra, az egészséges életmódra, valamint a vallás és Isten szerepére vonatkozik az orvostudományban és egészségügyi ellátásban.

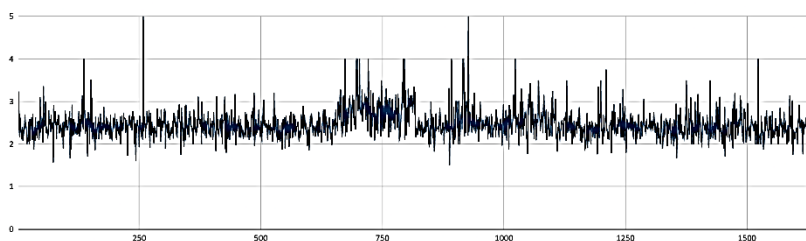
### 3.2. Szentimentelemzés

A páciens tapasztalatok osztályozása során a finomhangolt huBERT modell a 10 éves periódus során az ötosztályos változat esetében a tapasztalatok 94,4%-át negatívan osztályozta. Hasonló mintázat mutatkozott a másik két változattal is. Az ötosztályos modellben a vágóértéket a 3-as, vagyis a semleges értéknél határoztuk meg a polaritás eldöntéséhez. Egy páciens tapasztalatban több emocionális kifejezés is megjelenhetett, mivel az ellátás különböző aspektusai és folyamata során a páciens az eseményeket különbözőképpen észlelheti, illetve ugyanaz a kórházi ellátás az egyik páciensnek pozitív, míg egy másiknak negatív tapasztalattal végződhet. A páciens tapasztalatokat 10 éves periódusra osztva az ötosztályos modell alapján az íráskor valenciája hasonló tendenciát mutat, többségében negatív szentiment felé tolódva (3. ábra).

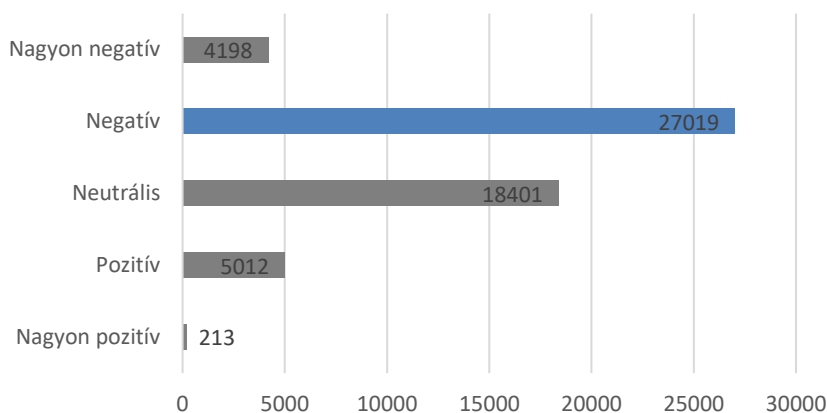


3. ábra. Szentimentelemzés a 10 éves periódusra.

A blogbejegyzések emocionális mintázatának évek során történő változására trendelemzést alkalmaztunk. Az osztályozott mondatok szentiment értékeit egy blogbejegyzésre átlagolva és évek szerint sorba rendezve ábráztuk. Amint a 4. ábrán látható, a blogbejegyzések legnagyobb része negatív szentimentet tükröz. Hasonló mintázat látható a 5. ábrán, amelyen az osztályozott mondatok érzelmi töltetének összesített eloszlását ötfokozatú Likert skálán ábráztuk.



**4. ábra:** Az évek szerint sorbarendezett blogbejegyzések megoszlása egy átlagos szentiment érték szerint.



**5. ábra:** Az osztályozott mondatok összesített eloszlása.

Az eredmények azt mutatják, hogy a fórumra beküldött páciens tapasztalatok túlnyomó többsége negatív. Ez valószínűleg részben a negativitási hibával hozható összefüggésbe, vagyis azzal a pszichológiai jelenséggel, hogy a negatív események érzelmi töltése és emlékképei intenzívebbek, mint a pozitív eseményeké, és ezek hosszabb ideig tartó hatást fejtenek ki (Ito és mtsai., 1998).

A negatív szentimentek dominanciája arra is utalhat, hogy ez a nyilvános fórum azokat a betegeket vonzza, akik nem tudnak vagy nem akarnak a betegellátásban létező legális úton, a betegjogi képviselőknél panaszt tenni. Fontos megemlíteni, hogy az elemzésre használt adatok forrása miatt az eredmények nem alkalmasak arra, hogy azokból bármilyen általános következtetést lehessen levonni a hazai betegellátás minőségére. Sőt, azt figyelembe véve, hogy évente több mint egymillió ápolási nap kerül elszámolásra az összes hazai fekvőbeteg-intézményben<sup>6</sup>, a 10 év alatt beküldött 1663, zömmel negatív betegtapasztalat akár kevésnek is tartható.

<sup>6</sup> Nemzeti Egészségbiztosítási Alapkezelő - Fekvőbeteg Statisztikák. Letöltve: 2021. november 21  
[http://www.oep.hu/akadalymentes/virtualis\\_rovat/altfin\\_virt\\_dok2/besorolo/fekvo\\_stat?pagenum=2](http://www.oep.hu/akadalymentes/virtualis_rovat/altfin_virt_dok2/besorolo/fekvo_stat?pagenum=2).

A narratívákban azonosított témák és kategóriák a betegellátásnak a páciensek által fontosnak vélt, gyakran előforduló problémáit mutatják. A panaszokban azonosított témák közötti differenciálás nehéz, mivel egy beteg több problémát említhet, valamint egy probléma leírása több mondaton keresztül is történhet.

Kutatásunk fontossága abban rejlik, hogy ráirányítja a figyelmet a betegek élményeiről spontán írt szövegek elemzésének fontosságára, mivel az lehetőséget ad a jellemzően legnagyobb mértékű frusztrációt és elégedetlenséget okozó konfliktushelyzetek és témák feltárására. A betegek tapasztalatainak elemzése – különösen nagyobb volumenben – fontos adalékkal szolgálhatna az ellátás minőségének javításához, különösen ha az elemzés nevekre és lokalizációkra is kiterjeszhető lenne.

A jelen tanulmányban ismertetett kutatás kezdeti eredményeket mutat be, amelyek a BERT témamodellezés alkalmazhatóságát támasztják alá páciensek narratíváinak automatikus elemzésére. További kutatások szükségesek a módszertani részletek kidolgozására, így például a témák megfelelő számának megállapítására vagy a témák közötti differenciálásra a pontosabb eredmények érdekében.

#### 4. Következtetés

A páciensek tapasztalatainak feltárása és megértése kulcsfontosságú az egészségügyi ellátás javításához, mivel azok egyrészt tükrözik az ellátás minőségét, másrészt befolyással vannak más jelenlegi és leendő betegek attitűdjeire és véleményeire, valamint az ellátással kapcsolatos döntéseikre.

Kutatásunk során egy évtized alatt keletkezett beteg-narratívák automatizált feldolgozását és elemzését végeztük el. Témamodellezést alkalmaztunk BERT alapú modellekkel a nagy gyakorisággal előforduló témák és kategóriák azonosítására. Az írásokból öt jelentőségteljes kategóriát azonosítottunk: 1) diagnózis, tünetek és betegségek; 2) struktúra; 3) ellátás kimenete; 4) család és gyerek; és a 5) nőgyógyászat és szülés. A kategóriák közül kettő, a struktúra és az ellátás kimenete megegyezik a Donabedian-féle modell két dimenziójával, ezért is alkalmaztunk azonos elnevezést. Emberi interpretációt használtunk a modell teljesítményének értékelésére, valamint a témák kategóriákká való összevonására. Szentimentelemzést alkalmaztunk a mondatok, egyben a blogbejegyzések polaritásának meghatározására. Az írások döntő részét a modell negatívan osztályozta, utalva a negatív páciens tapasztalatokra. A betegellátás problematikus témáinak feltárására és az orvos-beteg találkozások mélységi megértésére hatékonyan használhatók a természetesnyelv-feldolgozás eszközei. Az alkalmazott módszereket azonban feltétlenül finomítani kell annak érdekében, hogy az eredmények értelmezése során minél kevesebb manuális közreműködésre legyen szükség.

#### Hivatkozások

- Ács, J., Lévai, D., Nemeskey, D. M., & Kornai, A. (2021). *Evaluating Contextualized Language Models for Hungarian*. <http://arxiv.org/abs/2102.10848>
- Angelov, D. (2020). Top2Vec: Distributed Representations of Topics. *arXiv*. <http://arxiv.org/abs/2008.09470>



- Bidmon, S., Elshiewy, O., Terlutter, R., & Boztug, Y. (2020). What Patients Value in Physicians: Analyzing Drivers of Patient Satisfaction Using Physician-Rating Website Data. *Journal of medical Internet research*, 22(2), e13830. <https://doi.org/10.2196/13830>
- Brito Fernandes, Ó., Baji, P., Kringos, D., Klazinga, N., Gulácsi, L., Lucevic, A., Boncz, I., & Péntek, M. (2019). Patient experiences with outpatient care in Hungary: results of an online population survey. *European Journal of Health Economics*, 20(Suppl 1), 79–90. <https://doi.org/10.1007/s10198-019-01064-z>
- Burkle, C. M., & Keegan, M. T. (2015). Popularity of internet physician rating sites and their apparent influence on patients' choices of physicians. *BMC Health Services Research*, 15(1), 1–7. <https://doi.org/10.1186/s12913-015-1099-2>
- Carinci, F., Van Gool, K., Mainz, J., Veillard, J., Pichora, E. C., Januel, J. M., Arispe, I., Kim, S. M., & Klazinga, N. S. (2015). Towards actionable international comparisons of health system performance: Expert revision of the OECD framework and quality indicators. *International Journal for Quality in Health Care*, 27(2), 137–146. <https://doi.org/10.1093/intqhc/mzv004>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm), 4171–4186.
- Doing-Harris, K., Mowery, D. L., Daniels, C., Chapman, W. W., & Conway, M. (2016). Understanding patient satisfaction with received healthcare services: A natural language processing approach. *AMIA ... Annual Symposium proceedings. AMIA Symposium, 2016*, 524–533. <https://europepmc.org/articles/PMC5333198>
- Donabedian, A. (2005). Evaluating the quality of medical care. In *Milbank Quarterly* (Köt. 83, Szám 4, o. 691–729). Blackwell Publishing Inc. <https://doi.org/10.1111/j.1468-0009.2005.00397.x>
- Doyle, C., Lennox, L., & Bell, D. (2013). A systematic review of evidence on the links between patient experience and clinical safety and effectiveness. In *BMJ Open* (Köt. 3, Szám 1, o. 1570). British Medical Journal Publishing Group. <https://doi.org/10.1136/bmjopen-2012-001570>
- Emmert, M., Meier, F., Heider, A. K., Dürr, C., & Sander, U. (2014). What do patients say about their physicians? An analysis of 3000 narrative comments posted on a German physician rating website. *Health Policy*, 118(1), 66–73. <https://doi.org/10.1016/j.healthpol.2014.04.015>
- Fernandes, Ó. B., Péntek, M., Kringos, D., Klazinga, N., Gulácsi, L., & Baji, P. (2020). Eliciting preferences for outpatient care experiences in Hungary: A discrete choice experiment with a national representative sample. *PLoS ONE*, 15(7 July). <https://doi.org/10.1371/journal.pone.0235165>
- Giordano, L. A., Elliott, M. N., Goldstein, E., Lehrman, W. G., & Spencer, P. A. (2010). Development, implementation, and public reporting of the HCAHPS survey. In *Medical Care Research and Review* (Köt. 67, Szám 1, o. 27–37). Med Care Res Rev. <https://doi.org/10.1177/1077558709341065>
- Greaves, F., Ramirez-Cano, D., Millett, C., Darzi, A., & Donaldson, L. (2013). Use of sentiment analysis for capturing patient experience from free-text comments posted online. *Journal of Medical Internet Research*, 15(11), 1–9. <https://doi.org/10.2196/jmir.2721>
- Hanauer, D. A., Zheng, K., Singer, D. C., Gebremariam, A., & Davis, M. M. (2014). Public awareness, perception, and use of online physician rating sites. *JAMA - Journal of the American Medical Association*, 311(7), 734–735. <https://doi.org/10.1001/jama.2013.283194>
- Ito, T. A., Larsen, J. T., Smith, N. K., & Cacioppo, J. T. (1998). Negative information weighs more heavily on the brain: the negativity bias in evaluative categorizations. *Journal of personality and social psychology*, 75(4), 887–900. <https://doi.org/10.1037//0022-3514.75.4.887>
- Jenkinson, C., Coulter, A., Bruster, S., Richards, N., & Chandola, T. (2002). Patients' experiences and satisfaction with health care: Results of a questionnaire study of specific aspects of care. *Quality and Safety in Health Care*, 11(4), 335–339.

- <https://doi.org/10.1136/qhc.11.4.335>
- Khanbhai, M., Anyadi, P., Symons, J., Flott, K., Darzi, A., & Mayer, E. (2021). Applying natural language processing and machine learning techniques to patient experience feedback: A systematic review. In *BMJ Health and Care Informatics* (Köt. 28, Szám 1, o. 100262). BMJ Publishing Group. <https://doi.org/10.1136/bmjhci-2020-100262>
- Lagu, T., Goff, S. L., Hannon, N. S., Shatz, A., & Lindenauer, P. K. (2013). A mixed-methods analysis of patient reviews of hospital care in England: Implications for public reporting of health care quality data in the United States. In *Joint Commission Journal on Quality and Patient Safety* (Köt. 39, Szám 1, o. 7–15). Joint Commission Resources, Inc. [https://doi.org/10.1016/s1553-7250\(13\)39003-5](https://doi.org/10.1016/s1553-7250(13)39003-5)
- Li, J., Liu, M., Li, X., Liu, X., & Liu, J. (2018). Developing embedded taxonomy and mining patients' interests from web-based physician reviews: Mixed-methods approach. *Journal of Medical Internet Research*, 20(8). <https://doi.org/10.2196/jmir.8868>
- López, A., Detz, A., Ratanawongsa, N., & Sarkar, U. (2012). What patients say about their doctors online: A qualitative content analysis. *Journal of General Internal Medicine*, 27(6), 685–692. <https://doi.org/10.1007/s11606-011-1958-4>
- Lucevic, A., Péntek, M., Kringos, D., Klazinga, N., Gulácsi, L., Brito Fernandes, Ó., Boncz, I., & Baji, P. (2019). Unmet medical needs in ambulatory care in Hungary: forgone visits and medications from a representative population survey. *European Journal of Health Economics*, 20(1), 71–78. <https://doi.org/10.1007/s10198-019-01063-0>
- Mattke, S., Epstein, A. M., & Leatherman, S. (2006). The OECD Health Care Quality Indicators Project: History and background. *International Journal for Quality in Health Care*, 18(SUPPL. 1), 1–4. <https://doi.org/10.1093/intqhc/mzl019>
- McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), 205. <https://doi.org/10.21105/joss.00205>
- McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. <http://arxiv.org/abs/1802.03426>
- Nawab, K., Ramsey, G., & Schreiber, R. (2020). Natural Language Processing to Extract Meaningful Information from Patient Experience Feedback. *Applied Clinical Informatics*, 11(2), 242–252. <https://doi.org/10.1055/s-0040-1708049>
- Nemeskey, D. M. (2020). Natural Language Processing Methods for Language Modeling. In *Eötvös Loránd University*.
- Nemeskey, D. M. (2019). Introducing huBERT. *XVII. Magyar Számítógépes Nyelvészeti Konferencia*, 3–14.
- Price, R. A., Elliott, M. N., Zaslavsky, A. M., Hays, R. D., Lehrman, W. G., Rybowski, L., Edgman-Levitan, S., & Cleary, P. D. (2014). Examining the role of patient experience surveys in measuring health care quality. In *Medical Care Research and Review* (Köt. 71, Szám 5, o. 522–554). SAGE Publications Inc. <https://doi.org/10.1177/1077558714541480>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. <http://arxiv.org/abs/1908.10084>
- Saha, B., Lisboa, S., & Ghosh, S. (2020). Understanding patient complaint characteristics using contextual clinical BERT embeddings. *42nd IEEE Annual Conference of Engineering in Medicine and Biology Society*.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *7th International Conference on Learning Representations, ICLR 2019*. <http://arxiv.org/abs/1804.07461>
- Zhang, W., Deng, Z., Hong, Z., Evans, R., Ma, J., & Zhang, H. (2018). Unhappy Patients Are Not Alike: Content Analysis of the Negative Comments from China's Good Doctor Website. *Journal of Medical Internet Research*, 20(1), e35. <https://doi.org/10.2196/jmir.8223>

## Nagyot mondó képviselők? Fokozás a politikai kommunikációban

Szabó Martina Katalin<sup>1,2</sup>, Vincze Veronika<sup>3</sup>, Ring Orsolya<sup>4</sup>, Guba Csenge<sup>4,5</sup>

<sup>1</sup>Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup>Társadalomtudományi Kutatóközpont, CSS-RECENS

<sup>3</sup>ELKH-SZTE Mesterséges Intelligencia Kutatócsoport

<sup>4</sup>Társadalomtudományi Kutatóközpont, Politikatudományi Intézet

<sup>5</sup>Szegedi Tudományegyetem, Nyelvtudományi Doktori Iskola

{martina,vincze}@inf.u-szeged.hu

Ring.Orsolya@tk.hu

csenge.guba@gmail.com

**Kivonat** A politikai kommunikációban megjelenő érzelmkifejezés kutatása az utóbbi évtizedekben egyre nagyobb hangsúlyt kap. Dolgozatunkban azt vizsgáljuk, hogy a nyelvi fokozás milyen kvantitatív és kvalitatív sajátságokkal jelenik meg a politikai kommunikációban. A fokozó elemek csoportján belül külön figyelmet fordítunk az ún. negatív emotív fokozókra, amelyek prior szemantikai tartalma valamely negatív érzelmehez kapcsolódik, azonban fokozó elemekként funkcionálhatnak. Ezzel összefüggésben a szövegek szentimentjét is elemezzük: azt vizsgáljuk, hogy mely szentimentértékű szavak jelentését intenzifikálják a politikusok a fokozó szavak segítségével. Mindezek segítségével megmutatjuk, hogy milyen kvalitatív és kvantitatív sajátságok jellemzik a nyelvi fokozás tekintetében a kormánypárti és az ellenzéki kommunikációt, valamint, hogy hogyan befolyásolja mindezt a Covid19-járvány.

**Kulcsszavak:** fokozás, negatív emotív fokozók, beszélt nyelvi korpusz, magyar, politikai kommunikáció, szentimentelemzés, Covid19

### 1. Bevezetés

Ebben a dolgozatban a fokozó értelmű kifejezéseket vizsgáljuk politikai diskurzusban nagy méretű beszélt nyelvi korpusz alapján, NLP-vizsgálati módszerek segítségével. A munka során különös figyelmet fordítunk az ún. negatív emotív fokozókra.

Bolinger (2013), Méndez-Naya (2008) és Chang és Bin (2020) alapján a *fokozók* azok a szavak, amelyek képesek egy skálán felfelé vagy lefelé módosítani a melléknevek, igék, határozószók, ritkábban a főnevek jelentését, pl. *nagyon*, *eléggé*, *fantasztikusan* stb. A fokozók csoportja az egyik leggyorsabban változó elemkészlete a nyelvnek, és számos olyan nyelvi elem létezik, amely képes fokozóként funkcionálni (Ito és Tagliamonte, 2003; Dressler és Barbaresi, 2011).

Ennek egyik oka lehet a beszélő törekvése arra, hogy eredeti legyen, felhívja és fenntartsa a hallgató figyelmét, valamint hogy növelje kifejezései újszerűségét (Tagliamonte és Roberts, 2005; Szabó, 2018).

A fokozás szociolingvisztikai szempontból is érdekes problémakör, ugyanis több társadalmi tényező is befolyásolja a használatukat. A szakirodalom például régóta tárgyalja, hogy a nők több fokozót használnak, mint a férfiak (Jesperesen, 1922; Lakoff, 1973; Fuchs és Gut, 2012). A perzsa nyelvet vizsgáló kutatók emellett arra a megállapításra jutottak, hogy az eltérésnek gyakran kulturális oka is lehet (Sardabi és Afghari, 2015): Rasekh és Saeb (2015) szerint például a nők az alacsonyabb szociális státusz miatt szeretnék a fokozók használatával nagyobb hangsúlyt adni a beszédüknek. De a nemi szerepeken túl az életkor, a szociális státusz és egyéb változók is befolyásolják, hogy milyen fokozókat használunk, valamint azt is, hogy milyen gyakran használjuk azokat (Fuchs és Gut, 2012). Xiao és Tao (2007) britek körében végzett kutatásukban például rámutatnak, hogy a magasabb iskolai végzettséggel rendelkező emberek gyakrabban használnak fokozókat, illetve a 30 év alattiak körében ritkább a használatuk.

A nyelvi fokozás szerepének megértéséhez tekintsük Pintili és mtsai (2010) megállapítását: A szerző a meggyőzés, rábeszélés két fő stratégiájának nevezi a megerősítést és a kisebbitést (an. *intensification* és *minimization*), ami a saját erősségek és ellenfél gyengeségeinek felerősítését, valamint az opponensi pozitívumok és a saját negatívumok „halványítását” jelenti. Belátható, hogy mindezeknek fontos eszközei lehetnek a fokozók (mind a növelő, mind a csökkentő értelműek, részletesebben l. később). Trajkova és Neshkovska (2018) vizsgálata alapján a gyűlöletpropaganda gyakori elemei a növelő értelmű fokozók (a negatív töltetű szavak metaforák és ironia mellett). Mindezen túl, pl. Pant és mtsai (2020) gépi tanulási kísérleteikben is felhasználják a fokozókat arra, hogy a szövegekben automatikusan torzítást (an. *bias*) azonosítsanak.

A fokozó elemek csoportján belül különös figyelmet érdemelnek az ún. *negatív emotív fokozó szavak* (a továbbiakban csak NEF). A jelenség megértéséhez vegyük alapul Balogh (2009) megállapításait: a magyar nyelvben a fokozó elemek csoportján belül megkülönböztethetők azok, amelyek elsődleges funkciójuk szerint fokozó szerepet töltenek be, például *nagyon*, *felettébb* stb., és azok, „amelyeknek van elsődleges jelentése, és csak másodlagosan szolgálnak fokozásra”, például *mértéktelenül*, *kimondhatatlanul* stb. Shmelev (2012) rámutat, hogy az erős emocionális értékelést kifejező szavak gyakran nyernek ezzel ellentétes szemantikai tartalmat (vö. az enantioszémia jelensége, bővebben l. pl. Klégr (2013); Szabó (2019)). A NEF-ek lexikai szinten negatív szemantikai tartalommal rendelkeznek, ugyanakkor képesek más szavak módosítóiként funkcionálni, és ilyenkor ezt a tartalmat részben vagy egészben elveszíteni (Szabó és Bibok, 2019). Ilyen például az *iszonyú* és a *rettenetesen* szó is az *iszonyú szép* és a *rettenetesen okos* kifejezésekben. A jelenség számos nyelvben adathozható, valószínűleg nyelvfüggetlen sajátság (Nemesi, 1998; Jing-Schmidt, 2007; Andor, 2003, 2011; Szabó és Bibok, 2019).

Mivel a politikai kommunikáció egyik fontos törekvése, hogy érzelmi hatás kiváltása által, különböző eszmék propagálásán keresztül politikai cselekvésekre

ösztönözzön (részletesebben l. később), a nyelvi fokozás témakörén belül érdekes külön figyelmet fordítanunk azokra az elemekre, amelyek negatív emotív szemantikai tartalommal rendelkeznek a fokozáson túl.

E munkában részletesen elemezzük az 1998 és 2021 közötti időszak napirend előtti parlamenti felszólalásait, különös tekintettel a fokozók és a NEF-ek használatára, valamint külön figyelmet fordítunk arra, hogy a Covid19-járvány hogyan befolyásolta a fokozók használatát a felszólalásokban.

A dolgozat hiánypótlónak tekinthető, ugyanis - bár vannak a fokozás szerepét, megjelenését a politikai szövegekben elemző kutatások (l. többek között (Nir, 1988; Schneider és Eitelmann, 2020; Vlasova, 2021)). Olyan dolgozatot is találunk, ahol a szerző a fokozók szerepét kifejezetten a Covid19-járvánnyal összefüggésben vizsgálja (Admiral, 2020). Ugyanakkor nincs tudomásunk olyan kutatásról, amely politikai diskurzusban kifejezetten a negatív emotív fokozókra fókuszálna. Mindemellett, a magyar nyelvvel kapcsolatosan már magáról a nyelvi fokozásra irányuló kutatást sem ismerünk.

## 2. Kapcsolódó irodalom

### 2.1. A nyelvi fokozásról, azon belül a NEF-ről röviden

A fokozóknak számos altípusa van, és a nemzetközi terminológia is igen változatos képet mutat, pl. *intensive adverbs* (Stoffel (1901), idézi Bolinger (2013)), *intensives* (Ito és Tagliamonte, 2003), *amplifiers* és *downtoners* (Quirk és mtsai, 1985). A fentebbi definícióból kiindulva, mivel a fokozók lefelé és felfelé is képesek módosítani a szerkezeti alaptag jelentését a skálán, *növelők* (an. *amplifiers*) és *csökkentők* (an. *downtoners*) osztályaira bonthatóak (Fuchs és Gut, 2016; Strohm és Klinger, 2018). Az utóbbi csoporttal a jelen kutatásban nem foglalkozunk.

A növelő értelmű fokozók további altípusokra bonthatóak, úgymint *erősítők* (an. *boosters*), amely egy magasabb fokra mutat egy bizonyos skálán, valamint a *maximalizálók* (an. *maximizers*), amelyek pedig a skála legfelsőbb pontját jelölik meg (Quirk és mtsai, 1985). Quirk és mtsai (1985) megkülönböztet még egy harmadik csoportot is, az ún. *hangsúlyozók* (an. *emphasizers*), amelyek pedig a megnyilatkozásban kifejezett tartalom igazságértékét hangsúlyozzák.

A jelen dolgozatban a fentebb tárgyalt jelentéskülönbségeket nem vizsgáljuk. Egyetlen szócsoporthoz fordítunk külön is figyelmet a növelő értelmű fokozók osztályán belül, a negatív emotív fokozókra. Ugyanakkor megjegyezzük, hogy Szabó és mtsai (2021a) alapján a NEF-ek a növelők (*amplifiers*) (azon belül mind az erősítők (an. *boosters*), mind a maximalizálók (an. *maximizers*), valamint a hangsúlyozó (an. *emphasizers*) szerepű fokozók között találhatóak meg.

A NEF-eket sok szerző vizsgálja (l. többek között Jing-Schmidt (2007); Kugler (2014); Chang és Bin (2020), azonban kevés az olyan kutatás, amely a jelenséget nagyobb mennyiségű korpuszadat alapján, NLP-módszerekkel elemzi (Hamilton és mtsai, 2016; Szabó és Bibok, 2019).

E helyütt a szakirodalom részletes áttekintésére nincs mód, ezért itt csak egyetlen, a jelen kutatás szempontjából fontos sajtóságról tennénk említést. A

magyar nyelvre vonatkozó tanulmányok a NEF-ek prior szemantikai tartalma kapcsán érzelmek közül kizárólag a félelmet említik (Nemesi, 1998; Kugler, 2014). Nemesi (1998) szerint például mindezek a kifejezések „az ijesztés, ijedés képzetkörébe tartoznak”. Ugyanakkor Szabó és Guba (2022) szerint, a NEF-ek lexikai szintű szemantikai tartalma helyenként nem a félelemmel, hanem az undor, a düh vagy a bánat érzélemmel függ össze, sőt, némely esetben egyszerre több emócióval is.

## 2.2. A beszélt nyelvi korpuszok

A különböző, így többek között a társadalomtudományi és a nyelvészeti tárgyú kutatások egyik legfontosabb vizsgálati eszközét az NLP-módszerekkel és eszközökkel elemezhető formátumú beszélt nyelvi szövegtörzsek jelentik. Az elmúlt évtizedekben több spontán beszélt nyelvi korpusz keletkezett több nyelven is (Crowdy, 1993; Hemphill és mtsai, 1990; Maekawa és mtsai, 2000; Oostdijk, 2000; Van Bael és mtsai, 2007), és közöttük immáron a magyar nyelvet reprezentáló korpuszokat is találunk (Gósy, 2013; Kugler, 2015; Váradi, 2003; Pápay és mtsai, 2011; Bodó és mtsai, 2017; Galántai és mtsai, 2018; Szabó és mtsai, 2021b). Ugyanakkor a politikai tárgyú beszélt nyelvi korpuszok száma szerény (Guerini és mtsai, 2008; Savoy, 2010; De Smedt és Jaki, 2018; Álvarez-Mellado, 2020).

Fontos ugyanakkor itt megemlíteni, hogy az általunk vizsgált korpusz a klasszikus értelemben nem tekinthető egy beszélt nyelvi korpusz leiratozott változatának, lévén, bár a leiratozás szöveghű, megakadások, hezitációk, újrakezdések stb. nincsenek jelölve benne (l. lentebb).

## 2.3. Érzelmek vizsgálata a politikai kommunikációban

A politikai kommunikáció érzelmi hatás kiváltásának segítségével, illetve különböző eszmék propagálásán keresztül politikai cselekvésekre ösztönöz. Az elmúlt évtizedek technikai és társadalmi változásai következtében a politikai kommunikáció is jelentős változásokon ment át, például megnőtt a kommunikációban résztvevő aktorok és a rendelkezésre álló csatornák száma, mindez pedig hatást gyakorolt a politikai kommunikáció jellegére, intenzitására is. A kommunikáció szerepének felértékelődésére, politikai szereplők a kommunikációjuk professzionalizálásával válaszolnak (Bene és Nábelek, 2019). A politikai beszéd és szövegek jól megtervezett akciók, mivel a politikusok nem csak arra törekcsenek, hogy a közélet valamely aspektusáról tájékoztassanak, hanem hogy meggyőzzék a hallgatóságot és megszerezzék támogatásukat. A politikai kommunikáció egyik fontos színtere a Parlament, ahol a választott képviselők megvitatják a benyújtott törvényjavaslatokat és más országos jelentőséggel bíró ügyeket. A benyújtott törvényjavaslatok vitái során különféle témák merülnek fel, érvek és ellenérvek ütköznek és ezek révén formálódik a politikai napirend, ami a közvitákat is tematizálja.<sup>1</sup> A politikai kommunikáció fontos része a pro és kontra érvelés,

<sup>1</sup> <https://k-monitor.github.io/>

melyben kiemelkedő szerepet kap a kommunikáció intenzitása. Így a politikai beszéd jellegzetes eszköze az egyes álláspontok és tények felerősítése (Siddikova és Zubareva, 2020), melynek nyelvi megjelenésének kvantitatív elemzését teszi lehetővé a fokozás vizsgálata nagy méretű politikai beszédkorpuszon NLP-eszközök segítségével.

A politikai kommunikációban megjelenő érzelmkifejezés kutatása az utóbbi évtizedekben egyre nagyobb hangsúlyt kap mind a nemzetközi, mind a magyar társadalomtudományi vizsgálatokban (Szabó, 2020; Richards, 2004; Crigler és Just, 2012; Wagner és Morisi, 2019; Settle, 2020; Haselmayer és Jenny, 2017) és ugyancsak jelentős a Covid19 járvány politikai kommunikációjához kötődő vizsgálatok száma (Gollust és mtsai, 2020; Mariani és mtsai, 2020). Több kutatás elemzi a járványhoz kötődő politikai és egészségkommunikációs válságot, melyben kitüntetett szerep jut a megszólalások érzelmi töltésének (Gollust és mtsai, 2020). Ezek a kutatások jórészt a politikusok médiában és közösségi médiában megjelenő megszólalásait elemzik (Aparicio és mtsai, 2021; Wang és mtsai, 2021; Rufai és Bunce, 2020), azonban a parlamenti felszólalások a vilá járvánnyal kapcsolatos érzelmi töltésének NLP-eszközökkel történő elemzése újszerűnek számít.

### 3. Vizsgálati módszerek és eszközök

Ebben a részben bemutatjuk a kutatásunkban felhasznált korpuszt és a vizsgálati módszereket.

#### 3.1. A korpusz

A korpuszt az 1998 és 2021 közötti időszakban keletkezett napirend előtti parlamenti felszólalásokból állítottuk össze, melyek szabadon letölthetők a Magyar Országgyűlés honlapjáról.<sup>2</sup> A vizsgálatunk alapjául szolgáló metaadatokkal ellátott adatbázis a Társadalomtudományi Kutatóközpont Politikatudományi Intézetének Hungarian Comparative Agendas Projektje<sup>3</sup> keretében készült, kutatási célokra, regisztráció ellenében szabadon hozzáférhető. A napirend előtti felszólalások az országgyűlés plenáris ülésén hangoznak el országos jelentőségű, halaszthatatlan és rendkívüli ügyekben 2-5 perces időterjedelemben.

A műfajból adódóan beszélt nyelvi adatokról van szó, ugyanakkor stílusuk hivatalos, formális – ebben mindenképpen különbözik a korpusz a magyar nyelvre elérhető, beszélt nyelvi korpuszoktól, melyek spontán beszédet tartalmaznak és/vagy informális jellegűek (pl. (Vincze és mtsai, 2021; Szabó és mtsai, 2021b)). A felszólalások hivatalos jellegűek, a képviselők sok esetben előre megtervezték/megírták beszédjüket, azaz tervezett beszédéről van szó. Sok megszólítást és köszönetnyilvánítást tartalmaznak (*Tisztelt Ház!, Köszönöm a szót.*), szinte kizárólag magázó formulákat használnak, ugyanakkor a leiratok nem tartalmazzák az élőbeszédre jellemző hezitációkat, kisebb szüneteket, téves szókezdeteket.

<sup>2</sup> <https://parlament.hu>

<sup>3</sup> <https://cap.tk.hu/hu>

### 3.2. Szótárillesztés

A korpuszt alapvető számítógépes nyelvészeti feldolgozásnak vetettük alá (tokenizálás, lemmatizálás, morfológiai elemzés, szófaji egyértelműsítés és dependenciaelemzés) a magyarlanc eszköz (Zsibrita és mtsai, 2013) segítségével. Első lépésben egy általunk készített fokozószótár szavait illesztettük automatikusan a szóalakokra, majd megvizsgáltuk, hogy az adott szóalakok mely más szavakkal állnak közvetlen szintaktikai kapcsolatban, azaz mely szavakat módosítanak. Itt arra voltunk elsősorban kíváncsiak, hogy milyen szentimentértékű szavakat módosítanak a vizsgált fokozók. Ez utóbbi feladathoz tehát egy szentimentszótárra volt szükségünk, és az elemzési eredményél ugyancsak a kvantitatív sajátságokra fókuszáltunk, azaz gyakorisági adatokat gyűjtöttünk az egyes kollokációkról.

A fokozószótárakat a kutatás egy korábbi fázisában állítottuk össze, mégpedig úgy, hogy az egyszerű fokozókat, valamint a NEF-et külön listákba rendeztük (Szabó és Guba, 2022). A szótárépítő munkához a kiinduló listákat egy szótár (Tukacs, 2015), az MNSZ2 korpusz (Oravecz és mtsai, 2014), valamint webes keresések segítségével állítottuk össze. Ezt követően a listákat két lépésben bővítettük: Először a magyar Wordnet (Prószéky és Miháltz, 2008) synseteit használtuk, és azokból a szinonimakapcsolatokat vettük ki, majd a találatokat a szótárunk szempontjából kézzel ellenőriztük. A kapott listákat ezután egy magyar nyelvű helyesírási szótár<sup>4</sup> segítségével bővítettük tovább, és az így kapott eredményt ugyancsak végigellenőriztük. A végső szótárak összesen 343 szót tartalmaznak, 113-at a nem negatív emotív és 226-at a NEF szótár.

Kollokációelemzésünkhöz a Társadalomtudományi Kutatóközpont Politikatudományi Intézetének poltextLAB projektje által speciálisan a magyar politikai doménre létrehozott szentimentszótárat használtuk fel, amely 2585 pozitív és 2566 negatív szót tartalmaz (Ring és mtsai, 2022). A szentimentszótár szavait a magyarlanc által visszaadott lemmákra illesztettük, és megvizsgáltuk, hogy az illeszkedő lemmákat módosítja-e egy fokozó vagy NEF. A továbbiakban az így kinyert fokozó/NEF + szentimenttel bíró szó kollokátumok vizsgálatára összpontosítottunk.

Az adatok elemzéséhez a politikai szereplőket ciklusonként két csoportra, ellenzéki és kormányoldalra osztottuk, és a függetleneket és nemzetiségieket kihagytuk. Emellett külön vizsgáltuk a Covid19-et megelőző és a Covid19-időszak alatt keletkezett szövegeket.

## 4. Eredmények

Vizsgálatainkat két szöveghalmazon is elvégeztük. Egyfelől a teljes, 1998 és 2021 között elhangzott felszólalások anyagát elemeztük, másrészt különös figyelmet fordítottunk a jelenlegi, azaz a 2018 és 2022 közötti ciklusra, azon belül is a Covid19-járvány megjelenése előtti és utáni időszakokra. Mindkét vizsgálatban külön elemeztük a kormánypárti és az ellenzéki felszólalásokat. Az alábbiakban ismertetjük eredményeinket.

<sup>4</sup> <http://hunspell.github.io/>



#### 4.1. 1998 és 2021 közötti felszólalások

A fokozók és a NEF-ek számát és gyakoriságát az 1. táblázat mutatja.

Ciklus	Párt	mondat	token	fokozó	NEF	fokozó%	NEF%
1998-2002	ellenzék	9331	180057	1333	81	0,7403	0,0450
1998-2002	kormány	17152	350626	2846	223	0,8117	0,0636
2002-2006	ellenzék	17769	332816	2414	176	0,7253	0,0529
2002-2006	kormány	21417	433841	3978	175	0,9169	0,0403
2006-2010	ellenzék	20796	400195	3415	277	0,8533	0,0692
2006-2010	kormány	18618	394460	3937	177	0,9981	0,0449
2010-2014	ellenzék	26926	565982	5010	383	0,8852	0,0677
2010-2014	kormány	19615	388874	3035	193	0,7805	0,0496
2014-2018	ellenzék	22166	460168	4392	384	0,9544	0,0834
2014-2018	kormány	14974	296504	2418	142	0,8155	0,0479
2018-2022	ellenzék	23752	488275	4674	500	0,9572	0,1024
2018-2022	kormány	10345	200085	1427	114	0,7132	0,0570
Összesen		222861	4491883	38879	2825	0,8655	0,0629

1. táblázat. A vizsgált korpusz adatai: a fokozó és a NEF-ek gyakorisága.

Ha a fokozó szavakat vizsgáljuk, azt látjuk, hogy a vizsgált időszak legelejétől egészen 2010-ig a kormánypártok használtak arányaiban több fokozót, ugyanakkor 2010 óta egyértelműen az ellenzék használ többet, minden ciklusban, kivétel nélkül. Ezek alapján azt mondhatjuk, hogy – a vizsgált időszakban – a baloldali parlamenti pártokra jellemző inkább a fokozók használata a napirend előtti felszólalásokban. A NEF-ek gyakorisága ugyanakkor az adatok alapján valamelyest más képet mutat: azok eleve, rendre az ellenzéki képviselők felszólalásaiban a gyakoribbak minden ciklusban a legelsőt kivéve, tehát az 1998 és 2002 közötti időszakot. Mi több, a NEF-ek gyakorisági különbsége a kormány- és az ellenzéki oldal között fokozatosan egyre jelentősebbé is válik, a 2014-es ciklustól már szembeszökő az eltérés. Mondhatjuk, hogy a változás látványos eltérést hoz a teljes korszak végére a két oldal kommunikációs stílusában. Együtt tekintve a két vizsgált elemcsoportot, megállapítható, hogy a teljes időszak végére mindkét nyelvi eszköz inkább az ellenzéki kommunikáció jellemzőjévé vált.

Külön megvizsgáltuk azt is, hogy a fokozók és a NEF-ek milyen szentiment-értékű szavakat módosítanak a korpuszban. Így a mondatok szintaktikai szerkezetére építve kigyűjtöttük azokat a szókapcsolatokat, melyek egy fokozóból vagy negatív emotív kifejezésből, valamint az általuk módosított szóból állnak. A módosított szavakat külön vizsgálat alá vetettük, hogy szerepelnek-e a rendelkezésünkre álló politikai szentimentszótárakban. A 2. és a 3. táblázatokban láthatjuk, hogy az összes fokozó, illetve negatív emotív kifejezéshez képest hány százalékuk módosít pozitív, illetve negatív alaptagot.

A fenti adatok alapján általánosságban elmondható, hogy a fokozó + pozitív töltetű szó kombináció, valamint a NEF + pozitív töltetű szó kombináció gyakrabban szerepel kormánypárti felszólalásokban, mint az ellenzékiekben. A fokozó

Ciklus	Párt	fokozó	POS alaptag	%	NEF	POS alaptag	%
1998-2002	ellenzék	1333	417	31,2828	81	14	17,2840
1998-2002	kormány	2846	851	29,9016	223	36	16,1435
2002-2006	ellenzék	2414	747	30,9445	176	36	20,4545
2002-2006	kormány	3978	1336	33,5847	175	41	23,4286
2006-2010	ellenzék	3415	1021	29,8975	277	42	15,1625
2006-2010	kormány	3937	1410	35,8141	177	39	22,0339
2010-2014	ellenzék	5010	1524	30,4192	383	48	12,5326
2010-2014	kormány	3035	1012	33,3443	193	36	18,6528
2014-2018	ellenzék	4392	1383	31,4891	384	60	15,6250
2014-2018	kormány	2418	782	32,3408	142	28	19,7183
2018-2022	ellenzék	4674	1492	31,9213	500	77	15,4000
2018-2022	kormány	1427	476	33,3567	114	12	10,5263
Összesen		38879	12451	32,0250	2825	469	16,6018

2. táblázat. A pozitív alaptagú fokozók és NEF-ek száma és gyakorisága.

+ negatív töltetű szó kombináció ugyanakkor az ellenzéknél fordul elő gyakrabban. A fokozók ilyen használata arra enged következtetni, hogy valószínűleg a kormánypárti képviselők jobban szeretik kihangsúlyozni a pozitívumokat, míg az ellenzék inkább a negatívumokat emeli ki nyelvi fokozó eszközökkel. Ezzel kapcsolatban érdemes visszautalni az 1. táblázat adataira is, ahol azt láttuk, hogy a NEF használata a kormányoldallal szemben inkább az ellenzékre jellemző, ami arra mutat, hogy a kormány még a NEF-ben rejlő prior negatív tartalmat is igyekszik mellőzni a kommunikációjában (a prior negatív tartalomról részletesebben l. lentebb). Elmondható még, hogy NEF-ek sokkal inkább kollokálódnak negatív töltetű szavakkal, mint pozitív töltetűekkel (29,7%, valamint 16,6%), ami nyelvészeti szempontból különösen figyelemre méltó: Partington (1993) és Dér (2013) kutatásai alapján a NEF-ek delexikalizációs folyamaton mennek keresztül, miközben elveszítik prior lexikai jelentésüket, ezzel együtt pedig negatív emotív szemantikai tartalmuk is elhalványul és eltűnik, miközben grammatikai elemekké válnak (a delexikalizációs folyamatról részletesebben l. pl. Bonelli (2000)). Ugyanakkor mindez egy hosszú nyelvi változási folyamat, és ezzel együtt nem minden NEF van a fejlődési folyamat ugyanazon fokán egy adott időszakban (Bolinger, 2013; Partington, 1993). Ezzel összefüggésben például, a prior használat is jelen van az új fokozó funkciójú használat mellett, pl. *brutálisan rossz*, *brutálisan jó* és *brutálisan bánik vki vkivel*. Az, hogy a negatív emotív tartalom kiüresedése hosszú időszak alatt megy végbe, valamint, hogy a negatív jelentéskomponens bizonyos kontextusokban aktív lehet az interpretációban, nyilvánvalóan befolyásolja a NEF-ek kollokációs preferenciáit, hiszen, az ún. kongruencia jelensége alapján egy poláris kifejezés tipikusan a vele azonos polaritású kifejezésekkel kollokálódik (Dilts, 2010). Korpuszvizsgálati eredményeink arra mutathatnak, hogy a NEF-ek esetében, azok kollokációs preferenciáik alapján a delexikalizálódás folyamata még nem ment végbe teljesen, azaz ere-

deti szemantikai tartalmuk visszaköszön kollokációs kapcsolódási mutatóikban. Eredményeink statisztikailag is szignifikánsak ( $\chi^2$ -próba,  $p < 0,001$ ).<sup>5</sup>

Ciklus	Párt	fokozó	NEG alaptag	%	NEF	NEG alaptag	%
1998-2002	ellenzék	1333	342	25,6564	81	23	28,3951
1998-2002	kormány	2846	650	22,8391	223	74	33,1839
2002-2006	ellenzék	2414	621	25,7249	176	61	34,6591
2002-2006	kormány	3978	830	20,8648	175	43	24,5714
2006-2010	ellenzék	3415	832	24,3631	277	72	25,9928
2006-2010	kormány	3937	845	21,4630	177	63	35,5932
2010-2014	ellenzék	5010	1082	21,5968	383	107	27,9373
2010-2014	kormány	3035	588	19,3740	193	70	36,2694
2014-2018	ellenzék	4392	952	21,6758	384	106	27,6042
2014-2018	kormány	2418	454	18,7758	142	48	33,8028
2018-2022	ellenzék	4674	1072	22,9354	500	137	27,4000
2018-2022	kormány	1427	300	21,0231	114	36	31,5789
Összesen		38879	8568	22,0376	2825	840	29,7345

3. táblázat. A negatív alaptagú fokozók és NEF-ek száma és gyakorisága.

#### 4.2. A koronavírus-járvány előtti és alatti időszak

Amint korábban említettük, külön figyelmet fordítunk a koronavírus-járványt közvetlenül megelőző és az alatta zajló politikai kommunikációra. A 2018-2022 közti parlamenti ciklusban elhangzott felszólalásokat ezért két részre osztottuk: egyrészt a ciklus elejétől (2018) a 2019 decemberéig tartó időszakra, valamint a 2020 elejétől a 2021 őszeig tartó időszakra. A kormánypártokra és ellenzékre az eddigiekhez hasonlóan külön összpontosítunk.

Ciklus	Párt	mondat	token	fokozó	NEF	fokozó%	NEF%
2018-2019	ellenzék	13025	267291	2539	266	0,9499	0,0995
2018-2019	kormány	5321	100217	696	41	0,6945	0,0409
2020-2021	ellenzék	10727	220984	2135	234	0,9661	0,1059
2020-2021	kormány	5024	99868	731	73	0,7320	0,0731

4. táblázat. A fokozó és a NEF-ek gyakorisága a Covid19 előtt és alatt.

A 4. táblázat adataiból látszik, hogy az ellenzék és a kormánypártok kommunikációja esetében egyaránt nőtt a fokozó elemek és a NEF-ek használata a

<sup>5</sup> A NEF-ek delexikalizálódási folyamatáról, időbeli szemantikai és kollokációs változási folyamatairól jelenleg is folytatunk kutatásokat több, nagy méretű korpusz alapján, mint például Magyar Nyelvtörténeti Szövegtár, Ludas Matyi-korpusz, saját Twitter-korpuszunk, HuTongue-korpusz

Covid19-járvány idején. Ez az eredmény azt valószínűsíti, hogy a járványhelyzet a kommunikációs stílust is befolyásolja, a képviselők gyakrabban élnek a fokozással való erősítés, nyomatékosítás eszközével. A kormányoldal és az ellenzék kommunikációja között a legjelentősebb eltérést egyébként a NEF-ek használatában találjuk. Azt látjuk ugyanis, hogy azok gyakorisága a kormánypártoknál majdhogynem megkétszereződött.

Ciklus	Párt	fok + POS	NEF + POS	fok + POS%	NEF + POS%
2018-2019	ellenzék	767	34	30,2087	12,7820
2018-2019	kormány	233	3	33,4770	7,3171
2020-2021	ellenzék	725	43	33,9578	18,3761
2020-2021	kormány	243	9	33,2421	12,3288

5. táblázat. A pozitív alaptagú fokozók és NEF-ek száma és gyakorisága a Covid19 előtt és alatt.

Ciklus	Párt	fok + NEG	NEF + NEG	fok + NEG%	NEF + NEG%
2018-2019	ellenzék	561	71	22,0953	26,6917
2018-2019	kormány	141	8	20,2586	19,5122
2020-2021	ellenzék	511	66	23,9344	28,2051
2020-2021	kormány	159	28	21,7510	38,3562

6. táblázat. A negatív alaptagú fokozók és NEF-ek száma és gyakorisága a Covid19 előtt és alatt.

Az eddig tárgyaltakon túl elvégeztünk egy olyan vizsgálatot is, ahol az alaptagok szentimentértékeit a járványhelyzettel összefüggésben vizsgáltuk. Azt elemeztük tehát, hogy a pozitív, illetve negatív alaptaggal bíró fokozók és NEF-ek gyakorisága hogyan változott a járvány kitörése után. Az 5. és 6. táblázatok mutatják az erre vonatkozó adatokat. Eredményeink statisztikailag is szignifikánsak ( $\chi^2$ -próba,  $p < 0,05$ ).

Tekintsük először a fokozó elemek használati sajátságait! Az adatokból az világlik ki, hogy a járvány kezdete óta összességében nőtt a szentimenttel bíró alaptagok használata. A legjelentősebb eltérést az ellenzéki kommunikációban látjuk, mégpedig abban, hogy – meglepő módon – a pozitív szavakat hajlamosabbak fokozni a járvány előtti időszakhoz képest. A fokozóknál azonban sokkal nagyobb eltérést mutatnak a NEF-ek kollokációs sajátságai: mind a pozitív, mind a negatív szentimentértékű alaptagok gyakorisága jelentősen megnövekedett (a semleges alaptagokkal szemben). Különösen szembeötlő ez az eltérés a kormányoldali beszédekben: amíg 2018-19-ben a NEF-ek 26,8%-a módosított valamilyen polaritással rendelkező (tehát nem semleges szentimentértékű) szót a kormánypárti felszólalásokban, addig 2020-21-ben már a felszólalások 50,7%-a. Még behatóbban vizsgálva az adatokat, a negatív polaritású alaptagok mellett

emelkedett meg a legnagyobb mértékben (közel a duplájára) a NEF-ek száma a kormány kommunikációjában. Az ellenzék kommunikációjában is növekedést látunk a NEF + negatív polaritású kollokátumoknál, hasonlóan a kormányoldali beszédekhez, ugyanakkor itt a NEF + negatív szavak kapcsolatai nem szaporodtak meg olyan jelentős mértékben, mint a kormánypártoknál.

Fontos megemlítenünk, hogy a szentimentet hordozó kifejezések aránya változatlanul mondható a két vizsgált időszakban, sőt, enyhe csökkenés mutatkozik a pozitív szavak frekvenciájában, ahogyan azt a 7. táblázat mutatja alább.

Ciklus	Párt	POS	NEG	POS%	NEG%
2018-2019	ellenzék	9246	28676	3,4592	10,7284
2018-2019	kormány	4049	11123	4,0402	11,0989
2020-2021	ellenzék	7427	24334	3,3609	11,0117
2020-2021	kormány	3582	11357	3,5867	11,3720

7. táblázat. A pozitív és negatív szavak száma és gyakorisága a Covid19 előtt és alatt.

Mindezek alapján tehát a pozitív és negatív szentimentértékű kifejezések növekvő használata kifejezetten a fokozókkal, illetve NEF-vel összefüggésben figyelhető meg. Mindezekből az következik, hogy a kormánypártok felszólalásai nyelvileg is lekövetik a társadalmi veszélyhelyzetet: a fokozók, illetve NEF-ek segítségével a különböző pozitív és negatív tartalmak jóval nagyobb hangsúlyt kapnak, mint korábban.

## 5. Összegzés

Ebben a dolgozatban azt vizsgáltuk, hogy a nyelvi fokozás milyen kvantitatív és kvalitatív sajátosságokkal jelenik meg a politikai kommunikációban. Kutatásunkhoz egy nagy méretű, beszélt nyelvi szövegekből álló korpusz írott változatát, a napirend előtti parlamenti felszólalások szövegeit használtuk az 1998 és 2021 közötti időszakból. Az adatokat választási ciklusonként elemeztük. A fokozó elemek csoportján belül külön figyelmet fordítottunk az ún. negatív emotív fokozókra (amelyekre a dolgozatban NEF-ekként utaltunk). Emellett a fokozással összefüggésben a szövegek szentimentjét is elemeztük. Azt vizsgáltuk, hogy a politikusok mely szentimentértékű szavak jelentését teszik intenzívebbé, illetve nyomatékosítják a fokozás eszközével. Külön, összehasonlító jelleggel megvizsgáltuk a jelenlegi, azaz a 2018 és 2022 közötti ciklust, azon belül is a Covid19-járvány megjelenése előtti és utáni időszakot. Mindezen túl, minden elemzésünkben figyelmet fordítottunk a kormánypárti és az ellenzéki kommunikáció hasonlóságaira és eltéréseire a nyelvi fokozás, továbbá az azzal összefüggésben megjelenő szentimenttartalmak tekintetében.

Legfontosabb megállapításaink a következők voltak: Megállapítható, hogy napjainkra mind a fokozók, mind a NEF-ek inkább az ellenzéki kommunikáció jellemzőjévé váltak. A fokozó szavakat tekintve 2010-ben fordul meg az arány

az ellenzék javára, innentől az ellenzék használ több fokozót. A NEF-ek ugyanakkor eleve az ellenzéki képviselők felszólalásaiban voltak gyakoribbak minden ciklusban, a legelsőt kivéve, ráadásul az eltérés a kormány- és az ellenzéki oldal között fokozatosan egyre jelentősebbé is válik.

Megvizsgálva azt, hogy a fokozók és a NEF-ek milyen szentimentértékű szavakat módosítanak a korpuszban, azt találtuk, hogy a fokozó + pozitív töltetű szó kombináció, valamint a NEF + pozitív töltetű szó kombináció gyakrabban szerepelt a kormánypárti felszólalásokban, mint az ellenzékiekben. A fokozó + negatív töltetű szó kombináció ugyanakkor az ellenzékénél fordult elő gyakrabban. Arra következtethetünk tehát, hogy a kormánypárti képviselők jobban szeretik kihangsúlyozni a pozitívumokat, míg az ellenzék inkább a negatívumokat emeli ki nyelvi fokozó eszközökkel.

A koronavírus-járvány tekintetében azt láttuk, hogy az ellenzék és a kormánypártok kommunikációja esetében egyaránt nőtt a fokozó elemek és a NEF-ek használata a járvány idején. Ez az eredmény rámutat, hogy a járványhelyzet valószínűleg a kommunikációs stílust is befolyásolja, a képviselők gyakrabban élnek a fokozással való erősítés, nyomatékosítás eszközével. A kormányoldal és az ellenzék kommunikációja között a legjelentősebb eltérést egyébként a NEF-ek használatában találtuk. Azt látjuk ugyanis, hogy azok gyakorisága a kormánypártoknál majdhogynem megkétszereződött.

Végül, az alaptagok szentimentértékeit a járványhelyzettel összefüggésben vizsgálva azt találtuk, hogy a járvány kezdete óta összességében nőtt a szentimenttel bíró alaptagok használata. Emellett a fokozóknál azonban sokkal nagyobb eltérést mutattak a NEF-ek kollokációs sajátságai, ugyanis mind a pozitív, mind a negatív szentimentértékű alaptagok gyakorisága jelentősen megnövekedett (a semleges alaptagokkal szemben), különösen a kormányoldali beszédekben, és a két polaritás közül a negatív alaptagok gyakorisága ugrott meg jobban.

A jelen dolgozatban a fokozókat és NEF-eket nem vizsgáltuk az egyes elemek szintjén, azonban ez fontos lépését fogja képezni a kutatásnak. Szeretnénk feltárni, hogy pontosan hogyan változik az egyes elemek gyakorisága az időben. Például, ha az egyes NEF-ek gyakorisági sajátságait longitudinálisan vizsgáljuk, árulkodik-e a korpusz a NEF-ek delexikalizációs folyamatairól az egyes szavak szintjén. (Ahogyan azt korábban említettük, e vizsgálatok jelenleg is aktívan folynak több korpusz alapján.) Emellett, ahogyan arról korábban (l. 2.1) említést tettünk, a NEF-ek prior szemantikai tartalma különböző negatív emóciókkal hozható összefüggésbe. Kíváncsiak vagyunk, felfedezhetünk-e valamilyen összefüggést a Covid19-járvány, a politikai szerep és a NEF-használat között ebben a tekintetben. Emellett szeretnénk megvizsgálni a parlamenti kommunikáció más időbeli tendenciáit, így például a szókészlet és egyéb nyelvi jellemzők változásait is. Továbbá, mivel a korpusz lehetőséget ad a formális beszélt nyelv részletesebb vizsgálatára is, tervezzük, hogy a korpuszadatok alapján kísérletet teszünk e szövegtípus részletesebb, kvantitatív jellegű leírására. Végül szeretnénk összetettebb NLP-módszerek alkalmazási lehetőségeit is megvizsgálni a korpuszon.

Az itt bemutatott és felhasznált fokozószótárakat a kutatás elvégzését követően kutatási célra szabadon hozzáférhetővé tesszük.

## Köszönetnyilvánítás

A kutatást, amelyet a Szegedi Tudományegyetem és a Társadalomtudományi Kutatóközpont valósított meg, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében. Külön köszönet illeti a Társadalomtudományi Kutatóközpont Comparative Agendas Project kutatócsoportjának tagjait és gyakornokait a felhasznált korpusz előkészítéséért.

Szabó Martina Katalin kutatásait a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal OTKA posztdoktori kiválósági programja támogatta (NKFI-azonosító: 132312).

The research was supported by the European Union's Horizon 2020 research & innovation programme under Grant Agreement no. 951832.

## Hivatkozások

- Admiral, S.: The use of intensifiers by politicians' concerning coronavirus instructor: Vian ahmed karam. In: Proceedings of the first virtual scientific conference of the English Language Department in the College of Basic Education - Al-Mustansiriyah University in cooperation with the University alturath College. pp. 93–104. the Lebanese University and the Turkish University, Turkey (2020)
- Álvarez-Mellado, E.: A Corpus of Spanish Political Speeches from 1937 to 2019. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 928–932 (2020)
- Andor, J.: De durva ez a téma!–megfigyelések a melléknévi polaritásváltásról. *Hungarológiai Évkönyv* 12(1), 33–42 (2011)
- Andor, J.: Functional studies in the polarity and gradation of amplifier adjectives and adverbs in english. *Studies in English theoretical and applied linguistics* pp. 43–59 (2003)
- Aparicio, J.T., de Sequeira, J.S., Costa, C.J.: Emotion analysis of Portuguese Political Parties Communication over the covid-19 Pandemic. In: 2021 16th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6. IEEE (2021)
- Balogh, P.: Gender-markerek a nyelvben. *Wiener Elektronische Beiträge des Instituts für Finno-Ugristik (WEBFU)* pp. 1–7 (2009)
- Bene, M., Nábelek, F.: A politikai kommunikáció története a külföldi szakirodalomban . In: Kiss, B. (szerk.) *A szavakon túl. Politikai kommunikáció Magyarországon, 1990-2015*, pp. 11–29. L'Harmattan Kiadó (2019)
- Bodó, Cs., Kocsis, Zs., Vargha, F.: A Budapesti Egyetemi Kollégiumi Korpusz. Elméleti és módszertani kérdések. In: Benő, A., Fazakas, N. (szerk.) *Élőnyelvi kutatások és a dialektológia: Válogatás a 19. Élőnyelvi Konferencia - Marosvásárhely, 2016. szeptember 7-9. - előadásából*. pp. 169–177 (2017)
- Bolinger, D.: Degree words. *De Gruyter Mouton* (2013)
- Bonelli, E.T.: Corpus classroom currency. *Darbai ir dienos* 24, 205–243 (2000)
- Chang, Y., Bin, S.: Literature Review on Negative Emotive Intensifiers in English. *Higher Education of Social Science* 18(2), 55–59 (2020)

- Crigler, A.N., Just, M.R.: Measuring affect, emotion and mood in political communication. *The Sage handbook of political communication* pp. 211–224 (2012)
- Crowdy, S.: Spoken corpus design. *Literary and Linguistic Computing* 8(4), 259–265 (1993)
- De Smedt, T., Jaki, S.: The Polly corpus: Online political debate in Germany. In: of the 6th Conference on Computer-Mediated Communication (CMC) and Social Media Corpora (CMC-corpora 2018). p. 33 (2018)
- Dér, Cs.: Grammaticalization: A specific type of semantic, categorical, and prosodic change. *Berliner Beiträge Zur Hungarologie. Schriftenreihe Des Fachgebiets Für Ungarische Literatur Und Kultur an Der Humboldt-Universität Zu Berlin* 18, 160–179 (2013)
- Dilts, P.: Good nouns, bad nouns: what the corpus says and what native speakers think. In: *Corpus-linguistic applications*, pp. 103–117. Brill Rodopi (2010)
- Dressler, W.U., Barbaresi, L.M.: Morphopragmatics: Diminutives and intensifiers in Italian, German, and other languages, vol. 76. Walter de Gruyter (2011)
- Fuchs, R., Gut, U.: Do women use more intensifiers than men?—Investigating gender and age-specific language use with the International Corpus of English. ICAME332012 (2012)
- Fuchs, R., Gut, U.: Register variation in intensifier usage across Asian Englishes. *Discourse-pragmatic variation and change: Insights from English* pp. 185–210 (2016)
- Galántai, J., Pápay, B., Kubik, B.Gy., Szabó, M.K., Takács, K.: A pletyka a társas rend szolgálatában – Az informális kommunikáció struktúrájának mélyebb megértéséért a computational social science eszközeivel. *Magyar Tudomány* 179(7), 964–976 (2018)
- Gollust, S.E., Nagler, R.H., Fowler, E.F.: The emergence of COVID-19 in the US: a public health and political communication crisis. *Journal of health politics, policy and law* 45(6), 967–981 (2020)
- Gósy, M.: BEA—A multifunctional Hungarian spoken language database. *Phonetician* 105, 50–61 (2013)
- Guerini, M., Strapparava, C., Stock, O.: Corps: A corpus of tagged political speeches for persuasive communication processing. *Journal of Information Technology & Politics* 5(1), 19–32 (2008)
- Hamilton, W.L., Clark, K., Leskovec, J., Jurafsky, D.: Inducing domain-specific sentiment lexicons from unlabeled corpora. In: *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing*. vol. 2016, p. 595. NIH Public Access (2016)
- Haselmayer, M., Jenny, M.: Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. *Quality & quantity* 51(6), 2623–2646 (2017)
- Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania* (1990)



- Ito, R., Tagliamonte, S.: Well weird, right dodgy, very strange, really cool: Layering and recycling in English intensifiers. *Language in society* 32(2), 257–279 (2003)
- Jespersen, O.: *Language: Its nature, development and origin*, vol. 68. H. Holt (1922)
- Jing-Schmidt, Z.: Negativity bias in language: A cognitive-affective model of emotive intensifiers. *Cognitive Linguistics* 18(3), 417–443 (2007)
- Klégr, A.: The limits of polysemy: enantiosemy. *Linguistica Pragensia* 23(2), 7–23 (2013)
- Kugler, N.: A nyelvi polaritás kifejezésének egy mintázata, avagy milyen a félelmetesen jó? *Magyar Nyelvőr* 138(2), 129–139 (2014)
- Kugler, N.: *Megfigyelés és következtetés a nyelvi tevékenységben*. Tinta Könyvkiadó (2015)
- Lakoff, R.: Language and woman’s place. *Language in society* 2(1), 45–79 (1973)
- Maekawa, K., Koiso, H., Furui, S., Isahara, H.: Spontaneous Speech Corpus of Japanese. In: *LREC*. pp. 947–9520. Citeseer (2000)
- Mariani, L.A., Gagete-Miranda, J., Retti, P.: Words can hurt: How political communication can change the pace of an epidemic. *Covid Economics* 12, 104–137 (2020)
- Méndez-Naya, B.: Special issue on English intensifiers. *English Language & Linguistics* 12(2), 213–219 (2008)
- Nemesi, A.: Társalgási divatkifejezések a benyomáskeltés szolgálatában. *Magyar Nyelvőr* 122, 24–35 (1998)
- Nir, R.: Electoral rhetoric in israel—the televised debates a study in political discourse. *Language Learning* 38(2), 187–208 (1988)
- Oostdijk, N.: The Spoken Dutch Corpus. Overview and First Evaluation. In: *LREC*. pp. 887–894. Athens, Greece (2000)
- Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian gigaword corpus. In: Calzolari, N.e.a. (szerk.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, Reykjavik, Iceland, 26–31 May. pp. 1719–1723. ELRA (2014)
- Pant, K., Dadu, T., Mamidi, R.: Towards detection of subjective bias using contextualized word embeddings. In: *Companion Proceedings of the Web Conference 2020*. pp. 75–76 (2020)
- Pápay, K., Szeghalmy, Sz., Szekrényes, I.: Hucomtech multimodal corpus annotation. *Argumentum* 7, 330–347 (2011)
- Partington, A.: Corpus evidence of language change. *Text and technology*. In honour of John Sinclair pp. 177–192 (1993)
- Pintilii, M.C., és mtsai: Intensification And Minimization – Persuasive Techniques In The Present Political Discourse. *Comunicare Interculturală și Literatură* 10(2-II), 282–289 (2010)
- Prószéky, G., Miháلتz, M.: Magyar WordNet: az első magyar lexikális szemantikai adatbázis. *Magyar Terminológia* 1(1), 43–57 (2008)
- Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: *A comprehensive English grammar*. London and New York: Longman (1985)

- Rasekh, A.E., Saeb, F.: Gender differences in the use of intensifiers in Persian. *International Journal of Applied Linguistics and English Literature* 4(4), 200–204 (2015)
- Richards, B.: The emotional deficit in political communication. *Political Communication*, 21(3), 339–352 (2004)
- Ring, O., Szabó, M.K., Váradi, B., Guba, Cs., Üveges, I.: Approaches to Sentiment Analysis of Hungarian Political News at Sentence Level with Dictionary-based Method and with Machine Learning (2022), bírálat alatt
- Rufai, S.R., Bunce, C.: World leaders' usage of Twitter in response to the COVID-19 pandemic: a content analysis. *Journal of public health* 42(3), 510–516 (2020)
- Sardabi, N., Afghari, A.: Gender differences in the use of intensifiers. *Journal of Applied Linguistics and Language Research* 2(7), 203–213 (2015)
- Savoy, J.: Lexical analysis of US political speeches. *Journal of Quantitative Linguistics* 17(2), 123–141 (2010)
- Schneider, U., Eitelmann, M.: Linguistic Inquiries into Donald Trump's Language: From 'Fake News' to 'Tremendous Success'. Bloomsbury Publishing (2020)
- Settle, J.: Moving beyond sentiment analysis: Social media and emotions in political communication. *The Oxford Handbook of Networked Communication*. Oxford: Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780190460518.013.20> (2020)
- Shmelev, A.: Cognitive and communicative sources of enantiosemey. In: Cantero, C. (szerk.) *Proceedings of the 10th World Congress of the International Association for Semiotic Studies (IASS/AIS)*. Spain: Universidade da Coruña. pp. 837–844. Universidade da Coruña (2012)
- Siddikova, I., Zubareva, N.: Translation of Intensity in Political Discourse. *Rupkatha Journal on Interdisciplinary Studies in Humanities* 12(5) (2020)
- Stoffel, C.: *Intensives and down-toners: A study in English adverbs*. C. Winter's universitätsbuchhandlung (1901)
- Strohm, F., Klinger, R.: An empirical analysis of the role of amplifiers, downtoners, and negations in emotion classification in microblogs. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 673–681. IEEE (2018)
- Szabó, G.: Emotional Communication and Participation in Politics. *Intersections. East European Journal of Society and Politics* 6(2) (2020)
- Szabó, M.K.: A szentimentérték módosulásának a problémája a magyar nyelvű szövegek szentimentelemzésében, különös tekintettel az értékvesztésre és az értékváltásra. Ph.D.-értekezés, szte (2018)
- Szabó, M.K.: Az értékváltás jelensége a magyar nyelvben. A negatív emotív elemek egy sajátos használatáról. *Magyar Nyelv* 115(3), 309–323 (2019)
- Szabó, M.K., Bibok, K.: Értékvesztésre és értékváltásra képes lexémák újabb vizsgálata. *Argumentum* 15, 639–649 (2019)
- Szabó, M.K., Guba, Cs.: Analysis of Negative Emotive Intensifiers in Hungarian Tweets (2022), kézirat
- Szabó, M.K., Vincze, V., Bibok, K.: Analysis of Hungarian Negative Emotive Words, with Special Regard to The Phenomenon of Polarity Shift (2021a), kézirat

- Szabó, M.K., Vincze, V., Ring, O., Üveges, I., Vít, E., Samu, F., Gulyás, A.E., Galántai, J., Szvetelszky, Zs., Bodor-Eranus, E.H.: StaffTalk: magyar nyelvű spontán beszélgetések korpusza. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 211–224. Szegedi Tudományegyetem, Szeged (2021b)
- Tagliamonte, S., Roberts, C.: So weird; so cool; so innovative: The use of intensifiers in the television series Friends. *American speech* 80(3), 280–300 (2005)
- Trajkova, Z., Neshkovska, S.: Online hate propaganda during election period: The case of Macedonia. *Lodz Papers in Pragmatics* 14(2), 309–334 (2018)
- Tukacs, T.: Túlzásba vitt szavak. A fokozó értelmű szókapcsolatok magyar angol szótára. Budapest, Tinta (2015)
- Van Bael, C., Baayen, R.H., Strik, H.: Segment deletion in spontaneous speech: a corpus study using mixed effects models with crossed random effects. In: INTERSPEECH. pp. 2741–2744 (2007)
- Váradi, T.: A Budapesti Szociolingvisztikai Interjú. In: Kiefer, F., Siptár, P. (szerk.) A magyar nyelv kézikönyve. pp. 339–359. Akadémiai Könyvkiadó, Budapest (2003)
- Vincze, V., Üveges, I., Szabó, M.K., Takács, K.: A magyar beszélt és írott nyelv különböző korpuszainak morfológiai és szófaji vizsgálata. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 169–182. Szegedi Tudományegyetem, Szeged (2021)
- Vlasova, E.: Exaggeration in a political text (on the material of modern british and american newspapers). Available at SSRN 3932128 (2021)
- Wagner, M., Morisi, D.: Anxiety, fear, and political decision making. In: Oxford research encyclopedia of politics. Oxford University Press (2019)
- Wang, Y., Croucher, S.M., Pearson, E.: National Leaders' Usage of Twitter in Response to COVID-19: A Sentiment Analysis. *Frontiers in Communication* p. 183 (2021)
- Xiao, R., Tao, H.: A corpus-based sociolinguistic study of amplifiers in British English. *Sociolinguistic studies* 1(2), 241–273 (2007)
- Zsibrita, J., Vincze, V., Farkas, R.: magyarul: A toolkit for morphological and dependency parsing of Hungarian. In: Proceedings of RANLP. pp. 763–771 (2013)



# Syntax-based data augmentation for Hungarian-English machine translation

Attila Nagy<sup>1</sup>, Patrick Nanys<sup>1</sup>, Balázs Frey Konrád<sup>1</sup>, Bence Bial<sup>1</sup>, Judit Ács<sup>2</sup>

<sup>1</sup>Department of Automation and Applied Informatics  
Budapest University of Technology and Economics

<sup>2</sup>Institute for Computer Science and Control  
Eötvös Loránd Research Network

**Abstract.** We train Transformer-based neural machine translation models for Hungarian-English and English-Hungarian using the Hunglish2 corpus. Our best models achieve a BLEU score of 40.0 on Hungarian-English and 33.4 on English-Hungarian. Furthermore, we present results on an ongoing work about syntax-based augmentation for neural machine translation. Both our code and models are publicly available<sup>1</sup>.

## 1 Introduction

Machine Translation (MT) is a subfield of natural language processing, where the task is to perform translation automatically from one language to another. To be able to translate corpora between arbitrary languages, one needs to develop a deep understanding of the underlying structure of language and for this reason MT has been considered one of the hardest problems in NLP.

Our contributions with this work are twofold. First, we train neural machine translation models for English-Hungarian and Hungarian-English using Transformer models. Second, we propose a new data augmentation technique for machine translation using syntactic parsing. To the best of our knowledge, no prior work has been published on Transformer-based machine translation for Hungarian-English.

## 2 Related work

Early approaches tried to model translation by deriving translation rules based on our knowledge of linguistics. A rule-based method, however, is insufficient for covering the countless edge cases in language. With the increasingly available parallel datasets, data-driven approaches gained dominance in the previous decades. Statistical machine translation (SMT) (Koehn et al., 2003; Brown et al., 1990) outperforms rule-based methods by learning latent structures in the data with the help of statistical methods. Although better than its predecessor, SMT struggles to capture long-term dependencies. Neural machine translation (NMT)

<sup>1</sup> <https://github.com/attilanagy234/syntax-augmentation-nmt>

(Bahdanau et al., 2014; Cho et al., 2014) tackles this problem by modeling translation as an end to end process using neural networks. In current machine translation research, the Transformer architecture (Vaswani et al., 2017) is almost exclusively used in supervised settings (Tran et al., 2021; Germann et al., 2021; Oravecz et al., 2020). For the Hungarian-English language-pair, published methods followed the same evolution of rule-based systems (Prószéky and Tihanyi, 2002), statistical methods (Laki et al., 2013) and neural models (Tihanyi and Oravecz, 2017).

Data augmentation is particularly important in machine translation research, because many language-pairs have insufficient resources to build complex models. Classical augmentation methods that are used in NLP are hard to apply to machine translation, because it is very hard to augment both the source and target sentence such that the parallelism of the sentence-pair holds. Wang and Yang (2015) stochastically select words for replacement based on a distance metric in an embedding space. Kobayashi (2018) train a language model to predict new words based on its surrounding context and use this model to replace words. Xie et al. (2017) avoid overfitting to specific contexts by randomly replacing words in the training data with a blank token. Methods that use the dependency parse tree of a sentence for augmentation were proved useful in a number of tasks such as word relation classification (Xu et al., 2016), POS tagging (Şahin and Steedman, 2019) and dependency parsing (Vania et al., 2019; Dehouck and Gómez-Rodríguez, 2020). Duan et al. (2020) use the depth of words in a dependency parse tree as a clue of importance for selecting words for augmentation in machine translation. The syntax-aware data augmentation that we discuss in this work was first proposed by Nagy (2021). In machine translation, backtranslation is the most common data augmentation method, which creates pseudo-parallel sentences from monolingual data using a baseline translation model (Sennrich et al., 2015).

### 3 Methodology

We discuss two main experiments in this work. Firstly, we train competent neural machine translation models based on state-of-the-art architectures for HU-EN and EN-HU and provide a solid baseline for future NMT research in Hungarian by releasing the trained model. Secondly, we propose a novel data augmentation technique for machine translation using dependency parsing. As data augmentation is particularly useful when training data is insufficient, we perform these experiments in a simulated low-resource setting, using a subset of the Hungarian2 corpus.

#### 3.1 Formulation

We formulate machine translation on the sentence level. Given a dataset  $\mathcal{D}$  that contains parallel sentences from the source and target language  $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ , we

define the goal of neural machine translation as estimating the unknown conditional probability  $P(\mathbf{y}|\mathbf{x})$ . This is a classical sequence-to-sequence problem: an encoder can be used to create a representation of the source sentence, which is fed into a decoder with the purpose of extracting relevant information from this representation. The decoder then generates the output symbols from left to right. This way the decoder can be thought of as a language model conditioned on the output of the encoder and the already generated symbols.

### 3.2 Dataset

We use the Hunglish2 corpus for our experiments, which is a sentence-aligned corpus consisting of 4.1M Hungarian-English bisentences (Varga et al., 2007). The dataset was constructed by scraping and aligning bilingual data in several domains from the internet. The distribution of each domain that make up Hunglish2 is shown in Table 1.

Subcorpus	Tokens	Bisentences
Modern literature	37.1M	1.67M
Classical literature	17.2M	652k
Movie subtitles	3.2M	343k
Software docs	1.2M	135k
Legal text	56.6M	1.351M
Total	115.3M	4.151M

Table 1: Statistics of the Hunglish2 corpus.

We applied thorough preprocessing before training the models. First, we removed seemingly incorrect data points: sentence pairs, which contained HTML code or were outliers with respect to sentence length. We also remove sentence-pairs, where either side is an empty string. A large number of sentences were wrapped in quotation marks, so in order to avoid overfitting on this behaviour, we also remove leading and trailing quotation marks. Next, we filter the dataset with length-based heuristics. We drop sentences if either the source or target sentence contains more than 32 words. Furthermore, we filter with the relative word counts of the source and target sentences using the below rule:

$$(|WC(x) - WC(y)| < 7) \vee (WC(x)/WC(y) < 1.6)$$

where  $WC(x)$  and  $WC(y)$  are the word counts of the source and target sentences respectively. The threshold parameters were determined by exploratory data analysis and a series of experiments. See Figure 1 for the distribution of length difference and length ratio of bisentences in the raw Hunglish2 corpus. The post-processed dataset contains 3.4M bisentences. Finally, we split the data to train,

development and test sets, with a 99-0.5-0.5 ratio. We do this with stratified sampling with respect to the the subcorpora in Hunglish2, to ensure that all splits have a similar distribution.

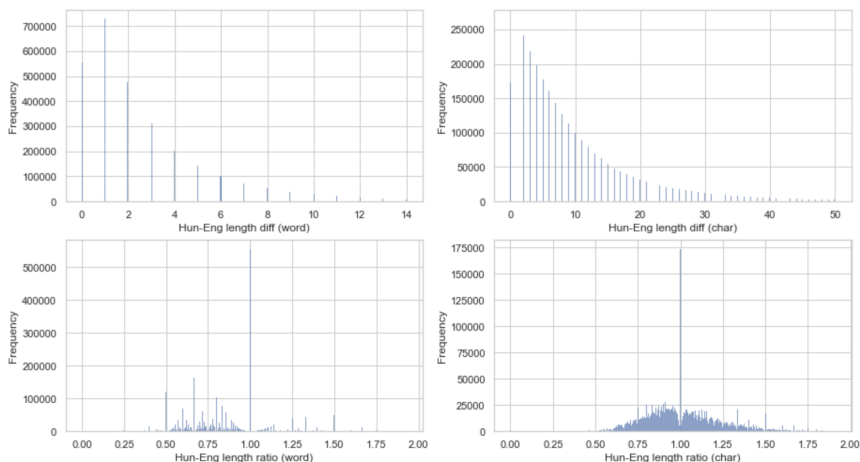


Fig. 1: Word- and character-level distributions of the length difference and ratio between source and target sentences in the Hunglish2 corpus.

### 3.3 Training

All of our models use identical architecture and hyperparameters: a vanilla Transformer-based encoder-decoder model. We tokenize the input with the unigram sentencepiece subword tokenizer (Kudo and Richardson, 2018), which works particularly well with morphologically rich languages, such as Hungarian. We found that using a shared vocabulary of size 32000 yields the best results. We perform gradient descent using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 2, 8000 warmup steps and Noam learning rate decay. To avoid overfitting, we perform early stopping based on the perplexity computed on the validation set. Due to computational constraints, we only perform a manual search in the hyperparameter space. The complete set of hyperparameters of our best model can be found in Table 2. All models were trained using the OpenNMT framework (Klein et al., 2017) with standalone Nvidia Tesla V100 GPUs.

### 3.4 Syntax-aware data augmentation

One of the greatest challenges in data augmentation for machine translation is preserving parallelism. Taking the example from Duan et al. (2020) (see Table



3.) it is visible, that replacing words only on the source side of the bisentence can easily lead to noisy translation pairs. The success of backtranslation can be partially contributed to the fact that it is good at generating parallel bisentences. Backtranslation, however, might not always be an option, especially in low-resource scenarios, where there is not enough parallel data to train a model that can be used for backtranslation in the first place.

Parameter	Value	Parameter	Value
batch type	tokens	batch size	4096
accumulation count	4	average decay	0.0005
train steps	150000	valid steps	5000
early stopping	4	early stopping criteria	ppl
optimizer	adam	learning rate	2
warmup steps	8000	decay method	noam
adam beta2	0.998	max grad norm	2
label smoothing	0.1	param init	0
param init glorot	true	normalization	tokens
max generator batches	32	encoder layers	8
decoder layers	8	heads	16
RNN size	1024	word vector size	1024
Transformer FF	2096	dropout steps	0
dropout	0.1	attention dropout	0.1
share embeddings	true	position encoding	true

Table 2: Hyperparameters of our best model.

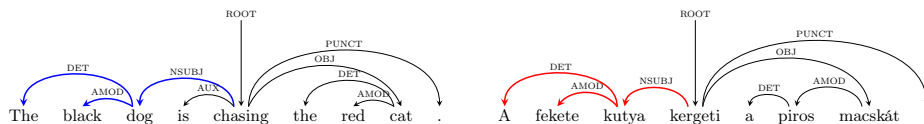


Fig. 2: An example of finding the same substructure in the dependency trees of a bisentence.

We propose a novel syntax-aware data augmentation technique, which is based on a hypothesis, that dependency parse trees of the source and target sentences contain subtrees, which carry the same meaning. If we can identify these subtrees simultaneously in the source and target language (see Figure 2 for an example), we have the possibility to generate new bisentences by swapping subtrees.

Original-EN	We shall fight on the beaches.
Original-HU	A tengerparton kellene küzdenünk.
Replacement-EN	We shall fight with the sandy.
Replacement-HU (Google Translate)	Harcolni fogunk a homokkal.

Table 3: Error analysis of a data augmentation method for NMT (Duan et al., 2020).

Finding any subtree-pair, which correspond to the same part of the source and target sentence is a very hard task, so we limit our work to finding two substructures that are common across languages: subjects and objects (see Figure 3 and Table 4). As the problem space of new sentence-pairs that we can generate explodes quickly with respect to the size of the dataset, we filter sentences based on their dependency parse trees with two conditions. First, the dependency trees must contain only one subject and object for both the source and target sentences. Second, the subtrees corresponding to objects and subjects must be a consecutive sequence of words with respect to the original word order. In our experiments, we found that about 5% of the sentence pairs satisfy the above two conditions and therefore are eligible for augmentation. It is convenient to have the same dependency relations for both English and Hungarian, so we use the Universal Dependencies<sup>2</sup> (Nivre et al., 2016) tag set. Implementation-wise, we chose the Stanza<sup>3</sup> dependency parser (Qi et al., 2020) for English and the parser in the Hungarian Spacy<sup>4</sup> model for Hungarian.

We perform a series of simulated low-resource experiments on a subcorpus of Hunglish2. We constrain our experiment to only one subcorpus, because this better simulates the lack of diversity in parallel corpora of low-resource languages. We subsample from the Modern Literature subcorpus of Hunglish2 from 5k to 500k. For each dataset, we perform three experiments: train a baseline Transformer model and train the same model with additional data from the two proposed augmentation methods. In every iteration, we extend the datasets with 50% augmented data. Every model is evaluated on the same held-out test set, that is a fraction of our original test set with samples only from the Modern Literature subcorpus.

We acknowledge that sampling from a medium- or high-resource corpus is not the same as working on a truly low-resource language pair. The latter is likely a much worse representation of the overall population, but for the course of this work we limit our experiments to Hungarian-English. Extending these experiments to actual low-resource language pairs is a promising direction for future work.

<sup>2</sup> <https://universaldependencies.org/>

<sup>3</sup> <https://stanfordnlp.github.io/stanza/>

<sup>4</sup> <https://github.com/huspacy/huspacy>

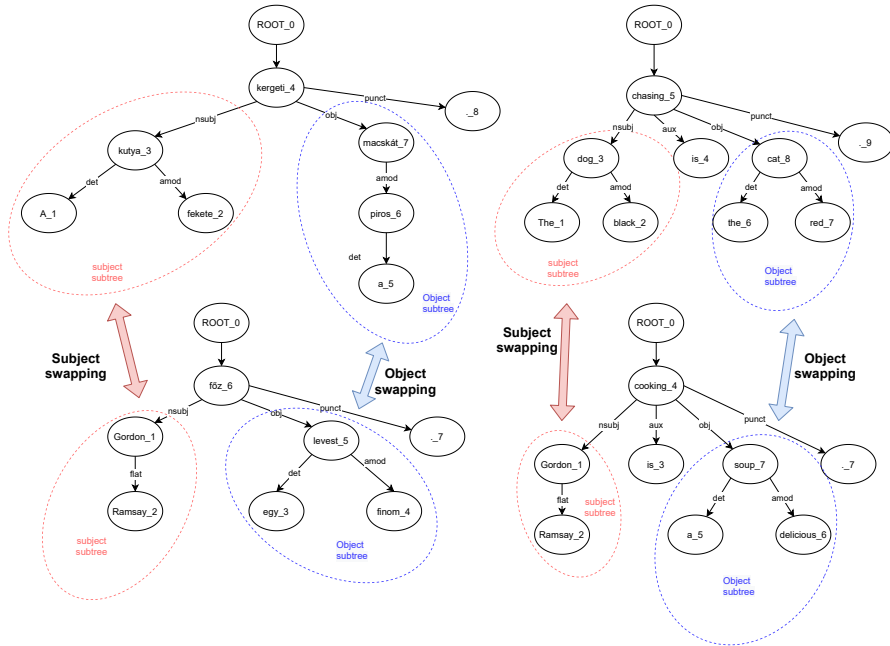


Fig. 3: The process of swapping subject and object dependency subtrees between bisentences. The augmented sentences can be seen in Table 4.

Sentence1-EN	The black dog is chasing the red cat.
Sentence1-HU	A fekete kutya kergeti a piros macskát.
Sentence2-EN	Gordon Ramsay is cooking a delicious soup.
Sentence2-HU	Gordon Ramsay egy finom levest főz.
EN-OBJ-AUG-1	The black dog is chasing a delicious soup.
HU-OBJ-AUG-1	A fekete kutya kergeti egy finom levest.
EN-OBJ-AUG-2	Gordon Ramsay is cooking the red cat.
HU-OBJ-AUG-2	Gordon Ramsay a piros macskát főz.

Table 4: Augmentation via subtree swapping of objects.

## 4 Results

In this section, we provide a detailed evaluation of both the Transformer-based machine translation models and the proposed augmentation method. For quantitative evaluation, we use the BLEU score.

**NMT models for EN-HU and HU-EN** Our baseline model without augmentation achieved a BLEU score of **33.4** for EN-HU and **40.0** for HU-EN on the held-out test set. In Table 5, we collected a few example translations from the test set provided by our best models.

**Syntax-aware data augmentation** The scores of our simulated low-resource experiments can be found in Table 7 for both HU-EN and EN-HU. The absolute performance gain due to the augmentation is visualized in Figures 4 and 5 for each sample size. With a smaller sample size (5k, 10k, 25k), the models had a close to 0 BLEU score. This can likely be contributed to the fact that the model is fixed for all experiments and it is probably too complex for datasets of this size regardless of augmentation. In the 50k-100k range, we observe visible improvements in the BLEU score of models trained with augmentation compared to the baseline models. With a 75k sample size, the baseline BLEU scores of 0.9 and 1.4 are significantly outperformed by the model trained with object swapping augmentation with BLEU scores of 6.1 and 8.8 for EN-HU and HU-EN respectively. In the 100k-500k range, we do not see improvement in BLEU score with the augmentation methods. With two exceptions, the models using augmented data perform slightly worse than the baseline model. As the augmentation ratio is also fixed to 0.5 during all experiments, it is possible that at this scale we inject too many and too noisy new data points into the training set. We also examined the reason behind the noisiness of the augmentation by manual analysis. We collected a few common error types, which are listed in Table 6. Apart from the ones listed below, we found that most of the errors propagate from an incorrect dependency parsing, especially for Hungarian. We observed many falsely identified subjects, especially in cases, where the subject of the sentence was dropped (pronoun-dropping).

## 5 Conclusion

We presented Transformer-based NMT models for Hungarian-English and English-Hungarian. Our best models achieve a BLEU score of **40.0** and **33.4** for HU-EN and EN-HU respectively. We also shared results of an ongoing work on a potential data augmentation method alternative to back-translation in lower-resource scenarios. We briefly discussed this syntax-aware method, which creates new data points by swapping specific subtrees of dependency parse trees in parallel for both the source and target sentences. Regarding our future work, we plan to fix some of the common errors listed in Table 6 and therefore enhance the

Example #	Sentence	
1	<b>Source</b>	Villefort ezeket az utolsó szavakat olyan lázas dühvel ejtette ki, ami egészen vadul ékesen szólóvá tette.
	<b>Reference</b>	Villefort pronounced these last words with a feverish rage, which gave a ferocious eloquence to his words.
	<b>Predicted</b>	Villefort pronounced these last words with a feverish rage which rendered him passionately eloquent.
2	<b>Source</b>	Amióta Merytonba szállásolták az ezredet, csak a szerelem, az udvarlás, a tiszték jártak az eszében.
	<b>Reference</b>	Since the shire were first quartered in Meryton, nothing but love, flirtation, and officers have been in her head.
	<b>Predicted</b>	He had been thinking of love, of courting, of officers, ever since the regiment came to Meryton.
3	<b>Source</b>	Malfoy, Crak és Monstro Csikócsőrrel próbálkoztak.
	<b>Reference</b>	Malfoy, Crabbe, and Goyle had taken over Buckbeak.
	<b>Predicted</b>	Malfoy, Crabbe, and Goyle had tried Buckbeak.
4	<b>Source</b>	His remembrance shall be sweet as honey in every mouth, and as music at a banquet of wine.
	<b>Reference</b>	Mint nyünknek a méz, édes az emléke, vagy mint a nótaszó borozgatás közben.
	<b>Predicted</b>	Emlékezete édes lesz, mint a méz minden szájban, és mint a zene a bor lakomáján.
5	<b>Source</b>	His eyes moved toward the hunting knife that had been slung over the mosquito-net bar by the dead man the day he arrived.
	<b>Reference</b>	Szeme a vadászkésre siklott, amelyet a halott ember a moszkítóháló keretére dobott azon a napon, amikor megérkezett.
	<b>Predicted</b>	Szeme a vadászkés felé fordult, melyet a halott férfi a moszkítóháló rácsára dobott az érkezése napján.
6	<b>Source</b>	He was frozen stiff in the weeds beside the track.
	<b>Reference</b>	Csonttá fagyva feküdt a vágány mellett a gazos földön.
	<b>Predicted</b>	Merevre fagyott a vágány melletti gazban.

Table 5: Example translations produced by the best EN-HU and HU-EN translator models.

<b>Error type</b>	<b>Sentences</b>
Article definiteness	<p><b>Source (aug)</b> Outside the apothecary, Hagrid checked <b>the</b> weapons again.</p> <p><b>Target (aug)</b> Mikor végeztek a patikában, Hagrid még egyszer ellenőrizte <b>a</b> fegyvert.</p>
Coreference	<p><b>Source (aug)</b> Two other companies claimed only <b>it</b>.</p> <p><b>Target (aug)</b> Két másik vállalat csak <b>egyéni elbánást</b> kérelmezett.</p>
Conjugation	<p><b>Source (aug)</b> Member states had also played an ironic role here.</p> <p><b>Target (aug)</b> A tagállamok ismét ironikus játékot <b>űzött</b> vele.</p>
Different subject	<p><b>Source (aug)</b> <b>Captain</b>, how many men did the wind leave on mars?</p> <p><b>Target (aug)</b> <b>A szél</b>, hány embert hagytál a Marson?</p>
Pronoun dropping	<p><b>Source (orig)</b> It shall submit the reports to the european parliament and to the council.</p> <p><b>Target (orig)</b> <b>Ő</b> A jelentéseket az Európai Parlamenthez és a Tanácshoz nyújtja be.</p>

Table 6: Common error types of the syntax-aware data augmentation.

Sample size	Method	EN-HU BLEU	HU-EN BLEU
5k	base	0.1	0.0
	object swapping	0.0	0.0
	subject swapping	0.0	0.0
10k	base	0.2	0.1
	object swapping	<b>0.3</b>	0.1
	subject swapping	<b>0.3</b>	0.0
25k	base	0.4	0.1
	object swapping	0.3	0.1
	subject swapping	0.3	<b>0.3</b>
50k	base	0.4	0.8
	object swapping	<b>1.6</b>	<b>2.4</b>
	subject swapping	<b>1.7</b>	<b>2.6</b>
75k	base	0.9	1.4
	object swapping	<b>6.1</b>	<b>8.8</b>
	subject swapping	<b>5.9</b>	<b>8.2</b>
100k	base	3.0	5.4
	object swapping	<b>7.4</b>	<b>9.7</b>
	subject swapping	<b>7.8</b>	<b>10.0</b>
200k	base	12.7	14.5
	object swapping	12.1	14.5
	subject swapping	12.2	<b>14.9</b>
300k	base	14.4	16.5
	object swapping	14.1	16.4
	subject swapping	14.1	16.1
400k	base	15.5	17.0
	object swapping	15.3	16.7
	subject swapping	15.3	16.9
500k	base	15.5	17.7
	object swapping	15.5	17.6
	subject swapping	<b>15.6</b>	17.2

Table 7: Results of our low-resource experiments. The BLEU scores in bold indicate an experiment, where the model with augmentation outperformed the baseline for that particular sample.

augmentation technique by making the generated samples less noisy. We also plan to extend our experiments to other languages.

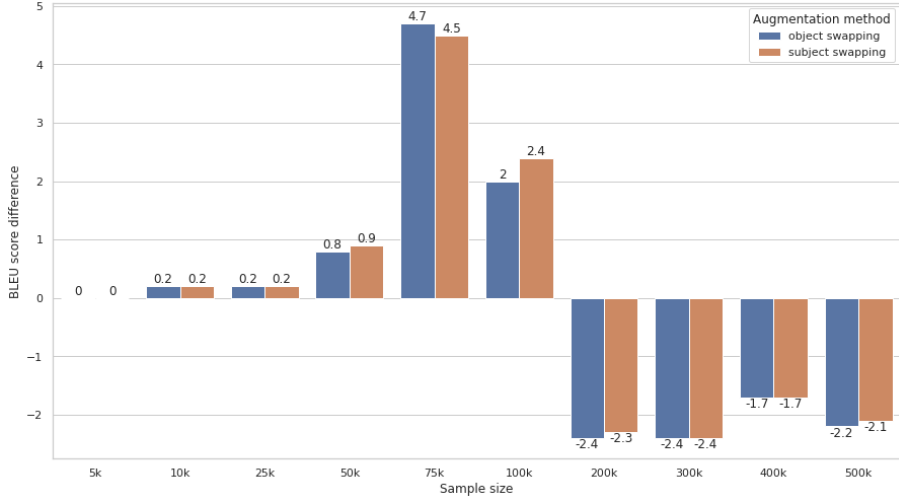


Fig. 4: Absolute BLEU score differences compared to the baseline model for each sample size. (EN-HU)

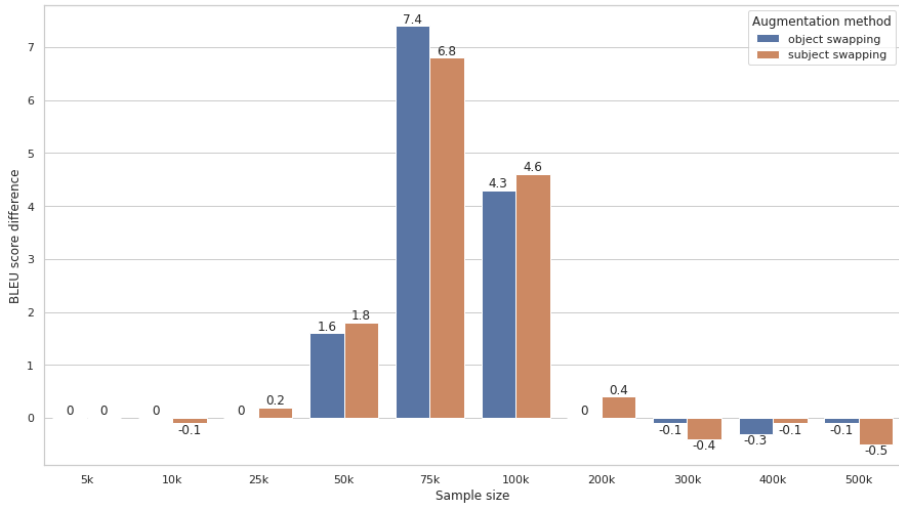


Fig. 5: Absolute BLEU score differences compared to the baseline model for each sample size. (HU-EN)



## Acknowledgements

The authors would like to thank András Kornai for discussions on the syntax-aware data augmentation.

## Bibliography

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
- Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. *Computational linguistics* 16(2), 79–85 (1990)
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
- Dehouck, M., Gómez-Rodríguez, C.: Data augmentation via subtree swapping for dependency parsing of low-resource languages. In: *Proceedings of the 28th International Conference on Computational Linguistics*. pp. 3818–3830 (2020)
- Duan, S., Zhao, H., Zhang, D., Wang, R.: Syntax-aware data augmentation for neural machine translation. arXiv preprint arXiv:2004.14200 (2020)
- Germann, P.C.J.H.U., Bogoychev, L.B.N., Waldendorf, A.V.M.B.J., Heafield, A.B.K.: The University of Edinburgh’s English-German and English-Hausa submissions to the WMT21 news translation task (2021)
- Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014), <https://arxiv.org/abs/1412.6980>
- Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: Open-source toolkit for neural machine translation. In: *Proceedings of ACL 2017, System Demonstrations*. pp. 67–72. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), <https://www.aclweb.org/anthology/P17-4012>
- Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201 (2018)
- Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. Tech. rep., UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST (2003)
- Kudo, T., Richardson, J.: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226 (2018)
- Laki, L., Novák, A., Siklósi, B.: English to Hungarian morpheme-based statistical machine translation system with reordering rules. In: *Proceedings of the Second Workshop on Hybrid Approaches to Translation*. pp. 42–50 (2013)
- Nagy, A.: Developing neural machine translation models for Hungarian-English. arXiv preprint arXiv:2111.04099 (2021)
- Nivre, J., De Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al.: Universal dependencies v1: A multilingual treebank collection. In: *Proceedings of the Tenth*

- International Conference on Language Resources and Evaluation (LREC'16). pp. 1659–1666 (2016)
- Oravecz, C., Bontcheva, K., Tihanyi, L., Kolovratnik, D., Bhaskar, B., Lardilleux, A., Klocek, S., Eisele, A.: etranslation's submissions to the WMT 2020 news translation task. In: Proceedings of the Fifth Conference on Machine Translation. pp. 254–261 (2020)
- Prószéky, G., Tihanyi, L.: Metamorpho: A pattern-based machine translation system. In: Proceedings of the 24th 'Translating and the Computer' Conference. pp. 19–24. Citeseer (2002)
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082 (2020)
- Şahin, G.G., Steedman, M.: Data augmentation via dependency tree morphing for low-resource languages. arXiv preprint arXiv:1903.09460 (2019)
- Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709 (2015)
- Tihanyi, L., Oravecz, C.: First experiments and results in English-Hungarian neural machine translation. XIII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Hungary (2017)
- Tran, C., Bhosale, S., Cross, J., Koehn, P., Edunov, S., Fan, A.: Facebook AI WMT21 news translation task submission. arXiv preprint arXiv:2108.03265 (2021)
- Vania, C., Kementchedjhieva, Y., Sogaard, A., Lopez, A.: A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. arXiv preprint arXiv:1909.02857 (2019)
- Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., Trón, V.: Parallel corpora for medium density languages. Amsterdam Studies In The Theory And History Of Linguistic Science Series 4 292, 247 (2007)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
- Wang, W.Y., Yang, D.: That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2557–2563 (2015)
- Xie, Z., Wang, S.I., Li, J., Lévy, D., Nie, A., Jurafsky, D., Ng, A.Y.: Data noising as smoothing in neural network language models. arXiv preprint arXiv:1703.02573 (2017)
- Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. arXiv preprint arXiv:1601.03651 (2016)

## Jobban fordítunk magyarra, mint a Google!

Laki László János, Yang Zijian Győző

Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.

{laki.laszlo, yang.zijian.gyozo}@nytud.hu

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport

Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar  
1083 Budapest, Práter u. 50/a.

{laki.laszlo, yang.zijian.gyozo}@itk.ppke.hu

**Kivonat** Kutatásunkban célul tűztük ki, hogy áttekintjük az elérhető gépfordító architektúrákat és megvizsgáljuk, hogyan teljesítenek angol-magyar környezetben. A magyar, mint célnyelv nagy kihívás a fordító-rendszerek számára, mivel mind grammatikai, mind szórendi szempontból eltért az angoltól. Kipróbáltunk ipari és kutatásra szánt gépfordító rendszereket egyaránt. Kutatásunk legfontosabb eredménye, hogy az általunk létrehozott modellek (MarianNMT és BART) nagymértékben jobban teljesítettek, mint a piacvezető multinacionális cégek rendszerei. Végül sikerrel finomhangoltunk (fine-tuning) angol-magyar fordításra egy többnyelvű T5 nyelvmoddelt, aminek az eredménye összehasonlítható volt az elérhető modellekével.

**Kulcsszavak:** neurális gépi fordítás, Marian-NMT, BART, mT5

### 1. Bevezetés

Az Infokommunikációs és Információtechnológiai Nemzeti Laboratóriumot (InfoLab) létrehozó konzorcium tagjai (a Nemzetbiztonsági Szakszolgálat és az IdomSoft Zrt.) kettős célt tűztek ki maguk elé: kutatásaikkal támogatni szeretnék a feltörekvő infokommunikációs és információs technológiák biztonságos bevezetését és alkalmazását, valamint az államigazgatás digitális átalakulását.

Az IdomSoft Zrt. egyik kiemelt célja a Mesterséges Intelligencián (MI) alapuló technológiák közigazgatási célú alkalmazási lehetőségeinek kutatása és alkalmazása, mely lehetővé teszi, hogy az ügyfelek mentesüljenek minden olyan adat szolgáltatása alól, amely adat a közigazgatásban már rendelkezésre áll. A fejlesztések megkímélik az ügyfeleket mindazon szervezési és ügyintézési feladatoktól, amelyek a közigazgatási szervek közötti, belső igazgatásszervezéssel megoldhatóak. Cél mellett az érintésmentes, teljes mértékben digitalizált és automatizmusokkal segített, biztonságos és zökkenőmentes ügyintézés megteremtése is.

Ennek a stratégiai innovációnak részét képezi többek között a magyar nyelv sajátosságait készség szinten kezelő, a XXI. század elvárásainak megfelelő közigazgatási szolgáltatások megvalósíthatósága. Az IdomSoft Zrt. e célkitűzések

megvalósításában együttműködik hazai egyetemekkel, hogy azok K+F folyamatban megvalósított termékeit az általa fejlesztett közigazgatási informatikai megoldások kapcsán, a gyakorlatban alkalmazza.

A közigazgatás egyik fontos feladata az idegen nyelvű dokumentumok fordítása és a szakfordítók támogatása. Napjainkra a gépi fordítás napi szintű használati eszközzé vált. Az utóbbi években a neurális módszerek, azon belül is a transzformer modellek érték el a legjobb eredményeket a legtöbb nyelvtechnológiai feladatban, beleértve a gépi fordítást is. A neurális gépi fordítás eszközei látványosan jobb, emberek számára olvashatóbb fordítást produkálnak elődjeikhez képest, ennek köszönhetően alkalmassá váltak arra, hogy az emberi fordítók számára előfordítóként használják őket. A nemzetközi publikációk kiemelt figyelmet szentelnek az angol nyelvnek, és különböző architektúrákkal kísérleteznek egyre jobb fordítási teljesítmény elérése érdekében. Kutatásunk során ezeket a különböző modelleket vizsgáljuk fókuszba a magyart, mint célnyelvet állítva. Arra keressük a választ, hogy vajon miben különböznek ezek a módszerek egymástól, mennyire tudnak magyarra fordítani, illetve kvantitatív módszerekkel állapítjuk meg a közöttük fennálló különbséget. Modelljeink és szkriptjeink megtalálhatóak a Github<sup>1</sup> és Hugging Face<sup>2</sup> oldalainkon.

## 2. Neurális gépfordító rendszerek és módszerek

A cikkünkben összehasonlítottunk kutatásra szánt és ipari gépfordító rendszereket egyaránt. Az ipari termékek közül nem mindegyik rendszernél sikerült felderíteni a modell architektúráját és paramétereit. Ebben a fejezetben részletesen bemutatjuk a vizsgált rendszereket. A felsorolt lista tekinthető irodalomkutatásnak is, mivel szinte teljesen lefedjük az elérhető magyar nyelvű fordítórendszereket.

### 2.1. Marian NMT

A Marian NMT (Junczys-Dowmunt és mtsai, 2018) nevű keretrendszer egy C++ nyelven íródott szabadon hozzáférhető programcsomag. Könnyen telepíthető, jól dokumentált, memória- és erőforrás-optimalis implementációjának köszönhetően<sup>3</sup> az akadémiai felhasználók és fejlesztők által leggyakrabban használt eszköz (Barrault és mtsai, 2019). A Marian NMT egy figyelmi (attention) modellel támogatott enkóder-dekóder architektúrájú neurális gépfordító modell. Legnagyobb előnye a többi módszerhez képest, hogy előtanított nyelvmodellek használata nélkül is a leggyorsabb futási idejű tanítást eredményezi. Két méretű transzformer modell tanítható be:

- Marian small: 6 réteg enkóder és 6 réteg dekóder; 8 figyelmi fej; 512 szóbeágyazás dimenzió; bementi hossz: 512; előre csatolt háló méret: 2048
- Marian big: 6 réteg enkóder és 6 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; előre csatolt háló méret: 4096

<sup>1</sup> <https://github.com/nytud/machine-translation>

<sup>2</sup> <https://huggingface.co/NYTK>

<sup>3</sup> <https://marian-nmt.github.io/>

## 2.2. BART és mBART

A BART (Lewis és mtsai, 2020) modell egy enkóder-dekóder architektúrán alapuló transzformer modell, amelyet a Fairseq (Facebook AI Research Sequence-to-Sequence Toolkit) fejlesztett<sup>4</sup>. Az enkóder kétirányú (Bidirectional), a dekóder autoregresszív (Autoregressive). A BART gyakorlatilag ötvöz egy BERT (Devlin és mtsai, 2019) és egy GPT (Radford és Narasimhan, 2018) típusú modellt. A kutatások alapján a BERT típusú modellek kiválóan alkalmasak szószintű és mondat szintű osztályozási feladatokra, azonban szöveggenerálás esetében gyengén teljesítenek. Ezzel ellentétben a GPT típusú autoregresszív modellek elsősorban szövegek generálására működnek jól, mint például szövegösszegzés vagy szöveggenerálás. A BART a két architektúra előnyeit ötvözi, ezért kiválóan alkalmas gépi fordításra. Jelenleg két különböző méretű BART érhető el: 1.) BART-base: 6 réteg enkóder és 6 réteg dekóder; 12 figyelmi fej; 768 szóbeágyazás dimenzió; bementi hossz: 512; 140 millió paraméter 2.) BART-large: 12 réteg enkóder és 12 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; 400 millió paraméter

Az mBART (Liu és mtsai, 2020) egy több nyelven előtanított, zajtalanító (denoising) autoenkóder modell, mely a seq2seq koncepción alapul. A többnyelvű zajtalanító előtanítás alkalmazható mind a felügyelt (supervised), mind pedig a felügyelet nélküli (unsupervised) gépi tanulás teljesítményének javítására. Az mBART felépítését tekintve a BART sémáját követi. A modell szerzői az előtanítás során fókuszálnak a többnyelvűsége, a betanított modell ezt követően kutatásukban kétnyelvű beállításban kerül finomhangolásra. Az előtanításhoz a Common Crawl adatbázisból kivonatolt 25 nyelvet tartalmazó CC25 (Wenzek és mtsai, 2020; Conneau és mtsai, 2020) korpuszt használták. Mind mondat szintű, mind pedig dokumentumszintű gépi fordításra alkalmazták a több nyelven előtanított modellt. Kiemelkedő jelentőségű, hogy kizárólag a seq2seq koncepció használatával tudták javítani a dokumentumszintű gépi fordítás minőségét, ez korábbi hasonló munkákhoz (Miculicich és mtsai, 2018; Li és mtsai, 2019) képest jelentős előrelépést jelent. Az mBART modellel végzett munka rámutat a többnyelvű előtanításban rejlő lehetőségek transzfer tanulós (transfer learning) irányban való felhasználhatóságára.

Az mBART nem tartalmazza a magyar nyelvi tudást, ezért erre a célra saját angol-magyar nyelvű BART modellt tanítottunk.

## 2.3. T5 és mT5

A T5 (Text-To-Text Transfer Transformer) (Raffel és mtsai, 2020), a Google kutatócsapata által publikált, modell és keretrendszer újfajta lehetőségeket kínál a nyelvfeldolgozási feladatok terén. A természetes nyelvfeldolgozás eszköztárában kiemelt szereppel bír a transzfer tanulás (transfer learning), melynek során a nyelvi modell egy adatokban gazdag feladaton van tanítva, majd ezt követően kerül finomhangolásra egy soron következő célfeladatra. Ideális esetben a modell

<sup>4</sup> <https://github.com/pytorch/fairseq/tree/master/examples/bart>

az előtanítás folyamán olyan általános tudásra tesz szert, amely átvihető és sikeresen alkalmazható a célfeladatok esetében is. A T5 projekt a transzfer tanulási alapelveket alkalmazza a szövegből szöveg (text-to-text) problémamegközelítés kontextusában. Kiindulási ötletként az szolgált, hogy minden szövegelemzési feladatot (fordítás, kérdések megválaszolása, osztályozás) szövegből szöveg problémaként közelít meg, azaz szöveg a bemenet és egy újabb szöveg lesz a kimenet. A szövegből szöveg eljárások nagy előnye a széleskörű alkalmazhatóság, ugyanis gyakorlatilag bármilyen természetes nyelvelemzési feladatra felhasználhatók, így például gépi fordításra, összefoglaló készítésére, kérdések megválaszolására vagy épp szentiment analízisre.

Az ilyen nagy volumenű kísérletekhez speciális korpusz szükséges, ennek érdekében lett létrehozva az ún. Colossal Clean Crawled Corpus (rövidítve C4), amely egy több száz gigabájtnyi világhálóról összegyűjtött és tisztított angol nyelvű szöveget tartalmaz. A C4 korpusz alapját a Common Crawl<sup>5</sup> adatbázis képezi. A transzfer tanulási módszereknek fontos jellegzetessége, hogy az előtanításhoz jelöletlen adathalmazra van szükség. További kívánalmak egy ilyen keretrendszerben alkalmazható korpusz felé, hogy nagy méretű, változatos és magas minőségű legyen. Összehasonlításképpen, az alkalmazott C4 korpusz kétszer akkora, mint a Wikipédia, vagyis lényegesen nagyobb mennyiségű adatot biztosít. Az T5 esetében a paraméterek száma alapján 5 különböző méretű modell került kialakításra, ezek a következők: Small (300 millió paraméter), Base (580 millió paraméter), Large (1,2 milliárd paraméter), XL (3,7 milliárd paraméter), XXL (13 milliárd paraméter)

A T5 projekt eredményeként létrehozott keretrendszer kiváló eredmények elérését tette lehetővé. A 11 milliárd paraméterrel futatott legnagyobb csúcsteljesítményt nyújtott több tekintetben is, így például GLUE, SuperGLUE, SQuAD és a CNN/Daily Mail referencia feladatokban.

Az **mT5** (Xue és mtsai, 2021) a korábban tárgyalt T5 több nyelvre kiterjesztett verziója. Az mT5 létrehozása során a szerzők törekedtek arra, hogy minél inkább megőrizték a kísérletek során több ízben is kiemelkedően teljesítő T5 strukturális jegyeit, ennek megfelelően az mT5 örökölte a szövegből szöveg (text-to-text) alapú problémamegközelítést és az általános előtanítás menetét is, amelyhez szintén rendkívül nagy méretű korpuszt használtak.

Az mT5 betanításához az mC4 korpuszt használták, amely a T5 tanítására alkalmazott C4 többnyelvű változata, és 101 különböző nyelvből szerepelnek benne szövegek. Többnyelvű modellek esetében általánosan felmerülő probléma, hogy amennyiben egy nyelv kevesebb forrással rendelkezik, akkor előfordulhat, hogy a gyakoribb mintavétel miatt a modell illesztése nem megfelelően alakul. Ezen probléma kiküszöbölésére korábbi modellek esetében is alkalmazott gyakoriság-alapú mintavételi eljárást (Devlin és mtsai, 2019; Aharoni és mtsai, 2019) használtak a szerzők. Mivel az mT5 modell több, mint száz nyelvből álló korpuszon lett betanítva, ezért szükséges volt egy nagyobb méretű –250 ezer szóelemből álló –szótár alkalmazása is.

<sup>5</sup> <https://commoncrawl.org>

Az mT5 teljesítményének kiértékeléséhez az XTREME többnyelvű referenciarendszer (Hu és mtsai, 2020) 6 feladata lett alkalmazva. A keretrendszerben szerepelnek mondatpár, névelemfelismerés és kérdés megválaszolás feladatok is. Az eredmények tekintetében a legnagyobb modell, az mT5-XXL csúcsteljesítményt nyújtott kérdés megválaszolásos feladatokban.

Az mT5 projekt megmutatta, hogy a T5 modellrendszer kiválóan alkalmazható többnyelvű kontextusban is, és rendkívül erős eredményeket tud elérni különböző referenciafeladatokban. Az mT5 tartalmazza a magyar nyelvi tudást is, ezért kutatásunkban alkalmaztuk az mT5 small modellt gépi fordításra.

## 2.4. M2M100

Az M2M100 (Aharoni és mtsai, 2019) a Fairseq többnyelvű gépi fordítás projektje<sup>6</sup>. A többnyelvű gépi fordítás (multilingual machine translation) célja egy olyan átfogó modell megalkotása, amely bármely nyelvről bármely nyelvre képes fordítani. A gépi fordítás sokáig meglehetősen angolközpontúnak számított, azaz zömében olyan nyelvi modellek születtek, amelyek angolról vagy angolra fordítottak. A valóságban azonban a fordítás nem ilyen kizárólagos módon van felhasználva, tehát sok egyéb más nyelvről és más nyelvekre történő fordításra is komoly igény mutatkozik. A M2M100 projekt keretében 100 nyelvre készült el egy olyan adathalmaz, amely egy nagyfokú diverzitást hozó, az angol nyelv gépi fordítás tekintetében elfoglalt egyeduralmát megtörő módszertani újítás felé nyitja meg az utat. Ez végeredményben lehetővé teszi az újgenerációs többnyelvű gépi fordító modellek megalkotását. A sok nyelvről sok nyelvre történő gépi fordításhoz nagy méretű adathalmaz létrehozására volt szükség. Az ilyen nagy volumenű soknyelvű adathalmazok generálásához szükséges az adatbányászat (Artetxe és Schwenk, 2019) és a visszafordítás (Sennrich és mtsai, 2016) alkalmazása. Az M2M100 tartalmazza a magyar nyelvi tudást, ezért kutatásunkban megmértük a teljesítményét.

## 2.5. Helsinki Marian NMT

A HNMT (Helsinki Neural Machine Translation) (Tiedemann és Thottingal, 2020) egy Marian NMT (Junczys-Dowmunt és mtsai, 2018) (small) rendszer, amely jelenlegi legjobban teljesítő fordító eljárás angolról finnre, ezen nyelvpár esetén a legmagasabb BLEU értékeket képes elérni. A HNMT mögött álló kutatók tesztelték a fordító rendszert angol-lett, angol-kínai és kínai-angol irányokban is, azonban ezeknél a nyelvpároknál szerényebb eredményeket értek el. A tanított gépi fordító rendszer kifejezetten jól teljesít morfológiailag gazdag nyelvekre, például a finn nyelvre is.

A Helsinki Egyetem munkatársai céljai között volt, hogy lehetőleg a legtöbb nyelvre készítsenek gépi fordító modelleket. Angol-magyar nyelvpárra több small modelljük is létezik. Ezeket teszteltük kutatásunkban.

<sup>6</sup> [https://github.com/pytorch/fairseq/tree/main/examples/m2m\\_100](https://github.com/pytorch/fairseq/tree/main/examples/m2m_100)

## 2.6. DeepL Fordító

A DeepL Fordító<sup>7</sup> egy ingyenes elérhető internetes fordító rendszer (DeepL GmbH, Köln, Németország). A fordító rendszer mögött álló vállalatot 2009-ben alapították Linguee néven, és az azonos névvel ellátott fordításokra specializálódott keresőmotort dobták piacra (deepl.com). A DeepL Fordító konvolúciós neurális hálókat (Kim, 2014) használ, és architektúrájának köszönhetően sokszor természetesebben hangzó fordításokat tud produkálni a piacon elérhető versenytársak megoldásaihoz képest. A 2018-ban elindított DeepL Pro egy tovább optimalizált verzióként lehetővé teszi, hogy a vállalat által fejlesztett mesterséges intelligencián alapuló megoldások még magasabb minőségű gépi fordításra legyenek képesek. 2021-ben 13 újabb európai nyelvvel, köztük a magyarral bővült a DeepL repertoárja.

## 2.7. Google Fordító

A Google Fordító<sup>8</sup> (Wu és mtsai, 2016) 2003-ban indult, akkor még statisztikai gépi fordítás elvén alapulva, majd 2016-ban ezt felváltotta a neurális hálón alapú gépi fordítás. A neurális hálón alapuló megközelítés bevezetése lényegesen javított a fordítási minőségen, mivel szélesebb kontextus alapján következteti ki a jobban illeszkedő és ezáltal hitelesebb lefordított verziót<sup>9</sup>. A Google Fordító többféle lefordított verziót is listáz, így például az angolban nem specifikált, de franciában vagy spanyolban nőnemű és hímnemű megkülönböztetéssel létező szavak esetén először a nőnemű, majd a hímnemű verziót mutatja (Rescigno és mtsai, 2020). A Google Fordító 109 különböző nyelvet képes kezelni<sup>10</sup>. 2020-tól kezdve a szóban elhangzott szövegek lefordítása is lehetséges<sup>11</sup>.

## 2.8. Yandex Fordító

A Yandex<sup>12</sup> egy orosz technológiai vállalat, amely gépi fordításon alapuló szolgáltatásokat értékesít a digitális termékek piacán. A Yandex által kifejlesztett fordító<sup>13</sup> két önálló gépfordító rendszeren alapul<sup>14</sup>. Az egyik egy statisztikai gépfordító, amely több százezer ugyanazon információkat tartalmazó, de különböző nyelveken íródott szövegek statisztikai összehasonlítása révén tanul. A Yandex statisztikai fordító mögött egy három komponensű gépfordító rendszer áll, melyek a fordító modell, a nyelvi modell és a dekóder. Magát a tényleges

<sup>7</sup> <https://www.deepl.com/translator>

<sup>8</sup> <https://translate.google.com>

<sup>9</sup> <https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate>

<sup>10</sup> <https://translate.google.com/intl/en/about/languages>

<sup>11</sup> <https://blog.google/products/translate/transcribe-speech>

<sup>12</sup> <https://yandex.com>

<sup>13</sup> <https://translate.yandex.com>

<sup>14</sup> <https://yandex.com/dev/translate/doc/dg/concepts/how-works-machine-translation.html>



fordítási folyamatot a dekóder végzi. Ennek során a fordító modell által felkínált különböző fordítási verziókat kombinálja össze és sorrendbe állítja az előfordulási gyakoriság alapján, amely a nyelvi modell révén kerül megállapításra. A másik fő komponens egy enkóder-dekóder architektúrájú neurális gépfordító. Tudomásunk szerint RNN architektúrájú (Cho és mtsai, 2014). A rendszer a két fordító rendszer által kiadott fordítást a CatBoost algoritmus (Prokhorenkova és mtsai, 2018) segítségével hasonlítja össze, majd a jobb fordítást adja végső kimenetként.

## 2.9. Bing Fordító - Microsoft fordító

A Bing Fordító a Microsoft Cognitive Services termékcsalád része, amely szövegek fordítását teszi lehetővé több, mint 100 különböző nyelven<sup>15</sup>. 2021-től már teljes dokumentumok lefordítására is használható. Kezdetben statisztikai megközelítést alkalmaztak a fejlesztők, majd 2018-ban átváltottak neurális háló alapú gépi fordításra. A Microsoft részéről is komoly érdeklődés övezi a többnyelvű gépi fordítást és számos kutatást végeznek a hatékonyság és a pontosság növelése érdekében. A többnyelvű és az egynyelvű modellek közötti fordítási pontosság tekintetében mutatkozó különbség leküzdésére Xu Tan és mtsai egy tudás desztilláción (knowledge distillation) (Bucila és mtsai, 2006) alapuló módszert dolgoztak ki (Tan és mtsai, 2019). A tudás desztilláció eredetileg modellek karcsúsítására szolgáló módszer, amelynek keretében egy ún. diák modell kerül kialakításra, amely képes a tanár modell vagy több modell együttesének teljesítményét (adott esetben pontosságát) hozni. Ennek mintájára, egy-egy nyelvpárra szakosodott ún. tanár modellek tanítják be a diák modellt, amely az összes nyelvpárt egyetlen modellben kezeli. Két különböző eljárást fejlesztettek ki, az egyik a szelektív desztilláció, melynek során a desztilláció alkalmazását teljesítményalapú küszöbértékhez kötik, a másik pedig a Top-K desztilláció, amely a tanár modellek valószínűségi eloszlását vizsgálja és csak a legjobb együtthatóval rendelkező modelleket tölti be a memóriába. A módszer eredményességét tanúsítja, hogy a TED előadások átiratát tartalmazó szövegbankon tesztelve 44 nyelvről angolra fordítva, az összes nyelvre nézve 1 körüli vagy annál magasabb BLEU érték javulást hozott a kifejlesztett módszer alkalmazása.

## 2.10. eTranslation

Az eTranslation<sup>16</sup> egy automatizált fordítóeszköz, amellyel szövegrészeteket vagy akár teljes dokumentumokat lehet lefordítani az Európai Unió tagállamaiban használatos hivatalos nyelvekre, valamint izlandi, norvég, orosz, illetve egyszerűsített kínai nyelvre is. Az Európai Bizottság által rendelkezésre bocsátott gépfordító eljárás segítséget kíván nyújtani az Európai Unió kis- és középvállalkozásainak, közszolgáltatóinak, hivatalnokainak a gördülékeny kommunikáció és

<sup>15</sup> <https://www.microsoft.com/en-us/translator/blog/2021/10/11/translator-now-translates-more-than-100-languages>

<sup>16</sup> <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation>

ügyintézés elősegítése érdekében. Az eTranslation könnyen integrálható más digitális rendszerekbe, amennyiben fordítási kapacításra van szükség. A gépi fordítás folyamatának megkönnyítése érdekében számos előfeldolgozási és szövegtisztítási lépés is elérhetővé vált a CEF eTranslation Building Block projekt keretében. Példaként említhető, hogy hosszú mondatok könnyebb kezelhetősége érdekében ezek a fordítás előtt kisebb részekre vannak osztva, majd a fordítást követően koherens szöveggé kerülnek összeállításra. Az eTranslation rendszer nem általános, hanem specifikusabb szövegeken lett tanítva, mint például pályázati dokumentumok, jogi és orvosi szövegek stb. A modell több mint 1 milliárd mondaton tanult 24 nyelvre.

### 3. Mérések

#### 3.1. Felhasznált korpusz

A gépi fordításhoz magunk építettünk egy angol-magyar párhuzamos korpuszt. A korpusz építéséhez az OPUS (Tiedemann, 2012) korpuszból vettünk angol-magyar (en-hu) párhuzamos alkorpuszokat. Felhasznált alkorpuszok: ParaCrawl, OpenSubtitles, Tatoeba, WikiMatrix, EUbookshop, PHP manual, TED2020, KDEdoc, KDE4. A korpusz méretei a 1. táblázatban láthatóak (nem tokenizált szövegen):

	Szegmens	Token		Type		Token átlag / mondat	
		en	hu	en	hu	en	hu
OpenSubtitles	42 655 519	272 571 665	209 481 645	2 382 239	6 519 406	6,39	4,91
ParaCrawl	12 681 746	196 278 983	172 671 171	3 555 484	5 713 776	15,48	13,62
WikiMatrix	488 319	8 978 943	7 673 323	627 814	1 057 487	18,38	15,71
TED2020	308 341	5 194 871	3 982 056	158 210	495 452	16,85	12,91
EUbookshop	438 264	9 406 548	7 847 111	360 311	648 778	21,46	17,90
KDE4	12 0657	622 959	649 457	62 257	98 940	5,16	5,38
Tatoeba	109 041	639 834	505 838	30 759	84 570	5,86	4,64
PHP	35 423	169 610	157 583	17 215	25 854	4,79	4,45
KDEdoc	861	10 904	9 474	2 402	2 987	12,66	11,00
Összesen	56 838 171	493 874 317	402 977 658	5 873 336	11 770 992	8,69	7,09

1. táblázat. Felhasznált részkorpuszok méretei.

#### 3.2. Saját tanított gépi fordító modellek

Kutatásunk során kettő Marian, egy BART és egy mT5 gépi fordító modellt tanítottunk angol-magyar nyelvre.

A **BART** kísérlet esetében elsőként előtanítottunk egy angol-magyar kétnyelvű BART base modellt, ami elérhető a Hugging Face oldalon<sup>17</sup>. Az előtanításhoz az angol WikiText-103 (Merity és mtsai, 2017) és a Webcorpus 2.0 (Nemeskey, 2020) magyar Wikipédia részét használtuk. Az eredeti BART kutatáshoz hasonlóan csak azokat a bekezdéseket hagytuk meg, amelyek legalább egy darab pont írásjellel rendelkeztek. A felhasznált korpuszok méretei az 2. táblázatban láthatók (tokenizált szövegen).

	Angol WikiText-103	Magyar Wikipédia
Szegmensek száma	707.391	1.098.156
Tokenek száma	96.534.563	90.349.849
Type	596.820	3.137.980
Átlagos mondatszám / bekezdés	5	4
Átlagos tokenszám / bekezdés	125	69

2. táblázat. BART előtanításhoz használt korpuszok méretei.

Az angol-magyar BART-base modellünk előtanításához 4 darab GeForce GTX 1080 Ti (12GB) videokártyát használtunk, az alábbi paraméterekkel: batch méret / GPU: 12; szótár méret: 40.000; tanulási ráta: 2e-8; tanítási lépésszám: 170.000. Az előtanításhoz a Hugging face Transformers könyvtárban található Seq2SeqTrainer<sup>18</sup> és BartForCausalLM<sup>19</sup> függvényeket használtuk.

Az így előtanított BART modellünket tovább finomhangoltuk angol-magyar gépi fordítás feladatára. A finomhangoláshoz 4 darab GeForce GTX 1080 (12GB) videokártyát használtunk az alábbi paraméterekkel: batch méret / GPU: 26; maximum szöveghossz (bemeneti és kimeneti): 128; warmup: 15.000; fp16; epoch: 10; tanulási ráta: 5e-5. A finomhangoláshoz a Hugging Face Transformers könyvtárban található példakódot<sup>20</sup> használtuk fel.

A **Marian NMT** esetén a keretrendszer által alapértelmezetten biztosított paraméter beállításokat alkalmaztuk<sup>21</sup>. Az első esetben egy small modellnek megfelelő beállítást választottunk (Marian small), másodjára pedig egy kétszer több paraméterrel rendelkezőt (Marian big), amely elérhető a Hugging Face oldalon<sup>22</sup>. Subword tokenizáláshoz a beépített Sentence Piece (Kudo és Richardson, 2018) tokenizálót használtuk. A szótér mérete: 32.000.

Az **mT5** kutatásunkban az előtanított mT5 small<sup>23</sup> modellt finomhangoltuk angol-magyar nyelvre, amely elérhető a Hugging Face oldalon<sup>24</sup>. A tanításhoz 4

<sup>17</sup> <https://huggingface.co/NYTK/translation-bart-128-en-hu>

<sup>18</sup> [https://huggingface.co/transformers/main\\_classes/trainer.html#seq2seqtrainer](https://huggingface.co/transformers/main_classes/trainer.html#seq2seqtrainer)

<sup>19</sup> [https://huggingface.co/transformers/model\\_doc/bart.html#bartforcausallm](https://huggingface.co/transformers/model_doc/bart.html#bartforcausallm)

<sup>20</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/translation>

<sup>21</sup> <https://github.com/marian-nmt/marian-dev/blob/master/src/common/aliases.cpp>

<sup>22</sup> <https://huggingface.co/NYTK/translation-marianmt-en-hu>

<sup>23</sup> <https://huggingface.co/google/mt5-small>

<sup>24</sup> <https://huggingface.co/NYTK/translation-mt5-small-128-en-hu>

darab GeForce GTX 1080 (12GB) videokártyát használtunk az alábbi paraméterekkel: batch méret: 6; prefix: „translate English to Hungarian: ”; maximum szöveghossz (bemeneti és kimeneti): 128; epoch: 1; tanulási ráta: 5e-5. Sajnos az epoch szám csak 1, amellyel közel egy hónap volt a futási idő. A finomhangoláshoz ugyanazt a könyvtárat használtuk, mint a BART finomhangolásnál.

### 3.3. Kipróbált gépfordítók és modellek

Kutatásunkhoz kipróbáltunk különböző kutatásra és ipari alkalmazásra szánt gépfordító rendszereket és modelleket, amelyek képesek angolról magyarra fordítani. Kísérleteinkben az alábbi rendszereket, modelleket próbáltuk ki:

- **M2M100**: Két modelljük is elérhető, egy kisebb (418M) és egy nagyobb (1.2B) modell. A fordításhoz a Hugging Face Transformers könyvtár M2M100Tokenizer és M2M100ForConditionalGeneration függvényeit használtuk.
- **Helsinki Marian NMT**: több modelljük is tartalmaz angol-magyar tudást: angol-magyar (en-hu); angol- finnugor (en-fiu); angol-uráli (en-urj); angol-multi (310 nyelv) (en-multi). A fordításhoz a Hugging Face Transformers könyvtárMarianTokenizer és MarianMTModel függvényeit használtuk.
- **eTranslation**: Akadémia számára ingyenesen elérhető szolgáltatás. Regisztráció után beküldtük a teszt fájlunkat, amelynek a fordítását e-mailben kaptuk meg.
- **deepL**: Az online fájl-fordító funkcióval fordítottunk, 500 mondatonként.
- **Google**: Az online dokumentum-fordító funkcióval fordítottunk, 500 mondatonként.
- **Microsoft**: Azure Translator szolgáltatás<sup>25</sup> felhőalapú dokumentum-fordító moduljával fordítottunk.
- **Yandex**: Az online dokumentum-fordító funkcióval fordítottunk, 500 mondatonként.

## 4. Eredmények

A különböző modellek, rendszerek kiértékelésére a SacreBLEU (Papineni és mt-sai, 2002; Post, 2018) és a chrF (Popović, 2015) metrikákat használtuk. A BLEU metrika mellett azért választottuk a chrF metrikát, mert az karakteralapú, ami a ragozó nyelvek esetében, mint magyar nyelv, pontosabb kiértékelést eredményez. A chrF kiértékelésnél a 6-gram mellett a 3-gram értékeket is megmértük.

A 4. táblázatban láthatók a különböző gépfordítók eredményei. Az ipari alkalmazások közül az eTranslation és a deepL teljesítettek a legjobban és a közöttük lévő különbség statisztikailag nem szignifikáns mértékű. A második minőségi kategóriába a nagy cégek rendszerei (Google, Microsoft) kerültek, míg a Yandex rendszere nagymértékben elmarad ezektől.

<sup>25</sup> <https://docs.microsoft.com/hu-hu/azure/cognitive-services/translator/document-translation/overview>

forrás	- Oh, no. If you think you're tucking me away somewhere, you've got another think coming.
referencia	Ha azt tervezi, hogy bedug valahová, akkor terveljen ki valami mást.
google	- Óh ne. Ha azt hiszed, hogy elrejtess valahova, akkor más gondolat jön.
M2M100	Ha azt hiszed, hogy valahol elrejtess engem, van egy másik gondolkodásod.
mT5	Ha azt hiszed, hogy elrángatsz valahol, akkor jön egy másik gondolat.
BART	Ha azt hiszed, hogy el akarsz dugni valahova, akkor másra is gondolhatsz.
Marian big	Ha azt hiszed, hogy eldughatsz valahova, akkor tévedsz.

3. táblázat. Példamondat a fordító rendszerek összehasonlítására.

	BLEU	chrF-3	chrF-6	Tanítási idő
Marian big	<b>37,30</b>	61,61	56,80	21 nap
BART	<b>36,89</b>	60,77	56,48	43,5 nap
mT5	27,69	53,73	48,57	26 nap
Marian small	26,99	51,31	46,07	9 óra
Helsinki en-hu	27,21	55,03	49,82	-
Helsinki en-fiu	24,23	52,68	47,16	-
Helsinki en-urj	24,16	52,56	47,09	-
Helsinki en-multi	14,39	43,69	36,74	-
M2M100 - 1.2B	21,62	50,93	45,73	-
M2M100 - 418M	18,75	48,40	42,72	-
eTranslation	28,29	56,00	51,27	-
deepL	26,54	56,06	51,01	-
Google	25,30	54,09	49,06	-
Microsoft	25,22	53,02	48,00	-
Yandex	19,22	49,78	43,94	-

4. táblázat. A gépfordító rendszerek eredményei.

A kutatásra szánt modellek közül egyértelműen a Helsinki en-hu modell teljesített a legjobban, ami nem meglepő, hiszen a tanítóanyag szempontjából átfedés volt a használt tanítóanyaggal, továbbá kétnyelvű és nem több nyelvű modell. Az M2M100 nagyobbik modellje, annak ellenére, hogy 100 nyelvet tud, versenyképes eredményt ért el angol-magyar nyelvpárra.

Az általunk tanított modellek közül a Marian big és a BART modellek kerültek a legjobb minőségi osztályba szinte azonos eredménnyel. Az elvártaknak megfelelően a Marian big modellje ért el legjobb eredményt köszönhetően annak, hogy a legtöbb paraméterrel rendelkező hálózattal dolgozik. Másfelől az implementációjának köszönhetően gyorsabban érte el ezt az eredményt, mint vetélytársai. Mindezek ellenére érdemes megemlíteni a BART modellt is mivel egy base modellként sikerült összemérhető teljesítményt nyújtania a sokkal nagyobb hálózattal rendelkező társával szemben.

Az mT5 modellünk erőforrás hiányában csak 1 epochon finomhangoltuk, ezért nem sikerült olyan magas eredményt elérni vele. Azonban így is versenyképes teljesítményt nyújtott, megelőzve a legtöbb ipari és kutatási modellt, annak el-

lenére, hogy az mT5 vegyes feladatokra tanították elő és csak 1-szer látta a teljes anyagunkat.

A 3. és az 5. táblázatokban 1-1 példamondat olvasható, ahol az érdekesebb rendszerek fordításait emeltük ki. A fordításokat vizsgálva látható, hogy mindegyik rendszer kimenetén viszonylag olvasható szövegek voltak, továbbá a közöttük lévő eltérések nagyrészt nyelvtani szerkezetbeli különbségek. Ez a jelenség a példamondatokban is megmutatkozik: a hibás fordítások fő oka a ragozásból fakadó tartalmi eltérés. A bemutatott példákon megfigyelhető, hogy a BART és a Marian big modellek fordításai adják vissza legjobban a forrásmondatok mondanivalóját annak ellenére, hogy ez nem felel meg karakter szinten a referencia mondatnak.

forrás	This may not make much sense to you, sir, but I'd like to ask your permission to date your daughter.
referencia	Szeretném megragadni az alkalmat uram, hogy az engedélyét kérjem, hogy találkozhatnék a lányával.
google	Lehet, hogy ennek nem sok értelme van, uram, de szeretném engedélyét kérni a lányával való randevúzáshoz.
M2M100	Lehet, hogy ez nem sok értelme, uram, de szeretném kérni az engedélyét, hogy dátumot a lánnyal.
mT5	Talán nem sok értelme van, uram, de szeretném kérni az engedélyét, hogy randizzon a lányával.
BART	Lehet, hogy önnek nincs sok értelme, uram, de szeretném az engedélyét kérni, hogy randizhatnék a lányával.
Marian big	Ennek talán nincs sok értelme, uram, de szeretném az engedélyét kérni, hogy randizhatnék a lányával.

5. táblázat. 2. példamondat a fordító rendszerek összehasonlítására.

## 5. Összegzés

Kutatásunkban különböző neurális gépfordító modelleket és rendszereket tanítottunk be és próbáltunk ki angol-magyar nyelvpárra. Egyaránt kísérleteztünk kutatásban használt és iparban alkalmazott gépfordító módszerekkel, rendszerekkel is. Kísérleteink során a meglévő modellek mellett saját gépfordító rendszereket is tanítottunk. Betanítottunk kettő Marian NMT rendszert, egy kicsi és egy nagy modellt. Továbbá betanítottunk egy saját BART modellt, amelyet ezután finomhangoltuk gépi fordításra. Végül egy előtanított mt5 modellt finomhangoltunk angol-magyar gépi fordításra. Eredményeinkben megmutattuk, hogy az általunk tanított nagy Marian NMT modell és a BART modell szignifikánsan magasabb eredményt értek el az összes többi modellhez képest. Kettőjük versenyében a BART minimális értékkel marad csak le a Marian Big modelltől, ami érdekes eredmény, mivel a BART kevesebb paraméterekkel volt képes versenyképes eredményt elérni.

## Köszönetnyilvánítás

A publikációban szereplő kutatást, amelyet a Pázmány Péter Katolikus Egyetem és az IdomSoft Zrt. valósított meg, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta az Infokommunikációs és Információtechnológiai Nemzeti Laboratórium keretében.

## Hivatkozások

- Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3874–3884. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Artetxe, M., Schwenk, H.: Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Transactions of the Association for Computational Linguistics 7, 597–610 (Mar 2019)
- Barrault, L., Bojar, O., Costa-jussà, M.R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., MÁzller, M., Pal, S., Post, M., Zampieri, M.: Findings of the 2019 conference on machine translation (wmt19). In: Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1). pp. 1–61. Association for Computational Linguistics, Florence, Italy (August 2019)
- Bucila, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 535–541. KDD '06, Association for Computing Machinery, New York, NY, USA (2006)
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar (Oct 2014), <https://aclanthology.org/D14-1179>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

- Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., Johnson, M.: Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In: ICML. pp. 4411–4421 (2020)
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Hermann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018)
- Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014)
- Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Li, L., Jiang, X., Liu, Q.: Pretrained language models for document-level neural machine translation (2019)
- Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., Zettlemoyer, L.: Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics* 8, 726–742 (11 2020)
- Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. In: 5th International Conference on Learning Representations. Palais des Congrès Neptune, Toulon, France (2017)
- Miculicich, L., Ram, D., Pappas, N., Henderson, J.: Document-level neural machine translation with hierarchical attention networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)
- Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002), <https://aclanthology.org/P02-1040>
- Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation. pp.



- 392–395. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), <https://aclanthology.org/W15-3049>
- Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers. pp. 186–191. Association for Computational Linguistics, Brussels, Belgium (Oct 2018)
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: Unbiased boosting with categorical features. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 6639–6649. NIPS’18, Curran Associates Inc., Red Hook, NY, USA (2018)
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020)
- Rescigno, A.A., Monti, J., Way, A., Vanmassenhove, E.: A case study of natural gender phenomena in translation: A comparison of Google Translate, Bing Microsoft translator and DeepL for English to Italian, French and Spanish. In: Workshop on the Impact of Machine Translation (iMpacT 2020). pp. 62–90. Association for Machine Translation in the Americas, Virtual (Oct 2020)
- Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany (Aug 2016)
- Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., Liu, T.Y.: Multilingual neural machine translation with knowledge distillation. In: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019). New Orleans, LA, USA (2019)
- Tiedemann, J., Thottingal, S.: OPUS-MT — Building open translation services for the World. In: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT). Lisbon, Portugal (2020)
- Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12). European Language Resources Association (ELRA), Istanbul, Turkey (May 2012)
- Wenzek, G., Lachaux, M.A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., Grave, E.: CCNet: Extracting high quality monolingual datasets from web crawl data. In: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France (May 2020)
- Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google’s neural machine

translation system: Bridging the gap between human and machine translation. CoRR abs/1609.08144 (2016)

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021), <https://aclanthology.org/2021.naacl-main.41>

# KORPUSZ, SZÖVEGFELDOLGOZÁS



## ELTE Verskorpusz – a magyar kanonikus költészet gépileg annotált adatbázisa

Horváth Péter<sup>1</sup>, Kundráth Péter, Indig Balázs<sup>1</sup>, Fellegi Zsófia<sup>2</sup>, Szlávich Eszter<sup>1</sup>, Bajzát Tímea Borbála<sup>12</sup>, Sárközi-Lindner Zsófia<sup>1</sup>, Vida Bence<sup>1</sup>, Karabulut Aslihan<sup>1</sup>, Timári Mária<sup>1</sup>, Palkó Gábor<sup>12</sup>

<sup>1</sup> Eötvös Loránd Tudományegyetem, Bölcsészettudományi Kar  
1088 Budapest, Múzeum krt. 4., 4–6.  
{horvath.peter, indig.balazs, szlavich.eszter,  
lindner.zsafia, vida.bence, karabulut.aslihan, timari.maria,  
palko.gabor}@btk.elte.hu

<sup>2</sup> Bölcsészettudományi Kutatóközpont, Irodalomtudományi Intézet  
1118 Budapest, Ménési út 11–13.  
{fellegi.zsafia, bajzat.timea}@abtk.hu  
peter.kundrath@gmail.com

**Kivonat:** Az ELTE Verskorpusz a magyar kanonikus költészet vizsgálatára létrejött, a versek szerkezeti egységeinek, a szavak bizonyos grammatikai és fonológiai jellemzőinek, valamint a vershangzás bizonyos jellemzőinek az automatikusan létrehozott annotációit tartalmazó, online elérhető lekérdezőfelülettel rendelkező adatbázis. A tanulmányban bemutatjuk az ELTE Verskorpusz fontosabb mennyiségi jellemzőit, a verskorpusz létrehozásának főbb lépéseit, valamint az egyes annotálási lépésekhez használt eszközöket. Részletesebben ismertetjük a vershangzás annotálásának módját, valamint a verskorpusz formátumát. Emellett bemutatjuk a korpuszhoz készült lekérdezőfelület főbb funkcióit is.

### 1. Bevezetés és kapcsolódó munkák

Az ELTE Verskorpusz<sup>1</sup> egy olyan online lekérdezőfelülettel<sup>2</sup> rendelkező adatbázis, amely különböző típusú annotációkkal ellátva tartalmazza a kanonikusnak tekintett magyar költészet jelentős részét. Korábbi, magyar nyelvű versek hangzásjellezőit is tartalmazó, automatikusan annotált korpusz létezéséről nincs tudomásunk. A korpusz létrehozása során ugyanakkor támaszkodhattunk az olyan általánosabb célú, magyar nyelvű szövegeket tartalmazó korpuszokra, mint a Magyar Nemzeti Szövegtár<sup>3</sup> (Oravecz és mtsai, 2014) vagy a Magyar Történelmi Szövegtár<sup>4</sup> (Csengery, 2006; Sass, 2017). Bár nem tartalmaz annotált szövegeket, előzményként megemlítendő *A régi magyar vers leltára a kezdetektől 1600-ig/Répertoire de la poésie hongroise ancienne* című

---

<sup>1</sup> <https://github.com/ELTE-DH/poetry-corpus>

<sup>2</sup> <https://verskorpusz.elte-dh.hu>

<sup>3</sup> <http://clara.nytud.hu/mnsz2-dev>

<sup>4</sup> <http://clara.nytud.hu/mtsz>

adatbázis is, amely magyar versek különböző adatait, többek között a metrikai jellemzőit tartalmazza kereshető formában.<sup>5</sup> A verskorpusz létrehozása során fontos kiindulópont volt számunkra a Cseh Tudományos Akadémia által fejlesztett Cseh verskorpusz (Korpus českého verše)<sup>6</sup>, amely közel 80 000 annotált verset tartalmaz a 19. századból és a 20. század elejéről. A korpusz a szavak lemmája, szófaja, morfológiai és fonológiai jellemzői mellett a ritmusra és a rímekre vonatkozó automatikusan létrehozott annotációkat is tartalmaz (Plecháč és Kolár, 2015; Ibrahim és Plecháč, 2011). Ugyancsak támaszkodtunk az ELTE Verskorpuszhoz hasonlóan TEI XML formátumban készült, 16. és 17. századi spanyol szonettek tartalmazó Corpus de Sonetos del Siglo de Oro (Corpus of Spanish Golden-Age Sonnets)<sup>7</sup> nevű verskorpusz megoldásaira, amelyben ritmusra vonatkozó, automatikusan létrehozott annotációk is szerepelnek (Navarro-Colorado, 2015; Navarro-Colorado és mtsai, 2016).<sup>8</sup>

Az ELTE Verskorpusz jelenleg 49 költő összes versét tartalmazza.<sup>9</sup> Ez összesen 13063 verset jelent, amelyek együttes szószáma durván 2,7 millió. A korpusz fő formátuma TEI XML, amely a különféle digitális bölcsészeti és nyelvészeti adatbázisok egyik legelterjedtebb formátuma. A versek annotációit gépileg hoztuk létre. A verseket három szinten annotáltuk. Egyrészt annotáltuk a versek szerkezeti egységeit: a címeiket, a versszakokat és a verssorokat. Emellett tokenizáltuk a versszövegeket, és annotáltuk a szavak grammatikai jellemzőit, azaz a szavak lemmáját, szófaját és morfoszintaktikai tulajdonságait. Végezetül a versek vershangzáshoz kapcsolódó, formailag egyszerűen megragadható jellemzőit is címkéztük, ami a rímképletek, a rímpárok, a ritmus, az alliterációk és a szavak fontosabb fonológiai jellemzőinek az annotálását jelentette.

A 2. részben bemutatjuk a korpusz létrehozásának lépéseit és az egyes lépésekhez használt eszközöket. A 3. részben ismertetjük a vershangzás automatikus annotálásának a módját, a 4. részben pedig a korpusz formátumát. Végezetül az 5. részben röviden ismertetjük a korpuszhoz készített, online elérhető lekérdezőfelület főbb funkcióit.

## 2. A korpusz létrehozásának lépései

A korpusz jelenlegi tartalmának forrását a Magyar Elektronikus Könyvtár<sup>10</sup> adatbázisában megtalálható, a public domain körébe tartozó szerzők összes versét tartalmazó dokumentumfájlok adták. Ezek a dokumentumok jellemzően többféle formátumban is megtalálhatók a MEK oldalán. Elsődlegesen az RTF-formátumú fájlokat használtuk, ha ez nem állt rendelkezésre, akkor a HTML-formátumút. Első lépésben a MEK oldaláról leszedett fájlokból egy szkripttel létrehoztuk a versek szerkezeti egységeinek az annotációit tartalmazó TEI XML fájlokat. Az RTF-formátumú dokumentumok esetében ehhez egy XQuery szkriptet, a HTML-formátumú szövegek esetében pedig egy

<sup>5</sup> <https://f-book.com/rpha>.

<sup>6</sup> [https://versologie.cz/v2/web\\_content/corpus.php?lang=en](https://versologie.cz/v2/web_content/corpus.php?lang=en); <https://github.com/versotym/corpusCzechVerse>

<sup>7</sup> <https://github.com/bncolorado/CorpusSonetosSigloDeOro>

<sup>8</sup> További verskorpuszokról és azok típusairól lásd Dodé et al. 2018.

<sup>9</sup> A korpusz Radnóti kivételével tartalmazza többek között a középiskolai tananyagban szereplő, hetvenn

<sup>10</sup> <http://mek.oszk.hu>

Python szkriptet használtunk. A szkriptek futtatásának eredményeképpen mindegyik vers belekerült egy külön TEI XML fájlba, amely XML elemekként tartalmazza a címekre, versszakokra és verssorokra vonatkozó annotációkat. Ugyancsak ebben a fázisban hoztuk létre automatikusan a TEI XML fájlok <teiHeader> elemeit, amely tartalmazza a fontosabb metaadatokat, például a vers szerzőjét és címét, valamint az eredeti MEK-forrás fontosabb adatait.

Ezt követően a szerkezeti egységek annotációit tartalmazó TEI XML fájlokat kézzel ellenőriztük, azaz összevetettük őket a kiinduló RTF- vagy HTML-formátumú dokumentumokkal. Erre azért volt szükség, mert a szkriptek bizonyos speciális eseteket nem tudtak helyesen annotálni, illetve a kiinduló dokumentumfájlokban is voltak olyan inkonzisztens megoldások, amelyek annotációs hibához vezettek. A kézi ellenőrzéshez az Oxygen XML Editor programot<sup>11</sup> használtuk.

Ezt követően annotáltuk, úgyszintén automatikusan, a vers szavainak a grammatikai jellemzőit, azaz a lemmát, a szófajt és a morfoszintaktikai tulajdonságokat. Ehhez az e-magyar elemzőlánc emtsv változatát használtuk (Váradai és mtsai, 2017; Indig és mtsai, 2019; Simon és mtsai, 2020), amelyet egy saját Python szkriptbe ágyazva futtattunk. Erre azért volt szükség, mert az e-magyar bemenete TXT, a kimenete pedig TSV, a verskorpusz formátuma azonban TEI XML. A szkript kiszedi a TEI XML-ből a szöveget, a szövegen lefuttatja az e-magyart, majd az e-magyar TSV-formátumú elemzését visszaalakítja TEI XML-lé. A szófaji és morfoszintaktikai jellemzőket az e-magyar Universal Dependenciesnek (UDv1)<sup>12</sup> megfelelő kimenetével annotáltuk, amely a grammatikai jellemzők annotálásának a legelterjedtebb címkézési rendszere (Vincze és mtsai 2017).

Az annotációs folyamat következő lépése a vershangzáshoz kapcsolódó, formailag egyszerűbben megragadható jellemzőknek, a rímnek, a rímpárokknak, a sorok ritmusának, az alliterációknak és a szavak fontosabb fonológiai jellemzőinek az annotálása volt. Ehhez egy saját fejlesztésű, kifejezetten a projekt számára készített, Python nyelvben írt programot használtunk.

A korpusz utolsó annotálási lépéseként egy XSLT stíluslap segítségével elvégeztünk néhány átalakítást az XML fájlokban szereplő annotációk pozícióján, átneveztünk bizonyos XML elemeket és attribútumokat, illetve további, a versek, versszakok és sorok szó- és szótagszáma vonatkozó információkkal bővítettük a már meglévő annotációkat.<sup>13</sup> Az így előálló XML fájlok bár TEI-közeli, de nem felelnek meg a TEI specifikációjának. Erre az utolsó annotálási fázisra azért volt szükség, mert a TEI által specifikált formátum a részletesebb annotációk tárolására kevésbé alkalmas. E lépés eredményeképpen előállt egy olyan formátuma is a verskorpusznak, amelyben az annotációkat tartalmazó elemek és attribútumok a lehető legegyszerűbb módon utalnak a tartalmukra, illetve amely esetében az annotációk logikusabb elrendezése, valamint a bevezetett további annotációk miatt egyszerűbben meg lehet írni a keresőkifejezéseket, és azokat gyorsabban le lehet futtatni.

Az alábbi felsorolás mutatja be a korpusz létrehozásának a leírt lépéseit.

<sup>11</sup> <https://www.oxygenxml.com>

<sup>12</sup> [https://universaldependencies.org/treebanks/hu\\_szeged/index.html](https://universaldependencies.org/treebanks/hu_szeged/index.html)

<sup>13</sup> A bevezetett, illetve átnevezett elemekről és attribútumokról lásd a <https://github.com/ELTE-DH/poetry-corpus> oldalon szereplő leírást.

level0: szerkezeti egységek annotálása

Bemenet: RTF, HTML

Kimenet: TEI XML

Eszköz: XQuery szkript, Python szkript

level1: szerkezeti egységeket tartalmazó TEI XML-ek kézi ellenőrzése

Kimenet: TEI XML

Eszköz: manuális, Oxygen XML Editor használatával

level2: tokenizálás, lemmatizálás, szófaji és morfológiai annotálás

Kimenet: TEI XML

Eszköz: Python szkriptbe ágyazott e-magyar

level3: a vershangzás jellemzőinek annotálása

Kimenet: TEI XML

Eszköz: a projekthez fejlesztett Python program (hunpoem\_analyzer-TEI)

level4: formátumátalakítás és az annotációk bővítése

Kimenet: XML

Eszköz: XSLT stíluslap

A korpusz annotálására kialakított folyamat lehetővé teszi, hogy a szerkezeti annotációkat tartalmazó, kézzel ellenőrzött TEI XML fájlokon bármikor újrafuttassuk a további annotációs lépéseket, amennyiben azok valamelyikén valamilyen fejlesztés vagy hibajavítás történt.

### 3. A vershangzás gépi annotálása

Angol nyelvű versek hangzástulajdonságainak a gépi felismertetésére számos programot írtak. Ilyen például a Scandroid (Hartman, 2005) és a ZuScansion (Agirrezabal és mtsai, 2016) nevű eszköz, amelyek angol nyelvű versek ritmusát és metrumát ismerik fel, vagy az AnalysePoem (Plamondon, 2006) nevű eszköz, amely angol nyelvű versek ritmusának, metrumának és rímképletének a felismerésére képes. Több, angol nyelvű versek hangzástulajdonságainak az automatikus elemzésére épülő kutatás is született az utóbbi években. Kao és Jurafsky (2012) kutatása például professzionális és amatőr amerikai versek különbségeit vizsgálta, amelynek során a szókincs mellett olyan vershangzáshoz kapcsolódó, automatikusan elemzett jellemzőket is felhasználtak, mint a versekben szereplő alliterációk vagy rímpárok. De megemlíthető Tanasescu és mtsai (2016) kutatása is, amely angol nyelvű versek ritmus és rím alapján történő automatikus osztályozására irányult.

Magyar nyelvű versek hangzástulajdonságainak az automatizált, valamilyen számítógépes programmal végzett elemzésére csak kevés példát találhatunk, ugyanakkor ezek között meglepően koraiak is vannak. Voigt Vilmos 1972-es tanulmánya mutatja



be az első kísérletet magyar nyelvű versek számítógépes ritmuselemzésére. A létrehozott programmal három Szabó Lőrinc-szonettnek ismertették fel a megvalósuló időmértékes ritmusát (Voigt, 1972). Saját korát megelőzte Jékel és Papp (1974) könyve, amely Ady összes versének az algoritmikus úton előállított fonémastatisztikai adatait tartalmazza. Ugyancsak a korai példák között tartható számon Jékel és Szuromi (1980) műve, amely Petőfi 300 verse esetében tartalmazza a szótagok részben gépi úton meghatározott, különböző típusú nyomatékértékeit, valamint a nyomatékértékek automatikusan előállított összegzését és különféle statisztikáit. Szorosan kapcsolódik a tanulmány témájához Lesi (2006, 2008) kutatása is, aki tudomásunk szerint elsőként hozott létre olyan többfunkciós programot, amely magyar nyelvű versek rímképletének, alliterációinak és metrumának a gépi elemzésére is alkalmas. Érdeemes utalni Labádi (2018) Berzsényi verseiről írt tanulmányára is, amelyben automatizált módszerekkel végzett, a versek szóképzetere és fonémajellemzőire (szóhosszúság, magánhangzók és mássalhangzók eloszlása) vonatkozó vizsgálatok eredményei szerepelnek.

Az ELTE Verskorpuszban szereplő versek hangzásának gépi annotálásához a *hunpoem\_analyzer-TEI* elnevezésű, Python nyelvben írt, saját fejlesztésű programot használtuk (Horváth, 2020a, 2020b). A programmal a sorok időmértékes ritmusát, a versszakok rímképletét, a rímpárokat, az alliterációkat és a szavak fontosabb fonológiai jellemzőit annotáltuk.

A sorok időmértékes ritmusának, azaz a hosszú és rövid szótagoknak az annotálása néhány egyszerű, a magyar verstanban jól ismert szabály alapján elvégezhető, így nem volt szükséges kiejtésszótárakat beépíteni az algoritmusba. Ezek a szabályok a következők: (1) a program rövid szótagként elemzi azokat a szótagokat, amelyekben rövid magánhangzó van, és közvetlenül a rövid magánhangzó után nem áll mássalhangzó, vagy csak egy rövid mássalhangzó áll; (2) a program hosszú szótagként elemzi azokat a szótagokat, amelyekben hosszú magánhangzó áll, valamint azokat a rövid magánhangzós szótagokat, amelyekben hosszú vagy egynél több mássalhangzó követi a magánhangzót. E szabályok alkalmazása során a program a bevett verstani hagyománynak megfelelően nem veszi figyelembe az esetleges szóhatárokat. Az annotáló programba be lett építve az a verstani szabály is, miszerint a szó eleji mássalhangzó-torlódások (pl. *krákog, trottyos, strigula*) nem nyújtják meg az előtte lévő rövid magánhangzóra végződő szótagot, vagyis azok nem hosszúnak, hanem rövidnek számítanak. Az elemzés kimenete minden sor esetében egy 0 és 1 karakterekből álló karaktorsor, amelyben a 0 a rövid, az 1 pedig a hosszú szótagokat reprezentálja (pl. *Húnyt szemmel bérceken futunk* – 11110101). Mivel a szótagok hosszúságának a megállapításában fontos információ az, hogy a magánhangzót egy vagy több mássalhangzó követi-e, szükséges volt a programba beépíteni annak vizsgálatát is, hogy egy két- vagy háromjegyű mássalhangzónak tűnő karaktorsorozat valóban két- vagy háromjegyű mássalhangzónak, azaz egy fonémának tekintendő. Ehhez az e-magyar program morfológiai elemzőjét használtuk. Amennyiben a két- vagy háromjegyű mássalhangzónak tűnő karaktorsorozat közé morfémahatár esik, az nem tekinthető egy fonémának.<sup>14</sup>

<sup>14</sup> Például a *gázság* szóra az e-magyar morfológiai elemzője többek között megadja a következő részletes morfológiai elemzési lehetőséget: *gaz[/Adj]=gaz+ság[\_Nz\_Abstr/N]=ság+[Nom]=*. Az elemzésből kiderül, hogy a *gaz* és a *ság* karaktorsorok között morfémahatár van, azaz a *z* és az *s* nem tekinthető egy *zs* hangot alkotó rövid mássalhangzónak.

A program a versszakok rímképleteinek az annotálását a hagyományos módon végzi el: minden versszak esetében egy olyan karaktersor jelöli a rímképletet, amelyben a rímelő sorok az ábécé azonos betűjével reprezentálódnak (pl. aabbcb). A program azokat a sorvégeket tekinti egymással rímelőnek, amelyek megegyeznek egymással a sorvégi záró mássalhangzó megléte vagy meg nem léte tekintetében, amelyekben az utolsó szótag magánhangzója a hosszúságot nem számítva megegyezik, illetve amelyekben megegyezik az utolsó előtti szótagok hosszúsága. A rímelés e szabályának alkalmazásában a fő szempont az volt, hogy a szabály ne legyen túl szűkös, de ne is generáljon túl. Mindkét eset ugyanis ahhoz vezet, hogy a konzisztensen, azaz azonos rímképletű versszakokkal is leelemezhető verseket a program nagyobb eséllyel kezelné inkonzisztens módon, vagyis a túl specifikus vagy túl általános szabály alkalmazása miatt bizonyos versszakokat a többihez képest eltérő rímképlettel annotálna.

A rímképlet mellett a program annotálja az egymással rímpárt alkotó, azaz hívó- és felelőrim viszonyban lévő szavakat. A program jelenlegi verziója csak versszakon belül elemzi rímpárokat. Egy rímpár szavai között maximum négy sor lehet, például egy hat soros abbbca rímképletű versszakban az első és az utolsó sor rímhelyzetben lévő szavait rímpárként azonosítja a program, de például egy abbccca rímképletű hétsoros versszak első és utolsó sorának rímhelyzetben lévő szavait már nem elemzi rímpárként. Egy rímhelyzetben lévő szó két rímpárnak is a része lehet, az első rímpárban felelő-, a másodikban hívórimként. Például egy négysoros bokorrím, azaz egy aaaa rímképletű versszak második sorának a rímhelyzetben lévő szava felelőrimként az első sor, hívórimként pedig a harmadik sor sorvégi szavával is rímpárt alkot, ugyanakkor a program jelenlegi elemzésében nem alkot rímpárt a negyedik sor sorvégi szavával, vagyis egy rímelő szó hívórimként mindig csak a hozzá legközelebbi rímelő szóval alkothat rímpárt.

Az alliterációk annotálása során a program nem csupán azokat a szerkezeteket elemzi alliterációként, amelyekben egymást követő szavak ugyanazzal a hanggal kezdődnek, hanem azokat is, amelyekben két ugyanolyan hanggal kezdődő szó közé beékelődik egy másik hanggal kezdődő szó. Minden alliterációként elemzett szerkezet kap egy "a" és "n" betűből álló karaktersort annotációként, amelyben az "a" betű az egymással alliteráló szavakat, az "n" betű pedig az alliteráló szavak közé beékelődő nem alliteráló szavakat jelöli (pl. „Bus donna barna balkonon” – anaa). Az alliterációk elemzése során a program az *a*, *az*, *és*, *s* szavakat stopwordökként kezeli, azaz ezek a szavak egy kételemű alliterációnak nem lehetnek részei, ugyanakkor részei lehetnek olyan kettőnél több elemű alliteráló szerkezeteknek, amelyekben legalább kettő nem stopword előfordul alliteráló szóként.<sup>15</sup>

A szavak fonológiai tulajdonságainak annotálása a szótagszám, a hangrend (magas, mély vagy vegyes), valamint a szó egyszerűsített fonológiai szerkezetének a megadására terjedt ki. A fonológiai szerkezet megadása során a Magyar Nemzeti Szövegtárban alkalmazott megoldást követtük némi módosítással (Oravec és mtsai, 2014): minden szó kap egy karaktersort, amely c, b, f, B és F karakterekből állhat. Az egyes karakterek a szó hangjainak néhány fontosabb fonológiai tulajdonságát jelölik. Ezek a következők:

<sup>15</sup> Csak versszakon belül elemeztünk alliterációkat, ami azt jelenti, hogy egy versszak utolsó és a következő versszak első, azonos hanggal kezdődő szava nem annotálódik alliterációként.

c – mássalhangzó, b – hátul képzett rövid magánhangzó, f – elől képzett rövid magánhangzó, B – hátul képzett hosszú magánhangzó, F – elől képzett hosszú magánhangzó (pl. *szerszámaival* – cfccBcbfcbc).

## 4. A korpusz formátuma

A korpusz formátuma – leszámítva az utolsó annotációs fázissal létrejövő XML-eket – TEI XML. A TEI XML a digitális bölcsészeti és nyelvészeti korpuszok egyik széles körben elterjedt sztenderdje, amely számos szövegtípus, többek között versek annotálására is felkínál elemkészletet (TEI Consortium, 2021). Az alábbiakban bemutatandó level1, level2, level3 és level4 formátumok a korpusz egyre több annotációs réteget tartalmazó verzióinak a formátumai. Ezek megfelelnek a korpusz gitHub oldalán található könyvtáraknak.

### 4.1. A level1 formátuma

A versek szerkezeti egységeinek az annotálása, illetve az annotációk kézi ellenőrzése révén létrejövő level1-es TEI XML-ek az alábbi módon tartalmazzák a verseket.

```
<text>
  <body>
    <div type="poem">
      <head>Húnyt szemmel...</head>
      <lg>
        <l>Húnyt szemmel bérceken futunk</l>
        <l>s mindig csodára vágy szivünk:</l>
        <l>a legjobb, amit nem tudunk,</l>
        <l>a legszebb, amit nem hiszünk.</l>
      </lg>
      <lg>
        <l>Az álmok síkos gyöngyeit</l>
        <l>szorítsd, ki únod a valót:</l>
        <l>hímezz belőlük</l>
        <l>fázó lelkedre gyöngyös takarót.</l>
      </lg>
    </div>
  </body>
</text>
```

A versek címei a <head> elembe, a versszakok az <lg> elembe, a sorok pedig az <l> elembe kerülnek. A mottók, a különböző szeparátorelemek, illetve a versek keletkezésének helyére és idejére vonatkozó megjegyzések pedig a <p> elembe vannak.

## 4.2. A level2 formátuma

Az e-magyar lefuttatásával létrejövő, a szavak grammatikai annotációit tartalmazó level2-es TEI XML fájlok esetében minden szó belekerül egy külön <w> elembe, amelyek attribútumokként tartalmazzák a szavak e-magyarral felismertetett jellemzőit. A @lemma attribútumba kerül a szó szótári alakja, a @pos attribútumba a szó szófajának a címkéje, az @msd attribútumba pedig a szó morfoszintaktikai jellemzői az Universal Dependencies rendszerének megfelelő tulajdonság-érték párokként.<sup>16</sup> A központozások <pc> elembe kerülnek, amelynek a @join attribútuma jelzi a tapadás irányát. Az <lg>, <l>, <w> és <pc> elemek az @xml:id attribútum értékeként egy egyedi azonosítót is kapnak ebben az annotációs fázisban. Az alábbi XML-részlet az idézett Babits-vers harmadik sora alapján mutatja be az említett annotációk korpuszbeli szerepeltetésének módját.

```
<l xml:id="l3">
  <w lemma="a" msd="Definite=Def|PronType=Art"
    pos="DET" xml:id="w10">a</w>
  <w lemma="jó" msd="Case=Nom|Degree=Sup|Number=Sing"
    pos="ADJ" xml:id="w11">legjobb</w>
  <pc join="left" pos="PUNCT" xml:id="pc2">,</pc>
  <w lemma="ami" msd="Case=Acc|Number=Sing|Person=3|
    PronType=Rel" pos="PRON" xml:id="w12">amit</w>
  <w lemma="nem" msd="PronType=Neg" pos="ADV"
    xml:id="w13">nem</w>
  <w lemma="tud" msd="Definite=Ind|Mood=Ind|
    Number=Plur|Person=1|Tense=Pres|VerbForm=Fin|
    Voice=Act" pos="VERB" xml:id="w14">tudunk</w>
  <pc join="left" pos="PUNCT" xml:id="pc3">,</pc>
</l>
```

## 4.3. A level3 formátuma

A hunpoem\_analyzer-TEI program lefuttatásával létrejövő level3-as TEI XML fájlokba további, a vershangzás annotációit tartalmazó elemek és attribútumok kerülnek bele.

```
<lg rhyme="abab" xml:id="lg1">
  <l n="8" real="11110101" xml:id="l1">
    <w lemma="húnyt" msd="Case=Nom|Degree=Pos
      Number=Sing" pos="ADJ" xml:id="w1">Húnyt</w>
    [...]
```

<sup>16</sup> Az attribútumok megnevezéseit a TEI specifikáció írja elő.

Az <lg> elemek @rhyme attribútumában szerepel a versszak rímképlete, az <l> elem @n attribútumában a sor szótagszáma, a @real attribútumban pedig a sor időmértékes ritmusa. A TEI alapsémája nem teszi lehetővé, hogy a szavak fonológiai jellemzőit a <w> elem attribútumaiként annotáljuk, így azokat standoff módon, azaz a versszövegtől leválasztva, az XML fájlunk egy későbbi részén annotáltuk az alábbi módon.

```
<spanGrp type="phonStructures">
  <span subtype="1" target="#w1" type="low">cBcc</span>
  <span subtype="2" target="#w2" type="high">cfccfc
</span>
  [...]
</spanGrp>
```

A <spanGrp type="phonStructures"> elemben szereplő <span> elemek egy-egy szó főbb fonológiai jellemzőinek az annotációit tartalmazzák. A @target attribútum értéke annak a szónak az xml:id-je, amire az annotáció vonatkozik. A @subtype attribútum értéke a szó szótagszáma, a @type attribútum értéke a szó hangrendje, magának a <span> elemnek a tartalma pedig a szó egyszerűsített fonológiai reprezentációja.

Ugyanígy, standoff módon annotáltuk a rímpárokat és az alliterációkat is. A rímpárokat a <linkGrp type="rhymePairs"> elemekben lévő <link> elemek annotálják. Minden <link> elem egy adott rímpárt annotál oly módon, hogy a @target attribútum értékeként szereplő két xml:id utal a rímpárt alkotó két szóra.

```
<linkGrp type="rhymePairs">
  <link target="#w4 #w14"/>
  <link target="#w9 #w19"/>
  <link target="#w28 #w34"/>
</linkGrp>
```

Az alliterációk annotációi hasonló módon szerepelnek, a <spanGrp type="alliterations"> elem egyes <span> elemei egy-egy alliterációt annotálnak oly módon, hogy a @target attribútum értékei utalnak az alliteráló szerkezetet alkotó szavak xml:id-ire. A @type attribútumban egy "a" és "n" betűből álló karaktersor áll. Az "a" karakterek az alliteráló, azaz ugyanolyan hanggal kezdődő szavakat reprezentálják, az "n" karakterek pedig az alliteráló szavak közé beékelődő nem alliteráló, más hanggal kezdődő szavakat.

```
<spanGrp type="alliterations">
  <span target="#w10 #w11 #w12 #w13" type="anaa"/>
  <span target="#w29 #w30 #w31" type="ana"/>
  <span target="#w34 #w35" type="aa"/>
  [...]
</spanGrp>
```

#### 4.4. A level4 formátuma

Az annotációs folyamat utolsó lépésével létrehozott level4-es fájlokban több szempontból is eltértünk a TEI XML sémától annak érdekében, hogy a versek olyan formátumban is meglegyenek, amelyen egyszerűbben és gyorsabban lehet lekérdezéseket végrehajtani. Egyrészt az érthetőség kedvéért több elemnek és attribútumnak is megváltoztattuk a nevét úgy, hogy az egyértelműen utaljon az általa tartalmazott annotáció típusára. Másrészt a level3 TEI XML fájljaiban standoff módon annotált fonológiai jellemzőket áthelyeztük a szavak <w> elemeiben szereplő attribútumokba. Harmadrészt bővítettük a versek szerkezeti egységeinek az annotációit versszakszámra, sorszámra, szószámra és szótagszámra vonatkozó annotációkkal.

```
<div type="poem" div_numStanza="2" div_numLine="8"
div_numWord="34" div_numSyll="63" div_numShorSyll="24"
div_numLongSyll="39" div_rhyme="abab|abcb"
div_syllPattern="8-8-8-8|8-8-5-10">
  <head type="title">Húnyt szemmel...</head>
  <lg xml:id="lg1" lg_numLine="4" lg_numWord="19"
lg_numSyll="32" lg_numShortSyll="11"
lg_numLongSyll="21" rhyme="abab" lg_syllPattern="8-
8-8-8">
    <l xml:id="l1" l_numWord="4" l_numSyll="8"
l_numShortSyll="2" l_numLongSyll="6" real="11110
101">
      <w xml:id="w1" lemma="Húnyt" pos="ADJ"
msd="Case=Nom|Degree=Pos|Number=Sing"
w_numSyll="1" phonType="low" phonStruct="cBcc">
        Húnyt</w>
```

A fenti level4-es XML példából látható, hogy a <div> elemben szereplő @div\_numStanza attribútum értékeként tüntettük fel a versszakok számát, a @div\_numLine attribútum értékeként a vers sorainak a számát, a @div\_numWord attribútum értékeként a vers szavainak a számát, a div\_numSyll attribútum értékeként pedig a vers szótagjainak a számát. A @div\_numShortSyll és a @div\_numLongSyll attribútumokban szerepel a rövid és a hosszú szótagok száma. Az egész versnek a rímképletét a @div\_rhyme attribútum tartalmazza. A @div\_rhyme attribútum értékeként megadott karaktersorokban virgula választja el az egyes versszakokra vonatkozó rímképleteket. Az azonos betűk csak egy versszakon belül jelölnek egymással rímelő sorokat. A @div\_syllPattern attribútumban szerepel a vers szótagmintája, amely egy kötőjelekkel és virgulákkal elválasztott, számokból álló karaktersor, amelyben a számok a vers egyes sorainak a szótagszámát jelölik. A versszakokat tartalmazó <lg> elemeket és a sorokat tartalmazó <l> elemeket hasonló módon bővítettük további, sorszámra, szószámra és szótagszámra vonatkozó attribútumokkal.

A szavaknak a level3-as TEI XML verzióban standoff módon annotált fonológiai jellemzői a level4-es verzióban – ahogyan a példában is látható – a <w> elemek attribútumai közé kerülnek. A @w\_numSyll attribútumban szerepel a szó szótagszáma, a

@phonType attribútumban a szó hangrendje, a @phonStruct attribútumban pedig a szó fonológiai reprezentációja.

A rímpárok standoff annotációiban az egyértelműség kedvéért megváltoztattuk az elemek neveit, illetve a lekérdezések megkönnyítése érdekében a rímpárok tagjaira vonatkozó elemek tartalmaként feltüntettük a szóalakokat, attribútumokként pedig a szavak annotált jellemzőit. Az alliterációk esetében is megváltoztattuk az alliterációkat tartalmazó elemek neveit, a lekérdezések megkönnyítése érdekében pedig az elemek tartalmaiként itt is feltüntettük az alliterációt alkotó szavakat, attribútumokként pedig az alliterációban szereplő szavak szófaját, morfoszintaktikai jellemzőit és lemmáit.

## 5. A lekérdezőfelület

A verseket és azok annotációit tartalmazó level4-es XML fájlokból létrehoztunk egy MariaDB-alapú SQL-adatbázist, ebben keres a verskorpusznak a <https://verskorpusz.elte-dh.hu> oldalon elérhető, bárki által szabadon használható online lekérdezőeszköze. A lekérdezőeszköz mellett, hogy az egyes versek annotált jellemzőit megjelenítse, számos keresőfunkcióval is rendelkezik. A keresőfunkciók megtervezése során nagymértékben támaszkodhattunk a már létező magyar nyelvű korpuszok lekérdezőfelületeire, különösen a Magyar Nemzeti Szövegtár lekérdezőfelületére (Oravecz és mtsai, 2014). Kereshetünk szóalakokra, lemmákra, szófajokra, morfoszintaktikai jellemzőkre, illetve ezek tetszőleges kombinációjából alkotott szókapcsolatokra is. A fonológiai jellemzők és a ritmus annotálásának köszönhetően szavak szótagszáma, hangrendje, egyszerűsített fonológiai reprezentációja és a szavak szótagjainak hosszúsága alapján is végezhetünk lekérdezéseket, illetve ezeket kombinálhatjuk a szóalakokra, lemmára és morfoszintaktikai jellemzőkre vonatkozó keresőkifejezésekkel. A szerzők mellett rímképletek alapján is szűrhetjük a verseket.

A lekérdezőfelülettel a megadott keresési feltételek alapján szóalakokra és lemmákra vonatkozó gyakorisági listákat is generálhatunk. Amennyiben a keresőmezőben több szóból álló szerkezetre vonatkozó keresési kifejezést adunk meg, akkor e szerkezetre vonatkozóan is létrehozhatunk gyakorisági listát. Az 1. táblázat három, a lekérdezőfelülettel generált gyakorisági listát mutat be: a verskorpuszban szereplő leggyakoribb öt rímhelyzetben lévő főnevet, a leggyakoribb öt *bús* + főnév szókapcsolatot, valamint a leggyakoribb öt alliteráló melléknév + főnév szókapcsolatot. A megadott szófaji kategóriákba tartozó szavak lekérdezése lemmák alapján történt.

	rímhelyzetben lévő főnevek		<i>bús</i> + főnév szókapcsolatok		alliteráló melléknév + főnév szókapcsolatok	
1	élet	2585	<i>bús</i> szív	63	szép szem	221
2	szem	2362	<i>bús</i> szem	56	szép szó	129
3	világ	2350	<i>bús</i> lélek	46	nagy név	93
4	ég	2099	<i>bús</i> fej	45	kis kéz	50
5	kéz	2002	<i>bús</i> dal	36	szép szerelem	30

**1. táblázat.** A verskorpusz leggyakoribb rímhelyzetben lévő főnevei, *bús* + főnév szókapcsolatai, valamint alliteráló melléknév + főnév szókapcsolatai.

A keresési találatokat, az azokhoz kapcsolódó fontosabb kvantitatív adatokat, a gyakorisági listákat, illetve a vizsgálat számára kijelölt alkorpusz legfontosabb kvantitatív jellemzőit TSV-formátumban letölthetjük, és bármelyik táblázatkezelő programban megnyithatjuk. A lekérdezőfelülethez tartozik egy részletes használati útmutató is, amely bemutatja az egyes keresési funkciókat.

## 6. Összegzés

Az ELTE Verskorpusz építésével a célunk egy olyan, bárki számára elérhető annotált korpusz létrehozása volt, amely reményeink szerint mind az irodalomtudományos, mind pedig a nyelvészeti kutatásokat segítheti. A korpuszhoz készült lekérdezőfelület lehetővé teszi, hogy különösebb informatikai tudás nélkül is olyan információkhoz juthassunk a magyar kanonikus költészetéről, amelyhez a szoros olvasás eljárásai révén nem juthatnánk. Bízunk abban, hogy a lekérdezőfelület révén a korpusz nemcsak a kutatásban, hanem egyéb színtereken, például a közoktatásban is hasznosítható lesz. Az annotált XML fájlok a projekt gitHub oldaláról (<https://github.com/ELTE-DH/poetry-corpus>) letölthetőek, és kutatás számára szabadon felhasználhatóak. Az XML fájlok közzétételének köszönhetően a programozási tudással rendelkező kutatónak lehetősége van arra, hogy olyan összetettebb lekérdezéseket is elvégezhesen a korpuszon, amelyeket az online lekérdezőfelület nem tesz lehetővé. Az ELTE Verskorpusz nem egy lezárt projekt, a jövőben szeretnénk további szerzőkkel és annotációs rétegekkel bővíteni a korpuszt, illetve további funkciókkal kiegészíteni a lekérdezőfelületet.

## Köszönetnyilvánítás

Az ELTE Verskorpusz elkészítését a Felsőoktatási Intézményi Kiválósági Program és a Digitális Örökség Nemzeti Laboratórium támogatta.

## Hivatkozások

- Agirrezabal, M., Astigarraga, A., Arrieta, B., Hulden, M.: ZeuScansion: A Tool for Scansion of English Poetry. *Journal of Language Modelling* 4(1), 3–28 (2016)
- Csengery, K.: Az elektronikus korpusz. In: Ittész, N. (szerk.) *A magyar nyelv nagyszótára 1. Segédletek*. pp. 18–19. MTA Nyelvtudományi Intézet, Budapest (2006)
- Dodé, R., Ludányi Zs., Falyuna, N., Kuna, Á.: Poétika és korpusz. Hogyan nyújthat segítséget a korpusznyelvészet poétikus szövegek vizsgálatához? In: Domonkosi, Á., Simon, G. (szerk.) *Nyelv, poétika, kogníció. Elmélet és módszer a poétikai kutatásban*. pp. 175–196. Líceum Kiadó, Eger (2018)
- Hartman, C., O.: *The Scandroid*. Version 1.1. [User guide] <http://charlesohartman.com/verse/scandroid/ScandroidManual.pdf> (2005)
- Horváth, P.: A vershangzás jellemzőinek automatikus feltárása József Attila verseiben. *Digitális Bölcsészlet* 3, M:3–M:27 (2020a)



- Horváth, P.: Az ELTE Verskorpusz automatikus annotációs eljárásai révén nyerhető kvantitatív adattípusok. In: Simon, G., Tolcsvai Nagy, G. (szerk.) *Nyelvtan, diskurzus, megismerés*. pp. 313–332. Eötvös Kiadó, Budapest (2020b)
- Ibrahim, R., Plecháč, P.: Toward Automatic Analysis of Czech Verse. In: Scherr, B. P., Baily, J., Kazartsev, E. V. (szerk.) *Formal Methods in Poetics*. pp. 295–305. RAM, Lüdenscheid (2011)
- Indig, B., Sass B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: emtsv – egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) *XV. Magyar Számítógépes Nyelvészeti Konferencia*. pp. 235–247. Szegedi Tudományegyetem TTIK, Informatikai Intézet, Szeged (2019)
- Jékel, P., Papp, F.: *Ady Endre összes költői műveinek fonémastatisztikája*. Akadémiai Kiadó, Budapest (1974)
- Jékel, P., Szuromi, L.: *Petőfi metrumai*. Kossuth Lajos Tudományegyetem, Debrecen (1980)
- Kao, J., Jurafsky, D.: A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry. In: *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*. pp. 8–17. Association for Computational Linguistics, Montréal (2012)
- Labádi, G.: Az olvasó gép: Berzsenyi Dániel versei távolról. *Digitális Bölcsészlet* 1, 17–34 (2018)
- Lesi, Z.: Automatikus verselemzés tanuló algoritmusok alkalmazásával. In: Alexin, Z., Csendes, D. (szerk.) *IV. Magyar Számítógépes Nyelvészeti Konferencia*. pp. 402–407. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2006)
- Lesi, Z.: Automatikus formai verselemzés. *Alkalmazott Nyelvtudomány* 8(1-2), 197–208 (2008)
- Navarro-Colorado, B.: A Computational Linguistic Approach to Spanish Golden Age Sonnets: Metrical and Semantic Aspects. In: *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*. pp. 105–113. Association for Computational Linguistics (ACL), Denver (2015)
- Navarro-Colorado, B., Ribes Lafoz, M., Sánchez, N.: Metrical Annotation of a Large Corpus of Spanish Sonnets: Representation, Scansion and Evaluation. In: Calzolari, N., Choukri, K., Declerck, T. et al. (szerk.) *Proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC 2016)*. pp. 4360–4364. European Languages Resources Association (ELRA), Portorož (2016)
- Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. pp. 1719–1723. European Languages Resources Association (ELRA), Reykjavik (2014)
- Plamondon, M. R.: Virtual Verse Analysis: Analysing Patterns in Poetry. *Literary and Linguistic Computing* 21(1), 127–141 (2006)
- Plecháč, P., Kolár, R.: The Corpus of Czech Verse. *Studia Metrica et Poetica* 2(1), 107–118 (2015)
- Sass, B.: Keresés korpuszban: a kibővített Magyar történeti szövegtár új keresőfelülete. In: *A nyelvtörténeti kutatások újabb eredményei*. pp. 267–277. Szegedi Tudományegyetem Magyar Nyelvészeti Tanszék, Szeged (2017)
- Simon, E., Indig, B., Kalivoda, Á., Mittelholcz, I., Sass, B., Vadász, N.: Újabb fejlemények az e-magyar háza táján. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) *XVI. Magyar Számítógépes Nyelvészeti Konferencia*. pp. 29–42. SZTE Informatikai Intézet, Szeged (2020)
- Tanasescu, C., Paget, B., Inkpen, D.: Automatic Classification of Poetry by Meter and Rhyme. In: *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*. Florida Artificial Intelligence Research Society. <https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/view/12923/12883> (2016)
- TEI Consortium: *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 3.5.0. <https://tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf> (2019)

- Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholtz, I., Novák, A., Indig, B., Prószéky, G., Vincze, V.: Az e-magyar digitális nyelvfeldolgozó rendszer. In: Vincze, V. (szerk.) XIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 49–60. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2017)
- Vincze, V., Simkó, K., Szántó, Zs., Farkas, R.: Universal Dependencies and Morphology for Hungarian – and on the Price of Universality. In: Mirella, L., Phil, B., Alexander, K. (szerk.) Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017): Volume 1. Long papers. pp. 356–365. Association for Computational Linguistics (ACL), Valencia (2017)
- Voigt, V.: Számítógépes ritmuselemzési kísérlet. Irodalomtörténeti Közlemények 76(2), 203–211 (1972)

## NerKor 1.41e

Novák Attila<sup>1,2</sup>, Novák Borbála<sup>1,2</sup>

<sup>1</sup>Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar

<sup>2</sup>MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport

Budapest, Práter u. 50/a.

{vezetéknév.keresztnév}@itk.ppke.hu

**Kivonat** Cikkünkben a legfrissebb és mind ezidáig legnagyobb magyar névelemkorpuszban, az NYTK-NerKorban szereplő annotáció automatikus kibővítésével kapcsolatos eredményeinket mutatjuk be. Az eredeti korpuszban megkülönböztetett négy entitástípust 30 körüli típusra bővítettük. Az új annotációt nyelvi transzferrel hoztuk létre. A korpuszt kiegészítettük egy kis méretű 12000 tokenes gépkocsi témájú részkorpuszal is. Ugyanakkor egyelőre sajnos a korpusz méretéből adódóan nem volt módunk a teljes anyagot alapos kézi ellenőrzésnek alávetni. Ezért az eredeti változattal ellentétben az új verzió nem tekinthető gold standardnek, különösen az eredetiben egyáltalán nem jelölt (többségében nem névelem típusú) entitások tekintetében. Teljes körű kézi ellenőrzésen csak a korpusz 2,5%-a esett át<sup>1</sup>, amelyet tesztanyagként használtunk a cikkben bemutatott modellek kiértékeléséhez. Mindazonáltal reméljük, hogy a korpusz ezen változata hasznos anyagnak bizonyul a magyar nyelvtechnológiai közösség számára.

**Kulcsszavak:** névelem-felismerés, nyelvi transzfer, annotált korpusz, gépi annotáció, többnyelvű nyelvmodellek

### 1. Bevezetés

Cikkünkben a mind ezidáig legnagyobb magyar névelemkorpuszban, az NYTK-NerKorban (Simon és Vadász, 2021) szereplő annotáció automatikus kibővítésével kapcsolatos eredményeinket mutatjuk be. Az eredeti korpuszban megkülönböztetett négy entitástípussal szemben az itt bemutatott bővített változatban 30 körüli entitástípus szerepel. Az entitástípusok száma részben a meglévő osztályok finomításával, részben új, nem névelem jellegű entitások (pl. idő- és mennyiségkifejezések) hozzáadásával nőtt. Az újraannotálás során elsősorban az angol OntoNotes 5 korpusz (Weischedel és mtsai, 2013) entitástípusaira, illetve azon betanított modellekre támaszkodtunk, de új entitástípusokkal és egy 12000 tokenes gépkocsi témájú részkorpuszal is kiegészítettük azt. Az új verzió nem tekinthető gold standardnek, különösen az eredetiben egyáltalán nem jelölt (többségében nem névelem típusú) entitások tekintetében, mert teljes körű kézi ellenőrzésen csak a korpusz 2,5%-a esett át, amelyet tesztanyagként használtunk

<sup>1</sup> valamint az autókorpusz

a cikkben bemutatott modellek kiértékeléséhez (ez az eredeti korpuszban definiált teszhalmaztól különbözik). Ebből adódóan a korpusz verziószámát 1.0-ról csak nagyjából  $\sqrt{2}$ -re növeltük, és hozzáadtunk egy e-t, ami azt hivatott jelölni, hogy nem tiszta arany, amit csináltunk, hanem elektrum.<sup>2</sup> Mindazonáltal reméljük, hogy a korpusz ezen változata is, mely az eredeti korpuszsal azonos licensszel elérhető,<sup>3</sup> hasznos anyagnak bizonyul a magyar nyelvtechnológiai közösség számára.

## 2. Előzmények

A névelem-felismerés alapvető nyelvfeldolgozási feladat, ami számos ráépülő feladatban fontos szerepet játszik, pl. információkivonatolás, dokumentumanonimizálás, különböző dialógusrendszerek, stb. A CONLL 2002-2003-as névelem-felismerési feladat annotációs sémáját követve a legtöbb hagyományos névelemkorpusz mindössze négy entitástípust különböztet meg: szervezet (ORG), személy (PER), hely (LOC), és az összes többi entitástípust lefedő egyéb (MISC) kategória. Minden korábban készült magyar nyelvű névelemkorpusz ugyanezt az annotációs sémát követi egészen az első ilyen erőforrástól, a Szeged NER korpusztól (Szarvas és mtsai, 2006) kezdve<sup>4</sup> a legfrissebb és legnagyobb NerKor korpuszig (Simon és Vadász, 2021), amelyen az itt bemutatott kutatásunk alapul.

A későbbi névelemkutatások során gazdagabb entitáskészleten alapuló annotációk is készültek. Az angol OntoNotes 5 korpusz (Weischedel és mtsai, 2013) például 18 entitástípust különböztet meg.<sup>5</sup> Ez az annotációs séma a helyeken belül például megkülönbözteti a geopolitikai entitásokat (országok, települések, stb.) és a létesítményeket (épületek, utak, repülőterek, stb.) a földrajzi helyektől (kontinensek, vizek, hegységek, stb.) Az egyéb kategórián belül különválasztják a termékeket, a jogszabályokat és egyéb jogi norma jellegű entitásokat, az eseményeket és az alkotások címeit. Ezek mellett az OntoNotes korpuszban alkalmazott annotációs séma idővel és kvantoros kifejezésekkel kapcsolatos típusokat is tartalmaz, illetve felőlel néhány olyan további „entitástípust”, amely nem neveknek felel meg, hanem – vélhetően amiatt, hogy az angol nyelvben ezeket a kifejezéseket mindig nagybetűvel írják – a nyelvek nevei, illetve a különböző embercsoportokat jelölő nemzetiségnevek, vallási és politikai hovatartozással kapcsolatos kifejezések is külön típust képeznek.

Az orvosbiológiai szakszövegek területét lefedő több tucat (angol nyelvű) szaknyelvi erőforrás más típusú elemekkel foglalkozik: számos különböző orvosi, illetve biológiai szempontból lényeges entitástípust különböztetnek meg. A

<sup>2</sup> Elektrum: természetes arany-ezüst ötvözet.

<sup>3</sup> <https://github.com/novakat/NYTK-NerKor-Cars-OntoNotesPP>

<sup>4</sup> Az itt bemutatott munka előzményeként elkészült a Szeged NER korpusz gazdagabb annotációt tartalmazó változata is (Novák és Novák, 2021) az itt bemutatotthoz hasonló módszertannal.

<sup>5</sup> CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PER, PERCENT, PROD, QUANTITY, TIME, WORK\_OF\_ART

legtöbb ilyen erőforrásban főszereplők a gének, illetve fehérjék. Ezek mellett fajnevek, betegségek, a kémiai nomenklatúrához tartozó elemek, sejtvonalak, illetve anatómiai egységek annotációja szerepel bennük. Az orvosbiológiai entitásfelismeréssel kapcsolatos kutatások egyik legfontosabb mérföldköve a GENIA-korpusz (Kim és mtsai, 2003) létrehozása volt, amely 47 entitástípust különböztetett meg.

Számos angoltól különböző nyelvre is készültek olyan általános tematikájú névelemkorpuszok, amelyek négy-nél több entitástípust különböztetnek meg. Ugyan a német NoSta-D korpusz (Benikova és mtsai, 2014) a négyosztályos névelemtípus-megkülönböztetés mellett marad, de ebben a korpuszban külön típusként annotálva vannak a nevekből képzett szavak (főleg melléknevek) és a neveket tartalmazó összetett szavak is tekintettel arra, hogy a németben ezek a konstrukciók az angollal ellentétben nem kapnának annotációt a két nyelv morfológiai és helyesírási különbségei miatt egy pusztán szótoken alapú annotáció esetén. Ez a korpusz, hasonlóan más erőforrásokhoz, például az orvosbiológiai entításokat tartalmazó GENIA-korpuszhoz, illetve a spanyol és katalán sajtószövegeket tartalmazó AnCora-korpuszhoz (Taulé és mtsai, 2008) egymásba ágyazott entításokat is tartalmaz. Az egyik talán leggazdagabban annotált névelemkorpusz, a cseh nyelvre készített Czech Named Entity Corpus (CNEC, Ševčíková és mtsai (2007)) is tartalmaz egymásba ágyazott entításokat, és gazdag entitástípus-hierarchián alapul. 4 eredendően komplex entitástípust is megkülönböztetnek, amelyekben belül tipikus a beágyazott entítások jelenléte: ilyenek a személynevek, az idővel kapcsolatos kifejezések, a (posta)címek, és a bibliográfiai adatok. Az alapvető típusok száma pedig 46, ezek 8 felső szintű kategória alá vannak besorolva.

### 3. A NerKor korpusz

Az NYTK-NerKor (továbbiakban: NerKor) korpusz 1 millió tokenből áll, jóval nagyobb, mint a korábbi magyar nyelvű NER-korpuszok. Emellett a korábbi egyetlen domént lefedő korpuszokkal ellentétben a műfajok és témák széles választékát öleli fel, így mindenképp hiánypótló erőforrás. Öt egyenként mintegy kétszázezer tokenes részkorpusz alkotja. A *szépirodalmi* alkorpusz részben 20. század eleji szövegekből áll, amelyek helyesírásukban és nyelvhasználatukban is viszonylag jelentős mértékben eltérnek a jelenlegi normától. Ennek a részkorpusznak a másik része magyar nyelvű filmfeliratok részleteit tartalmazza. A *jogi* részkorpusz európai uniós jogi szövegek részleteiből áll. Ezek sajátossága, hogy rengeteg jogszabályi hivatkozást tartalmaznak. Sajnos ezek nem összefüggő szövegek, hanem mondatok összekevert halmazáról van szó. Ugyanez igaz a magyar Webkorpusz 2-ből válogatott *webes* részkorpuszra, és a magyar *Wikipédiából* származó részkorpusz nagy részére is, amely részben a korábban készült hunNER-wiki korpuszon alapul (Simon és Nemeskey, 2012). Ehhez olyan mondatokat választottak a korpusz összeállítói, amelyek legalább egy névelemet tartalmaznak. Sajnos az automatikus mondatokra bontó algoritmus hibájából kifolyólag ebben a részkorpuszban nagyon gyakoriak a rövidítéseknél félbeszakadt

mondattöredékek. A *hírszövegeket* tartalmazó részkorpusz egy része a Global Voices magyar kiadásából<sup>6</sup> származik, amely amatőr fordítók által írt szövegeket tartalmaz. Ebben a részkorpuszban nagyon gyakoriak a közösségi oldalakra való hivatkozások. Ez a korpuszrész összefüggő szövegrészeket tartalmaz. A sajtókorpusz másik fele a WMT gépi fordítási konferenciákat szervező csapat<sup>7</sup> által készített NewsCrawl korpusz<sup>8</sup> 2019-es magyar anyagát tartalmazza. Ez a rész ismét megkevert mondathalmaz.

A korpusz egészét tekintve megállapítható, hogy nagyjából a harmada tartalmaz összefüggő szövegeket, a másik része a korpusznak nem összefüggő mondat- vagy mondattöredék-halmaz. Ebből a szempontból különbözik a korábbi magyar nyelvű gold standard névelemkorpuszoktól, amelyek jobbra összefüggő szövegekből vagy szövegrészekből álltak. Emellett csak a szövegeknek csak nagyjából a fele tekinthető gondozott és a jelenlegi helyesírási normáknak megfelelő szövegnek (és éppen ezekre a részekre jellemző, hogy véletlenszerű sorrendben következő mondatokból állnak).

A mondatok összekeverése korábban gyakran alkalmazott módszer volt az esetleges copyright-problémák elkerülésére, és amíg a nyelvtechnológiában alkalmazott modellek meg sem próbálták szövegszintű összefüggéseket kezelni, nem tűnt különösebben fájdalmasnak ez az ár, amit az esetleges jogi hercehurcák elkerüléséért a korpuszok szabad felhasználhatósága érdekében fizetni kellett. Maga a névelem-felismerési feladat sem követel meg az esetek túlnyomó részében a mondatnál nagyobb kontextust. Az érdekes entitásokat nem tartalmazó szövegrészek eldobása is indokolható az annotátorokra rótt feladat csökkentésével (mi is éltünk ezzel az autós korpuszrész létrehozásakor). Azonban tekintve az annotációba beleölt munkát, jó lenne, ha a korpuszt más célokra is tovább lehetne fejleszteni, ahol szükség van a tágabb összefüggésekre (pl. koreferencia-annotáció). Ilyen célokra a korpusz harmada használható: örömteli, hogy részben már koreferencia-annotált korpuszból, a KorKor pilotkorpuszból (Vadász, 2020) származik ezeknek a szövegeknek egy része.

A korpuszban szereplő szövegek egyötöd része morfológiai annotációt is tartalmaz, hogy hagyományos gépi tanuló algoritmusok is betaníthatóak legyenek rajta. Jelen átdolgozás során azonban a morfológiai annotációt nem vettük át.

## 4. Annotációs módszerünk

A korpusz előannotálásakor két az angol OntoNotes 5 névelemkorpuszon betanított modellt alkalmaztunk a magyar nyelvű korpuszra. Az egyiket a DeepPavlov csapat készítette multilingual BERT nyelvmodell finomhangolásával (Burtsev és mtsai, 2018). A másik modellt a Flair eszközkészlet része (Akbik és mtsai, 2019), és az XLM-RoBERTa többnyelvű nyelvmodellen (Conneau és mtsai, 2019) alapul, amely sokkal nagyobb többnyelvű korpuszon lett betanítva, mint a multilingual BERT. A két modell különböző tokenizálást végez a nyelvmodell

<sup>6</sup> <https://hu.globalvoices.org/>

<sup>7</sup> <http://www.statmt.org/wmt19/>

<sup>8</sup> <http://data.statmt.org/news-crawl/>

tokenizálási modelljéhez alkalmazkodva: az XLM-RoBERTa SentencePiece alapú (Kudo és Richardson, 2018), míg a BERT hagyományos tokenizálást követő BPE tokenizáláson alapul. Ennek megfelelően a modellek által visszaadott tokenszekvencia különbözött az eredeti korpusz tokenszekvenciájától. Erre tekintettel kellett lenni, amikor a modellek kimenetét egyesítettük az eredeti annotációval. Az összefésülést végző algoritmus átfedő entitások esetén az eredeti annotációban szereplő entitástartományokat tartotta meg. Ha az újraannotálást végző modell az eredeti annotáció egy altípusának megfelelő entitást javasolt, akkor ennek megfelelően módosította az entitás típusát (pl. LOC *hely* → LOC *földrajzi hely*/FAC *létesítmény*/GPE *geopolitikai entitás*). Az eredeti annotációban nem szereplő típusok, mint például a *dátumok*, *mennyiségek*, *nemzetiségek* a modellek kimenete alapján kerültek bele az annotációba.

#### 4.1. Hibaelemzés és automatikus hibajavítás

A modellek által generált annotáció jellegzetes hibákat tartalmaz. Tipikus probléma például az angol–magyar transzfer esetében, hogy a művek címei, a szervezetek, újságok, események stb. nevei a magyarban általában határozott névelővel állnak, mikor a mondat szerkezetbe integrálódnak, míg az angolban soha nem áll határozott névelő az ilyen típusú nevek előtt. Ezért a modellek a határozott névelőket az esetek nagy részében a név részének jelölték. Ez a hiba azonban egyszerű reguláris kifejezések segítségével felismerhető és javítható. Hasonlóképpen automatikusan javítható volt a számok, illetve mennyiségek annotációjával kapcsolatos anomáliák egy része.

#### 4.2. A nyelvi transzfer hozadékai és a felvetett dilemmák

Bár a nyelvek közötti leképezés hibákkal is jár, hasznos mellékhatásai is vannak. Például mivel a magyar nevek közül képzett mellékneveknek az angolban többnyire prepozíciós frázisok felelnek meg, amelyek a neveket eredeti formájukban tartalmazzák, és így az eredeti angol annotációban névelemként megjelölve szerepelnek, ezért a modellek az ilyen melléknevek legtöbb előfordulását megfelelő típusú névelemként annotálták. Szemben azzal, ahogy ezt az esetet a német NoSta-D névelemkorpuszban kezelték, az összes hagyományos magyar névelemkorpuszban ezek a szavak jelöletlenül maradtak, annak ellenére, hogy névelemként való azonosításuk a névelem-felismerésre épülő alkalmazásokban, mint például az információ-visszakeresés, vagy anonimizálás hasznos volna. Ezért a modellek kimenetében ezeket az annotációkat hasznosnak ítéltük, és megtartottuk őket.

Ez a lépés ugyanakkor olyan dilemmákat is felvet, amelyek a melléknevek annotációjától tartózkodás esetén nem nehezítik meg az ember életét. Míg az eredeti OntoNotes korpuszban pusztán szófaji alapon könnyen elhatárolhatóak voltak a NORP<sup>9</sup> címkével jelölt melléknévi elemek azoktól az esetektől, ahol a hasonló jelentésű elemek birtokos vagy prepozíciós szerkezet formájában jelennek meg (ezek esetében GPE vagy ORG annotáció jelenik meg), ezek magyar képe

<sup>9</sup> Nationality, other, religion, political affiliation

összemosódik, és ez sokszor nehezen feloldható többértelműségekhez vezet. Bár az *angol-angliai* kettősség esetében viszonylag egyértelműen eldönthető, hogy az előbbire a NORP, az utóbbira inkább a GPE címke illik, a legtöbb analóg esetben (pl. *kínai*) nincs lexikai kettősség. Ezek esetében a kézi egyértelműsítés, illetve annotáció vezérfonala az *angol-angliai* kettősség analógiaként való használata lehet, de mindenképp könnyű hibázni ezeknek a döntéseknek a meghozatalakor.

A nem név jellegű entitások bevezetésekor az annotálandó kifejezések pontos körét is viszonylag nehéz koherensen definiálni, és konzekvensen betartani (ez az időkifejezésekre és a mennyiségekre is igaz). Emellett az új elemek megjelenése problémákat vet fel azzal kapcsolatban is, hogy az elemek elszaporodásával még inkább megnő az igény az egymásba ágyazott entitások jelölésére, amitől egyelőre tartózkodtunk.

Egyelőre ezeket a problémákat a gondos kézi ellenőrzés későbbre halasztásával a korpusz 97,5%-ában a szőnyeg alá söpörhettük, de az ellenőrzött 2,5%-ban azért szembesültünk velük.

A két modell kimenetére alkalmazott automatikus javítóminták által javított annotációváltozatokat manuálisan egyesítettük áttekintve a közöttük lévő különbségeket.

### 4.3. Mennyire vonzó a cseh modell?

A két angol nyelvű modell mellett a korábban említett cseh névelemkorpuszon (CNEC2) betanított NameTag2 neurális névelem-felismerő modellt is alkalmaztuk a korpuszra. Ez a modell egy olyan részletes entitáshierarchián alapul, amelyben az általánosabb kategóriákon belül alosztályok is szerepelnek, például a szervezeteken belül megkülönbözteti a gazdasági társaságokat az állami/politikai intézményeket, a tudományos, kulturális, oktatási és sport szervezeteket, valamint a konferenciákat és versenyeket (az utóbbiakat szintén szervezetnek tekintve). A NameTag2 eszköz egymásba ágyazott annotációkat is képes létrehozni: maximum két egymással átfedő entitást rendel egy-egy tokenhez. A modell web service-ként érhető el. Bár a részletes osztályozás vonzónak tűnt, a cseh anyagon betanított modellt magyarra alkalmazva az kevésbé pontos elemzést adott, mint az angol modellek.<sup>10</sup> A csehben egyáltalán nincsen határozott névelő, így arra számítottunk, hogy ennek a modellnek is problémája lehet a névelők névelemekhez ragasztásával, azonban ez csak a mondateleji nagybetűs névelők esetén történt így. Ugyanakkor ugyanannak az entitásnak a különböző előfordulásai gyakran különböző elemzéseket kaptak, és az esetek többségében ez hiba volt, nem valódi többértelműségről vagy metonimikus használatról volt szó. A nevek határait is kevésbé pontosan találta el ez a modell, mint az angol korpuszon betanított modellek.

<sup>10</sup> Tény, hogy az OntoNotes korpusz 1,5 millió tokenjével szemben a CNEC 2.0 csak kétszázezer tokenből áll, miközben az utóbbi jóval több entitástípust különböztet meg.



#### 4.4. Tövesített névelemlista és automatikus javítóminták

Ennek ellenére a cseh modell által generált annotációk hasznosnak bizonyultak. Az összes modell által generált névelem-annotációkat lemmatizáltuk (csak a név utolsó elemét) és létrehoztunk egy névelemosztályok szerint csoportosított listát (egy gazetteer jellegű erőforrást) a korpuszban szereplő összes tövesített névelemről, amelyben az egyes entitások alternatív elemzéseit azok gyakoriságával együtt felsoroltuk. A listákat áttekintve úgy találtuk, hogy a cseh modell viszonylag gyakran téved az elemek osztályozásával kapcsolatban, ezért nem vettük át a CNEC2-ben alkalmazott taxonómiát.

Ugyanakkor az automatikusan generált névelemlista segítségével azonosítani tudtuk az OntoNotes modellben szereplő entitástípusok közül gyakran félreosztályozott elemeket, illetve olyan entitásokat, amelyek jól megkülönböztethető alosztályba tartoznak, mint például a sajtóorgánumok, vagy a közösségi oldalak. Ezek a magyar korpuszban MISC annotációt kaptak, mikor az újságról vagy közösségimédia-platfomról, és ORG-ot, mikor a megfelelő cégről, kiadóról, szerkesztőségről van szó, tehát a NerKor korpuszban az annotáció a tag-for-meaning elvet követi. Az OntoNotes annotációban ugyanakkor ezek mind ORG típusúak, ami nem túl jó megoldás. Mi itt új entitástípust (MEDIA, illetve SMEDIA) rendeltünk a MISC olvasatokhoz a CNEC korpuszban használt eljárás mintájára. A másik osztályba átsorolandó elemeket kézzel megjelöltük a listában, és ebből automatikusan generáltunk olyan reguláriskifejezés-alapú javítószkripteket, amelyek a felsorolt elemek ragozott alakjainak annotációját is automatikusan javítják a korpuszban. Ezeknek a mintáknak a segítségével automatikusan tömegesen javítottuk az annotációt.

Az OntoNotes modellek által generált annotációban a mennyiségek külön osztályt alkotnak, ezek egy számból és egy mértékegységből állnak. Ugyanakkor az időtartamokat jelölő kifejezések, mint például a *két napra*, vagy *három éves* dátumként annotálódnak. Az ilyen típusú kifejezéseket kézzel átnézve időtartam, illetve életkor típusúvá alakítottuk át (részben automatikusan).

#### 4.5. Kézi hibajavítás

A korpusz egészében kézi hibajavítást három pontról elindulva végeztünk. Egyrészt a tövesített névelemlistában szereplő anomáliákat néztük át és javítottuk az előző részben leírt módon főleg automatikus minták alkalmazásával, de részben kézi ellenőrzéssel. Másrészt az eredeti korpusz és a transzfermodellek által adott annotáció ellentmondásait (illetve a különböző transzfermodellek kimenetének különbségeit) az egyes esetek kézi ellenőrzésével oldottuk fel. Harmadrészt elsősorban a jogi részkorpuszban nagyon gyakori jogszabály-hivatkozások kapcsán figyeltük meg, hogy az egyes jogszabályok részeire való hivatkozások esetében (pl. *[1260/2001/EK rendelet] [1. cikke (1) bekezdésének a) pontjában]*) a modellek kimenete sokszor fragmentált volt, és változatos annotációs mintákat produkáltak. Ezeket olyan formában egységesítettük, hogy a jogszabály nevét egy entitásnak, majd a konkrét jogszabályi helyre való hivatkozást másik, az előzőt követő entitásnak jelöltük. Minden bármilyen formában előkerült kézi javítás környékén

észrevett bármely egyéb hibát is javítottunk, és a folyamat közben előkerült tipikusnak tűnő hibákra is rákerestünk, és javítottuk őket. Azonban mint azt a bevezetőben jeleztük, a teljes korpusz szisztematikus ellenőrzésére az elejétől a végéig a munkálat jelenlegi fázisában nem került sor.

#### 4.6. Metonimikus nyelvhasználat

Az eredeti NerKor korpuszban a nevek metonimikus használata az éppen aktuálisan használt értelem szerinti címkét kapta (tag-for-meaning annotáció). Nemcsak a korábban említett sajtótermék/szerkesztőség kettősség esetében történt ez így, hanem például amikor az országnevek ágensként jelennek meg a szövegben, szervezetként és nem helyként vannak az eredeti korpuszban annotálva. Ez a típusú metonímia teljesen produktív minden geopolitikai entitás esetében, és az összes előfordulás geopolitikai entitásként való annotációja véleményünk szerint megnyugtatóan kezelné a metonimikus névhasználat ezen gyakori típusának problémáját. Az országneveket csak akkor jelöltük egyszerűen szervezetnek, amikor a szövegben nemzeti válogatott (sportcsapat) értelemben szerepeltek (hasonlóan más sportcsapatok annotációjához). Ez az eset véleményünk szerint viszonylag markánsan különbözik attól, amikor egy cégre és annak termékeire ugyanazt a nevet használjuk, és a kevésbé produktív metonimikus mintáktól, mint pl. egy utcanev (*Wall Street*) használata egy intézmény (a New York-i tőzsde) vagy egy városnév (*Brüsszel*) használata egy jóval kiterjedtebb geopolitikai entitás (az EU) jelölésére, melyeknél teljesen indokolt az elválasztás, mert egyértelműen nem azonos entitástípusról van szó.

Az eredeti korpuszban a legtöbb esetben MISC osztályúnak jelölték azokat az eseteket, ahol az állítás arról szólt, hogy valakinek vagy valaminek mi a neve. Ezekben az esetekben első körben úgy jártunk el, hogy a MISC címkéhez hozzácsatoltuk a név valódi osztályára utaló címkét, például egy személynév esetében MISC-PER. Praktikus szempontból azonban ezek az esetek is az adott személyre/szervezetre való referenciának tekinthetők, tehát célszerűnek érezzük őket egyszerűen személy típusúnak tekinteni (az 1. táblázatban és a kiértékelésnél így tettünk, ugyanakkor a korpusz publikált változatában ezt az egyszerűsítést nem tettük meg, hogy akár az eredeti annotációban kódolt információ is visszanyerhető legyen belőle). Hasonló a szerző-mű metonímia esete (pl. *Marxot olvas*), ahol bár nyilván nem személy, hanem mű az olvasás témája, *Marx* a szerző neve (aki személy, és őt említjük) nem az olvasott műé (Marx semelyik konkrét művét nem hívják Marxnak; *Marxot olvas* = ‘Marx<sub>PERS</sub> valamelyik művét olvassa’ ≠ ‘a Marx<sub>WORK\_OF\_ART</sub> című művet olvassa’).

Mint fentebb említettük, a sajtóorgánumok annotációja a négyentitásos korpuszokban általában MISC. Az OntoNotes-alapú modellek szervezetként annotálják őket (akkor is, mikor nem úgy kéne). Ezek, illetve a közösségi oldalak valamint a projektek annotációjára külön típusokat vezettünk be. Vannak ezek mellett azonban további entitások (pl. a Wikipédia), amelyek határesetnek tekinthetők a szervezet-média-műalkotás-termék-kontinuum homályosabb régióiban. Ezek annotációja egyelőre jobb híján MISC-ORG lett az eredeti és az

OntoNotes-alapú annotáció kombinációjaként, vagy egyszerűen MISC maradt, mikor végképp nem tudtuk eldönteni, hogy miről is van szó.

#### 4.7. Az autókörpusz

Elvégeztünk egy kísérletet arra vonatkozóan, hogy milyen eljárással lehetne új altípust bevezetni valamelyik, a korpuszban már megkülönböztetett típuson belül. Kísérleti osztályunknak a gépjárműveket választottuk a *termék* osztályon belül. A betanításhoz a HVG archívumából<sup>11</sup> választottunk autó témájú kulcsszócímek alapján keresve cikkeket, és ezekből a cikkekből választottunk ki olyan mondatokat, amelyben egy autókereskedelmi oldal menürendszeréből letöltött autótípus- és modell-listában előforduló szavak szerepelnek. Ezt az 12000 tokenes korpuszt a Flair OntoNotes modelljével annotálva majd kézzel kijavítva az annotációs hibákat a benne szereplő autónevek típusát *termék*-ről *gépjármű*-re alakítottuk át. Ezt a speciális részkorpuszt hozzáadtuk a korpuszhoz mint a gépjárművekre vonatkozó tanítóanyagot.

#### 4.8. A korpusz jellemzői

Az eredeti NerKor korpusz és a létrejött új változat címkeeloszlását az 1. táblázatban mutatjuk be.

PER <sup>a</sup>	15266	PER <sup>a</sup>	15239	LOC	2242	FAC	816	PROJ	252
LOC	12988	GPE <sup>b</sup>	13809	WORK_OF_ART	1973	MONEY	681	MISC	119
ORG <sup>b</sup>	12343	DATE	12067	QUANTITY	1910	EVENT	613	ID	80
MISC	5751	ORG <sup>b</sup>	9516	CAR	1378	LANGUAGE	496	AWARD	63
		CARDINAL	6757	PERCENT	1252	MISC-ORG	313		
összesen		NORP	4489	PROD	1212	AGE	298	összesen	
<b>token</b>	<b>elem</b>	ORDINAL	3252	MEDIA	1063	DUR	271	<b>token</b>	<b>elem</b>
<b>1027218</b>	<b>46348</b>	LAW	3231	TIME	951	SMEDIA	268	<b>1038997</b>	<b>84641</b>

<sup>a</sup> A korpuszban néhány zenekar neve tévesen PER-nek volt annotálva.

<sup>b</sup> Az eredetileg ORG-nak annotált elemek egy része GPE lett az új változatban, illetve sok névből képzett melléknév is GPE annotációt kapott.

1. táblázat. Az eredeti és az új változat mérete és címkeeloszlása.

## 5. A továbblépés lehetséges irányjai

Az elmúlt időszakban az egymásba ágyazott entitások kezelésére is alkalmas modellek kifejlesztése lendületet kapott, és létrejött néhány nyílt forráskódú neurális névelemcímkező eszköz (pl. Wang és mtsai (2020), Shibuya és Hovy (2020)). Ugyanakkor ezeknek a modelleknek a pontossága az egymásba ágyazott entitásokat nem tartalmazó adathalmazokon betanítva és mérve egyelőre elmarad a legjobb teljesítményt nyújtó modellektől. A NameTag2 címkező esetében is azt

<sup>11</sup> [www.hvg.hu](http://www.hvg.hu)

találtuk, hogy kevésbé pontos annotációt generált, mint az OntoNotes 5 alapú modellek. Ezért a korpusz itt bemutatott változatában maradtunk az egymásba ágyazott entitásokat nem tartalmazó annotációnál, annak ellenére, hogy számos esetben megkönnyítené az annotációs döntések meghozatalát, ha megengednénk az egymásba ágyazott entitásokat, illetve információ-visszakeresési szempontból is sokkal motiváltabb lenne az ilyen annotáció.

## 6. Modellek és teljesítmény

A korpuszból leválasztott és kézzel kijavított tesztanyagon kiértékeljük a nyelvi transzfert alkalmazó modellek teljesítményét. A kiértékelést úgy is elvégeztük, hogy a címkekészletet visszánormalizáltuk azokra az elemekre, amelyek az eredeti modellben szerepelnek. Így értékeltük ki az OntoNotes 5 korpuszon alapuló Flair és DeepPavlov modelleket, valamint a cseh NameTag2 címkézőt. Emellett betanítottunk a HuggingFace Transformers könyvtár (Wolf és mtsai, 2020) felhasználásával egy a magyar huBERT kontextuális nyelvmodellen (Nemeskey, 2021) alapuló neurális tokencímkéző modellt az egyelőre nem gold standard minőségű korpuszrészén. Ezek teljesítményét a 2. táblázatban mutatjuk be.

változat	eredeti			az jav.			közös címkék			k. cím., az jav.		
	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>
CZ	15,82	11,39	13,25	15,89	11,44	13,30	64,57	52,92	58,16	64,63	52,97	58,22
DP	66,32	60,41	63,23	71,66	65,27	68,31	68,79	63,42	65,99	74,63	68,81	71,60
FL	74,81	70,73	72,71	80,59	76,19	78,33	77,68	74,34	75,97	<b>83,90</b>	<b>80,29</b>	<b>82,06</b>
NKA	<b>87,59</b>	<b>86,32</b>	<b>86,95</b>				<b>88,49</b>	<b>87,52</b>	<b>88,00</b>			
teszt	91,59	89,69	90,63									

2. táblázat. A modellek teljesítménye a tesztkorpuszon, CZ: NameTag2 cseh modell, DP: DeepPavlov OntoNotes/m-BERT, FL: Flair-OntoNotes-Large/XLM-RoBERTa, NKA: NerKor 1.41e/huBERT, teszt: a tesztkorpusz kézi javítás előtti pontossága.

A nyelvi transzfert alkalmazó modellek elég jól teljesítményt nyújtottak, de látható, hogy az azonos korpuszon betanított angol modellek közül az XLM-RoBERTa-alapú Flair modell sokkal jobban teljesített (kb. 10% F-mértékkel). Az „erősebb” nyelvmodellen alapuló Flair modell minden névelemtípus tekintetében jobb pontosságot és fedést produkált, mint a gyengébb modell. Ezeknek a modelleknek a teljesítményén nagyon sokat (5-6% F-mértéknyit) javított a névelők egyszerű reguláriskifejezés-mintákon alapuló automatikus javítása. A Flair modell végső változattal közös címkéken mért teljesítménye (tehát ha az általunk bevezetett MEDIA, SMEDIA, PROJ stb. címkéket figyelmen kívül hagyjuk) igazán jónak mondható.

A korpuszon betanított egynyelvű magyar nyelvmodellen alapuló modell azért a teljes végső tagkészleten mérve is szignifikánsan jobban teljesít, mint a legjobb

transzfer alapú modell csak a közös címkéken. A cseh modell látszólag gyenge teljesítményét részben magyarázza, hogy a végső annotációnknál részletesebb címkékészlettel dolgozik, ezért a valódi teljesítmény felméréséhez itt mindenképp szükségyszerű volt a címkék normalizálása. A teljesítménye azonban a címkenormalizálás után is messze elmarad a többi modellétől. A cseh tanítóanyag jóval kisebb, mint a többi modell esetében, valamint a bonyolultabb, beágyazott entitásokat is megengedő algoritmus is szerepet játszhat a gyengébb teljesítményben.

Az összehasonlíthatóság kedvéért csak a névelemeken (tehát a mennyiség-, és idő kifejezések, nyelvnevek és a NORP melléknevek nélkül) mérve a huBERT-alapú modell teljesítményét ( $F_1=90.42$ ), az hasonlóan adódott a korpusz eredeti változatán betanított hasonló (emBERT) modell Simon és mtsai (2022) által publikált teljesítményéhez ( $F_1=91.44$ ; bár a két méréshez használt tesztalalmaz nem azonos, tehát az eredmények nem teljesen összehasonlíthatóak). Ez arra utal, hogy egyes névelemosztályok (különösen a MISC) több alosztályra bontása (ráadásul alapvetően automatikus módszerekkel) nem okozott számottevő romlást a rendszer pontosságában. Hasonló hatást (illetve kifejezett teljesítményjavulást) figyeltünk meg korábban függőségi annotáció algoritmikus finomítása során (Tálas és Novák, 2019).

## 7. Összefoglalás

Cikkünkben a nagy és változatos műfajokat felölelő NYTK-NerKor korpusz annotációjának automatikus bővítésére irányuló kutatásunkat mutattuk be: az annotált elemek számát csaknem megdupláztuk, a megkülönböztetett osztályok számát meghétszereztük. Ehhez nyelvi transzfert alkalmaztunk, amely a mérések szerint hatékonyan működött, de a transzfermodellek teljesítménye jelentős eltéréseket mutatott. Az annotációt félautomatikus módszerekkel javítottuk. A korpusz 2,5%-át kézzel is ellenőriztük, ezt a részt használtuk tesztkorpusznak a transzfermodellek és a tanítóanyagon betanított huBERT-alapú modell kiértékeléséhez. Az utóbbi a jóval részletesebb névelem-osztályozás mellett az eredeti korpuszon mérthez hasonló teljesítményt mutatott.

## Köszönetnyilvánítás

Jelen kutatás a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással az FK 125217 és a PD 125216 számú projekt keretében az FK 17 és a PD 17 pályázati program valamint az IdomSoft Zrt.-vel való együttműködésben az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatásával az Infokommunikációs és Informatótechnológiai Nemzeti Laboratórium keretében valósult meg.

## Hivatkozások

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: FLAIR: An easy-to-use framework for state-of-the-art NLP. In: Procee-

- dings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 54–59. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://aclanthology.org/N19-4010>
- Benikova, D., Biemann, Ch., Reznicek, M.: NoSta-D named entity annotation for German: Guidelines and dataset. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 2524–2531. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014), [http://www.lrec-conf.org/proceedings/lrec2014/pdf/276\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf)
- Burtsev, M., Seliverstov, A., Airapetyan, R., Arkhipov, M., Baymurzina, D., Bushkov, N., Gureenkova, O., Khakhulin, T., Kuratov, Y., Kuznetsov, D., Litinsky, A., Logacheva, V., Lymar, A., Malykh, V., Petrov, M., Polulyakh, V., Pugachev, L., Sorokin, A., Vikhрева, M., Zaynutdinov, M.: DeepPavlov: Open-source library for dialogue systems. In: Proceedings of ACL 2018, System Demonstrations. pp. 122–127. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), <https://aclanthology.org/P18-4021>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. CoRR abs/1911.02116 (2019), <http://arxiv.org/abs/1911.02116>
- Kim, J., Ohta, T., Tateisi, Y., Tsujii, J.: GENIA corpus - a semantically annotated corpus for bio-textmining. In: Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia. pp. 180–182 (2003), [http://bioinformatics.oupjournals.org/cgi/content/abstract/19/suppl\\_1/i180?etoc](http://bioinformatics.oupjournals.org/cgi/content/abstract/19/suppl_1/i180?etoc)
- Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018), <https://aclanthology.org/D18-2012>
- Nemeskey, D.M.: Introducing huBERT. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021). pp. 3–14. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2021), [http://acta.bibl.u-szeged.hu/73353/1/msznykonf\\_017\\_003-014.pdf](http://acta.bibl.u-szeged.hu/73353/1/msznykonf_017_003-014.pdf)
- Novák, A., Novák, B.: Transfer-based enrichment of a Hungarian named entity dataset. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2021. pp. 1060–1067 (2021)
- Ševčíková, M., Žabokrtský, Z., Krůza, O.: Named entities in Czech: annotating data and developing NE tagger. In: Matoušek, V., Mautner, P. (szerk.) Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue. Lecture Notes in Computer Science, vol. 4629, pp. 188–195. Springer, Berlin / Heidelberg (2007)

- Shibuya, T., Hovy, E.: Nested named entity recognition via second-best sequence learning and decoding. *Transactions of the Association for Computational Linguistics* 8, 605–620 (2020), <https://aclanthology.org/2020.tacl-1.39>
- Simon, E., Nemeskey, D.M.: Automatically generated NE tagged corpora for English and Hungarian. In: *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*. pp. 38–46. Association for Computational Linguistics, Jeju, Korea (Jul 2012), <https://aclanthology.org/W12-4405>
- Simon, E., Vadász, N.: Introducing NYTK-NerKor, a gold standard Hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) *Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science*, vol. 12848, pp. 222–234. Springer (2021)
- Simon, E., Vadász, N., Lévai, D., Nemeskey, D., Orosz, Gy., Szántó, Zs.: Az NYTK-NerKor több szempontú kiértékelése. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2022). pp. 375–388. Szegedi Tudományegyetem, TTIK, Informatikai Intézet, Szeged (2022)
- Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.: A highly accurate named entity corpus for Hungarian. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA), Genoa, Italy (May 2006), [http://www.lrec-conf.org/proceedings/lrec2006/pdf/365\\_pdf.pdf](http://www.lrec-conf.org/proceedings/lrec2006/pdf/365_pdf.pdf)
- Taulé, M., Martí, M.A., Recasens, M.: AnCora: Multilevel annotated corpora for Catalan and Spanish. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco (May 2008), [http://www.lrec-conf.org/proceedings/lrec2008/pdf/35\\_paper.pdf](http://www.lrec-conf.org/proceedings/lrec2008/pdf/35_paper.pdf)
- Tálas, D., Novák, A.: Különböző függőségi elemzők teljesítményének vizsgálata magyar nyelven. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 345–354. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2019), [http://acta.bibl.u-szeged.hu/59097/1/msznykonf\\_015\\_345-354.pdf](http://acta.bibl.u-szeged.hu/59097/1/msznykonf_015_345-354.pdf)
- Vadász, N.: KorKorpusz: kézzel annotált, többretegű pilotkorpusz építése. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020). pp. 141–154. Szegedi Tudományegyetem, TTIK, Informatikai Intézet, Szeged (2020), [http://acta.bibl.u-szeged.hu/67674/1/msznykonf\\_016\\_141-154.pdf](http://acta.bibl.u-szeged.hu/67674/1/msznykonf_016_141-154.pdf)
- Wang, J., Shou, L., Chen, K., Chen, G.: Pyramid: A layered model for nested named entity recognition. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 5918–5928. Association for Computational Linguistics, Online (Jul 2020), <https://aclanthology.org/2020.acl-main.525>
- Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., Houston, A.: *OntoNotes Release 5.0* (2013), <https://hdl.handle.net/11272.1/AB2/MKJJ2R>

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), <https://aclanthology.org/2020.emnlp-demos.6>



## Az NYTK-NerKor több szempontú kiértékelése

Simon Eszter<sup>1</sup>, Vadász Noémi<sup>1</sup>, Lévai Dániel<sup>2,3</sup>, Nemeskey Dávid<sup>2,3</sup>,  
Orosz György<sup>4</sup>, Szántó Zsolt<sup>4</sup>

<sup>1</sup>Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.

<sup>2</sup>Eötvös Loránd Tudományegyetem Bölcsészettudományi Kar  
TI Digitális Bölcsészet Tanszék  
1088 Budapest, Múzeum krt. 6-8.

<sup>3</sup>Digitális Örökség Nemzeti Laboratórium  
1088 Budapest, Múzeum krt. 6-8.

<sup>4</sup>Szegedi Tudományegyetem Informatikai Intézet  
6725 Szeged, Tisza Lajos körút 103.

simon.eszterke@gmail.com, vadasz.noemi@nytud.hu  
levai.daniel@btk.elte.hu, nemeskey.david@btk.elte.hu  
gyorgy@orosz.link, szantozs@inf.u-szeged.hu

**Kivonat** Cikkünkben az NYTK-NerKor korpusz kiértékelését mutatjuk be több rendszer segítségével. Azt vizsgáljuk, hogy az egymillió tokenet tartalmazó, műfajilag heterogén, szabadon elérhető gold standard adathalmaz mennyire használható magyar nyelvű tulajdonnév-felismerő rendszerek fejlesztéséhez. A kiértékeléshez négy különböző rendszert használtunk: a CRFsuite-ot, a magyar spaCy-t, a Stanzát és az emBERT-et. Cikkünkben ismertetjük az egyes rendszerek által elért eredményeket, melyeket össze is hasonlítottunk. Az eredmények azt mutatják, hogy az NYTK-NerKor és a Szeged NER korpusz együttes használata még stabilabb modelleket eredményezhet, valamint hogy az NYTK-NerKoron tanítva a rendszerek nagyobb általánosító képességgel rendelkeznek, ami ahhoz kell, hogy egy azelőtt nem látott szövegben jól azonosítsák a nevekét.

**Kulcsszavak:** tulajdonnév-felismerés, kiértékelés, korpusz

### 1. Bevezetés

A felügyelt gépi tanuláson alapuló statisztikai és neurális rendszerek nagy mennyiségű gold standard adatot igényelnek. Ahhoz, hogy egy adathalmaz gold standard korpusznak minősüljön, több feltételnek is meg kell felelnie, úgymint reprezentativitásra kell törekednie, elég nagyoknak kell lennie gépi tanuló rendszerek tanításához és teszteléséhez, valamint kézzel hozzáadott pontos nyelvi annotációt kell tartalmaznia. Az ilyen korpuszok építése viszont sok időt és hozzáértést igényel, ezért a gold standard adathalmazokból kevés van, és rendkívül értékesek a természetesnyelv-feldolgozásban (Natural Language Processing, NLP). A magyar nyelvű tulajdonnév-felismerés (Named Entity Recognition, NER) terén is

hasonló volt a helyzet, mivel a meglévő névannotált korpuszok erősen domainspecifikusak (jellemzően csak híreket tartalmaznak), és korlátozottak méretükben, illetve hozzáférhetőségükben.

A jelenleg elérhető magyar nyelvű gold standard tulajdonnév-annotált korpuszok közül a legismertebb a Szeged NER korpusz (Szarvas és mtsai, 2006b), amely kizárólag gazdasági rövidhíreket tartalmaz, és összesen kb. 225 000 tokenből áll. A CoNLL2003 shared task (Tjong Kim Sang és De Meulder, 2003) annotációs sémáját és címkekészletét követi. A korpusz szövege a Szeged Treebankból (Csendes és mtsai, 2005) lett válogatva, annak egy alkorpusza, így annak a licencét örökíti tovább, vagyis csak kutatási célokra lehet használni.

A másik a Criminal NE Korpusz<sup>1</sup>, amely gazdasági bűncselekményekről szóló HVG-cikkekből áll, és kb. 560 000 tokent tartalmaz. Ez a korpusz a Magyar Nemzeti Szövegtár (Váradi, 2002) alkorpusza, ezért felhasználhatósága még inkább korlátozott. Ez is a CoNLL2003 címkekészletét követi, de azzal a különlegességgel, hogy az annotációnak két verziója van. Az egyik az ún. *tag-for-meaning*, a másik a *tag-for-tagging* elvet követi. Egyes nevek bizonyos kontextusokban metonimikusan viselkednek, ami számos érdekes kérdést vet fel már a korpusz címkézése során. Két megközelítés létezik ennek a jelenségnek a kezelésére. Az első szerint – ez a *tag-for-meaning* – a nevet az aktuális kontextusának megfelelően annotáljuk. Ebben az esetben abban a mondatban, hogy *Az esetek 90%-ában Brüsszel javára dönt ez a bíróság*, a ‘Brüsszel’ intézménynévként címkézendő, mivel itt egy jogi entitásként, egy cselekvő félként szerepel. A *tag-for-tagging* elv alapján ugyanez a név ugyanebben a mondatban földrajzi névként címkéződik, mivel az az elsődleges referenciája.

A fent leírt gold standard korpuszok mellett létezik egy silver standard korpusz is. A hunNERwiki korpusz (Simon és Nemeskey, 2012) automatikusan lett generálva a magyar Wikipédiából, ugyanazt az annotációs sémát követi, mint a Szeged NER korpusz, de több mint 19 millió tokenből áll. A Wikipédia licencét követve ez szabadon felhasználható CC-BY-SA 3.0 licenc alatt.

Az NYTK-NerKor korpusz (Simon és Vadász, 2021) a fentiekől több paraméterében is különbözik. Méretét tekintve egy nagyságrenddel nagyobb, mint az eddigiek, ugyanis 1 millió tokenből áll. Gold standard korpusz, vagyis a címkézés kézzel lett ellenőrizve. Kiegyensúlyozott válogatást nyújt többféle domainből: tartalmaz szépirodalmi, jogi és vegyes webes szövegeket, híreket, valamint Wikipédia cikkeket is. Egy kb. 200 000 tokennyi alkorpusz gold standard morfológiai címkézést is kapott, hogy a klasszikus statisztikai gépi tanuláson alapuló rendszerek morfológiai jegyei is biztosítva legyenek. Az adatformátum követi a nemzetközi sztenderdeket, ugyanis a széles körben ismert és alkalmazott CoNLL-U Plus<sup>2</sup> formátumban van, a névannotáció a CoNLL2002 (Tjong Kim Sang, 2002) címkézési szabványt követi, és a morfológiai információ a Universal Dependencies<sup>3</sup> v2 szófajkódjaival és jegy-érték párjaival van kódolva. Ezenfelül az NYTK-NerKor újdonsága, hogy CC-BY-SA 4.0 licenc alatt fel-

<sup>1</sup> <https://rgai.inf.u-szeged.hu/node/130>

<sup>2</sup> <https://universaldependencies.org/ext-format.html>

<sup>3</sup> <https://universaldependencies.org/>

használható bármilyen célra, és szabadon elérhető a GitHub repozitóriumából: <https://github.com/nytud/NYTK-NerKor>.

A META-NET Fehér könyvek sorozatának magyar nyelvről szóló kiadványa (Simon és mtsai, 2012) alapján a magyar az erőforrásokkal közepesen jól ellátott nyelvek közé tartozik, vagyis feltételezhetjük, hogy a helyzet a tulajdonnév-annotált korpuszok tekintetében is hasonló. Több olyan aggregátor weboldal is létezik, amely különféle nyelvekre elérhető erőforrásokat listáznak, mint például a CLARIN tudástára<sup>4</sup>. Ezt az oldalt áttanulmányozva sok névannotált korpuszt találunk a különféle európai nyelvekre, 46 000-tól 1 millió tokenig terjedő méretben, amibe az NYTK-NerKor kiválóan illeszkedik. Az erőforrásokkal leginkább ellátott angol nyelvre a legismertebb adathalmaz az OntoNotes 5.0<sup>5</sup>, amelynek az angol nyelvű része kb. 1,5 millió tokent tartalmaz.

Összefoglalva az NYTK-NerKor jelenleg a legnagyobb magyar gold standard tulajdonnév-annotált korpusz, amilyen azelőtt nem volt, annak ellenére, hogy igény lett volna rá. Ezt az is jól mutatja, hogy megjelenése után nem sokkal többen használatba is vették. A korpusz GitHub repozitóriumának kérdései alapján az ELTE RC2S2 kutatócsoportja<sup>6</sup> és a Stanford NLP Group<sup>7</sup> használják. Cikkünkben a korpusz alapos kiértékelését mutatjuk be több tulajdonnév-felismerő rendszert használva tanításra és tesztelésre. A 2. fejezetben először az NYTK-NerKor releváns tulajdonságait ismertetjük, majd a 3. fejezetben a kiértékelések eredményeit írjuk le. A cikket összegzés zárja a 4. fejezetben.

## 2. A korpusz ismertetése

Ebben a fejezetben az NYTK-NerKor korpusznak csak azon tulajdonságait ismertetjük, amelyek relevánsak a kiértékelés szempontjából. További részletekért a korpuszt bemutató cikkhez (Simon és Vadász, 2021) utaljuk az olvasót.

Az NYTK-NerKor korpusz 5 műfajból tartalmaz egyenletes szövegválogatást: szépirodalom, jogi szövegek, hírek, vegyes webes szövegek és Wikipédia. Az egymillió token egyenletesen oszlik el a műfajok között, vagyis minden műfaj kb. 200 000 tokent tartalmaz.

A korpusz fő annotációhalmaza a named entity (NE). A 2002-es és 2003-as CoNLL shared taskok sztenderd címkekészletét használja, ami 4 fő névkategóriát különít el: PER, ORG, LOC, MISC. A címkeprefixek tekintetében a CoNLL2002 annotációs formátumát, az ún. IOB2 formátumot követi, miszerint minden név első eleme 'B-' prefixet, míg minden nem első elem 'I-' prefixet kap. A nem neveket O betű jelöli.

A NE annotáció a teljes szöveget lefedi. Ez praktikus azt jelenti, hogy minden tokenhez tartozó cella ki van töltve. Minden ugyanolyan értékű tokennek számít, az írásjelek is. A folyó szöveg tokenekre bontásakor bizonyos tapadó

<sup>4</sup> <https://www.clarin.eu/resource-families/manually-annotated-cor-pora#Named%20Entity%20recognition>

<sup>5</sup> <https://catalog.ldc.upenn.edu/LDC2013T19>

<sup>6</sup> <https://rc2s2.elte.hu/>

<sup>7</sup> <https://nlp.stanford.edu/>

írásjeleket a tokenizáló leválaszt az előtte-utána álló szóról, így azok is önálló tokenné válnak. Minden, ami nem név, 0 annotációt kap, így az írásjelek is. A mondathatárt jelölő üres sorok viszont ténylegesen üres sorok.

Mivel a klasszikus gépi tanuló rendszerek számára a morfológiai információk hasznos jegyeket jelentenek, a korpusz egy részére morfológiai annotáció is került. A Szeged NER korpusz esetében a kb. 200 000 tokennyi tanítóanyag elégnak bizonyult 96% feletti F-mérték eléréséhez, ezért az NYTK-NerKor esetében is kb. ennyi szöveganyag kapott kézzel ellenőrzött egyértelműsített morfológiai elemzést.

A korpusz egyes alkorpuszai különféle előfeldolgozási lépéseken estek át, és került rájuk automatikus morfológiai és NE annotáció is. A szépirodalmi és jogi szövegek, valamint a hírek az `emtsv`-vel (Indig és mtsai, 2019) lettek elemezve, a mondatra bontástól a morfológiai egyértelműsítésig. A NE előelemzéshez szintén az `emtsv` volt használva, annak az `emBERT-NER`, illetve az `emNER` moduljai. A kézi annotáció így tehát az automatikus előannotáció ellenőrzésévé és javításává egyszerűsödött. A Wikipédia alkorpusz szövegei két forrásból származnak: egyrészt a KorKorpusz (Vadász, 2020) morfológiailag annotált szócikkeiből, másrészt a hunNERwiki korpuszból (Simon és Nemeskey, 2012). A KorKorpuszból érkezett szövegek eleve fel voltak dolgozva, csak NE előcímkézést igényeltek, amihez szintén az `emBERT-NER` lett használva. A hunNERwikiből érkező szövegek esetében sem volt szükség előfeldolgozásra, hiszen az ott szereplő szövegek eleve mondatokra és tokenekre voltak bontva. NE előcímkézésre sem volt szükség, hiszen a hunNERwiki korpusz tartalmaz silver standard névannotációt, ráadásul ugyanazokat a névkategóriákat használja, mint amiket az NYTK-NerKor. A webes szövegek forrása teljes egészében a Magyar Webkorpusz 2.0 (Nemeskey, 2020b), ami eleve elemezve volt az `emtsv`-vel, így csak NE előcímkézést igényelt, ami az `emBERT-NER`-rel készült.

A korpusz kétféle morfológiai annotációt is tartalmaz. Egyrészt az `emtsv emMorph` (Novák és mtsai, 2016) modulja által kiadott elemzést, másrészt a Universal Dependencies v2 szófajkódjait és morfoszintaktikai jegy-érték párpárjait.

A korpuszhoz készült egy hivatalos train–devel–test vágás is. Az egyes halmazok nagyjából aránya: 80%–10%–10%. A vágás törekszik a kiegyensúlyozottságra, vagyis minden műfaj, forrás és morfológiai annotáltság ugyanilyen arányban van képviseltetve. A vágásnál figyelembe lettek véve a dokumentumhatárok, vagyis csak egész fájlok kerültek az egyes halmazokba.

### 3. Kiértékelések

A kiértékeléshez négy különböző rendszert használtunk: a CRFsuite-ot (Okazaki, 2007), a HuSpaCy-t (Orosz és mtsai, 2022), a Stanzát (Qi és mtsai, 2020) és az `emBERT`-et (Nemeskey, 2020a). Az első rendszer klasszikus gépi tanulást valósít meg, míg az utóbbi három rendszer neurális architektúrát alkalmaz. A spaCy-nek és az `emBERT`-nek közös vonása, hogy már az NYTK-NerKor korpusz létrejötté előtt is volt magyar nyelvű tulajdonnév-felismerője, míg a Stanza fejlesztői csak az NYTK-NerKoron tanítva készítették magyar NER modult. Ez

utóbbi esetben a kiértékelést az indokolta, hogy képet kapjunk ennek az új rendszernek a teljesítményéről, míg az előbbi rendszerek esetében azt vártuk, hogy a nagyobb és heterogénebb korpuszon tanítva jobb teljesítményt fognak nyújtani, mint az eddig elérhető, de jóval kisebb és domainspecifikusabb korpuszokon. A CRFsuite-ot kimondottan baseline-ként használtuk, hogy összehasonlítsuk a neurális rendszerek teljesítményét egy klasszikus gépi tanuláson alapuló rendszerével.

Mindegyik rendszer esetében ugyanazokat a méréseket végeztük el: egyrészt tanítottuk és teszteltük az NYTK-NerKor hivatalos vágásán, ugyanezt megtettük a Szeged NER Korpuszon is, illetve ezeket keresztbe is mértük, továbbá a két korpuszt együtt is használtuk tanításhoz és teszteléshez. Ezenfelül az NYTK-NerKor egyes alkorpuszain is végeztünk méréseket.

Az NYTK-NerKornak van hivatalos vágása (lásd a 2. fejezetet) – ott azt használtuk. A Szeged NER korpusznak ugyan nincs hivatalos vágása, de a magyar NLP közösség hagyományosan azt a vágást használja tanításhoz–teszteléshez, amelyet Szarvas és mtsai (2006a) használtak a rendszerük építéséhez. Mivel a Szeged NER korpusz a Szeged Treebank egy alkorpusza, annak a morfológiai elemzését tartalmazta eredetileg, amit át kellett konvertálni az `emmorph` címke-készletére. Ezt a konvertált verziót használtuk minden kiértékelésnél.

Címke-készlet tekintetében a kiértékeléshez használt Szeged NER korpusz és az NYTK-NerKor némileg eltérnek. A fő névkegóriák (`PER`, `ORG`, `LOC`, `MISC`) megegyeznek, de a címkeprefixek terén van eltérés. A Szeged NER korpusz a `BIE1` címkeprefixálást követi, amely megkülönbözteti a kezdő (`B-`), a közbülső (`I-`), a záró (`E-`) és az egyelemű (`1-`) névelemeket. A Stanza és a `spaCy` is ezt használja, csak más névvel illeti: a Stanzánál ugyanez `BIOES` (`Beginning`, `Inside`, `Outside`, `Ending`, `Single`), a `spaCy`-nél `BILOU` (`Beginning`, `Inside`, `Last`, `Outside`, `Unit-length`). Ezzel szemben az NYTK-NerKor a `CoNLL2002-es IOB2` címkeprefixálási formátumot alkalmazza (lásd a 2. fejezetet). A két formátum információvesztés nélkül átjárható – a kiértékelésnél a korpuszokat természetesen átkonvertáltuk a kellő formátumra.

Az egyes rendszerek teljesítményét a 3.1., a 3.2., a 3.3. és a 3.4. fejezetekben ismertetjük, összehasonlító összegzést pedig a 3.5. fejezetben adunk. A rendszerek teljesítményét mindenhol a tulajdonnév-felismerésben szokásosan entitás szinten számoltuk, a szintén szokásos `F`-mértékkel, amelyet százalékos formában szerepeltettünk a táblázatokban. A tanítás fő paramétereit az 1. táblázat tartalmazza.

### 3.1. CRFsuite

A neurális rendszerekkel szemben baseline-ként egy manuálisan kinyert jellemzőkre építő CRF (Lafferty és mtsai, 2001) alapú megoldást választottunk. Ehhez a gyors implementációval rendelkező, C nyelven íródott CRFsuite könyvtárat használtuk fel.

Jellemzőként a szó kisbetűs változata mellett annak kettő és három hosszú szuffixe, valamint felszíni jellemzői (nagybetűs-e, nagybetűvel kezdődik-e, szám-e) voltak felhasználva, illetve ugyanezen tulajdonságok az azt megleelőző és rákö-

Rendszer	Architektúra	CPU/GPU	Tanítás		Tesztelés	
			Batch	Idő (min:s)	Batch	Idő (s:ms)
CRFsuite	CRF	Xeon 5218	1	1:25	1	1:18
emBERT	BERT	1db A100	10	40:20	16	15
HuSpaCy	CNN	1db A100	1024	120	1024	1:47
Stanza	Flair	1db A100	4096	360	4096	7:13

1. táblázat. A négy rendszer főbb tanítási paramétereit. A 'Tanítás' oszlop a teljes tanítás idejére vonatkozik, a 'Tesztelés' az NYTK-NerKor tesztkorpuszának egyszeri annotálására.

vetkező szóra. Tanításhoz az L-BFGS (Nocedal, 1980) algoritmust használtuk, 0,1-es L1 és L2 regularizációs paraméterekkel, és az iterációk számát 100-ban limitáltuk. Az így kapott eredmények a 2. táblázatban láthatók.

	NerKor	SzegedNER	Együtt
NerKor	75,22	67,81	73,65
SzegedNER	45,95	93,42	56,44
Együtt	75,12	92,89	79,04

2. táblázat. A CRFsuite teljesítménye különböző korpuszokon tanítva és kiértékelve. A sor mutatja a tanító-, az oszlop a tesztkorpuszt.

### 3.2. HuSpaCy

A természetesnyelv-feldolgozó keretrendszerek közül az egyik legelterjedtebb a spaCy. Népszerűségét a relatív korai indulásának, könnyű felhasználhatóságának és erőforrásbarát felépítésének köszönheti. Jelenleg 64 nyelvet támogat alap szinten, ami elsősorban nyelvspecifikus tokenizálást és stopszósűrűst jelent. Ezen belül 19 nyelven érhető el hivatalos modellek olyan magasabb szintű feladatokra, mint a szófaji egyértelműsítés, szintaktikai elemzés vagy esetünkben a tulajdonnév-felismerés. A magyar nem tartozik ezek közé, ugyanis hivatalos modellek nem léteznek rá, csak külső fejlesztések, amelyek között viszont elérhető a spaCy 3. főverziójához egy tulajdonnév-felismerő modul is. Mi a jelen kísérletekhez ezt az új spaCy-re épülő eszközt, a HuSpaCy-t használtuk fel (Orosz és mtsai, 2022).

A spaCy egy moduláris mély neuronhálós architektúrát alkalmaz, aminek a legalsó szintjén egy szóbeágyazási réteg található. A mi esetünkben ez a réteg két részre osztható: egy 300 dimenziós előre tanított szóbeágyazásra, ami a

CBOV (Mikolov és mtsai, 2013) algoritmus segítségével a Magyar Webkorpuszon (Halácsy és mtsai, 2004) és a magyar Wikipédián lett tanítva; továbbá egy 256 dimenziós, szóalakokra építő alrétetre, amely a szó mellett annak prefixét, szuffixét és alakját is kódolja 64–64 dimenzióban. Egy négyrétegű CNN-alapú encoder épül a beágyazások fölé. A hálózat legtetetjén pedig a névcímkek prediktálásáért egy átmenetalapú elemző felel. Bár a spaCy képes beolvasni mind az IOB2, mind a BILUO formátumokat, a háttérben a BILUO címkerendszert alkalmazza.

A 3. táblázat mutatja a HuSpaCy teljesítményét a különböző korpuszokon tanítva és tesztelve.

	NerKor	SzegedNER	Együtt
NerKor	80,75	79,13	80,39
SzegedNER	58,80	95,31	66,61
Együtt	80,59	93,86	83,46

3. táblázat. A HuSpaCy teljesítménye különböző korpuszokon tanítva és kiértékelve. A sor mutatja a tanító-, az oszlop a tesztkorpuszt.

### 3.3. Stanza

A Stanza a Stanford NLP Group egyik Python-alapú szövegfeldolgozó eszköze. A teljes Stanza pipeline neurális hálókön alapul. 66 nyelvet támogat jelen pillanatban, de könnyen hozzá lehet adni újabb nyelveket is, ugyanis teljesen a Universal Dependencies formátumán alapul minden eszköze.

A Stanza NER modulja kontextualizált sztringreprezentáción alapuló szekvenciátaggert (Akbik és mtsai, 2018) használ. Egy előre feltanított karakterszintű LSTM-modell (Hochreiter és Schmidhuber, 1997) már be van építve a Stanzába, amit a Ginter és mtsai (2017) adathalmazon feltanítottak, és címkézéskor konkatenáljuk a reprezentációkat szavanként mindkét irányból az előre feltanított szóbeágyazást használva. A reprezentáció ezután egy Bi-LSTM szekvenciátaggerbe kerül be, aminek a kimenetét egy CRF-alapú (Sutton és McCallum, 2007) dekóder alakítja névelemcímkeké.

Mivel a Stanza minden nyelvhez ugyanazt az architektúrát használja, egyszerűen lehet a tanítási folyamatot elindítani. A fejlesztők adnak receptet a NER modellek tanításához – mi is az előre megadott receptet használtuk. A modellek alapértelmezett bemeneti formátuma JSON, amiben egy-egy szóhoz meg van adva a *text* és a *ner* mező. A Stanza belső címkeprefixálási formátuma a BIOES, amire a konvertálást automatikusan el lehet végezni egy beépített függvénnyel<sup>8</sup> a `stanza` Python csomagból.

<sup>8</sup> `stanza.utils.datasets.ner.prepare_ner_file.process_dataset`

A Bi-LSTM rétegek számát és méretét lehet változtatni; adathalmazonként 3 modellt tanítottunk: egyet az alapbeállítással, azaz 1 Bi-LSTM réteggel, mely 256 egységből áll, egyet 2 réteggel és rétegenként 512 egységgel, és egyet 4 réteggel, szintén 512 egységgel rétegenként. A tanítást NVIDIA A100-as GPU-kon végeztük. A modellek kezdetben 1.0-es tanulási rátával tanultak, a minimum tanulási ráta 0.01 volt, amint ezt az ütemezővel elérték, leállt a tanulás. A tanítási folyamatok 6-24 órát vettek igénybe, mérettől függően. Az eredményeket a 4. táblázat tartalmazza.

	Méret	NerKor	SzegedNER	Együtt
NerKor	1*256	80,53	75,23	79,25
NerKor	2*512	79,42	73,06	77,90
NerKor	4*512	79,25	72,84	77,66
SzegedNER	1*256	49,18	91,78	60,57
SzegedNER	2*512	44,54	89,52	57,25
SzegedNER	4*512	50,85	90,32	61,15
Együtt	1*256	<b>80,66</b>	<b>92,90</b>	<b>83,75</b>
Együtt	2*512	80,07	91,65	82,97
Együtt	4*512	79,55	90,62	82,27

4. táblázat. A Stanza teljesítménye különböző korpuszokon tanítva és kiértékelve. A sor mutatja a tanító-, az oszlop a tesztkorpust.

Jól láthatóan a modellek által elért F-mérték független a modell méretétől. Erre a legvalószínűbb magyarázat az, hogy egyrészt a Stanza viszonylag egyszerű architektúrát használ osztályozásra, másrészt pedig a prediktált címke legfeljebb olyan jó, mint a beérkező reprezentáció.

### 3.4. emBERT

Az **emBERT** az **emtsv** egyik modulja, amely lehetővé teszi BERT (Devlin és mtsai, 2019) alapú tokenszintű osztályozók integrálását a szövegfeldolgozó láncba. A modul maga nyelvfüggetlen, azonban az **emtsv** részeként csak magyar tulajdonnév- és főnévcsoport-felismerésre lett feltanítva. Magyar nyelvre mindkét feladaton az **emBERT** számít a legjobb teljesítményű rendszernek (Nemeskey, 2021).

Az **emBERT** tanításkor egy kész BERT modellt vesz alapul, aminek a kimenetére egy softmax osztályozót köt. A tanítás folyamán a két komponenst együtt finomhangolja. Mi alapmodellnek a **huBERT**-et választottuk, amely a Magyar Webkorpusz 2.0-n lett előtanítva, és jelentősen jobb eredményt lehet vele elérni, mint a többnyelvű BERT-tel (Nemeskey, 2021). A jósolt címkeszekvencia konzisztenciáját egy Viterbi-algoritmus biztosítja, ami csak az érvényes címkeátmeneteket engedélyezi. Az **emBERT** mind az IOB2, mind a BIOES formátumot támogatja. Az



NYTK-NerKoron belüli méréseket a korpusz natív IOB2 formátumán végeztük. A korpuszok közötti keresztmérésekhez az NYTK-NerKort BIE1 formátumba konvertáltuk, hogy a címkekészlete megegyezzen a Szeged NER-ével.

Hogy teljesebb képet kapjunk a korpuszról, többféle kiértékelést csináltunk. Először a teljes tanítóanyagot tanítottunk, majd az egyes műfajoknak megfelelő teszhalmazokon teszteltünk. Ennek eredményét mutatja az 5. táblázat. Másodszor az egyes műfajokhoz tartozó alkörpuszok tanítóhalmazán tanítottunk és teszhalmazán teszteltünk – ennek eredményeit a 6. táblázat mutatja. A tanításhoz egy NVIDIA A100 GPU-t használtunk. Minden modell négy iterációt tanult; a futási idő az alkörpuszokon 5–12, a teljes korpuszon 40 perc volt.

	fiction	legal	news	web	wikipedia	teljes
LOC	90,32	86,62	92,01	89,72	96,11	93,42
MISC	52,94	85,46	80,54	68,34	76,83	76,57
ORG	62,50	95,87	87,89	82,30	86,80	90,72
PER	97,25	95,24	98,06	87,08	95,99	96,14
átlag	93,88	93,50	91,13	80,88	92,85	91,44

5. táblázat. A teljes NYTK-NerKoron tanított **emBERT** modell teljesítménye a különböző alkörpuszokon.

	fiction	legal	news	web	wikipedia
LOC	90,67	84,97	89,83	85,97	95,96
MISC	51,61	83,54	81,85	59,39	82,20
ORG	57,14	94,89	87,40	80,99	87,10
PER	96,74	100,00	97,88	88,64	96,00
átlag	93,29	92,40	90,64	77,57	93,37

6. táblázat. Az **emBERT** teljesítménye a különböző alkörpuszokon tanítva és tesztelve.

A táblázatok számaiból több következtetés is levonható. Az egyik legszembe-tűnőbb, hogy a MISC kategória felismerése a legnehezebb. Ez indokolható egyfelől a kategória heterogenitásával, másfelől a többenél alacsonyabb előfordulási gyakorisággal: a teljes tanítóadatban összesen 4 604 MISC típusú entitás található, ami kevesebb, mint a fele a második legritkább entitástípus, az ORG számának (9 657). Szintén a gyakoriság (jobban mondva annak hiánya) magyarázhatja a

szépirodalmi alkorpuszban a MISC és az ORG kiugróan alacsony, 50-60%-os F-mértékét. Itt a két entitástípus 134-szer, illetve 112-szer fordul elő, míg LOC típusú névelemből 695, PER-ből pedig 3 674 van.

Az alkorpuszok között a legrosszabb átlag F-mértéket a webes szövegeken mértük. Ez egyrészt meglepő lehet annak fényében, hogy a huBERT a Magyar Webkorpusz 2.0-n készült, másrészt bizonyítja a BERT architektúra általánosító képességét. Ez az eredmény feltehetően annak köszönhető, hogy a másik négy alkorpuszsal ellentétben ezek a szövegek nem szerkesztettek, és a minőségük hullámzó. Az egyes kategóriák közül itt is negatív irányba lóg ki a MISC, annak ellenére, hogy a webes alkorpuszon belül ez a második leggyakoribb entitástípus. A rendszer a Wikipédia alkorpuszon teljesített a legjobban, amit az magyarázhat, hogy az enciklopédia szövegek meglehetősen kötött nyelvezetűek.

Az NYTK-NerKor és a Szeged NER korpusz összehasonlítása céljából keresztméréseket is végeztünk a két korpusz, illetve az uniójuk között. A 7. táblázat mutatja az eredményeket. Az átlót megvizsgálva több dolog is feltűnik. Az NYTK-NerKor esetén elért F-mérték 0,65 százalékponttal meghaladja a 6. táblázatban lévőt, ami arra utal, hogy a BIE1 címkekészleten könnyebb tanulni. Ez valószínűleg annak köszönhető, hogy a részletesebb címkekészlettel a rendszer könnyebben meg tudja tanulni az egyes névelemekre jellemző tulajdonságokat. A Szeged NER-en mért érték hibahatáron belül van a Nemeskey (2021)-ben publikált értékhez (97,62) képest, ami felfogható egy „sanity check”-nek. Végül az unió F-mértéke nagyjából a két érték relatív korpuszmérettel súlyozott átlaga körül van.

	NerKor	SzegedNER	Együtt
NerKor	92,09	91,61	91,97
SzegedNER	81,01	97,40	84,39
Együtt	91,84	97,25	92,99

7. táblázat. Az emBERT teljesítménye különböző korpuszokon tanítva és kiértékelve. A sor mutatja a tanító-, az oszlop a tesztkorpuszt.

A korpuszok közötti keresztmérések az előre sejthető eredménnyel zárultak. Bár minden modell a saját korpuszán teljesít a legjobban, az NYTK-NerKor viszonylag jó pontszámot ért el a Szeged NER-en, míg fordítva ez nem áll. Az együtt tanított modell pedig, a két korpuszon kiértékelve, 0,15–0,25 százalékpontonra megközelíti azok saját modelljeit, mutatva, hogy az NYTK-NerKor és a Szeged NER korpusz együttes használata még stabilabb modelleket eredményezhet.

### 3.5. Diskusszió

A könnyebb összehasonlíthatóság kedvéért egy közös táblázatba rendeztük az egyes rendszerek által elért összehasonlítható eredményeket. A 8. táblázat sorában a tanító–teszt korpuszok, míg az oszlopaiban az egyes rendszerek szerepelnek. A Stanza esetében az alapbeállítással (1\*256 egység) elért eredményeket másoltuk ide, mert azok bizonyultak a legjobbnak.

	CRFsuite	spaCy	Stanza	emBERT
NerKor–NerKor	75,22	80,75	80,53	92,09
NerKor–SzegedNER	67,81	79,13	75,23	91,61
NerKor–Együtt	73,65	80,39	79,25	91,97
SzegedNER–NerKor	45,95	58,80	49,18	81,01
SzegedNER–SzegedNER	93,42	95,31	91,78	97,40
SzegedNER–Együtt	56,44	66,61	60,57	84,39
Együtt–NerKor	75,12	80,59	80,66	91,84
Együtt–SzegedNER	92,89	93,86	92,90	97,25
Együtt–Együtt	79,04	83,46	83,75	92,99

8. táblázat. Az egyes rendszerek teljesítménye az egyes korpuszokon.

A rendszerek összehasonlításából minden mérés esetében az **emBERT** kerül ki győztesen. Ez feltehetően köszönhető egyrészt a modell nagyobb kapacitásának (300 helyett 768 dimenziós beágyazás, a Stanza és a spaCy 2-4 rétege helyett 12, egyenként 3072 dimenzióban stb.), másrészt annak, hogy az attention modell az LSTM-mel és a CNN-nel szemben minden egyes tokennél az egész mondatra közvetlen rálátást biztosít. Az **emBERT**-től elmaradva, de szinte minden értékben a spaCy a második legjobb, a Stanza a harmadik, míg a CRFsuite produkálta a legalacsonyabb F-mértékeket. Ez rímel az angol tulajdonnév-felismerés tapasztalataira: az NYTK-NerKorral méretben nagyjából összevethető OntoNotes 5.0 eredménylistáját is egy BERT modell vezeti 92%-kal<sup>9</sup>, míg a spaCy 85%<sup>10</sup>, a Stanza pedig 89% körül teljesít (Qi és mtsai, 2020).

A korpuszok közötti mérések esetében az eredmények rendre az elvárásoknak megfelelően alakultak, vagyis a saját korpuszon való tanítás és tesztelés adta a legjobb eredményeket, és a korpuszok közötti keresztmérések adták a legrosszabbakat. A legalacsonyabb teljesítményt akkor kaptuk, amikor a Szeged NER korpuszon tanítottunk, és az NYTK-NerKoron teszteltünk, ami azzal magyarázható, hogy a Szeged NER korpusz kicsi és domainspecifikus – nem igazán ad lehetőséget arra, hogy a rendszer más műfajú szövegekre is általánosítson.

Bár a spaCy és a Stanza által elért eredmények az **emBERT** teljesítményéhez és az eddigi magyar NER eredményekhez hasonlítva alacsonynak tűnhetnek, de

<sup>9</sup> <https://paperswithcode.com/sota/named-entity-recognition-ner-on-ontonotes-v5>

<sup>10</sup> <https://spacy.io/models/en>

igazából hozzák a többi nyelvre is szokásosan elért átlagot. Az eddigi magas F-mértékek elsősorban annak voltak köszönhetőek, hogy kicsi és specifikus korpuszon lettek tanítva és kiértékelve a rendszerek, amelyen könnyebb magas pontszámokat elérni, mint egy nagyobb, de heterogén szövegen. Az NYTK-NerKoron elért alacsonyabb számok így tehát nem azt jelentik, hogy alacsonyabb lenne a korpusz minősége vagy hogy rosszabb lenne az adott rendszerek teljesítménye, hanem, hogy jobb általánosító képességgel rendelkeznek azelőtt nem látott szövegeken is.

#### 4. Összegzés

Cikkünkben az NYTK-NerKor korpusz alapos kiértékelését adtuk. A magyar tulajdonnév-felismerésben eddig elérhető gold standard korpuszok közös tulajdonsága, hogy kicsik és domainspecifikusak, ezért a domainen kívüli szövegek elemzéséhez kevésbé voltak jól használhatóak a rajtuk tanított rendszerek. Ugyanezen korpuszok tesztalmazán végezve a kiértékelést torz képet kaphattunk a rendszerünk teljesítményéről, hiszen a rendszer valószínűleg nem rendelkezett azzal az általánosító képességgel, ami ahhoz kell, hogy egy azelőtt nem látott szövegben jól azonosítsa a neveket. Az NYTK-NerKor kellően nagy és heterogén ahhoz, hogy ezt kiküszöbölje, amit a méréseink is alátámasztanak.

#### Hivatkozások

- Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1638–1649. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018)
- Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference, TSD 2005. pp. 123–131. Springer (2005)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (6 2019)
- Ginter, F., Hajič, J., Luotolahti, J., Straka, M., Zeman, D.: CoNLL 2017 shared task - automatically annotated raw texts and word embeddings (2017), <http://hdl.handle.net/11234/1-1989>, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
- Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of the 4th international conference on Language Resources and Evaluation (LREC2004) (2004)

- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* 9, 1735–80 (12 1997)
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` — egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019)
- Lafferty, J.D., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. pp. 282–289 (2001)
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
- Nemeskey, D.M.: Egy `emBERT` próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2020). pp. 409–418. Szeged (2020a)
- Nemeskey, D.M.: Natural Language Processing methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020b)
- Nemeskey, D.M.: Introducing `huBERT`. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). pp. 3–14. Szeged (2021)
- Nocedal, J.: Updating quasi-Newton matrices with limited storage. *Mathematics of computation* 35(151), 773–782 (1980)
- Novák, A., Siklósi, B., Oravecz, Cs.: A new integrated open-source morphological analyzer for Hungarian. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (szerk.) *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France (May 2016)
- Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) (2007), <http://www.chokkan.org/software/crfsuite/>
- Orosz, Gy., Szántó, Zs., Berkecz, P., Szabó, G., Farkas, R.: `HuSpaCy`: an industrial-strength Hungarian natural language processing toolkit. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (2020)
- Simon, E., Lendvai, P., Németh, G., Olaszy, G., Vicsi, K.: A magyar nyelv a digitális korban – The Hungarian Language in the Digital Age. Georg Rehm and Hans Uszkoreit (Series Editors): *META-NET White Paper Series*, Springer (2012)
- Simon, E., Nemeskey, D.M.: Automatically generated NE tagged corpora for English and Hungarian. In: *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*. pp. 38–46. Association for Computational Linguistics, Jeju, Korea (July 2012)
- Simon, E., Vadász, N.: Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) *Text, Speech, and Dialogue - 24th International Conference, TSD*

- 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)
- Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning (01 2007)
- Szarvas, Gy., Farkas, R., Kocsor, A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: Proceedings of Discovery Science 2006. pp. 267–278. Springer Verlag (2006a)
- Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.: A highly accurate Named Entity corpus for Hungarian. In: Electronic Proceedings of the 5th International Conference on Language Resources and Evaluation (May 2006b)
- Tjong Kim Sang, E.F.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Roth, D., van den Bosch, A. (szerk.) Proceedings of CoNLL-2002. pp. 155–158. Taipei, Taiwan (2002)
- Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Daelemans, W., Osborne, M. (szerk.) Proceedings of CoNLL-2003. Edmonton, Canada (2003)
- Vadász, N.: KorKorpusz: kézzel annotált, többrétegű pilotkorpusz építése. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 141–154. Szegedi Tudományegyetem (January 2020)
- Várad, T.: The Hungarian National Corpus. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002). pp. 385–389. European Language Resources Association, Las Palmas de Gran Canaria (2002)

## OCR-hibák javítása neurális technológiák segítségével

Laki László János<sup>1,2</sup>, Kőrös Ádám<sup>1</sup>, Ligeti-Nagy Noémi<sup>1,2</sup>, Nyéki Bence<sup>1</sup>,  
Vadász Noémi<sup>1</sup>, Yang Zijian Győző<sup>1,2</sup>, Váradi Tamás<sup>1</sup>

<sup>1</sup>Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.  
vezeteknev.keresztnev@nytud.hu

<sup>2</sup>MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport  
1083 Budapest, Práter utca 50/a  
vezeteknev.keresztnev@itk.ppke.hu

**Kivonat** Munkánk során párhuzamos optikai szövegfelismerővel digitalizált (OCR) szövegeken végeztünk utólagos korrektúrát. Kutatásunkban különböző OCR-hibákat tartalmazó szövegeket detektáló és javító modellt implementáltunk, valamint saját *Silver standard* párhuzamos korpuszt építettünk. Eredményeink azt mutatják, hogy abban az esetben, amikor kizárólag OCR-hibák javítását tűzzük ki célul, modelljeink a Context-based Character Correction (CCC) detekciós modellel való kombinációval a leghatékonyabbak. A létrehozott enkóder-dekóder alapú javító módszereink az OCR-hibák mellett nagy pontossággal javítják a szövegekben található koherenciahibákat (pl.: oldalszámok, elválasztások). Kutatásunk folytatásaként tervezzük a modellek tanítását és tesztelését a *Gold standard* korpuszon is.

**Kulcsszavak:** OCR-hiba detektálás, OCR-hiba javítás, OCR-hiba korpuszépités

### 1. Bevezetés

Napjainkban a mélyneurális-hálózatok sok más tudományterület mellett a nyelvtechnológia területén is felülmúlták az addigi piacvezető rendszerek teljesítményét. Az előtanított nyelvmodellek építésének sarokpontja a nagy mennyiségű és jó minőségű, egy- vagy többnyelvű tanítóanyag megléte. Ilyen korpuszok nem csak az internetes weblapok letöltéséből gyűjthetők, hanem kézenfekvő megoldásként kínálkozik a nyomtatott dokumentumok felhasználása is. Számptalan könyv, folyóirat, újság stb. áll rendelkezésünkre nyomtatott, de nem digitalizált formában. A nyomtatott művek digitalizálása sok emberi erőforrást igényel. Ezen feladatok támogatására kiválóan alkalmas az optikai szövegfelismerők (Optical Character Recognition – OCR) használata. Sajnálatos módon az OCR-technológiáknak vannak korlátai is, és különféle, változatos hibát generálnak a digitalizálás során.

Magyar vonatkozásban az Arcanum Adatbázis Kiadó<sup>1</sup> egyik célkitűzése a kulturális tartalmak nagy tömegű digitalizálása, adatbázisokba rendezése és a szövegek elérhetővé tétele online szolgáltatás révén. Az NYTK és az Arcanum Adatbázis Kiadó közötti együttműködés keretében végzett kutatásunk során különböző módszereket próbáltunk ki a Kiadó OCR-ezett szövegeinek javítására.

## 2. Kapcsolódó irodalom

Az OCR-ezett szövegek javításának jelentőségét mutatja, hogy az ICDAR konferencia<sup>2</sup> 2017 óta<sup>3</sup> külön versenyt rendez az OCR-hibák javítására. Az évek során a nyelvtechnológia fejlődésével az OCR-javító algoritmusok is szorosan követték az aktuális legjobb eredményeket elért módszereket.

A 2019-es év győztes OCR-javító módszere, a CCC (Context-based Character Correction, Rigaud és mtsai, 2019) egy BERT (Devlin és mtsai, 2019a) modellel kiegészített konvolúciós hálózaton alapuló detekciós, illetve egy kétirányú LSTM (Long short-term memory) rekurrens neurális hálózaton (figyelmi mechanizmussal) alapuló, enkóder-dekóder architektúrájú korrekciós részből áll.

Nguyen és mtsai (2020) a CCC módszert dolgozták át úgy, hogy a detekciós modellhez egy NER (Name Entity Recognition) modellt adaptáltak, míg a korrekciós modellhez nem rekurrens hálózatot használtak, hanem egy neurális gépi fordító rendszert, az OpenNMT keretrendszert (Klein és mtsai, 2017).

Schaefer és Neudecker (2020) szintén egy kétlépcsős módszert alkalmaztak, ahol az első egy detekciós, a második egy korrekciós modell. A detekciós modellük egy kétirányú LSTM hálózaton alapul, ami minden következő karakterre megjósolja, hogy az hibás vagy sem. A korrekciós modell a kutatásukban szintén egy LSTM alapú gépifordítóra épül.

Duong és mtsai (2021) kutatásukban transzformer (Vaswani és mtsai, 2017) alapú gépi fordítással oldották meg a problémát, detekciós modul nélkül. Ehhez viszont olyan modellt kellett tanítaniuk, amelyben egyaránt szerepelnek helyes és OCR-hibás szövegrészek, hogy a modell felügyelet nélkül megtanuljon mindent magától, mind a helyes, mind a hibás részeket. Ehhez automatikus módszerekkel saját, OCR-hibákat tartalmazó tanítókorpuszt hoztak létre. Kétféle módszert alkalmaztak az automatikus OCR-hibák generálására. Első módszerükben összegyűjtötték a lehetséges OCR-hibákat és véletlen módon belegenerálták azokat a helyes szövegbe. Második módszerükkel egy rekurrens neurális hálózaton alapuló gépi fordító modellt tanítottak be egy párhuzamos OCR-hibás korpusz segítségével. A bemenet egy hibátlan szöveg, a kimenet az OCR-hibás párja. Az így létrejött korpuszokkal tanítottak OCR-javító gépifordító modelleket.

Mei és mtsai (2016) foglalkozott azzal az eshetőséggel, ha több OCR-javító modell is rendelkezésre áll, melyek egyenként állítanak jelölteket az adott OCR-hiba javítására. A különböző modellek különböző jelöltjei közti hatékony választásra statisztikai eljárást dolgoztak ki, mely során a jelölteket OCR-tulajdonságaik

<sup>1</sup> <https://www.arcanum.com>

<sup>2</sup> <https://www.icdar.org>

<sup>3</sup> <https://sites.google.com/view/icdar2017-postcorrectionocr>



(Levenshtein távolság, legnagyobb közös szekvencia, gyakoriság, lexikon ellenőrzés stb.) alapján pontoszták, majd a kiválasztási feladatot regressziós modelleként felfogva súlyozták a jelöltek közötti valószínűségi sorrendet.

Kutatásunk során kipróbáltuk a kétlépcsős (detekció + korrekció) és az egylépcsős gépi fordítással történő megközelítést egyaránt, valamint saját párhuzamos korpuszt is építettünk.

### 3. Korpuszépítés

Az OCR javításához egy párhuzamos tanítókorpuszt építettünk, ahol a forrásnyelvi oldal egy OCR-hibás szöveg, míg a célnyelvi szöveg annak hibátlan párja. Ehhez olyan nagyobb mennyiségű szöveget kellett választanunk, ami elérhető mind elektronikus (helyes), mind OCR-ezett (hibás) változatban. Ezt követően ezeket manuálisan/fél-automatizálva annotáltuk, majd később annotátorokkal javítottuk. Az annotált korpusz lett a *Silver standard* korpuszunk, míg az annotátorokkal kijavított hibátlan változat a *Gold standard*.

A *Silver és Gold standard* korpusz építéséhez Jókai Mór<sup>4</sup>, illetve Mikszáth<sup>5</sup> (JiM) összes műveit használtuk fel. Választásunk azért esett rájuk, mert ezeknek elérhető a kiadó által elektronikusan közzétett (nem OCR) változata is.

Azonban a probléma az volt, hogy egyetlen fájlban volt az összes Jókai-mű és egyetlen fájlban az összes Mikszáth-mű. Ezért első feladatként a műveket össze kellett párosítani az Arcanum fájlokkal. Szerencsére a fájlokban a művek az Arcanum Kézíkönyvtár<sup>6</sup> oldalán található sorrendben voltak, ezért az adott sorrendben a címek alapján írtunk egy szkriptet, amely a címek mentén közel tökéletesen különválasztotta a műveket külön-külön fájlokba. A Mikszáth-művek szétválasztásánál voltak kezelendő esetek, mivel a fájlban a címek és az alcímek külön sorban voltak, míg az adott sorrendlistában ezek egy sorban álltak, ezért ezeket kézzel korrigálni kellett, és az alcímeket a címekkel egy sorba kellett tenni.

#### 3.1. Párhuzamos korpusz építése

Jókai és Mikszáth összes művének külön fájlba való bontása után következett a párhuzamos korpusz építése. A feladat az volt, hogy az OCR-ezett fájlok között megtaláljuk ugyanezeket a műveket, majd fájlszinten hozzájuk rendeljük. A munkát nehezítette, hogy a digitális kiadás csupán a szövegtörzset tartalmazta, míg az OCR-ezett dokumentum a teljes könyvet, beleértve a címoldalakat, tartalomjegyzékeket, valamint a dokumentum végi jegyzeteket. További nehézség volt, hogy egy OCR-ezett kötetben található könyvek vagy novellák nem mindegyike volt megtalálható a digitális kiadásban, valamint ami szerepelt benne, az sokszor nem azonos sorrendben. Ezen túl a JiM művek nem csak egy kiadó fájljaiban voltak, ezért a megtalálásukat egy Sketch engine motorját használó indexált

<sup>4</sup> <https://www.arcanum.com/hu/online-kiadvanyok/Jokai-jokai-mor-osszes-muvei-1>

<sup>5</sup> <https://www.arcanum.com/hu/online-kiadvanyok/Mikszath-mikszath-osszes-muve-2A85B>

<sup>6</sup> <https://www.arcanum.com/hu/online-kiadvanyok>

keresőbe töltöttük, és ennek a segítségével támogattuk. A fájlok párosítását kézi munkával végeztük. Ezzel előállt a művek szintjén a párhuzamos korpusz.

Következő feladatként ezeket a műveket további szegmensekre bontottuk, és kisebb egységű párhuzamosítást végeztünk rajtuk. Ehhez a feladathoz kétféle módszert is alkalmaztunk. Az egyik a mondatszintű szegmentálás (mondat), a másik egy 100 karakteres (100 karakter) egységekre bontott, gördülőablakos szegmentálás.

A művek kisebb egységű párhuzamosítása előtt a szövegeket előfeldolgozás alá vetettük. Első lépésként NFKC normalizálást<sup>7</sup> végeztünk, majd szólista alapján egyesítettük az elválasztott szavakat: ha kötőjel és egy vagy több szóköz választott el két betűfüzért, és két füzér egyesítése a szólistában szereplő szót adott, akkor az elválasztó karaktereket töröltük. Mindenhol töröltük továbbá a feltételes kötőjeleket, a sor elején és szóközök között álló kötőjeleket (U+2010) és a nagyköötőjeleket (U+2013) egységesen az Unicode szerinti U+2014 kódú nagyköötőjelre cseréltük le. Több egymást követő szóközt egyetlen szóközre cseréltünk, különböző idézőjel-karaktereket (U+201C, U+201D, U+201E) pedig az U+0022 kódú karakterre.

A fenti előfeldolgozó lépésekkel kezeltük mind a nyers OCR, mind a *Silver standard* korpusz alapjául szolgáló szövegeket. A mondatszintű tokenizáláshoz a *quntoken*<sup>8</sup> eszközt használtuk. Ennek alternatívájaként a szövegeket 100 karakter hosszú egységekre szegmentáltuk gördülőablakkal, melyet legalább 25 karakteres lépésekkel csúsztattunk: a lépést 25 karaktertől kezdve addig növeltük, amíg az ablak új szó elejére nem ért. Miután a szövegeket mondattokenizálással, illetve gördülőablakkal szegmentáltuk, a nyers és a standard szöveg egységei közötti megfeleltetést a Python programozási nyelv *difflib*<sup>9</sup> könyvtárának segítségével végeztük el.

	Tanító		Teszt	
	Elektronikus	OCR	Elektronikus	OCR
Mű	Jókai: 77 Mikszáth: 651			
Mondat	653.799		64.318	
Sor (100 karakter)	1.808.794		199.700	
Token	9.683.628	9.782.166	992.642	996.977
Type	554,541	599.327	127.321	130.062
Karakter	57.691.276	57.841.735	5.934.846	5.957.890
Átlagos mondat hossz	14,81	14,96	15,43	15,51

1. táblázat. A JiM korpusz tulajdonságai.

Az így létrejött párhuzamos korpuszt felbontottuk 90%-10% arányban tanító- és tesztkorpuszra. Az 1. táblázatban láthatóak a JiM korpusz főbb tulajdonságai.

<sup>7</sup> <https://unicode.org/reports/tr15>

<sup>8</sup> <https://github.com/nytud/quntoken>

<sup>9</sup> <https://docs.python.org/3/library/difflib.html>

### 3.2. *Silver standard* korpusz előállítása

A párhuzamos korpusz elkészültével következett a *Silver standard* korpusz építése, ügyelve arra, hogy elkülönítsük a teszt- és a tanítóanyagot. A feladathoz vettük a mondat alapú párhuzamos korpuszt és fél-automatikusan annotáltuk a szövegben felmerülő **OCR**, **koherencia** és a kiadások közötti **központozási** különbségekből eredő hibákat.

Fél-automatikus annotáláson a következő eljárást értjük: A párhuzamos korpusz tesztelésre elkülönített részénél összevetettük az OCR és *Silver* anyagot. Az anyagok közötti különbségeket listáztuk, gyakoriság szerint rendeztük. A gyakorisági listából látható volt, hogy a különbségek jelentős része nem OCR-eredetű hiba, vagy az OCR-hiba javítása szempontjából nem releváns, a kiértékeléskor nem veendő figyelembe. Példák:

- A könyvek oldalszámozása, mely az OCR-szövegben megvolt, a *Silverből* azonban hiányzott. Ez a kiadások közötti különbség, nyilvánvalóan nem OCR eredetű hiba. Ezeket a hibákat koherenciahibaként (K) címkéztük fel. Koherenciahiba származhat az előfeldolgozásból is: előfordult, hogy OCR tévesztés miatt a mondat tokenizálás eltért az OCR és a *Silver* szövegekben, ami rontotta a párhuzamosítást.
- A különbségek másik része az eltérő kiadások közötti központozásból származott. Ezeket pontuációs (P) eltérésként címkéztük fel. Előfordul például az OCR és a *Silver* korpusz között a *vessző* (,) és a *pontosvessző* (;) karakterek felcserélődése. Ennek javítását nem tekintjük OCR-hiba-javításnak, hiszen a szöveg minőségét nem javítaná, valamint könnyen lehet, hogy ezek is kiadásbeli különbségből származnak.

Ily módon rendszerezve a hibákat elértük, hogy a *silver standarden* történő kiértékeléskor csak az OCR-hibákon legyenek mérve a modellek teljesítménye.

A 2. táblázatban látható néhány példa a fent vázolt OCR-hiba címkézési stratégiájára.

A különbségek megtalálásakor a Pythonhoz írt *Levenshtein*<sup>10</sup> függvénytarát használtuk.

Miután a hibák felcímkézését ily módon elvégeztük, a címkéket karakter alapon rendeltük hozzá az OCR-*silver* párokhoz. Ehhez szükség volt arra, hogy a párhuzamos korpuszt mondatonként azonos hosszúságúra kitöltsük, majd a mondat párokhoz illesszük a hibaosztályokat. A CCC detekciós modell finomhangolásakor ugyancsak azonos hosszúságúra hozott mondatpárokat vár a bemenetre:

```
[OCR_toInput] mi rést nappal törtek, azt éjjel újra megépítetté,s a vár 196 ellen fűrt.
[OCR_aligned] mi rést nappal törtek, azt éjjel újra megépítetté,s a vár 196 ellen fűrt.
[GS_aligned] Ami rést nappal törtek, azt éjjel újra megépítetté,s a várKKKK ellen fűrt.
[OCRGD_errcl] O.....P.....KKKK
```

1. ábra: Azonos hosszúságúra töltött OCR-*silver*-hiba osztály címkesor. Az 'O' az OCR, a 'P' a központozási-, a K a koherencia hiba címkéje.

<sup>10</sup> <https://pypi.org/project/python-Levenshtein>

OCR	Gold	Hibaosztály	Előfordulás	Megjegyzés
é	e	ocr	581	karaktercserélődés
zd	v	ocr	1	több karakterből kevesebb
űié	üle	ocr	19	együtt többször előforduló összetett eset
szóköz	☐	ocr	3523	szóösszeragadás
☐	szóköz	ocr	360	szószétesés
y	☐	ocr	11	karakterelnyelődés
☐	r	ocr	232	karakterfelbukkanás
5	☐	koherencia	7	oldalszám az ocr-ben
-5	☐	koherencia	7	elválasztás és oldalszám az ocr-ben
ioo	☐	koherencia	4	hibásan ocr-ezett oldalszám
☐	(Hej	koherencia	1	előfeldolgozási hiba
:	,	központozási	123	eltérő kiadásból származó
,	;	központozási	50	eltérő kiadásból származó

2. táblázat. Példák OCR-silver között előforduló hibákra és osztályozásukra a JiM teszt párhuzamos korpuszában. A ☐ karakter a karakterhiányt szimbolizálja.

### 3.3. *Gold standard* korpusz előállítása

A párhuzamos korpuszt további manuális javításnak vetettük alá, hogy előállítsuk a *Gold standard* korpuszunkat. A különböző kiadások miatti eltéréseket négy annotátor kézzel javítja.<sup>11</sup> A javításhoz rendelkezésükre áll a – hibákat tartalmazó – OCR kimenet és az eltérő kiadásból származó hibátlan digitalizált szöveg (azaz a *Silver standard* korpusz). Az eredeti pdf-et felhasználva igazítják (azaz módosítják) a digitalizált változatot ahhoz a kiadáshoz, amiből az OCR kimenete készült. A végső, kézzel javított elektronikus korpusz a *Gold standard* korpuszunk. A javítás főbb alapelvei a következők:

- A javítás során nem a pdf fájlot követjük, hanem az OCR-ezett szöveget és a *Silver*-t. A fájlok ugyan követik a pdf-et, de akár egész bekezdések is hiányozhatnak belőlük a pdf-hez képest. Éppen ezért csak az OCR-ezett szöveg és a *Silver* közti eltérő szövegrészek kontextusára keresünk rá a pdf-ben, és azt javítjuk.
- A helyesírást ignoráljuk: a digitalizált változatban szereplő *feleletet* szót ki kell javítani *feletet*-re, ha a pdf-ben el van írva, függetlenül attól, hogy saját-hiba.
- Ha az OCR-ben van egy mondat, ami a digitalizált változatból hiányzik, de a pdf-ben megtalálható, akkor azt beírtuk a digitalizált változatba, oda, ahol a OCR-ezett változatban található.
- Két dolgot nem veszünk át a pdf-ből a digitalizált változatba: Ha egy szóban elválasztás van az OCR-ben, és tényleg el van választva a pdf-ben is, mert sortörés van benne, azt nem írjuk elválasztással a javított szövegben (tehát a pdf-ben *Magyaror-szág* van, az OCR-ezett szövegben *Magyaror-szág*,

<sup>11</sup> A cikk megjelenésének időpontjában 267 448 mondat ellenőrzése készült el (3 515 401 szó).

a digitalizált változatban *Magyarország*; ez utóbbit így is hagyjuk). Ha az oldalszám szerepel az OCR-ezett változatban és szerepel a pdf-ben is, akkor sem írjuk be a digitalizált változatba.

- Az OCR-ezett szövegben gyakran extra szóköz szerepel az írásjelek előtt. Hiába lehet a pdf-ben szellősen szedve a szöveg, a *Gold standard* változatban tapadnia kell az írásjelnek a szóvéghez.

Kutatásunk jelen a fázisában még csak a *Silver standard* korpusszal történő kiértékelések láthatóak.

#### 4. OCR-hibát detektáló és javító módszerek

Az OCR-javításhoz 4 különböző módszerrel kísérleteztünk. Kipróbáltunk egy detekciós-, illetve két különböző javítómódszert.

Kutatásunk során magyarra finomhangoltuk és implementáltuk a **CCC** többnyelvű BERT alapú detektáló módszerét, amely a 2019-es OCR-hiba javító verseny (Rigaud és mtsai, 2019) győztesének módszere. Betanítottunk továbbá egy transzformer alapú neurális gépfordító rendszert, és végül finomhangoltunk egy előtanított BART modellt a javítófeladat megoldásához.

A **CCC** (Context-based Character Correction, Rigaud és mtsai, 2019) módszer detekciós modulja (lásd 2. ábra) 3 fő részből áll: BERT enkódoló, konvolúciós háló, bináris osztályozó.

A BERT enkódoló segítségével tudja a modell megtanulni a környezetfüggő tulajdonságokat az adott szövegről, majd a BERT modellel enkódolt adatot 4 különböző kernellel rendelkező konvolúciós rétegekbe küldi be. A különböző konvolúciós rétegek gyakorlatilag n-gramm információkat osztanak meg egymással. Végül a konvolúciós rétegek által kiadott adatokat konkatenálva továbbadja egy teljesen összekapcsolt bináris osztályozást végző rétegnek, így létrehozva a token szintű értékelőt.

A CCC módszer kódja nem publikus, de egy TDS cikk<sup>12</sup> alapján reprodukáltuk, majd integráltuk a huBERT (Nemeskey, 2021) modellt, hogy magyar nyelvre tudjon detektálni. A finomhangoláshoz szükséges párhuzamos korpuszt a JiM nyers változata szolgáltatotta, míg a kiértékelés a JiM *silver* verzióján történt. A modell paramétereit módosítottuk úgy, hogy 250 szubtoken-t is képes legyen kezelni. Emellett kipróbáltunk több verziót (tanítottunk bert-base-multilingual-cased modellt, illetve az ICDAR2019-es több nyelvű tanító anyaggal is tanítottuk a modelleket), de az előzetes várakozásnak megfelelően a huBERT + JiM korpusz szolgáltatotta a legjobb kiértékelési eredményeket.

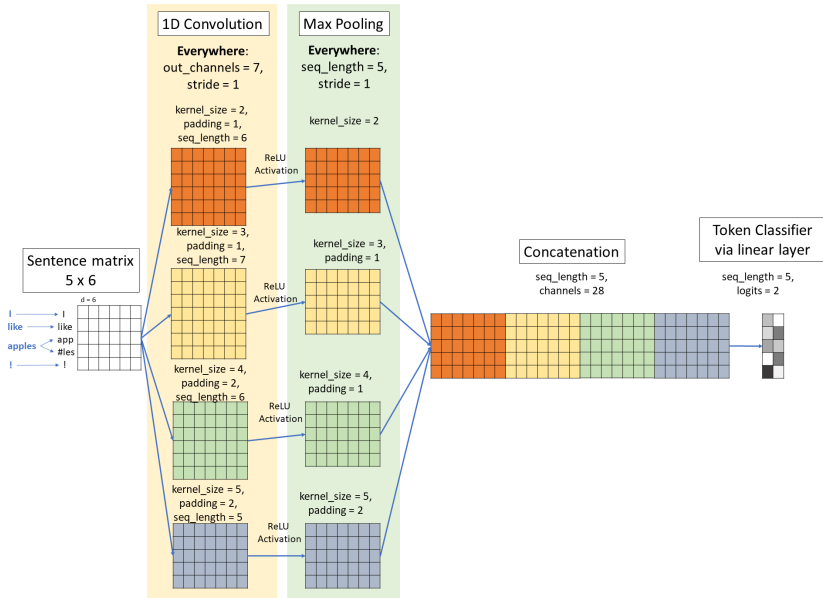
Megemlítendő még, hogy a detektáló *gold* anyagon vélhetően még jobban fog teljesíteni, mint *silveren*, mivel a koherenciát megbontó, oda nem illő szövegrészek láthatóan megzavarták a detektor viselkedését. Példa:

OCR eredeti mondat: *A csodabúvár szót fogadott tündér anyjának, s vágva egy 45 jó birssuhogót az erdőről, beveté magát a vízbe a sárga lilimokon túl.*

<sup>12</sup> <https://towardsdatascience.com/ocr-typo-detection-9dd6e396ecac>

A vastag betűs szavakat tartja a detektáló gyanúsnak, ami -legalábbis az **egy** és a **jó** szavak esetében - nyilván a közbeékelődő oldalszámnak (**45**) köszönhető.

Az osztályozó elvi felépítése:



2. ábra: CCC detekciós modell architektúrája

A **Marian NMT** (Junczys-Downumt és mtsai, 2018) egy neurális gépfordító keretrendszer, amely C++ nyelven íródott és szabadon hozzáférhető. Könnyen telepíthető, jól dokumentált, memória- és erőforrás-optimalis implementációjának köszönhetően<sup>13</sup> az akadémiai felhasználók és fejlesztők által leggyakrabban használt eszköz (Barrault és mtsai, 2019). A Marian NMT egy figyelmi (attention) modellel támogatott enkóder-dekóder architektúrájú neurális gépfordító modell, amely támogatja a transzformer architektúrát is. A legnagyobb előnye, hogy a többi módszerhez képest a leggyorsabb tanítás érhető el és ehhez nincs szükség előtanított nyelvmodellekre. A betanított modell paraméterei: 6 réteg enkóder és 6 réteg dekóder; 8 figyelmi fej; 512 szóbeágyazás dimenzió; bementi hossz: 512; előre csatolt háló méret: 2048.

A probléma egy gépi fordítás feladatként is értelmezhető, ahol a forrásnyelv egy hibás OCR-szöveg, a célnyelv a hibás OCR-szöveg hibátlan párja. A természetes nyelvek közötti gépi fordítás mondat szinten történik, ahol mind a tanítóanyag, mind a fordítandó szövegnek mondatokra bontottnak kell lennie. Az OCR-javítás feladathoz ezzel szemben nagyobb egységek vannak, és a dokumen-

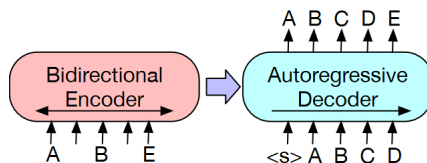
<sup>13</sup> <https://marian-nmt.github.io>

tum nincsen mondatokra bontva. Kézenfekvő megoldás, hogy fordítási egységnek a 100-as ablakra tagolt szöveget használjuk. Azonban a méréseink szerint a nagyon rövid szegmensek esetén a rendszer nem adott értelmezhető kimenetet. Ennek a jelenségnek a kezelése érdekében a végleges rendszert a 100-as ablakra és a mondatokra bontott szövegen vegyesen tanítottuk, így alkalmassá vált tetszőleges méretű szövegek javítására.

A **BART** (Lewis és mtsai, 2020) modell egy enkóder-dekóder architektúrán alapuló transzformer modell (lásd 3. ábra), amelyet a Facebook fejlesztett<sup>14</sup>. Az enkóder kétirányú (Bidirectional), a dekóder autoregresszív (Autoregressive). A BART gyakorlatilag ötvöz egy BERT (Devlin és mtsai, 2019b) és egy GPT (Radford és Narasimhan, 2018) típusú modellt. A kutatások alapján a BERT típusú modellek alkalmasak arra, hogy jó reprezentációt készítsenek egy adott szövegről, azonban szöveggenerálási feladatokra kevésbé megfelelőek, míg a GPT típusú autoregresszív modellek elsősorban szövegek generálására működnek jól. A BART a két architektúra előnyeit ötvözi, ezért kiválóan alkalmas például gépi fordításra. A BART rendszert két különböző méretben tették közzé:

- BART-base: 6 réteg enkóder és 6 réteg dekóder; 12 figyelmi fej; 768 szóbeágyazás dimenzió; bemeneti hossz: 512; 140 millió paraméter
- BART-large: 12 réteg enkóder és 12 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; 400 millió paraméter

A BART modell finomhangolható gépi fordításra, előnye a Marian NMT-vel szemben, hogy előtanított dekóderrel indítja a finomhangoló tanítást. A kutatásunkban egy magyar nyelvű BART-base modellt finomhangoltunk.



3. ábra: BART modell architektúrája (Lewis és mtsai, 2020)

## 5. OCR-hibajavító kísérletek

Kutatásunk során négy különböző javítási kísérletet végeztünk az OCR-hibák javítására:

- Marian NMT rendszerrel betanított gépifordító
- Magyar nyelvű BART finomhangolása gépifordító rendszerre
- CCC detekció és Marian NMT rendszerrel betanított gépifordító

<sup>14</sup> <https://github.com/pytorch/fairseq/tree/master/examples/bart>

- CCC detekció és Magyar nyelvű BART finomhangolása gépfordító rendszerre

Az OCR-hiba javító neurális hálózatok használatának általános mellékhatása, hogy sokszor kijavítanak olyan szövegrészeket is, amelyekben nem volt hiba (False Positive módosítás). Ezért – követve az utóbbi évek trendjeit – integráltuk a CCC detekció módszerét, hogy szűkítse le a helyeket, ahol vélhetően szükséges a javító alkalmazása.

Mivel az ICDAR versenyeken több európai nyelv is szerepelt a tanítóanyagban, a CCC detekciós modellje egy *bert-base-multilingual-cased* modellt vett alapul. Ezt – lévén a JiM magyar nyelvű – kicseréltük egy *huBERT-base-cc* modellre. Ezt tanítottuk be a JiM tanító adathalmazával, amelyet e célból az ICDAR megmérettetés sztenderd formátumára hoztunk. A CCC modell az osztályozást tokenszinten végezte el, ami ez esetben szavak, illetve központoszó elemek megjelölését tette lehetővé. Nem tudta jelölni azonban a szóközöket, azaz a szószét- eséseket ez a módszer nem tudja detektálni. Éppen ezért a szóközöket minden esetben potenciálisan hibásnak feltételeztük akkor, amikor a detektorral a javítók módosítási javaslatait megszürtük.

A Marian NMT tanításához és a BART finomhangolásához konkatenáltuk a mondataalapú és a 100 karakteres tanító korpuszokat, majd megkevertük a sorokat. A BART esetében további tokenizálást is végrehajtottunk az Orosz György-féle magyar nyelvű spaCy modellel (0.3.1).<sup>15</sup>

A BART finomhangolásához 4 darab GeForce GTX 1080Ti GPU kártyát használtunk, az alábbi módosított hiperparaméterekkel: batch méret / GPU: 14; epoch: 30; maximum bemeneti és kimeneti szöveg hossz: 256; fp16; tanulási ráta:  $2e^{-8}$ ; szótár méret: 40 000; warmup: 15 000. A BART esetében az epoch szám fixen 30.

A Marian NMT tanításához 2 darab GeForce GTX 1080Ti GPU kártyát használtunk, az alábbi módosított hiperparaméterekkel: epoch: 34; maximum bemeneti szöveg hossz: 100; tanulási ráta:  $3e^{-4}$ ; spm (Kudo és Richardson, 2018) szótár méret: 32 000; warmup: 16 000. A Marian NMT esetében az epoch szám a validációs halmazon való kiértékeléshez kötött, a megállási feltétel (early stopping) 10-re volt beállítva, ami 34 epoch számnál állította le a tanítást.

A CCC detekciós modell tanításához 4 darab GeForce GTX 1080Ti GPU kártyát használtunk, az alábbi hiperparaméterekkel: epoch: 3; maximum bemeneti szöveg hossz: 512; tanulási ráta:  $3e^{-5}$ ; warmup: 0.

## 6. Eredmények és kiértékelés

A 3. és a 4. táblázatban láthatóak az OCR-javító módszerek eredményei. A kiértékeléshez az alábbi metrikákat használtuk (Nguyen és mtsai, 2021): abszolút pontosság (*accuracy*), relatív pontosság (*precision*), fedés (*recall*), F-mérték (*F1-score*) és karakterszintű hibaarány (*character error rate* – CER). A kiértékelésben kétféle megközelítést választottunk. Az első a csak OCR-hibák vizsgálata,

<sup>15</sup> <https://github.com/spacy-hu/spacy-hungarian-models>



a másodikban az OCR-hibák mellett a koherencia- és központoszási hibákat is belevettük a kiértékelésbe, mint például írásjelek konzisztenciája, vagy oldalszámok beékelődése a szövegbe stb. Az eredmények azt mutatták, hogy a fordítók ezeket a nem OCR-típusú hibákat is tudták detektálni, javítani.

A 3. táblázatban láthatóak a csak OCR-hibák javításában elért teljesítmények. Az eredményben az látható, hogy a BART és a Marian NMT modellek tekintetében a BART magasabb fedéssel, míg a Marian NMT magasabb pontossággal javít, ezek mértéke határozta meg a végső F-mértékek eredményét, ahol a CCC nélküli versenyben a Mariann NMT győzött, míg a CCC integráció esetében a BART jött ki győztesen (F-értékben mérve). Ez azt jelenti, hogy amit a Marian NMT javít, azt pontosan javítja, azonban kevesebbet javít. Ezzel szemben a BART többet javít, ezáltal magasabb a fedése, ami nagyobb pontatlansághoz is vezet. Továbbá az látható még, hogy a CCC integrációja tovább növeli az összes modell teljesítményét.

	Pontosság	Fedés	F-mérték
BART	46,96	<b>42,61</b>	44,67
Marian NMT	53,08	42,11	46,95
CCC + BART	64,82	40,43	<b>49,80</b>
CCC + Marian NMT	<b>65,33</b>	40,09	49,69

3. táblázat. Modelljeink eredményei az OCR-hibák javításában

A 4. táblázatban láthatóak az OCR- és koherenciahibák javításának eredményei. Az eredményekből látszik, hogy a BART és a Marian NMT magas minőségben képesek a koherenciahibákat is javítani; mivel nagyobb szövegegységeken lettek tanítva, ezáltal a mondat szintű hibákat is képesek kezelni. Ebben a feladatban a Marian NMT érte el a legjobb eredményt, ami azt jelenti, hogy a Marian NMT jobban meg tudta tanulni a szövegszintű koherenciahibákat. A 4. táblázatban nem látható a CCC integráció, mivel a CCC csak OCR-hibákat tud detektálni.

	Pontosság	Fedés	F-mérték	Accuracy	1-CER
BART	59,78	70,25	64,59	99,35	0,9959
<b>Marian NMT</b>	<b>73,39</b>	<b>70,54</b>	<b>71,94</b>	<b>99,53</b>	<b>0,9969</b>

4. táblázat. Modelljeink eredményei az OCR- és koherenciahibák javításában

## 7. Összegzés

Jelen cikkben napjaink kurrens kihívására, a OCR-hibajavítás problémára szolgáltatunk több, mélytanuláson alapuló megközelítést, speciálisan magyar nyelvű szövegek javítására. Az irodalomkutatás, valamint a tárggyal kapcsolatos nemzetközi megmértetés eredményei azt mutatták, hogy a problémára adott *state-of-the-art* megoldások kétlépcsősök, detektor és javító egységek egymásra épüléséből állnak. Implementálásra a változatos detektálási lehetőségek közül a BERT-alapút választottuk, míg a javítást gépi fordítási feladatként értelmeztük.

Mindkét alegységhez, az egységek betanításához szükség volt tanítóanyagra, azaz megfelelő méretű párhuzamos korpuszra, ez azonban a projekt kezdetén nem állt rendelkezésre, annak megteremtése is a feladat része lett. Végül olyan párhuzamos korpuszt sikerült létrehozni, mely megfelel a rövid és hosszabb távú – a cikk tárgyán túlmutató – céljainknak:

- A párhuzamos korpusz egyik fele az Arcanum OCR-ezett szövegeiből
- másik fele az ennek megfelelő, OCR-hibák nélküli elektronikus alakból álljon.
- A párhuzamos korpusz kellően nagy méretű legyen a neurális modellek tanításához.
- A korpusz előfeldolgozása rövid időn belül elvégezhető legyen.

E szempontok alapján sikerült Jókai Mór és Mikszáth Kálmán életművének elemeiből előbb *Silver standard*, majd *Gold Standard* anyagot létrehozni.

A betanítás és kiértékelés lépéseit úgy terveztük meg, hogy a fordítómodulok több előnyös tulajdonságát is be tudjuk mutatni, nem pusztán a szűken vett OCR-hiba javításának minőségét. Eredményeink azt mutatják, hogy az általunk létrehozott architektúra nem csak OCR-hibákat tud javítani, hanem koherencia-problémákat is tud kezelni, mint például az oldalszámok automatikus negligálása, vagy az elválasztott szavak egyesítése.

## Köszönetnyilvánítás

Ezúton is szeretnénk megköszönni az Arcanum Adatbázis Kft.-nek, hogy rendelkezésünkre bocsátották erőforrásaikat, és ezzel lehetővé tették a hibajavító rendszerek fejlesztését.

## Hivatkozások

Barrault, L., Bojar, O., Costa-jussà, M.R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., MÁzller, M., Pal, S., Post, M., Zampieri, M.: Findings of the 2019 conference on machine translation (wmt19). In: Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1). pp. 1–61. Association for Computational Linguistics, Florence, Italy (August 2019)

- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019a)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019b)
- Duong, Q., Hämäläinen, M., Hengchen, S.: An unsupervised method for OCR post-correction and spelling normalisation for Finnish. In: Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa). pp. 240–248. Linköping University Electronic Press, Sweden, Reykjavik, Iceland (Online) (May 31–2 Jun 2021)
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018)
- Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: Open-source toolkit for neural machine translation. In: Proceedings of ACL 2017, System Demonstrations. pp. 67–72. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), <https://aclanthology.org/P17-4012>
- Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018), <https://www.aclweb.org/anthology/D18-2012>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Mei, J., Islam, A., Wu, Y., Moh'd, A., Milios, E.E.: Statistical learning for OCR text correction. CoRR abs/1611.06950 (2016), <http://arxiv.org/abs/1611.06950>
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

- Nguyen, T.T.H., Jatowt, A., Coustaty, M., Doucet, A.: Survey of post-ocr processing approaches. *ACM Computing Surveys* 1, 1 (March 2020), 36 (Mar 2021), <https://doi.org/10.5281/zenodo.4640070>
- Nguyen, T.T.H., Jatowt, A., Nguyen, N.V., Coustaty, M., Doucet, A.: Neural machine translation with bert for post-ocr error detection and correction. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. p. 333–336. JCDL '20, Association for Computing Machinery, New York, NY, USA (2020)
- Radford, A., Narasimhan, K.: *Improving language understanding by generative pre-training* (2018)
- Rigaud, C., Doucet, A., Coustaty, M., Moreux, J.P.: Icdar 2019 competition on post-ocr text correction. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. pp. 1588–1593 (2019)
- Schaefer, R., Neudecker, C.: A two-step approach for automatic OCR post-correction. In: *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. pp. 52–57. International Committee on Computational Linguistics, Online (Dec 2020)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems* 30, pp. 5998–6008. Curran Associates, Inc. (2017), <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>

# HuLU: magyar nyelvű *benchmark* adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából

Ligeti-Nagy Noémi<sup>1,2</sup>, Ferenczi Gergő<sup>1</sup>, Héja Enikő<sup>1</sup>, Jelencsik-Mátyus Kinga<sup>1</sup>,  
Laki László János<sup>1,2</sup>, Vadász Noémi<sup>1</sup>, Yang Zijian Győző<sup>1,2</sup>, Váradi Tamás<sup>1</sup>

<sup>1</sup>Nyelvtudományi Kutatóközpont

<sup>2</sup>MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport  
vezeteknev.keresztnev@nytud.hu

**Kivonat** A cikkben bemutatjuk a neurális nyelvmodellek nyelvértésének mérésére, kiértékelésére és a modellek összehasonlítására létrehozott adatbázisunkat. Az angol példát (GLUE – Wang és mtsai, 2018 –, illetve SuperGLUE – Wang és mtsai, 2020 – *benchmarkok*) követve, de a redundanciát mellőzve kiválasztottunk 13 olyan specifikus feladatot és az ehhez kapcsolódó alkorpuszokat, amelyekkel a neurális modellek teljesítménye mérhető. Mivel az angol *benchmarkokban* szereplő korpuszok közül egyik sem létezik a magyarra, ezeket megtervezzük, megépítjük, majd egységes tesztelési keretbe foglalva közrebocsátjuk ezeket.

**Kulcsszavak:** *benchmarking*, neurális modellek, korpuszpépítés, kiértékelés

## 1. Bevezetés

Az utóbbi években gombamód szaporodnak a neurális nyelvmodellek: évről évre egyre több, nagyobb, „okosabb” architektúrát mutatnak be. Ezeknek az összehasonlítására, kiértékelésére hozták létre a *benchmark* adatbázisokat, amelyek sokszor tulajdonképpen korpuszgyűjtemények, és változatos feladatokon mérik a modellek teljesítményét. Az angol GLUE és SuperGLUE *benchmarkokat* (Wang és mtsai, 2018, 2020) hamarosan követte a francia (FLUE, Le és mtsai, 2020), a spanyol (GLUES, Cañete és mtsai, 2020), vagy az orosz (Shavrina és mtsai, 2020) megfelelőjük, illetve az XGLUE, amely többnyelvű modellek kiértékelésére fókuszál (Liang és mtsai, 2020).

Kis késéssel ugyan, de megkezdődött a jelentősebb architektúrák magyar korpuszokon történő előtanítása (Nemeskey, 2021; Feldmann és mtsai, 2021). A jövőben várhatóan még több, magyarra tanított modell jelenik majd meg, melyeknek a nyelvértését ugyanúgy szükséges lesz mérni, összehasonlítani. Ezért döntöttünk úgy, hogy létrehozzuk a Hungarian Language Understanding Evaluation Benchmark Kit (HuLU) névre keresztelt adatbázis-gyűjteményünket.

A gépi tanulásban és a nyelvtechnológia területén a *benchmark* jellemzően egy vagy több adatbázisból, azokhoz tartozó megfelelő metrikákból és a teljesítmény

összesítésének módjából áll. A *benchmark* különböző rendszerek teljesítményének méréséhez biztosít olyan sztenderdet, melyben a szakmai közösség egyetért. Ez utóbbi kritériumnak történő megfelelés kényszerre eredményezte például, hogy az elmúlt pár év nagy *benchmark* adatbázisai már meglévő feladatok korpuszaiból igyekeztek reprezentatív gyűjteményt összeállítani (pl. a GLUE, vagy az XTREME, Hu és mtsai, 2020), mások pedig kifejezetten a szakmai közösség ajánlásai alapján válogatták össze az adatbázis elemeit (pl. a SuperGLUE, vagy a BIG-Bench, Ghazal és mtsai, 2017). A *benchmarkok* jelentőségét mutatja, hogy például az AI Index Report 2021 a SuperGLUE és a SQuAD (Stanford Question Answering Dataset, Rajpurkar és mtsai, 2016) alapján számol be az NLP területének általános előrehaladásáról (Zhang és mtsai, 2021). Szintén komoly szakmai érdeklődésre tart számot az, ha valamely modellnek sikerül valamelyik *benchmarkon* a humán teljesítményhez hasonló eredményt elérnie.

A magyarra *benchmark* adatbázis még nem készült. Mi kiindulópontunknak a széles körben alkalmazott, *multi-task* jellegű, mérőföldkőnek és meghatározó szerepű *benchmarknak* tartott GLUE-t, és utódját, a SuperGLUE-t választottuk. Tisztában vagyunk ugyanakkor ezeknek az adatbázisoknak a gyengeségeivel és hiányosságaival is (ezekről cikkünk összefoglaló fejezetében részletesen is szóltunk), ezért az alapvetőnek tartott, GLUE-beli és SuperGLUE-beli korpuszok létrehozásán túl a jövőben célunk majd egy az eddigieknél jobb, nyelvészeti és nyelvtechnológiai is megalapozott, átgondolt, bővebb *benchmark* adatbázis létrehozása.

A GLUE, a SuperGLUE, sőt még néhány, más nyelvre összeállított *benchmark* adatbázis esetében is már meglévő korpuszokból válogathattak a kutatók. Magyarra azonban nem elérhetőek ilyen specifikus, adott feladatra fókuszáló, megfelelően annotált korpuszok. Éppen ezért az itt bemutatott korpuszépítési munka feladata kettős: i) célunk, hogy előállítsunk több kisebb, specifikus, jól annotált, megbízható korpuszt, amelyek a nyelvmodellek számára jellemzően kihívást jelentő nyelvértési feladatokat céloznak meg, illetve ii) ezekből összeállítsunk egy *benchmark* adatbázist, amelyen a nyelvmodellek teljesítménye mérhető, összehasonlítható.

## 2. Az angol benchmarkok

A General Language Understanding Evaluation (GLUE) *benchmarkot* 2019-ben mutatták be. Az adatbázisba úgy válogatták a korpuszokat, hogy a neurális modellek nyelvértésének tesztelése a lehető legváltozatosabb, eltérő nehézségű és doménbe tartozó feladatokon váljon lehetővé. Törekedtek arra, hogy kifejezetten kevés tanítóadatot tartalmazó korpuszokat bocsássanak közre, ezáltal kedvezve a *transfer learningre* épülő modelleknek, mintegy ebbe az irányba orientálva a nyelvtechnológiai szakmai közösséget (Wang és mtsai, 2018).

A GLUE-ba kilenc, már meglévő adatbázist válogattak be, azok eredeti szerkezetét és az általuk képviselt feladatot néhol némileg módosítva. A kilenc al-korpusz a következő:

- A CoLA (Corpus of Linguistic Acceptability, Warstadt és mtsai, 2018) 10 657 angol mondatot tartalmaz, melyeket a nyelvészeti szakirodalomból gyűjtöttek. A bináris címkék a mondat elfogadhatóságát jelzik.
- Az SST (Stanford Sentiment Treebank, Socher és mtsai, 2013) az egyik legismertebb szentiment annotációt tartalmazó angol nyelvű korpusz. A korpusz összeállításához 10 662 mondatot gyűjtöttek a Rotten Tomatoes oldaláról. A mondatokat Stanford Parserrel elemezték, és az így kapott 215 154 frázist egyesével annotáltatták egy 25 fokú érzelmi skálán. A 25 fokú skálát az SST5-nek nevezett változatban 0-5-ig terjedő skálára konvertálták, az SST2-ben pedig binárisra. A GLUE részeként az SST2-t szerepeltetik a szerzők, és abból is hangsúlyosan csak az egész mondatokat, frázisszintű elemeket nem. A GLUE adatbázisából így több mint 70 000 „mondatot” és a hozzájuk tartozó címkét lehet letölteni. (A két szám – ti. a 70 000 és a 10 662 – közti különbségről ld. a magyar szentimentkorpusz létrehozásáról szóló alfejezetet.)
- Az MRPC (Microsoft Research Paraphrase Corpus, Dolan és Brockett, 2005) online hírportálok tartalmából automatikusan válogatott mondatpárok gyűjteménye, amelyekben humán annotátorok címkézték fel, hogy a két mondat szemantikailag ekvivalens-e.
- A QQP (Quora Question Pairs)<sup>1</sup> adathalmaz az MRPC-hez hasonlóan mondatpár-osztályozás a feladat, de ebben a Quora oldaláról nyert kérdések ekvivalenciájának megállapítása a feladat.
- Az STS (Semantic Textual Similarity Benchmark, Cer és mtsai, 2017) online hírekből, videó- és képfeliratokból, és más NLP-feladatokból nyert mondatpárokat tartalmaz, amelyeknél a két mondat közti jelentésbeli hasonlóság van címkézve (humán annotátorok által, 1-5 skálán).
- Az MNLI (Multi-Genre Natural Language Inference Corpus, Williams és mtsai, 2018) az SNLI (Bowman és mtsai, 2015) utódja, 10 különböző forrásból származó, humán annotációt tartalmazó mondatpárokból áll, melyeknél a feladat annak megállapítása, hogy a második mondat következik-e az elsőből, vagy ellentmond neki, esetleg semleges a viszonyuk.
- A QNLI adathalmaz a SQuAD következtetési feladattá alakított változata a GLUE-ban: a mondatpárok létrehozásához minden kérdést párosítottak a releváns szövegrészlet minden egyes mondatával.
- A GLUE-ba beválogatták a Recognizing Textual Entailment (RTE) kihívás adathalmazainak egy részét: az RTE1-ből (Dagan és mtsai, 2006), az RTE2-ből (Bar-Haim és mtsai, 2006), az RTE3-ből (Giampiccolo és mtsai, 2007) és az RTE5-ből (Bentivogli és mtsai, 2009) gyűjtöttek hírszövegekből és Wikipedia szócikkekből származó példákat. Ezekben egy (néha többmondatos) premisszáról és egy egymondatos hipotézisről kell eldönteni, hogy ez utóbbi következik-e az elsőből vagy sem. A feladat bináris címkézés, így az eredetileg hármas osztályozású példáknál a *semleges* és *ellentmondás* címkéket összevonták a konzisztencia érdekében.

<sup>1</sup> <https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs>

- A WNLI korpusz (Winograd NLI) a Winograd Schema Challenge (Levesque és mtsai, 2012) adatainak átalakított változata. A Winograd sémák lényege, hogy egy adott mondatban szereplő névmás referensét kell kiválasztani egy listából. A feladatban szereplő példákat a GLUE-ban mondatpárklasszifikálással formálták oly módon, hogy a névmás helyére beillesztettek minden egyes lehetséges referenst, és az így létrejött mondatról (mint hipotézisről) kell eldönteni, hogy az eredeti, névmást tartalmazó mondatból (a premisszából) következik-e.

A GLUE benchmark ezeken kívül tartalmaz egy ún. diagnosztikai adathalmazt is. A kézzel ellenőrzött adatbázis célja, hogy a modellek teljesítményét aprólékosan, nyelvi jelenségek széles körén lehessen elemezni. Ehhez többszáz mondatpárt válogattak össze, amelyek következtetési viszonyal vannak címkézve mindkét irányban (következés, ellentmondás, semleges viszony). Ezen kívül mindegyik mondatpár annotációja tartalmazza olyan nyelvi jelenségeknek a címkéit, amelyek a két mondat között fennálló következtetési viszonyt alátámasztják.

Végül a GLUE oldalán találunk egy dicsősegtáblát is, amelyen az egyes modellek eredménye látható, a 9 feladaton külön-külön, és összesítve is.<sup>2</sup>

A SuperGLUE (Wang és mtsai, 2020) létrehozását az motiválta, hogy a GLUE már túl könnyűnek bizonyult a nyelvmodellek számára. Így a szerzők igyekeztek nehezebb feladatokat célzó korpuszokat összeválogatni. Ennek eredményeképp a SuperGLUE a következő hat korpuszt tartalmazza (a GLUE-ban a legnehezebbnek bizonyuló, és ezért a SuperGLUE-ban is megtartott RTE és WNLI korpuszokon túl):

- QA feladatok
  - A BoolQ (Boolean Questions, Clark és mtsai, 2019) korpuszban egy szövegrészlet és egy eldöntendő kérdés alkot egy példát. A kérdéseket automatikusan gyűjtötték, és automatikusan rendelték hozzá Wikipédia-szócikkekhez.
  - A MultiRC korpusz (Multi-Sentence Reading Comprehension, Khashabi és mtsai, 2018) 871 bekezdéshez kapcsolódó 6 000, többmondatos (*multi-sentence*) kérdést tartalmaz. Többmondatos kérdés alatt azt értik a szerzők, hogy a kérdésre az információt több mondatból kell összegyűjteni. A kérdések feleletválasztós kérdések: több lehetőség közül kell kiválasztani a (több) helyes választ. A több lehetséges, de előre nem definiált darabszámú jó válasz arra kényszeríti a modelleket, hogy minden kérdés-válasz párt egyesével kiértékeljenek.
  - A ReCoRD (Reading Comprehension with Commonsense Reasoning Dataset; Zhang és mtsai, 2018) korpusz a szövegértési feladatok témakörén belül olyan kihívás elé állítja a modelleket, amelyben az emberi megértéshez hasonlóan kell értelmes következtetéseket levonni egy adott szövegből. Ehhez a CNN/Daily Mail<sup>3</sup> korpusz segítségével állítottak elő több

<sup>2</sup> A cikk írásakor (2021.10.06.) az első helyen az ERNIE modell áll (<https://github.com/PaddlePaddle/ERNIE>).

<sup>3</sup> <https://github.com/abisee/cnndailymail>



bekezdésből álló szövegeket, melyeknek a záró bekezdésében – mely az előző szövegrésznek egyfajta lezárása, konklúziója – egy már ismert tulajdonnevet elmaszkoltak: a feladat az elmaszkolt névvel kiválasztása egy listából. A korpusz több, mint 120 000 bejegyzést tartalmaz.

– jelentésegértelműsítés

- A WiC korpuszban (Word-in-Context, Pilevar és Camacho-Collados, 2019) mondatpárok bináris osztályozása a feladat: adott két mondat és egy poliszém szó<sup>4</sup>, amely mindkét mondatban előfordul; a feladat pedig annak eldöntése, hogy az adott szó mind a két mondatban ugyanabban a jelentésében szerepel-e.

– következtetési feladatok

- A CB (CommitmentBank, de Marneffe és mtsai, 2019) olyan rövid szövegrészekből áll, amelyekben legalább az egyik mondat tartalmaz egy alárendelő mellékmondatot. Mindegyik mellékmondat meg van címkézve azzal, hogy a szöveg írója milyen mértékben elkötelezett a mellékmondat igazsága mellett. A SuperGLUE-ban a feladatot egy hármas osztályozású következtetési feladattá alakították: a premissza a teljes szövegrészlet, a hipotézis pedig a beágyazott tagmondat. Csak azt a részét használták a korpusznak, ahol 80% fölötti volt a mért ITA.
- A CoPA (Choice of Plausible Alternatives, Roemmele és mtsai, 2011) ok-okozati összefüggésekre koncentrált: 1 000 kérdést tartalmaz, amelyekben egy premisszához két alternatíva közül kell kiválasztani azt, amelyik valószínűbben ok-okozati viszonyban áll a premisszával. A kérdések egy részében az okot, egy másik részében az okozatot kell az alternatívák közül kiválasztani.

A két részletesen bemutatott *benchmark* mellett természetesen további, hasonló céllal készült adatbázisok is léteznek szép számmal, melyeknek a részletes bemutatása túlmutat a jelen tanulmány keretein.

### 3. A HuLU alkorpuszai

A két legnagyobb, *multitask benchmarkba* beválogatott 15 korpusz közül mi 13-nak a létrehozását tűztük ki célul. Egyrészt a redundancia csökkentése miatt nem tervezünk QQP korpuszt (hiszen a feladat megegyezik az MRCP korpusszal, és részben feldolgozásra kerül az STS-ben), illetve nem foglalkozunk a SQuAD magyar párjának létrehozásával sem, mivel annak elkészítésén egy másik magyar nyelvtechnológiai műhely már dolgozik.

A 13 kiválasztott korpusz két csoportra osztható: egyrészt vannak azok az adatbázisok, melyeket fordítással létrehozhatunk, mert a bennük fókuszban lévő feladat nem nyelvspecifikus, és az adathalmaz jól fordítható; másrészt vannak

<sup>4</sup> A kitétel, hogy minden esetben poliszém szóról van szó, a SuperGLUE-t bemutató cikkben szerepel csak (Wang és mtsai, 2020, 6). A WiC korpuszt bemutató cikkben leírt módszertanból ilyen kitétel nem derül ki – erre a WiC-kel kapcsolatos nehézségeket bemutató alfejezetben bővebben is kitértünk.

olyan adatbázisok, amelyeket nem tudunk fordítani a jelenség nyelvspecifikus volta vagy a korpuszban szereplő szövegek összetettsége miatt. A következőkben bemutatjuk i) a saját erőforrásokból, magyar szövegekből, annotátorokkal előállított korpuszainkat, illetve ii) az angol korpuszok gépi fordításával, majd fordításellenőrzéssel és annotátori munkával előállított korpuszainkat.

A gépi fordításhoz az OPUS (Tiedemann, 2012) korpusztárból magunk építettünk egy angol-magyar párhuzamos korpuszt. A felhasznált alkorpuszok a következők: ParaCrawl, OpenSubtitles, Tatoeba, WikiMatrix, EUbookshop, PHP manual, TED2020, KDEdoc, KDE4. A párhuzamos korpuszból a Marian NMT (Junczys-Dowmunt és mtsai, 2018) nevű keretrendszerrel építettünk egy *transformer encoder-decoder* architektúrájú neurális fordítórendszert. A betanított modell paraméterei: 6 réteg enkóder és 6 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; előre csatolt háló méret: 4096.

### 3.1. Önállóan előállított korpuszok

**HuCOLA** A magyar COLA korpusz előállításához 9 944 példát gyűjtöttünk négy nagyobb, összefoglaló jellegű szakirodalmi tételből (Kiefer, 2015; Alberti és Laczkó, 2017a,b; É. Kiss és Hegedűs, 2021). A gyűjtés a következő szempontok szerint történt:

- Minden példamondatot kigyűjtöttünk a cikk írója által adott elfogadhatósági ítélet típusától függetlenül.
- A *Megnézzük (\*a) Budapest hídjait.* típusú példákából két bejegyzést készítettünk: *Megnézzük Budapest hídjait.* és *\*Megnézzük a Budapest hídjait.*
- Ha egy mondat azért nem elfogadható, mert egy adott jelentést nem “jelenthet”, azt nem gyűjtöttük, pl.: *\*Megver Péter.* ‘Péter megveri Pétert’ jelentésben (Kiefer, 2015, 49.)
- Ha egy mondatban értelmetlen szó szerepel, nem gyűjtöttük.
- Ha a fókusz helyzete miatt nem jó egy mondat, nem gyűjtöttük.
- Előíró szabályokat megsértő mondatokat nem gyűjtöttünk (*hátal* nem kezdünk mondatot).

A leírt gyűjtés során felvételre kerültek teljes mondatok és nem egész mondatok, frázisok, tagmondatok is. Mivel a megcélzott feladat mondatosztályozás, a nem teljes mondatnyi példákat egész mondatokra egészítettük ki.<sup>5</sup>

Az angol korpuszban a szerzők a gyűjtés után a leggyakoribb 100 000 angol szóra szűrték a korpuszt, és az ennél ritkább szavakat lecserélték. Mi nem alkalmaztunk ilyen szűrést a korpuszunkon, mert a *subword* alapú tokenizálás ezt már nem teszi szükségessé a mai nyelvmodellek esetében.

Minden egyes mondatot négy annotátor<sup>6</sup> címkézett fel. Az útmutató alapján azt kellett eldönteniük, hogy az adott mondat elfogadható-e, jó magyar mondatnak hangzik-e.

<sup>5</sup> A mondatkiegészítés alapelveit ld. az annotálási útmutatóban: <https://github.com/nytud/HuCOLA>.

<sup>6</sup> A feladathoz komolyabb nyelvészeti ismeretekkel nem rendelkező, nem nyelvészet szakon tanuló, vagy ott végzett annotátorokat választottunk. Összesen 12 annotátor dolgozott a korpuszon.

A mondatokat a gyűjtés során a bennük található nyelvi jelenségek alapján is felcímkéztük.

Bár a CoLA angol előzményénél a mondatok címkéi az eredetileg a nyelvész szerzők által meghatározott címkék voltak, mi a mondataink „eredő” címkéit kivettük az elemzésből. Ezzel garantáltuk, hogy a gyűjtés során hibásan feljegyzett címkék, vagy a nyomdahibák nem befolyásolják az adatok minőségét. A mondatok 69,2%-ában (6883 mondat) a négy annotátor ugyanazt a címkét rendelte a mondathoz. 22,2%-ban (2213 mondat) 3:1 arányban címkézték a mondatot. A 2:2 arányban annotált mondatokat (8,5%, 848 mondat) félretettük, ezek nem képezik részét az adatbázisnak. Ugyanakkor elérhetővé tesszük őket, mert értékes nyelvészeti kutatási anyagot jelentenek.

A mondatok végső címkéje a 3:1 arányú annotálás esetében a többség döntése alapján lett meghatározva. A GLUE-ban található arányokat követve az adatokat 80-10-10% arányban tanító-, validációs és teszhalmazra osztva adjuk közre.<sup>7</sup>

**HuRC** Az angol nyelvű ReCoRD alapján állítottuk elő a magyar nyelvű HuRC korpuszt. Zhang és mtsai (2018) automatikus módszerrel állították elő a ReCoRD-ot: több mint 120 000 példát nyertek ki a CNN/Daily News<sup>8</sup> korpuszból. A napi híreket több részre bontották (lásd 1. ábra bal oldali példa): főszöveg (*passage*), kérdés – az utolsó bekezdésben kimaszkolt tulajdonnév (*cloze-style query*), referenciaválasz (*reference answer*). A főszöveg a cikk első néhány bekezdéséből áll. A cikk utolsó bekezdésében, ami egyfajta cikklezáró passzus, szerepelnie kell egy olyan tulajdonnévnek, ami a főszövegben is előfordul. Ez a tulajdonnév a referenciaválasz. A konkrét szövegértési feladat során ezt a tulajdonnevet kimaszkolják, és a modellnek a megfelelő referenciaválaszt kell kiválasztania egy listából.

A magyar anyag előállításához a Népszabadság Online<sup>9</sup> napi cikkeit vettük alapul, ezek közül is azt a 396 886 cikket, amelyeknek volt címe, szövege és összefoglalója (*lead-je*) egyaránt. Ha valamelyik összetevő hiányzott egy cikkből, azt nem használtuk. Ezután kiválogattuk a 3-6 bekezdésből álló cikkeket. Fontos kritérium volt, hogy mind a főszöveg, mind a kérdés (az utolsó bekezdés) tartalmazzon tulajdonnevet.

A tulajdonnév felismeréséhez saját névelemfelismerő modellt tanítottunk a huBERT (Nemeskey, 2021) segítségével. A NER modell finomhangoláshoz a NerKor (Simon és Vadász, 2021) korpusz hivatalos tanító-validációs-teszt adathalmazait használtuk fel, valamint a Huggingface által nyújtott tokenszintű osztályozó könyvtárat.<sup>10</sup> NER modellünk 90,18 F-mértéket ért el a tesztanyagon.

Utolsó lépésben megkerestük azokat a tulajdonnévpárokat, amelyek a főszövegben és a kérdésben is egyaránt szerepeltek. Egy cikkben több tulajdonnévpár

<sup>7</sup> <https://github.com/nytud/HuCOLA>, illetve a korpusz elérhető Huggingface-en, a *dataset card* linkje: <https://huggingface.co/datasets/NYTK/HuCOLA>.

<sup>8</sup> <https://github.com/abisee/cnndailymail>

<sup>9</sup> <http://nol.hu>

<sup>10</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/token-classification>

<p><b>Passage</b> (CNN) -- A lawsuit has been filed claiming that the iconic <a href="#">Led Zeppelin</a> song "<a href="#">Stairway to Heaven</a>" was far from original. The suit, filed on May 31 in the <a href="#">United States District Court Eastern District of Pennsylvania</a>, was brought by the estate of the late musician <a href="#">Randy California</a> against the surviving members of <a href="#">Led Zeppelin</a> and their record label. The copyright infringement case alleges that the <a href="#">Zeppelin</a> song was taken from the single "<a href="#">Taurus</a>" by the 1960s band <a href="#">Spirit</a>, for whom <a href="#">California</a> served as lead guitarist. "Late in 1968, a then new band named <a href="#">Led Zeppelin</a> began touring in the <a href="#">United States</a>, opening for <a href="#">Spirit</a>," the suit states. "It was during this time that <a href="#">Jimmy Page</a>, <a href="#">Led Zeppelin</a>'s guitarist, grew familiar with '<a href="#">Taurus</a>' and the rest of <a href="#">Spirit</a>'s catalog. <a href="#">Page</a> stated in interviews that he found <a href="#">Spirit</a> to be 'very good' and that the band's performances struck him 'on an emotional level.' "</p> <ul style="list-style-type: none"> <li>• Suit claims similarities between two songs</li> <li>• <a href="#">Randy California</a> was guitarist for the group <a href="#">Spirit</a></li> <li>• <a href="#">Jimmy Page</a> has called the accusation "ridiculous"</li> </ul> <p><b>(Cloze-style) Query</b> According to claims in the suit, 'Parts of 'Stairway to Heaven,' instantly recognizable to the music fans across the world, sound almost identical to significant portions of 'X.'"</p> <p><b>Reference Answers</b> Taurus</p>	<p><b>Passage</b> "1968 lehetett, amikor először találkoztunk, gyakorlatilag váltottuk egymást az <a href="#">Omega</a> együttesben. Tamás akkor indult el az artista pályán, miközben zenélt is. Az <a href="#">Omegában</a> csak néhányszor játszottunk együtt, miután én beléptem, ő éveket töltött külföldön artistaként, aztán összefutottunk az <a href="#">LGT-ben</a>, ennek már 43 éve" - idézte fel <a href="#">Presser Gábor</a>.</p> <p>Mint kifejtette, <a href="#">Somló Tamás</a> színpadi jelenléte nagy hűzőerőt jelentett a zenekar számára és zenési képességeit mutatta az is, hogy amikor <a href="#">Frenreisz Károly</a> helyett belépett az <a href="#">LGT-be</a>, néhány hét alatt megtanult basszusgitarozni.</p> <p>A <a href="#">Locomotiv GT</a> utójára 2013 augusztusában lépett színpadra, az alsóörsi <a href="#">LGT-fesztiválon</a>.</p> <p>(Lead) <a href="#">Somló Tamás</a> nagyszerű egyénisége, énekhangja és éneklési stílusa egészen egyedülálló volt - fogalmazott <a href="#">Presser Gábor</a>, az <a href="#">LGT</a> vezetője a zenész halála kapcsán.</p> <p><b>(Cloze-style) Query</b> Nem ismerek olyan embert, aki Tamásra haragudott volna. Életét úgy fejezte be, ahogyan élt: utolsó fellépésére, amely talán egy hónappal ezelőtt lehetett, már nagyon nehezen tudott csak elmenni, de nem mondta le, mert Pécssett egy jótékonyági koncerten játszott beteg gyerekeknek - mondta <a href="#">[MASK]</a>.</p> <p><b>Reference Answers</b> PER: Presser Gábor</p>
--	--

1. ábra: Egy ReCoRD (Zhang és mtsai, 2018) és egy HuRC példa

is előfordulhatott. A példánkban (lásd 1. ábra jobb oldali példa) a *Presser Gábor* mellett a *Tamás* szerepel még mind a kérdésben, mind a főszövegben. Ilyen esetekben egy adott cikket többször is belevettük az adatbázisba, más-más tulajdonnév párral. Így összesen 49 782 különböző típusú cikk (*type*) került kiválasztásra, amelyekből összesen 88 655 instancia alkotja az adathalmazunkat a több tulajdonnévpár jelensége miatt. Az automatikus módszerekkel előállított korpuszunk kvantitatív tulajdonságai a következők: cikkek száma: 88 655, különböző cikkek száma (*type*): 49 782, token: 27 703 631, *type*: 1 115 260, szövegrész átlagos hossza (token): 249,42 (medián: 229), kérdés átlagos hossza (token): 63,07 (medián: 56).

Az előállított korpuszunkon ezután végeztünk néhány apró javítást. Az így létrejött, javított adathalmazunkat 100-as egységenként egy-egy annotátorral ellenőriztettük. Az annotáláshoz saját magunk által készített annotálófelületet biztosítottunk. Az automatikus maszkolást kellett validálni a következő szempontok alapján: i) jó-e a névelem-felismerés és -maszkolás (tehát *Ferenc pápa* lett maszkolva, és nem csak *Ferenc*, illetve a *Gödöllőre* az nem [\[MASK\]](#)re, hanem [\[MASK\]](#)), továbbá ii) szerepel-e a cikk korábbi részeiben is az elmaszkolt névelem.<sup>11</sup> Az ellenőrzés eredményeképpen 80 587, automatikusan előállított, kézzel validált szövegegység szerepel az adatbázisban.<sup>12</sup>

<sup>11</sup> Összesen 12 annotátor dolgozott a korpuszon.

<sup>12</sup> <https://github.com/nytud/HuRC>, <https://huggingface.co/datasets/NYTK/HuRC>

**A HuCommitmentBank** A HuCommitmentBank-et a feladat nyelvspecifikus volta miatt magyar nyelvű példák gyűjtésével lehet csak előállítani. Az angol korpusz 1 200 diskurzusszegmenst tartalmaz. Mindegyik szegmens egy alárendelt mellékmondatot tartalmazó mondatból és az azt megelőző 2-3 mondatos kontextusból áll. A célmondatban a mellékmondatot vonzó főige egy *entailment canceling operator* alá van beágyazva (ezek az operátorok a tagadás, a kérdés, a feltételes mód és a modális módosítók). A példákat automatikus módszerekkel gyűjtötték, majd kézzel validálták.

A magyar gyűjtést több irányban indítottuk el, az MNSZ2 (Oravecz és mtsai, 2014) beszélt nyelvi alkorpuszán: egyrészt mintázatillesztéssel próbálkoztunk a felszíni alakokon, másrészt az angol módszertanhoz hasonlóan szintaktikai elemzéssel kerestünk megfelelő jelölteket. 4 annotátor kétheti munkájával előállt kicsivel több mint 1 000 példa. Ezek egy részének<sup>13</sup> a validálása már megtörtént. A kutatás következő fázisában a példákat annotátoroknak osztjuk ki, hogy megcímkézzék a célmondatokat.

**HuWiC** A jelentés-egyértelműsítési feladat bináris osztályozási feladattá egyszerűsítve jelenik meg Pilevar és Camacho-Collados (2019) korpuszában. Itt azt kell eldönteni egy célszóról és az azt tartalmazó mondatpárról, hogy a szóelőfordulások ugyanazt jelentik-e a két különböző kontextusban, vagy sem. A feladat, látszólagos egyszerűsége ellenére, meglehetősen nehéznek bizonyult (vö. Véronis, 2001).

Az angol WiC korpusz építése során már meglévő jelentéstárakban (WordNet, VerbNet, Wiktionary) található példamondatokra támaszkodtak.<sup>14</sup> Mivel a WordNet-en kívül nem állnak rendelkezésünkre a magyarra hasonló adatbázisok, illetve Véronis (2001) kísérletei rávilágítottak a feladat komplexitására, úgy döntöttünk, az angol WiC korpusz létrehozásának módszertanától eltérő módon építjük meg a saját adatbázisunkat. Első lépésben a legegyszerűbbnek tűnő feladattal, az egyjelentésű főnevek csoportjával kezdtünk. Azokat a főneveket tekintettük egyjelentésűnek, amelyek a HuWN-ben (Miháltz és mtsai, 2008) egy jelentéssel szerepeltek, és a hozzájuk rendelt ÉKSz. (Pusztai, 2003) link a címszó alatti 1.1-es jelentésre mutatott.<sup>15</sup> Így egy 5 981 elemű lista állt elő.

Mivel a HuWN példamondatai nem tükrözik a természetes nyelvhasználatot, ezért a korpuszunkhoz a példamondatokat az MNSZ2-ből (Oravecz és mtsai, 2014) nyertük ki automatikus módszerekkel.

<sup>13</sup> A cikk leadásakor a példák mintegy 90%-a lett már validálva.

<sup>14</sup> Fontos megjegyezni, hogy a WiC korpusz létrehozásának leírásánál nem esik szó arról, hogy a gyűjtés során kifejezetten polyszém szavakra fókuszáltak volna; megfogalmazásuk szerint első lépésben mindent gyűjtöttek. Éppen ezért nem világos, hogy Wang és mtsai (2020) miért írják, hogy a korpusz polyszém célszók viselkedését vizsgálja (ld. a bevezetőben részletezett leírást.)

<sup>15</sup> Ezzel a HuWN-ben egyjelentésűként feltüntetett főnevek közül kiszűrtük azokat, amik az ÉKSz.-ben biztosan többjelentésűek. Az 1.1-es jelentésre való utalás azonban nem zárja ki egyértelműen, hogy az adott főnévnek az ÉKSz.-ben van más jelentése is.

A mondatpárok gyűjtésénél a GDEX (Good Dictionary EXamples, Kilgarrieff és mtsai, 2008) leírásból indultunk ki. Az abban megfogalmazott kritériumok lehetővé teszik, hogy kvantitatív módon megragadhatóak legyenek a jó példamondatok. Összefoglalva tehát, első lépésben az volt a célunk, hogy olyan mondatpárokat gyűjtünk automatikusan, amelyekben a célszó garantáltan ugyanazzal a jelentéssel szerepel.

125 célszóhoz tartozó 5-5 példamondat kvalitatív kiértékelése azt mutatta, hogy a gondos gyűjtés ellenére a célszavak jelentős része eltérő jelentéssel szerepel a példamondatokban. Szerencsére ezek a többjelentésű esetek csoportokba sorolhatók (szisztematikus metonímia, deverbális főnevek, kollokációk), így egy részletes annotálási útmutatóval a feladat jól definiálttá tehető. A korpusz előállításához fontosnak tartjuk a megfelelő annotálási alapelvek kidolgozását, elsősorban formai, szintaktikai kritériumok alapján, amennyiben ez lehetséges.

### 3.2. Fordított korpuszok

**HuCoPA** A CoPA 1000 kérdését először gépi fordítóval fordítottuk, ennek a kimenetét annotátorok ellenőrizték és javították, a fluenciára törekedve. Egy-egy annotátor pedig megjelölte a helyes választ a kérdésre. Ha az annotátor döntése és az eredeti címke között eltérést találtunk, akkor kézzel ellenőriztük az adott példát. Az annotálás harmadik lépésében derült fény például a CoPA egyik hibás címkéjére (a *training set* 380-as id-jú kérdése). Így végül előállt az 1 000 egységből álló HuCoPA korpusz.<sup>16</sup> Ebből az eredeti felosztást megtartva 400-at tanító-, 100-at validációs, 500-at pedig tesztanyagnak különítettünk el. Ahhoz, hogy a tesztanyag hibátlanságát biztosítsuk, minden, a tesztanyagban szereplő mondatot 4-4 annotátornak adunk, így egyszerre mérjük a humán teljesítményt a korpuszon és validáljuk is a címkéket az ITA vizsgálatával. Ez a lépés, ti. a tesztanyag annotálása még hátravan.

**A HuSST** A szentimentkorpusz lefordításakor annak a GLUE-ban szereplő formája helyett az ún. SST-5 adathalmazt vettük alapul. A GLUE alkorpuszai közt található SST-2 adatok letöltésekor szembesültünk azzal, hogy bár a GLUE-cikk szerzői kizárólag a teljes mondatok használatát jelzik (és mondatosztályozási feladatként fogalmazzák meg a kérdést),<sup>17</sup> a fájlokban rengeteg frázis található (néhány példa a tanítóanyagból: *of saucy, in world cinema, a doa*). Ez okozza tehát az SST bemutatásakor említett 10 662 mondat és a 70 600-as GLUE-s korpusz közti méretbeli különbséget. Megjegyzendő, hogy itt 11 855 mondatot találtunk, amiket gépi fordítóval magyarra fordítottunk. Ezt követték a HuCoPA korpusznál ismertetett ellenőrzési lépések. Végül minden magyar mondatot három-három annotátor címkézett a szentiment alapján egy hármas skálán. A szentimentcímkéket egy kurátor nézte át, aki végleges címkével látta

<sup>16</sup> <https://github.com/nytud/HuCoPA>; illetve a korpusz elérhető Huggingface-en, a *dataset card* linkje: <https://huggingface.co/datasets/NYTK/HuCoPA>

<sup>17</sup> „We use the two-way (positive/negative) class split, and use only sentence-level labels.”, (Wang és mtsai, 2018, 3).

el a mondatokat.<sup>18</sup> 7064 mondatnál (59.6%) teljes volt az egyetértés a három annotátor között, 4619 (38,96%) esetben pedig 2:1 arányban címkéztek. A végső címke minden esetben a kurátor döntése.<sup>19</sup> 172 mondatot nem használunk az adatbázisban, ezeknél a három annotátor három különböző címkével látta el a mondatot.

**A Winograd-sémák** A Winograd-sémák a referencia-feloldás feladatát célozzák. A feladat nem triviális: *The man couldn't lift his son because he was so (weak/heavy)*. A Winograd-sémákat már más nyelvekre is lefordították (japán, francia, portugál, kínai, héber). Az angol eredetit először a már ismertetett gépi fordító segítségével lefordítottuk, majd ennek kimenetét két ember validálta. Bizonyos sémákat elvetettünk, mert nem tudtuk őket úgy lefordítani, hogy megőrizzük bennük a Winograd-sémák jellemzőit (pl. *Lily spoke to Donna, breaking her (silence/concentration)*: a két angol kifejezés nem fordítható le magyarra úgy, hogy pusztán egy szóban térjen el a két mondat, de mindkettőben megmaradjon a birtokos szerkezet). Más esetekben kis módosítással ültettük át magyarra az eredeti sémát. A Winograd-sémák önálló kutatási témát jelentenek, amelynek részletes kifejtése túlmutat a jelen tanulmány keretein.

Az eredeti adathalmaz 150 mondatpárt tartalmaz. A fordítás és a validálás után 122 magyar séma lett.<sup>20</sup>

**HuRTE** Az RTE adathalmazok GLUE-ba beválogatott részét a már ismertetett gépi fordítóval fordítottuk magyarra. Ezt követően az így előállt 9 000 példát (amely nagyjából 18 000 mondatot jelent) az SST-2 magyar párjának előállításánál ismertetett módon annotátorokkal ellenőriztettük, javítottuk. Következő lépésként minden példát annotátorokkal címkéztetünk, hogy kiderüljön, ha a fordítás során elveszett a mondatpárok közti eredeti következtetési viszony. Az így validált adathalmazt a többi korpusznál ismertetett módon közzéteszük.

### 3.3. További alkorpuszok

A magyar *benchmark* korpuszba a fentiekén kívül egy *multi-sentence reasoning* korpuszt is készítünk majd. Az Oktatási Hivatallal történő együttműködés keretében rendelkezésünkre állnak majd a PISA-felmérések szövegértési feladatai az elmúlt 21 év felméréseiből. Ezek a szövegértési feladatok jellegükből adódóan éppen a MultiRC korpusz céljainak megfelelő szövegek: a kérdések megválaszolásához több bekezdésen átívelő, komplex szövegértésre van szükség. Ezen felül kézzel validált, válogatott szövegekről van szó. A feladatok kérdései közül pedig a korpusz céljainak megfelelően a feleletválasztásokat építjük majd be az adatbázisba.

<sup>18</sup> A fordításellenőrzést 12 annotátor végezte, a fluenciajavítást 8. 11-en végezték a szentimentannotálást, és 4 fő foglalkozott a kuratori feladattal.

<sup>19</sup> Az adatbázis elérhetősége: <https://github.com/nytud/HuSST>, illetve <https://huggingface.co/datasets/NYTK/HuSST>.

<sup>20</sup> <https://github.com/nytud/HuWSC>, <https://huggingface.co/datasets/NYTK/HuWSC>

A BoolQ korpusz magyar párját nem fordítással, hanem az angol korpusz létrehozásának módszertanát követve, magyar nyelvű Wikipédia-szócikkekből és annotátori munka segítségével állítjuk majd elő. A két nagy korpuszt, a szemantikai hasonlóságra fókuszáló STS-t és a következtetési feladatot képviselő MNLI-t csak komolyan nyelvészeti kutatómunka után tudjuk elkezdni megtervezni és előállítani, így ezek még a jövő feladatai közé tartoznak.

## 4. Összegzés

A tény, hogy a magyar modellek kiértékelésére létrehozott *benchmark* 1-2 év késséssel követi angol elődjét, az előnyünkre fordítható: több tanulmány is vizsgálta időközben a *benchmarkok* hiányosságait. Moradi és Samwald (2021) a BERT, az XLNet, a RoBERTa és az ELMo teljesítményét vizsgálták különböző feladatokon, elsősorban arra fókuszálva, hogy a modellek hogyan teljesítenek zajos inputon. Eredményeik azt mutatták, hogy az említett nyelvmodellek kifejezetten érzékenyek a bemeneti szövegek minőségére, és teljesítményük már a legapróbb változtatástól is jelentősen romlani kezd. Arra is rávilágítanak, hogy a jelenleg használt *benchmarkok* nem mérik megfelelően a modellek robusztusságát. Érvelésük szerint a jelenleg használt *benchmarkokat* ki kéne egészíteni a zajos inputokon történő kiértékeléssel, hogy az NLP rendszerek robusztusságáról realisabb képet kapjunk.

A feladatok kiválasztásán, és a feladathoz készített tanító- és tesztanyag megfelelő összeállításán túl természetesen az annotáció minősége is kulcsfontosságú. Mivel a *benchmarkok* előállításának folyamata egy vagy több ponton is tartalmaz automatizált lépéseket, a hiba lehetősége adott. Több tanulmány is vizsgálta, hogy a tanítóanyagban lévő hibás címkék milyen hatással vannak a modellekre. Általában az az álláspontja a kutatóknak, hogy valójában nem jelent nagy problémát a hibás tanítócímke, a neurális modellek elég robusztusak a hibás címkékkel kapcsolatban (erről ld. például Mahajan és mtsai, 2018; Rolnick és mtsai, 2017). Ezzel szemben a tesztadatok közt előforduló hibás címke komoly gondot okoz: Northcutt és mtsai (2021) 10 nagyobb adatbázis tesztanyagát vizsgálva 3.3%-osra becsülte a hibás címkék arányát – ami értelemszerűen túl sok, ha a tesztanyagon mért teljesítmény (például pontosság) alapján ítélnék megbízhatónak egy-egy nyelvmodelt.

Következő lépésként az előzőekben bemutatott korpuszokon szeretnénk human teljesítményt mérni, és eközben a korpuszok tesztanyagának elkülönített részét validálni, hogy megbízhatóságukról meggyőződve adhassuk azokat közre. Hosszabb távú célunk pedig az, hogy a magyar nyelvű nyelvmodellek előállításának és tesztelésének egyik hátráltató tényezőjét, a *benchmarkok* hiányát szisztematikusan felszámoljuk azzal, hogy a GLUE-t és SuperGLUE-t alkotó alkorpuszok mellett más, újabb, nehezebb feladatot megcélzó, robusztusabb modellek létrehozását ösztönző és karbantartott *benchmark* készlet álljon elő műhelyünkben.



## Hivatkozások

- Alberti, G., Laczkó, T. (szerk.): *Syntax of Hungarian. Nouns and Noun Phrases, Volume 1. Comprehensive Grammar Resources*, Amsterdam University Press (2017a)
- Alberti, G., Laczkó, T. (szerk.): *Syntax of Hungarian. Nouns and Noun Phrases, Volume 2. Comprehensive Grammar Resources*, Amsterdam University Press (2017b)
- Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*, Venice, Italy. pp. 1–9 (2006)
- Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: *Proceedings of the TAC Workshop* (2009)
- Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics (2015)
- Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish Pre-Trained BERT Model and Evaluation Data. In: *PML4DC at ICLR 2020* (2020)
- Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pp. 1–14. Association for Computational Linguistics, Vancouver, Canada (Aug 2017), <https://aclanthology.org/S17-2001>
- Clark, C., Lee, K., Chang, M.W., Kwiatkowski, T., Collins, M., Toutanova, K.: BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 2924–2936. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://aclanthology.org/N19-1300>
- Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Quinonero-Candela, J., Dagan, I., Magnini, B., d’Alché Buc, F. (szerk.) *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. pp. 177–190. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
- Dolan, W.B., Brockett, C.: Automatically Constructing a Corpus of Sentential Paraphrases. In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)* (2005), <https://aclanthology.org/I05-5002>
- É. Kiss, K., Hegedűs, V. (szerk.): *Syntax of Hungarian. Postpositions and Postpositional Phrases*. Comprehensive Grammar Resources, Amsterdam University Press (2021)

- Feldmann, A., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Ghazal, A., Ivanov, T., Kostamaa, P., Crolotte, A., Voong, R., Al-Kateb, M., Ghazal, W., Zicari, R.V.: BigBench V2: The New and Improved BigBench. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE). pp. 1225–1236 (2017)
- Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. p. 1–9. RTE '07 (2007)
- Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., Johnson, M.: XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization (2020)
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Hermann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast Neural Machine Translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018)
- Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., Roth, D.: Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In: NAACL (2018)
- Kiefer, F. (szerk.): Strukturális magyar nyelvtan 1. Mondattan. Akadémiai Kiadó, Budapest (2015)
- Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., Rychlý, P.: GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In: Elisenda Bernal, J.D. (szerk.) Proceedings of the 13th EURALEX International Congress. pp. 425–432. Institut Universitari de Linguística Aplicada, Universitat Pompeu Fabra, Barcelona, Spain (jul 2008)
- Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., Schwab, D.: FlauBERT: Unsupervised language model pre-training for French. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 2479–2490. European Language Resources Association, Marseille, France (May 2020), <https://aclanthology.org/2020.lrec-1.302>
- Levesque, H.J., Davis, E., Morgenstern, L.: The Winograd Schema Challenge. In: Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. p. 552–561. KR'12, AAAI Press (2012)
- Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, R., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J.H., Wu, W., Liu, S., Yang, F., Campos, D., Majumder, R., Zhou, M.: XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. arXiv abs/2004.01401 (2020)

- Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Barambe, A., van der Maaten, L.: Exploring the Limits of Weakly Supervised Pretraining (2018)
- de Marneffe, M.C., Simons, M., Tonhauser, J.: The CommitmentBank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung* 23(2), 107–124 (Jul 2019), <https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601>
- Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., Váradi, T.: Methods and Results of the Hungarian WordNet Project. In: *Proceedings of the Fourth Global WordNet Conference GWC 2008*. pp. 310–320 (2008)
- Moradi, M., Samwald, M.: Evaluating the Robustness of Neural Language Models to Input Perturbations. *Computer Science – Computation and Language* (2021)
- Nemeskey, D.M.: Introducing huBERT. In: *XVII. Magyar Számítógépes Nyelvészeti Konferencia*. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Northcutt, C.G., Athalye, A., Mueller, J.: Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. *ArXiv abs/2103.14749* (2021)
- Oravecz, C., Váradi, T., Sass, B.: The Hungarian Gigaword corpus. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014), [http://www.lrec-conf.org/proceedings/lrec2014/pdf/681\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2014/pdf/681_Paper.pdf)
- Pilevar, M.T., Camacho-Collados, J.: WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In: *NAACL 2019 (Minneapolis, USA)* (2019)
- Pusztai, F. (szerk.): Magyar értelmező kéziszótár. Akadémiai Kiadó, Budapest (2003)
- Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016), <https://aclanthology.org/D16-1264>
- Roemmele, M., Bejan, C., Gordon, A.: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. *AAAI Spring Symposium - Technical Report* (01 2011)
- Rolnick, D., Veit, A., Belongie, S.J., Shavit, N.: Deep Learning is Robust to Massive Label Noise. *ArXiv abs/1705.10694* (2017)
- Shavrina, T., Fenogenova, A., Emelyanov, A., Shevelev, D., Artemova, E., Malykh, V., Mikhailov, V., Tikhonova, M., Chertok, A., Evlampiev, A.: RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. *arXiv preprint arXiv:2010.15925* (2020)
- Simon, E., Vadász, N.: Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: *Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD*

- 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), <https://aclanthology.org/D13-1170>
- Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012)
- Véronis, J.: Sense Tagging: Does It Make Sense? In: Corpus Linguistics Conference (2001), <http://www.up.univ-mrs.fr/veronis/pdf/2001->
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (2020)
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018), <https://aclanthology.org/W18-5446>
- Warstadt, A., Singh, A., Bowman, S.R.: Neural Network Acceptability Judgments. arXiv preprint arXiv:1805.12471 (2018)
- Williams, A., Nangia, N., Bowman, S.: A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), <https://aclanthology.org/N18-1101>
- Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J.C., Sellitto, M., Shoham, Y., Clark, J., Perrault, R.: The AI Index 2021 Annual Report (2021)
- Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K., Durme, B.V.: ReCoRD: Bridging the Gap between Human and Machine Commonsense Reading Comprehension (2018)

# POSZTER, LAPTOPOS BEMUTATÓ



## A kétrétegű dinamikus pragmatika implementációja

Szeteli Ádám, Friedszám Attila, Szeteli Anna, Kárpáti Laura, Hagymási Judit,  
Kleiber Judit, Alberti Gábor

Pécsi Tudományegyetem BTK/TTK,

ReALIS Elméleti BTK/TTK, Számítógépes és Kognitív Nyelvészeti Kutatócsoport

{adamszeteli,f.attila975,anna.szeteli,lauranthalasa.1,hagymasi.judit.0}@gmail.com  
kleiber.judit@btk.pte.hu, alberti.gabor@btk.pte.hu

**Kivonat:** A dinamikus pragmatikai szakirodalom legfrissebb eredményeire alapozva a természetes nyelven történő információátadást kívánjuk modellálni egy olyan multiágens rendszer révén, ahol – egy változó külvilágmodellel párhuzamosan – az egymással kommunikáló Ágensek információállapotának releváns szegmenseit követhetik nyomon a Felhasználók, Fázisról Fázisra. Mindez egy olyan automataként is definiálható, amelyben egy átmenetszabály bemeneti feltétele az, hogy egy megnyilatkozás intenzionális profiljában definiált feladói és címzetti szerep unifikálható legyen rendre a beszélő, illetve a hallgató pillanatnyi (adott fázisbeli) információállapotával, a kimeneti Fázis pedig az unifikáció komplex eredményének a Felhasználó által kiválasztott szegmense lesz.

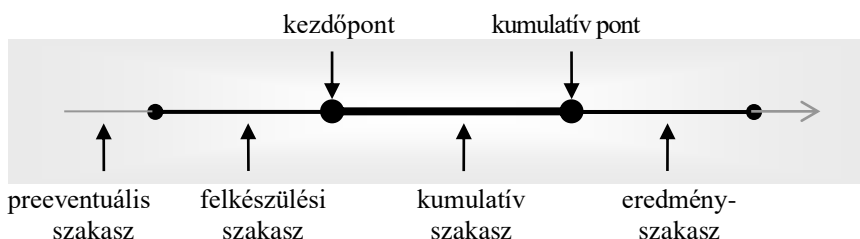
**Kulcsszavak:** kétrétegű dinamikus pragmatika, formális szemantika, automata-típusok, C#

### 1. Cél: multiágens rendszer az információátadás modellálására

A tanulmányban bemutatásra kerülő program és a háttérét képező kétrétegű dinamikus pragmatikai projekt egy négy évvel ezelőtti projekthez Szeteli és mtsai (2018) és Nőthig és Szeteli (2018) hasonló célokat tűz maga elé, de új programozói koncepciót követ, és a ReALIS-megközelítés (Alberti 2011), valamint a nemzetközi dinamikus pragmatikai szakirodalom (Farkas és Roelofsen 2017) legfrissebb eredményeit (Alberti és mtsai 2021, Szeteli és mtsai 2019, Gyuris 2019, Kárpáti és mtsai 2021) veszi alapul. A természetes nyelven történő információátadást kívánja modellálni egy olyan multiágens rendszer Benz (2000) révén, ahol – egy változó külvilágmodellel párhuzamosan – az egymással kommunikáló Ágensek információállapotának releváns szegmenseit követhetik nyomon a Felhasználók, általuk kidolgozott Témakörben.

### 2. Témakörök, Események, Állások, Fázisok

Egy Témakör időben elhelyezett Eseményekből áll (pontosabban eseményszerűségekből, amibe az egyszerűbb eseményszerkezetű állapotokat is beleértjük), amelyeknek a pillanatnyi Állásáról tehetőek állítások, az 1. ábrán bemutatott eseményszerkezet alapján (Farkas és Ohnmacht 2012). Amennyiben a sémát valakinek a hazautazási eseményére alkalmazzuk, akkor a négy szakaszhoz kapcsolódó Állások (balról jobbra) így fogalmazhatóak meg (többek között): *haza fog utazni, hazautazófélben van, éppen utazik haza, már hazautazott.*



1. **ábra:** Az általános eseményszerkezet és kapcsolata az intenzionális profillal.

A pillanatnyi Állások összessége egy temporális külvilágmodellt alkot, amit a kommunikációs lépések egységes kezelése érdekében érdemes a multiágens rendszer egy különleges Ágensére „bízni”: egy Orákulumra, aki a figyelembe vett időpillanatokban minden Állás igazságértékéről biztos tudással bír. A többi Ágens nem feltétlenül rendelkezik biztos tudással az Állásokról, hanem (lényegében montagoviánus) alternatív lehetséges világokban (Dowty és mtsai 1981) igazságértékeli azokat, amelyek különféle erősségű hipotéziseik, vágyaik, szándékaik gyűjtőhelyeinek tekintendők. Egy Ágens pillanatnyi információállapotát egy olyan diskurzus reprezentációjaként (Kamp és mtsai 2011) ragadhatjuk meg, amelyben a figyelembe vett Állásokra vonatkozó gondolatait összegeznék, attitűdök szerint csoportosítva (Maier 2016). Valahogy így, tipikusan több attitűdhez is társítva egy-egy Állást: „i. ebben *majdnem biztos* vagyok, de ii. *nem örülök* neki, ezért iii. a módosítására *vágyom*, aminek útja: iv. *szándékom* felébreszteni egy másik Ágensben a *szándékot* e módosításra.

A Felhasználók egy Témakörben az ahhoz tartozó Események néhány Fázisban fennálló Állásait tekinthetik át, illetve döntően azt, hogy az Ágensnek mondatnyi megnyilatkozásai – mint beszéddel való cselekvések (Austin 1975/1962) – hogyan állítanak elő egymást követő Fázisokat. A fentiek alapján egy Fázis egy olyan reláció, amelyben Állások társulnak a *ReALIS*-ben *világocskának* nevezett lehetségesvilágszerű objektumokkal (tekintettel ezen objektumok tipikusan csekély információtartalmára). Úgy is értelmezhetjük a Fázist, mint a releváns Állások igazságértékelését az Orákulum és a többi Ágens világocskáiban. Egy megnyilatkozás egy Állásról szól, olyan módon módosítva a Fázist, hogy az abban beszélőként és hallgatóként részt vevő Ágensnek az adott Állást a kimeneti Fázisban másképpen társítják világocskákhoz. Egy egyszerű példa: egy Ágens hallgatója lesz annak a megnyilvánulásnak, hogy *Tegnap este nyert a Barcelona*; és – nem látván a meccset – míg a bemeneti Fázisban csak vágyott rá, hogy ez így legyen, és elég valószínűnek ítélte ezt a kimenetelt, addig a kimeneti Fázisban már biztos lehet benne.

### 3. Véges állapotú automata a Fázisok közötti átmenetre?

A Fázisok formális megragadására kínálkozik az a lehetőség, hogy egy véges állapotú automata (Partee és mtsai 1990) állapotainak tekintsük őket, amelyek között az átmenetszabályok egy-egy Állásra vonatkozó megnyilatkozások „leolvasásával” járnak; vagyis azzal, hogy egy beszélő Ágens elmondja egy hallgató Ágensnek. Egy átmenetszabály bemeneti feltétele: az adott megnyilatkozás intenzionális profiljában definiált feladói és címzetti szerep unifikálható rendre a beszélő, illetve a hallgató pillanatnyi (adott fázisbeli) információállapotával. Az 1. táblázatban bemutatott három legalapve-



több intenzionális profil szemlélteti majd a fogalmat, mindenekelőtt azonban tisztáz-  
nunk kell, hogy a háttérben álló dinamikus pragmatikai felfogásra a tanulmány címé-  
ben is megjelenő kétrétegűség jellemző: a *beszélő* mint Ágens nem terminológiai  
alternatívája a *feladónak*, hanem az utóbbi egy nyelvi konvenciókban rögzült szerep  
(Oishi 2014). Egy megnyilatkozásával „a beszélő Ágens felveszi a feladói szerepet” –  
ami megfelel a searle-i illokúciós aktusnak (Searle 1976), felajánlva egyszersmind a  
hallgató Ágensnek a címzetti szerepet, aki „elfogadhatja eme szerepet” – ezeknek a  
mozzanatoknak egy része már a searle-i perlokúciós hatás mechanizmusának a részét  
képezi. A beszédcselekvés tehát ebben a felfogásban állapotok közötti átmenetsza-  
bály, amelynek bemeneti feltétele a szerepek *unifikálhatósága* a két beszédrésztvevő  
Ágens információállapotával, kimenete pedig az *unifikáció megvalósulásával* létrejő-  
vő új információállapot-pár, pontosabban az ezt tartalmazó módosult Fázis-verzió.

A következő példasorban bemutatott kijelentő, felszólító, illetve eldöntendő kérdő  
mondattípus intenzionális profilját tartják a legalapvetőbb három – (elegánsan fogal-  
mazva) *deklaratív*, *imperatív* és *interrogatív* (Lauer 2013) – mondathasználati kon-  
venciónak. Egy profil lényegében az érintett világocskákból és a bennük meghatáro-  
zott igazságértékekből áll.

- (1) a. Hazautazott.
- b. Utazzon haza!
- c. Hazautazott?

A táblázatban megadott profilokat terjedelmi okokból döntően a releváns adattípusok  
illusztrálása végett mutatjuk be; a pragmatikai részletek iránt érdeklődő olvasó egy-  
szerűsített leírást Alberti (2020) tanulmányban kaphat.

- 1. **táblázat.** A három fő mondattípus intenzionális profilja (Alberti és mtsai 2021),  
háttérében a társalgásnak mint együttműködő Ágensok beszédcselekvésének a grice-i  
maximái (Grice 1975)

Célorientált mentalizáció	Kijelentő	Felszólító	Eldöntendő kérdő
	Alap	Alap	Alap
e-re: iB	$iB = +5\alpha$	$iB = -5\gamma$	$iB \in "0"$
$iBuB \in "+5"$	$iBuB \in "0"$	$iBuB = "+5\bullet\gamma"$	$iBuBiB \in "+5\bullet+5\bullet"0"$
e'-re, ...•W, $r \in R \subseteq \{i, u, o\}$ ,	$W = uB\uparrow$	Default: $e' = res_e$	$W = iB\uparrow$
$(\Sigma iBrD \cdot iBrD) / (5 \cdot  R ) \in "+5"$	$iBrDuB\uparrow = 5\beta$	e'-re: iBrD	$iBrDiB\uparrow = \beta$
$iBuA \in "+5\bullet"+5"$	$iBuAuB\uparrow = \beta$	e'-re: iBuA	$iBuAiB\uparrow = \beta$ döntő elem: $iBuB = \xi \in \beta = 5$
e"-re: $iIuI\uparrow \in "+5\bullet+5"$	$iIuI\uparrow uB\uparrow = \alpha$	e"=e-re: $iIuI\uparrow$	$iIuI\uparrow iB\uparrow = \xi$
$iAiIuI\uparrow \in "+5"$	$iAiIuI\uparrow uB\uparrow = \alpha$	e"=e-re: $iAiIuI\uparrow$	$iAiIuI\uparrow iB\uparrow = \xi$

A világocskákat egy címkerendszer segítségével azonosítjuk és helyezzük el egy  
részben rendezett hálózatban. Egy *egyszerű* címke megmutatja, hogy kinek milyen  
attitűdjéről van szó (B: *belief*, D: *desire*, I: *intention*, A: *authority*, E: *experience*), és

hogy milyen időpillanatban tekintjük az igazságértékelést, ami a  $-5$ -tel jelölt *hamis* és a  $+5$ -tel jelölt *igaz* minősítések között<sup>1</sup> a dinamikus pragmatikában ma már bevett átmeneti szinteket is tartalmaz (Farkas és Roelofsen 2017); ami a hiedelem B dimenziójában például a teljes információhiány  $0$  értékén kívül a valószínűsítések  $+1$  és  $+4$ , illetve  $-1$  és  $-4$  közötti értékelését is lehetővé teszi. A kijelentő mondat feladójának tudását (iB: „I believe”) például a  $+5$  értékkel jellemezzük. Aki az (1a) megnyilatkozást teszi, az elkötelezi magát amellett, hogy jelenlegi tudása szerint (ahol a „jelenlegiséget” a ReALIS-publikációkban a jelöletlenség jelzi) egy szóban forgó személy hazautazott, semmi kétség ( $+5$ ). Felszólítás esetén (1b) ugyanez az Állás a feladó szerint egyértelműen hamis ( $-5$ ), hiszen éppen ennek megváltoztatására kíván lépéseket tenni. Kérdés esetén a feladó tudását úgy jellemezhetjük, hogy sem igaznak, sem hamisnak nem gondolja az adott Állást; amit a  $\{-4, -3, -2, \dots, +4\}$  számhalmazzal reprezentálhatunk, ezt jelöli a „0” „nulla körüli számhalmaz”. Az igazságértékelés általánosításába tehát még a tizenegy egész szám halmazának részalmazait is beleértjük. Ezen a ponton érdemes a korábban említett unifikálhatóságra példát mutatni: ha az (1c) megnyilatkozást olyan beszélő teszi, aki az adott hazautazási esemény eredményállapotához mint Álláshoz  $+2$  valószínűséget társít (olyasmit gondol tehát, hogy *talán hazautazott az az illető*), akkor ő jogosan vette fel a kérdezői feladói szerepet, hiszen  $+2$  eleme a fenti számhalmaznak.

Egy *összetett* címke rekurzívan definiálható: egyfelől egyszerű címkék összefűzéséből áll elő, másfelől már előállított címkék összefűzéséből. Mindhárom profilban releváns a címzett tudásáról való feladói tudás (iB= $+5$ ) vagy feltételezés; a két bizonyossági szint együtt így írható le:  $iB \in \{+5, +4, +3, +2, +1\}$ . Majd a három profilban elválik, hogy milyen értékelést kap az uB („you believe”) címkéjű egyszerű világocska (az iBuB címkéjű világocskán belül). Ahogy a táblázatból kiolvashatjuk, kijelentés esetén a címettről a biztos tudás hiánya feltételezendő, kérdés esetén viszont az akár pozitív, akár negatív biztos tudás (ezt jelöli az előjel nélküli ötös szám), a felszólításnak pedig az a jó alapja, ha mindkét beszédrésztvevő tudatában van az adott Állás hamis voltának.

Hogy ne maradjon a kép teljesen hiányos, összefoglaljuk az ismertetett episztemikus háttérhez képest a klasszikus pragmatikában döntőnek tartott szándékkritériumot (Searle 1976) (amelynek bevezető szelvénye mindhárom esetben: iIu↑: „szándékom, hogy a megnyilatkozás kimeneti hatásaként szándékoz legyen...”). Felszólítás esetén a feladó, aki úgy ítéli meg, hogy közös tudása a címettel, hogy egy Állás nem áll fenn, arra próbálja rábíjni, hogy szándékozzon a megnyilatkozás hatására elősegíteni az Állás fennállását. Kérdés esetén az az elvárás a címettől, hogy a feltételezhetően nála lévő biztos tudást az Állásról mutassa meg a feladónak is. Kijelentés esetén pedig azt a kimeneti szándékot várja el a feladó a címettől, hogy az fogadja be a tőle kapott igazságértéket. Mindhárom esetben a megnyilatkozásnak mint egy automata-átmenetszabály kimenetének az a legalapvetőbb formája, hogy a hallgató Ágens a címetti szerep részeként a kimeneti Fázisban immár rendelkezik az elvárt szándékkal. További *default* fázismódosulás: a felébredt szándék tette váltsa, ami a kijelentés és a kérdés esetében egy  $+5$  vagy  $-5$  igazságérték megfelelő világocskában való hozzárendelése egy Álláshoz, a felszólítás esetében pedig tipikusan a „külvilágban”

<sup>1</sup> Az Orákulum számára csak ez a két klasszikus igazságérték áll rendelkezésre, mivel a külvilágban egyértelműnek definiáljuk, hogy mely Esemény milyen Állásban van.

kell valaminek megtörténnie, például valakinek hazautaznia. Mivel az egységesség kedvéért a külvilág információtartalmát egy Orákulum információállapotaként reprezentáljuk, a külvilágbeli történés egy olyan megnyilatkozásként definiálható, amellyel a releváns szándékot hordozó Ágens „értesíti” az Orákulumot egy Állásról, az Orákulum pedig beépíti ezt a tudásába.

Felhívjuk azonban ezen a ponton a figyelmet arra, hogy az emberi kommunikációról nem kapnánk jó modellt az imént sugallt mechanizmussal, miszerint a Fázisok mint állapotok között a megnyilatkozások – intenzionális profiljuk unifikálódásával – mindig determinisztikusan állítják elő a kimenetet. Dinamikus pragmatikai megközelítésünk kértéteggősége azt is megengedi, hogy a Felhasználó önállóbb döntést tulajdonítson az Ágenseknek a fentiekben defaultként jellemzett átmenetek automatikus megvalósításánál. Az 5. szakaszban térünk erre vissza.

#### 4. A diskurzusjelölők mint átmenetszabály-módosító elemek

Ebben a szakaszban négy *diskurzusjelölő* intenzionális profiljának döntő elemeit mutatjuk be. Az a definitív eleme a diskurzusjelölőségnek (Aimer és mtsai 2006), hogy nem befolyásolja a mondat deskriptív, azaz igazságfeltételekkel jellemezhető tartalmát – amin a külvilághoz képest tekintendő igazságértékelés értendő. A beszéd-résztevői világocskákban nagyon is változik az értékelés a gazda-mondattípushoz képest; éppen ez a diskurzusjelölők feladata: bepillantást engedni a belső attitűdökbe. A programot gazdagító nyelvész felhasználó a gazda-profilok világocskacímke-halmazának módosításával vezethet be diskurzusjelölői profilokat, amelyek aztán a Fázisok között új átmenetszabályokként funkcionálnak. Lássuk hát a kiválasztott négy diskurzusjelölőt:

- (2) a. *Valószínűleg* hazautazott.
- b. *CSAK* hazautazott!
- c. *Szerintem* utazzon haza!
- d. Hazautazott, *ugye?*

2. táblázat. Diskurzusjelölők hozzájárulása a fő profilokhoz (Alberti és mtsai 2021, Szeteli és mtsai 2019, Gyuris 2019)

Kijelentő		Felszólító	Kérdő
<i>valószínűleg</i>	metakognitív <i>csak</i>	<i>szerintem</i>	<i>ugye</i>
$iB=\alpha' \in '+4$	$iB=+5$	$iB=-5\gamma$	$iB=\eta \in '+4$
$iBuB\leq\alpha'$	$iBuB=+5$	$iBuB\in "+5\bullet\gamma$	$iBuBiB\in "+5\bullet'+5\bullet\eta$
	$iB\geq iB\downarrow\downarrow\downarrow iB\downarrow$		
$iBrDuB\uparrow=5\beta$	$iD\downarrow\downarrow\sim iD\downarrow\sim iD\in '+5$	e'-re: $iBrD$	$iBrDiB\uparrow\in\beta=5$
$iBuAuB\uparrow=\beta$		e'-re: $iBuA$	$iBuAiB\uparrow\in'+5\bullet\beta$ döntő elem: $iBuB=\xi\in\beta$
$iIuI\uparrow uB\uparrow=\alpha'$		$iIuI\uparrow\in'+4\chi$	$iIuI\uparrow iB\uparrow\in'+5\bullet\xi$
$iAiIuI\uparrow uB\uparrow=\alpha'$		$iAiIuI\uparrow=\chi$	$iAiIuI\uparrow iB\uparrow\in'+5\bullet\xi$

A kérdésbe beépülő *ugye* diskurzusjelölő szerepe ragadható meg a legegyszerűbben. Azt *teszi hozzá* a gazdaprofilhoz, hogy a feladó a pozitív válaszalternatívát valószínűsíti, és ezt a címzettnek is a tudomására hozza. A következő szakaszban alaposabban áttekintjük még a profilt.

A *valószínűleg* hozzájárulása a jelöletlen kijelentés profiljához abban ragadható meg, hogy a feladó a +5-tel jellemezhető biztos tudásnál csekélyebb fokozatú ( $\alpha'$ ) tudással bír egy Állásról, de azt megosztani szándékozik a címzettel, akiről még alacsonyabb szintű információt feltételez az adott Állásra vonatkozóan. Programunk megragadhatja azt a kommunikációs stratégiát, hogy ugyanarról az Állásról több helyről gyűjtünk be információt, és végül elkötelezzük magunkat az igazsága mellett, ha azt több független forrás támogatja. Ehhez a véges állapotú automatát memóriával kell kiegészíteni, hogy tárolni lehessen az adott Állásra vonatkozó, több hullámban érkező információ egyes jellemzőit, például a minimális és a maximális bizonyosság-értéket.

A *szerintem* diskurzusjelölő (Szeteli és mtsai 2019) kijelentő mondatban a *valószínűleg* diskurzusjelölőhöz hasonló hatást fejt ki – nem biztos (nem közvetlen tapasztalatból) származó tudást kínál fel a feladó a címzettnek –, felszólításban is használható azonban. Ez esetben a szándék-profilelemet „enyhíti” ( $i\uparrow u\uparrow \in '+4$ ), tanácsa finomítva a jelöletlen mondatváltozat parancsoló karakterét.

A (2) példásor (b) pontjában hangsúlyosan ejtett *csak* diskurzusjelölő profilja a feladói elme három információállapotát hivatkozva meg (Gyuris 2019, Kárpáti és mtsai 2021), ugyanis annak konstatálását fejezi ki, hogy egy aktuális Fázisban igazolódó igazsága egy Állásnak (amihez tipikusan szélsőséges érzelemmel viszonyul a feladó) egy „bázis” Fázisban” ( $iB\downarrow\downarrow$ ) valószínűnek tűnt, de volt olyan köztes Fázis ( $iB\downarrow\downarrow$ ), amelyben akár az ellentette tűnt valószínűbbnek. A példában szereplő hazautazási eseményhez olyan történetet társíthatunk, miszerint a beszélő egy szóban forgó személytől erősen elvárja a hazautazást, esetleg úgy ítéli meg, hogy egy beszélgetés során a hazautazás szükségességéről meg is győzte, de aztán elbizonytalanodást tapasztal az ügyben – ami végül azért pozitív végkifejletet hoz. A program mint automata ismét valamilyen memóriával egészítendő ki az Állásra vonatkozó értékelések minimumának és maximumának rögzítésére.

## 5. Veremautomata a felhasználói döntési mechanizmus megragadására

Amellett érvelünk kidolgozott pragmatikai példákkal, hogy Alberti és mtsai (2021) 6. szakaszában áttekintett kommunikációs lehetőségek olyan számítástechnikai megoldást kívánnak, amelynek minimális feltétele az eddigiekben felvázolt véges állapotú automata kiegészítése egy legalább veremszerű („Left in First out”) memóriával.

A beszélő Ágens által kiválasztott intenzionális profilban (ha az címzetti szereppel rendelkező profil) arra nézve is vannak kritériumok, hogy a címzettnek kiválasztott hallgató alkalmas-e erre a címzetti szerepre. A hallgató „aktivizálhatja” is magát, opcionális profilelemek *akkommodálása* (információállapotában való létrehozása) révén. A tanulmány mellett érvel, hogy négyféleképpen állnak elő címzetti profilemek – és mind a négy eljárás valamennyi intenzionális profilra alkalmazható.

Az első módozatot címzeti megfelelésnek nevezik. Arról a kézenfekvő eljárásról van szó, hogy tekintjük a kiszemelt intenzionális profil azon  $iB$  prefixumszelvényű világait – azaz  $iB \bullet W$  címkefelépítésű világait –, amelyek címkéjében szerepel az  $u$  (l. 'you'), majd a  $W$  feletti eloszlást kirójuk a hallgatóra mint kritériumot. Vagyis arról a kézenfekvő dologról van szó, hogy azt tekintjük, hogy ahogyan a feladói szerep szerint a címzett mentalizációja elő van írva, az ténylegesen teljesül-e a hallgatóra.

Az előző szakaszban tárgyalt *ugye*-kérdés intenzionális profilja fogja ebben a szakaszban végig a példáinkat szolgáltatni, és egy anya és kamasz lánya közötti párbeszéd szemlélteti majd a formulákat, amit a külföldön konferenciázó anya indít el mobiltelefonon a (3) példaszor (a) pontjában bemutatott *ugye*-kérdéssel. A címzettnek kiválasztott hallgatójáról annyit kell tudni, hogy – az anyjának tett ígéret ellenére – anyja távollétét kihasználva elutazott egy fiúval egy nem éppen jóhírű több napos fesztiválra.

- (3) a. Esztikém, ugye otthon vagy?  
 b.  $iBuB=5 \rightarrow iB \bullet uB=5 \rightarrow uB \in 5$  (konkrétan  $uB=5$ )  
 b'.  $iBuBiB \in '+4 \rightarrow iB \bullet uBiB \in '+4 \rightarrow uBiB \in '+4$   
 b''.  $iBuDiB \uparrow = ? \rightarrow uDiB \uparrow = -5 \bullet 5$   
 c.  $iIu \uparrow iB \uparrow = \beta' \in 5 \rightarrow iIu \uparrow iB \uparrow = \beta' \rightarrow u \uparrow iB \uparrow = \beta' \rightarrow u \uparrow iB \uparrow = \beta' (= -5)$   
 d.  $iB \in '+4 \rightarrow uB \uparrow iB(\uparrow) \in '+4$   
 e.  $iB \in '+4 \rightarrow uB \uparrow i \uparrow uB \uparrow iB(\uparrow) \in '+4$  (közben  $uB \uparrow iB \uparrow = -5$ )

A (b-b'') sorokban az *ugye*-kérdés profiljának három olyan elemét tekintjük, amelyekben „szó esik” a potenciális címzetről, és lépésről lépésre végigkövetjük az  $iB$  címkeprefixumok törlését.

A (b) azt mutatja, ahogyan abból a profilelemből, hogy „a feladó biztos benne vagy valószínűsíti, hogy a címzett ismeri a választ a feltett kérdésre”, eljutunk az ideális hallgatóra vonatkozó azon profilelemig, miszerint ő tényleg „ismeri a választ a feltett kérdésre”. A kérdés jellegéből adódóan – regényes, horrorisztikus vagy fantasztikus helyzetektől eltekintve – triviálisan teljesül tetszőleges hallgatóra, hogy alkalmas a címzeti szerepre, hiszen azt tudnia kell mindenkinek, hogy otthon van-e éppen, vagy sem. Eszti (bár ő ennek nem örül) e kritérium alapján alkalmas a címzeti szerepre, hiszen pontosan tudja, hogy nincs otthon.

A (b')-ben azt a kritériumot vesszük górcső alá, miszerint a címzettel szemben az az elvárás, hogy ne lepődjön meg az *ugye*-kérdésen. Ne lepődjön meg azon, hogy a beszélő egyfelől nem ismeri az adott Állás igazságértékét, másfelől viszont erős elfogultsággal valószínűsíti annak igaz voltát. Nos, Eszti e kritériumnak is nyilvánvalóan megfelel – és ezt az anyja joggal feltételezheti is –, hiszen ígéretet tett arra, hogy alapvetően otthon lesz, de az nyilván nem zárható ki, hogy leugrik vásárolni, vagy átugrik egy barátjéhez.

A (b'') sorban egy súlyozott átlagérdeket veszünk szemügyre (amit e tanulmányban eddig még nem hoztunk szóba) – azt, hogy kinek is vágya/érdeke a beszélő információállapotának felfrissülése az adott Állás igazságértékével. Önmagában az  $iBuDiB \uparrow$  világra vonatkozóan nincsen kritérium, csak a beszélő által figyelembe vett  $R$  szereplőhalmazbeli emberek összességére. Amennyiben a konkrét példában az anya az  $R$  halmazt úgy határozta meg, hogy  $R = \{i\}$ , akkor Eszti címzeti alkalmasságát ez a profilelem nem is érinti. Amennyiben az anya abban reménykedik, hogy a lánya büszke

lesz, hogy igent mondhat a kérdésre ( $R = \{i, u\}$ , és  $iBuDiB \uparrow = +4 \bullet +5 \bullet 5$ ), akkor bizony hiba van a számításában (hiszen  $uDiB \uparrow = -5 \bullet 5$ , Eszti a legkevésbé sem vágyik anyja tájékoztatására), és ezzel a kalkulációk arra az eredményre vezettek, hogy Eszti mégsem alkalmas a címzetti szerepre. Bár ezt az alternatívát (misperint gyanakvó anya esetén Eszti jó címzett, naiv anya esetén nem) az első látásra a  $\Re$ ALIS-megközelítés előnytelen vonásának ítéelhetjük, egyáltalán nem az, hanem finom figyelembe vétele a legapróbb tényezőknek is.<sup>2</sup>

A hallgató címzetti profiljának felépítésében a második módozat egyetlen profilelemet érint: a szándék-kritériumot. A *centrális szándék megvalósulása* – így nevezik meg, kiszámításának módja pedig azon alapul, hogy töröljük a szándék-profilelem világcímkejéből az  $iL$  prefixumot. Ami az *ugye*-kérdés esetében marad (c): a kérdés hatására a címzettben felébred a szándék, hogy a feladó információállapotát felfrissítse az ő fejében meglévő, az Adott Állásra vonatkozó igazságértékkel.

Ezzel Eszti válaszára kerül. Ha őszintén elmondja, hogy nincs otthon, akkor ezzel felvette a címzetti szerepet, és elindul az az eseményszerkezeti láncreakció, amelyben az  $uL \uparrow$  törlésével a címzetti szándék (mint felkészülési szakasz) helyett az eredményállapothoz jutunk: az anya megtudja, hogy lánya nincs otthon. Ha hazudik a lány, akkor nem vette fel a címzetti szerepet, de ezt a kérdező (legalábbis közvetlenül) nem veheti észre.

A címzetti megfelelés és a centrális szándék megvalósulása kötelező elemei a címzetti profilnak. További két stratégiával opcionálisan vonhat le következtetéseket a hallgató (és akkommodálhatja azokat a kimeneti információállapotában).

Az egyik stratégiát *profilelem-tudatosításnak* nevezik. Mivel az intenzionális profilelok konvencionálisak, a hallgatónak lehetősége van úgy felfogni a beszélő által megmutatott profilt, mint a beszélő feltárulkozását, információállapota bizonyos elemeinek nyilvánossá tételét. Bármely profilelem tudatosítható, és ennek formális vetülete: az  $uB \uparrow$  prefixumot hozzáilleszteni az adott profilelemhez (megőrizve annak általánosított igazságértékelését is).

A (d) pont a beszélői tudás profilelemének tudatosítását szemlélteti. Eszti ennek révén ilyen gondolathoz juthat el: „anyám majdnem biztos abban, hogy otthon vagyok, tehát bízik az adott szavamban; egészen biztos persze nem lehet benne, hogy ebben a pillanatban is otthon vagyok (ami még nem feltétlenül gyanakvósból adódik)”.

Egy *ugye*-kérdésből lehet tehát tanulni! „Finnország fővárosa Helsinki, ugye”? – kérdezi mondjuk valaki a földrajzból amúgy igen gyenge hallgatót; aki számára ez a bizonytalan információ is előrelépés lehet ahhoz képest, hogy korábban tippje sem lett volna, ezért elraktározza az információt. A címzetti alkalmatlanság ellenére tehát nem marad hatás nélkül a megnyilatkozás. Ebből is kitűnik, hogy egy determinisztikus véges állapotú automata, amelyik egy ilyen helyzetben elutasítaná az állapotváltást (azaz egy új Fázisba való átlépést) még alkalmatlan az emberi kommunikáció finomabb jelenségeinek a megragadására.

<sup>2</sup> A  $\Re$ ALIS túllép azon az első látásra kézenfekvőnek látszó megközelítésen, hogy egy kérdésre való válasz elnyerése kizárólag a kérdezőnek áll érdekében; ehelyett a bekezdésben felvázolt meglehetősen összetett profilelem mellett érvel (Alberti és mtsai 2021). A programunk azt a felhasználóbarát eljárást követi, hogy defaultként a kérdező érdekltségét ajánlja fel, de ez felülírható.

A negyedik akkommodációs stratégia a *gyanakvás*. Bármely feladói profilelemet opcionálisan elláthatunk – a formalizmus nyelvén fogalmazva – egy  $uB\uparrow il\uparrow uB\uparrow$  háromemeletes címkeprefixummal. A forma is mutatja, tulajdonképpen a profilelem-tudatosítás továbbfejlesztéséről van szó: a hallgató a profilelem közvetlen tudatosítása helyett azt tudatosítja, hogy a beszélőnek lehet olyan szándéka, hogy az adott profilelemet elhitheti vele.

Az (e) sorban azt a lehetőséget vázoljuk fel, miszerint Eszti elkezd gyanakodni, hogy az anyja nagy aggodalmában előbb hazajött a konferenciáról, már otthon ül azon tanakodva, hogy hol lehet a lánya, de közben azt akarja vele elhithetni, hogy nincs otthon és bízik az adott szóban.

A tanulmány leszögezi, hogy a kommunikáció során figyelembe vehető „lehetséges lehetséges világaink” száma a bemutatott eljárások révén megsokszorozódik, a profilalkotó címkerendszerre alkalmazható szigorúan kompozicionális eljárásokkal. Ezeket egy program lokálisan könnyen kezeli, hiszen egyszerű szimbólummanipulációt igényelnek, globálisan azonban problémát jelent minden egyes megnyilatkozás mint átmenetszabály hatását a megfelelő Ágens információállapotának ilyen mértékű gyarapodásával megragadni. E ponton is egy legalább veremszerű memória szükségessége igazolódik, ahol a megelőző Fázisok bizonyos adatai még elérhetőek.

## 6. A program működése

A program C# nyelven íródik, alapvetően két főbb részre különíthető el a működés szempontjából.

Az első rész az adatok kezelésére, eltárolására vonatkozik. Mivel az egyik fő cél az egyes tudásállapotok szűrése személyre, eseményre, képletre, ezért egy sqLite adatbázisban tároljuk az adatokat, megkönnyítve a duplikátumok elkerülését, illetve magát a szűrést is. Továbbá az egyes adatmodellek közti kapcsolatok tárolására is az adatbázis szolgál.

A másik rész, az adatok elérésével és használatával foglalkozik, ahol az egyes személyekhez különböző tudásállapotokat tárolhatunk, szerkeszthetünk. Vizuális felület biztosítja a nyelvészek számára az adatok feltöltését (2. ábra), illetve minden felhasználónak az alapvető funkciókat (3. ábra). A már eltárolt tudásállapotokat felhasználva itt értékelést is kapunk a feladói, illetve hallgatói szerepeket felvevő személyek alkalmasságáról.

Ez a többretegű felosztás többek közt lehetőséget ad egy későbbi multiplatform rendszer létrehozására.

## 7. Összegzés, alkalmazások, a továbblépés útjai

A projekt elsődleges hasznát a *ReALIS*-megközelítés létjogosultságának alátámasztásában látjuk. Egy formális elmélet adekvátságának legfőbb igazolása az, ha az eszköztárára alapítva algoritmusok, majd programok írhatóak, amelyek az elmélet által leírt jelenségek szimulálásra is alkalmasak.

A *ReALIS*: 'reciprocal and lifelong interpretation system', egy kétrétegű dinamikus pragmatikaelmélet. Egy rekurziós alapon felépülő címkerendszer révén részben rendezett lehetségesvilág-halmazban folytat általánosított igazságértékelést a program, így megvalósítva a montagoviánus *interpretációt*. A *kölcsönösséget* az összetett címkejű

világocskák ragadják meg, ahol a multiágens rendszer Ágenseinek egymásról való gondolatai tárolódnak, a *dinamikus* karaktert pedig az, hogy alapjában véve egy olyan véges állapotú automata átmenetszabályaiként ragadja meg a program a megnyilatkozásokhoz társuló intenzionális profilokat, amelynek állapotai a külvilág és az arról (meg egymásról) gondolkodó Ágensek információállapotainak pillanatképeiből összeálló Fázisok.

A *kétretegűség* pedig abban ölt testet, hogy a nyelvi konvenciórendszer részének tekintett intenzionális profilokban definiált feladói és címzetti szerepek egyrészt a fázisonkénti átmenet feltételeként szolgálnak – unifikálhatónak kell lenniük a bemeneti fázisban bizonyos beszélői/hallgatói információállapot-szegmensekkel –, másrészt meg is határozzák a kimeneti fázist az érintett információállapotoknak a sikeres unifikáció eredményeivel való felfrissítése révén.

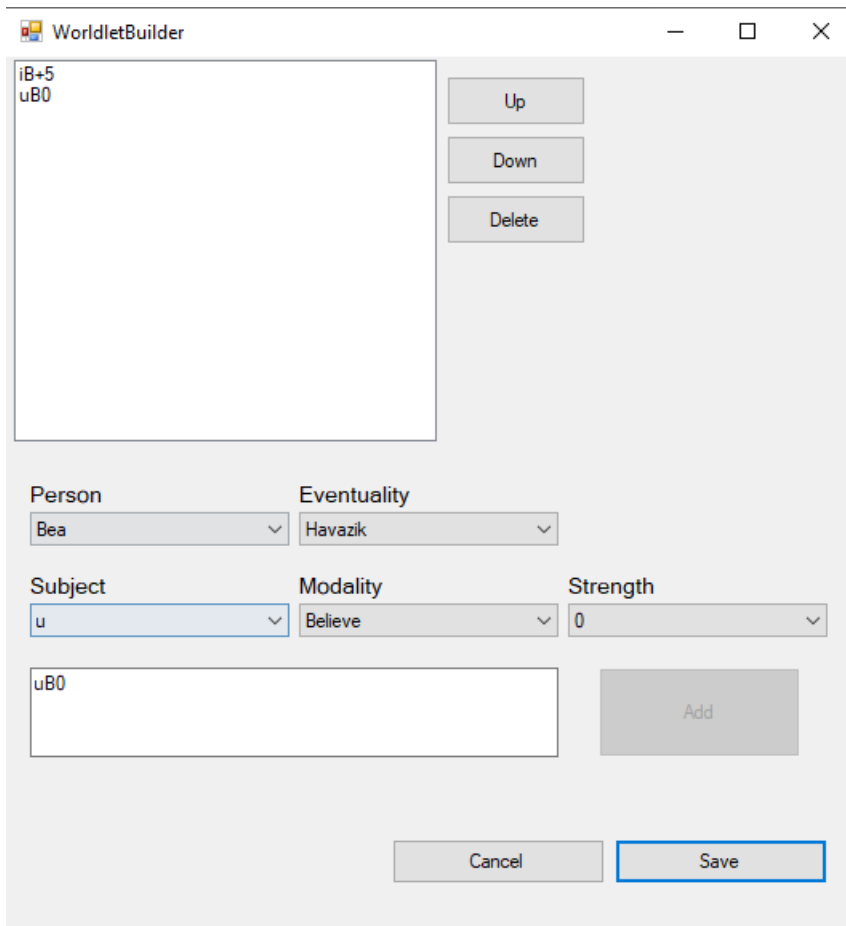
Egy véges állapotú automata azonban nem elégséges annak a jelentős szabadságnak a megragadására, amellyel egy beszélő dönthet, hogy milyen intenzionális profilt alkalmaz, illetve a hallgató is dönthet, hogy felveszi-e az abban meghatározott címzetti szerepet, illetve milyen profilelemeket akkommodál és generál a neki címzett profil alapján. Többféle pontozást is kidolgoztunk a „legilleszkedőbb” profil- és szerepválasztások jutalmazására, illetve a kevésbé vagy egyáltalán nem illeszkedők büntetésére (beleértve ezeket a blöffölést vagy akár a hazugságot is), ami alapján a felhasználók – avataruknak választva egy-egy Ágenst a multiágens rendszerből – játékként is működtethetik a programot. A Rizikóhoz hasonlóan, előre kihúzott titkos küldetések megvalósításán munkálkodhatnak (pl. „Hitessük el Rózsival, hogy Rezső hazautazott!”), miközben vigyázniuk kell, hogy nehogypontozódjanak.

Ami a továbblépés útjait illeti, a *ReALIS élethossziglani* karakterét kell majd beépíteni a programba. A (4) példasor (a) mondatának alkalmazását az legitimálja, ha a beszélő meghatározott tényekből meghatározott következtetési szabályokkal jut el az adott Állás valószínűsítéséig (Szeteli és mtsai 2019); például ha az alany kollégista, akkor bizonyos tárgyainak a kollégiumi szobájából való hiánya lehet a beszélő számára megfelelő kiindulópont. A (b) mondat pedig nem kérdés, mert az adott Állást tényként fogadja el; illetve mégis kérdés: „Hogy lehet ez?!” (Kárpáti és mtsai 2021). A válasz pedig olyan körülmény megadása lehet, ami elégséges indokként szolgálhat az Állásra; például az, hogy az előzőekben emlegetett kollégistának megbetegszik az édesanyja. A bekezdésben említett jelenségeket az Események közötti asszociációs hálózatok felépítésével kezelhetjük, aminek kézenfekvő módja lehet, hogy a program „tanulja meg” az általa elemzett szövegekből az Események társulási tendenciáit.

- (4) a. *Szerintem* hazautazott.  
b. *CSAK* hazautazott?!

Ami a program gyakorlati alkalmazását illeti, sok területen óriási jelentőséggel bír az, hogy ne csak rögzítsük a világ valamely területére vonatkozó információt, hanem aszerint (is) rendszerezzük, hogy ki miről honnan tud, ki mely részeihez jogosult hozzáférni, ki milyen módosításokban érdekelt vagy ellenérdekelt, kinek milyen módosításra van joga és lehetősége; mindehhez „érdemes a humán intelligenciából a lehető legtöbbet ellesni” (Alberti és Nóthig 2015).





2. **ábra:** Vizuális felület világoscakímkék szerkesztéséhez.



3. ábra: Vizuális felület az alapvető funkciókhoz.

## Köszönetnyilvánítás

Az Innovációs és Technológiai Minisztérium ÚNKP-21-2-I kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.

## Hivatkozások

- Szeteli, A., Alberti, G., Kleiber, J., Dóla, M.: The World is Built with our Words to Each Other – Basic and Fine-Tuned Intensional Profiles in Hungarian. XIV. MSZNY pp. 78–88 (2018)
- Nóthig L., Szeteli A.: Nagyfelbontású pragmaszemantikai igazságértékelés egy játékprogramban. XIV. MSZNY pp. 456–465 (2018)
- Alberti G.: *ReALIS*: Interpretálók a világban, világok az interpretálóban. Akadémiai Kiadó, Budapest (2011)
- Farkas, D., Roelofsen, F.: Division of labor in the interpretation of declaratives and interrogatives. *Journal of Semantics* 34(2) pp. 237–289 (2017)
- Alberti G., Dóla M., Kárpáti E., Kleiber J., Visket A., Szeteli A.: Lehetőség lehetőség világaink. *Jelentés és Nyelvhasználat* 8/1 pp. xx–yy (2021)
- Szeteli, A., Dóla, M., Alberti, G.: Pragmasemantic Analysis of the Hungarian Inferential – Evidential Expression 'szerint'. *Studies in Polish Linguistics* 14 pp. 207–225 (2019)
- Gyuris B.: A hangsúlyos *csak* diskurzuspártikula interpretációja. In: Maleczki M., Németh T. E. (szerk.): *A mai magyar nyelv leírásának újabb módszerei* 7. SZTE, Szeged, 157–179 (2009)
- Kárpáti, L., Kleiber, J., Alberti, G.: The polysemous system of the Hungarian *csak* 'only', with special emphasis on its metacognitive use. Talk at the Linguistics Beyond and Within Conference. Lublin, 14–15 October (2021)
- Benz, A.: Chains and the Common Ground. In: Poesio M., Traum D. (eds.) *GötaLog 2000 — Gothenburg Papers in Computational Linguistics* 00–5 pp. 181–184 (2000)
- Farkas, J., Ohnmacht, M. Aspect and Eventuality Structure in a Representational Dynamic Semantics. In Alberti, G., Farkas, J., Kleiber, J. (eds.) *Vonzásban és változásban* pp. 353–379. Doctoral School of Linguistics at Univ. of Pécs, Hungary (2012)
- Dowty, D. R., Wall, R. E., Peters, S.: *Introduction to Montague Semantics*. Reidel, Dordrecht (1981)
- Kamp, H., Genabith, J., Reyle, U.: Discourse Representation Theory. In: Gabbay D., Guenther F. (eds.) *Handbook of Philosophical Logic* 15 pp. 125–394 Springer-Verlag, Berlin (2011)
- Maier, E.: Attitudes and Mental Files in DRT. *Rev. Phil. Psych* 7 pp. 473–490 (2016)
- Austin, J. L.: *How to Do Things with Words*. Clarendon Press, Oxford (1975/1962)
- Partee, B. H., ter Meulen, A., Wall, R. P.: *Mathematical Methods in Linguistics*. Dordrecht, Kluwer (1990)
- Oishi, E.: Discursive functions of evidentials and epistemic modals. In: Cantarini, S. et al. (eds.) *Certainty-uncertainty – and the attitudinal space in between*. *Studies in Language Companion Series* 165 pp. 239–262 Benjamins, Amsterdam (2014)
- Searle, J. R.: A Classification of Illocutionary Acts. *Language in Society* 5(1) pp. 1–23 (1976)
- Grice, P.: Logic and conversation. In: Cole P., Morgan, J. L. (eds.) *Syntax and Semantics* 3. pp. 41–58 Academic Press, New York (1975)
- Lauer, S.: *Towards a Dynamic Pragmatics*. Dissertation at Stanford University (2013)
- Alberti G.: A világ nyelvei, a nyelv világai. In: Böhm G., Czeferner D., Fedeles T. (eds.) *Bölcsész Akadémia* 4. Pécs, BTK KTDI pp. 126–150 (2020)

XVIII. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2022. január 27–28.

- Aijmer, K., Foolen, A., Simon-Vandenberg, A-M.: Pragmatic markers in translation: a methodological proposal. *Approaches to Discourse Particles 1* pp. 101–114 (2006)
- Alberti, G., Nóthig, L.: HumInA projektsoport a ReALIS1.1 bázisán. *XI. MSZNY* pp. 326–332 (2015)

# "Az invazív medvék nem tolerálják a szukis agressziót"

## Magyar GPT-2 kísérleti modell

Yang Zijian Győző

Nyelvtudományi Kutatóközpont  
1068 Budapest, Benczúr u. 33.  
yang.zijian.gyozo@nytud.hu

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport  
1083 Budapest, Práter u. 50/a.  
yang.zijian.gyozo@itk.ppke.hu

**Kivonat** A GPT-2 egy kizárólag dekóderrel rendelkező autoregresszív transzformer modell, amely elsősorban szöveggeneráló feladatokra alkalmas. A kutatásomban betanítottam egy kísérleti GPT-2 modellt magyar nyelvre, majd azt tovább finomhangoltam különböző nyelvtechnológiai feladatokra. A kísérleteimben az előtanított GPT-2 modellel mondat-szintű és tokenszintű szövegosztályozás, absztraktív szövegösszefoglaló generálás, hírgenerálás és versgenerálás feladatokra finomhangoltam. Az eredmények alapján a GPT-2 kevésbé teljesít jól szövegosztályozás feladatokra. Absztraktív szövegösszegzés feladatában versenyképes eredményeket ért el, azonban a szöveggenerálás eredményei még érdekesebbek. A GPT-2 modellemmel létrehoztam egy hír- és egy versgenerátort.

**Kulcsszavak:** GPT-2, mondat-szintű osztályozás, tokenszintű osztályozás, szentiment analízis, NER, NP, absztraktív összefoglaló generálás, szöveggenerálás, hírgenerálás, versgenerálás

## 1. Bevezetés

A BERT alapú modellek kizárólag enkóderrel rendelkeznek, amelyek kiválóan alkalmasak különböző osztályozási feladatokra, mint például mondat-szintű vagy tokenszintű osztályozásra. Azonban dekóder hiányában kevésbé alkalmasak szöveggenerálásra, amelyhez általában egy dekóder hozzácsatlakoztatásával oldják meg a problémát. Ezzel ellentétben az autoregresszív modelleket elsősorban szöveggenerálásra tervezték, a közös tulajdonságuk, hogy dekóderrel rendelkeznek, amely a tanítás során mindig csak a szöveg bal oldalát látja, így megtanulja a következő szó/szavak predikciójának képességét. Ilyenek a Facebook OpenAI<sup>1</sup> GPT modelljei (Radford és Narasimhan, 2018), amelyek csak dekóderrel rendelkeznek.

<sup>1</sup> <https://openai.com>

A kutatásom során betanítottam egy kísérleti GPT-2 modellt, majd azt tovább finomhangoltam különböző feladatokra. A GPT-2 modellel mondat- és tokenszintű szövegosztályozás, absztraktív összefoglalás, hír- és versgenerálás feladatain tanítottam. A címben szereplő szokatlan mondatot az egyik finomhangolt GPT-2 modellem (SZUM modell) generálta. Külön érdekesség, hogy az idézőjeleket is a modell generálta.

Bár a GPT modellek egyik alapja a nagy mennyiségű adat használata, azonban erőforrások hiányában kisebb adatmennyiségen tanítottam a GPT-2 modellel, ezért is neveztem el kísérleti modellnek. Azonban kevesebb adattal is rendkívül látványos eredményeket értem el.

Modelljeim és szkriptjeim megtalálhatóak a Github<sup>2</sup> és Hugging Face<sup>3</sup> oldalakon.

## 2. GPT modellek és Kapcsolódó irodalom

A GPT (generative pre-training) (Radford és Narasimhan, 2018) egy kizárólag dekóder (decoder-only) tartalmazó transformer modell. A kizárólag dekóder architektúra abban különbözik az enkóder-dekóder architektúra dekóderétől, hogy hiányzik belőle az enkóder-dekóder figyelmi (encoder-decoder attention) réteg, mivel nincsen enkóder.

Az OpenAI<sup>4</sup> kutatócsoport projektje az úgynevezett félig felügyelt tanítás (semi-supervised learning) kategóriájába tartozik. A paradigma már eddig is komoly érdeklődésre tett szert, főleg az olyan alkalmazások terén, mint a szekvenciacímkezés (sequence labeling) vagy a szövegosztályozás (text classification). Nemrégiben számos kutatási eredmény rávilágított arra, hogy a szövegbeágyazásos (word embedding) (Collobert és mtsai, 2011) eljárások is komoly mértékben javíthatják a modellek teljesítményét. Mindezek azonban a szavak szintjén értelmezhető megközelítések, az OpenAI csapata azonban ennél magasabb szintű szemantikai vizsgálgódot tűzött ki célul. Ezeknek a megoldásoknak egyre nagyobb szerep jut, ugyanis a szavaknál magasabb szintű beágyazások a nagyobb szövegsegységek vektorrepresentációját teszik lehetővé. A felügyelet nélküli előtanítás (unsupervised pre-training) lingvisztikai információk megragadását teszi lehetővé, megfelelő transzformer háló kiválasztásával pedig hosszabb távú információkinyerés is megvalósítható. A tanítási segédobjektívák (auxiliary training objectives) használata a teljesítmény növekedésével járhat, mint például a Rei által bemutatott szekvenciacímkezés feladat esetén (Rei, 2017).

A nyelvi modellek képesek megtanulni számos nyelvelemzési feladat felügyelet nélküli megoldását, amennyiben kellően nagy méretű adathalmazon vannak betanítva, mint például a több millió honlap tartalmát felölelő WebText. A GPT-2 (Radford és mtsai, 2019) modell egy 1,5 milliárd paraméterrel rendelkező transzformer modell, amely nyolcból hét feladaton csúcsteljesítményt ért el az úgynevezett zero-shot üzemmódban. A GPT-2 választ próbál adni arra a

<sup>2</sup> <https://github.com/nytud/neural-models>

<sup>3</sup> <https://huggingface.co/NYTK>

<sup>4</sup> <https://openai.com>

kérdésre, hogy milyen módon hat a nagyobb adathalmazon való tanítás a nagy méretű nyelvi modellek teljesítményére. Számos hasonló munka próbálja ezt a megközelítést kísérleti úton átültetni a gyakorlatba. Például Jozefowicz és mtsai (2016) RNN-alapú (Recurrent Neural Network) nyelvi modelleket teszteltek az 1 milliárd szavas referencia feladaton (1 Billion Word Benchmark). A csoport kiemelkedő eredményei között szerepelnek, hogy az egyik modelljük 51,3-ról 30,0-ra javította a perplexitást, míg egy másik modellegyüttesük új csúcstól állított fel perplexitás terén 41,0-ról 23,7-re csökkentve azt (Jozefowicz és mtsai, 2016). Bajgar és mtsai (2017) egy Attention-Sum Reader modellt használtak, amelyet a Book Test adathalmazon tanítottak be, amely hatvanszor akkora méretű, mint a korábban megjelent, hasonló felépítésű Children’s Book Test<sup>5</sup> (CBT) adathalmaz. Kiemelendő azonban, hogy a Book Test adathalmazon betanított modell pontosabb eredményeket ért el a betanítást követően a hatvanszor kisebb CBT-n (Bajgar és mtsai, 2017). Hestness és mtsai (2017) több gépi tanulós modell teljesítményét vizsgálták a modell kapacitásának és a tanító adathalmaz méretének függvényében. A csoport eredményei összhangban állnak a GPT-2 esetében megfigyelt tendenciákkal (Hestness és mtsai, 2017). Karpathy és mtsai (2016) az RNN-alapú modellek, azon belül is a Long-Short Term Memory (LSTM) variáns kiemelkedő teljesítményének hátterét kutatták, továbbá az utóbbi modellvariánszt idézetek, zárójelkelemek valamint sorhosszúság vizsgálatára alkalmazták (Karpathy és mtsai, 2016). Fontos kiemelni azt is, hogy a Liu és mtsai (2018) által jegyzett publikációban a Wikipédia cikkek generálására betanított kizárólag de-kóder architektúrájú modell megtanulta lefordítani a neveket nyelvek között (Liu és mtsai, 2018).

A nyelvi modellek generatív vagy egyéb feladatok megoldása során mutatózó teljesítményének javítására többféle stratégia is lehetséges. Az egyik út a paraméterszám és a számítási kapacitás együttes növelése, a másik kizárólag a paraméterszám emelésére törekszik, a harmadik stratégia pedig csak a számítási kapacitást variálja. Számos példa létezik transzformer-alapú modellek paraméterszámának és azzal együtt közel arányosan a FLOPS per token arány szisztematikus növelésére, és ezáltal jobb teljesítmény elérésére. Kísérleteinkben azt vizsgálták, hogy a 300 milliós (Devlin és mtsai, 2019), 8,3 milliárdos (Shoeybi és mtsai, 2020) és 11 milliárdos (Raffel és mtsai, 2020) paraméterszámú modellek milyen hatást gyakorolnak a teljesítményre nézve. Számos munka törekedett a teljesítmény javítására kizárólag a paraméterszám növelésével, azaz a modell információtároló kapacitását megnövelték a számítási kapacitás emelése nélkül. Erre példa a TED előadások szöveges átiratait tartalmazó korpuszon betanított több nyelvről több nyelvre fordító modell, amely 50 illetve 100 milliárd paraméterrel üzemel (Aharoni és mtsai, 2019). A harmadik stratégiát képviselő, kizárólag számítási kapacitás növelésével operáló modellfejlesztésre példa az adaptív számítási idő (adaptive computation time, ACT) algoritmus alkalmazása RNN-alapú modelleknél oly módon, hogy optimalizálja a számítási lépések számát a bemenet beérkezése és a kimenet kiküldése között (Graves, 2017). Hasonló módon számítási kapacitást növelnek a Transzformerek és RNN-alapú modellek

<sup>5</sup> <https://paperswithcode.com/dataset/cbt>

előnyös tulajdonságait ötvöző Univerzális Transzformerek (Dehghani és mtsai, 2019).

A GPT-3 (Brown és mtsai, 2020) az első stratégiát követi, vagyis a paraméterszám és a számítási kapacitás együttes növelésével éri el a kívánt teljesítményt. 175 milliárd paraméterével, finomhangolás nélkül képes state-of-the-art eredmények elérésére számos feladat tekintetében. Nyelvi modellek hatékony önfelügyelt tanításához kiemelten fontos, hogy megfelelően nagy méretű, annotálatlan korpuszok álljanak rendelkezésre. Erre a célra számos lehetőség mutatkozik, például korábbi munkák különböző nyelvi modellek betanítására használták a Wikipédiát, a Gigaword-öt (Graff és mtsai, 2003), nem publikus Google News korpuszt, RealNews adatbázist (Zellers és mtsai, 2019), vagy a WikiText korpuszt (Merity és mtsai, 2017). A nagy méretű adathalmazok létrehozása egyre inkább a webes adatkivonatolás (internet scraping) technikával történik. Az egyik legtöbbit használt ilyen módon előállított szöveges adathalmaz a Common Crawl<sup>6</sup>. A Common Crawl-t használó projektek sokszor rávilágítanak az ezzel kapcsolatos olyan jellegű felmerülő problémákra, mint az adattisztítás vagy az adatok szűrése. Egyre inkább bevett gyakorlat, hogy több adatforrást kombinálnak össze nyelvi modellek betanításához.

A GPT modellek általános jellemzője, hogy szótárként BPE kódolást használnak. A jelenlegi 3+1 féle GPT modell összehasonlítva:

- GPT: 12 réteg, 12 figyelmi fej; 768 szóbeágyazás méret; 512 szöveghossz; 117 millió paraméter;
- GPT-2: 48 réteg, 12 figyelmi fej; 1600 szóbeágyazás méret; 1024 szöveghossz; 1,5 milliárd paraméter;
- GPT-3: 96 réteg, 96 figyelmi fej; 12888 szóbeágyazási méret; 2048 szöveghossz; 175 milliárd paraméter;
- GPT Neo (Black és mtsai, 2021): mesh-tensorflow könyvtár implementáció, hogy GPT-3 típusú modelleket tudjunk tanítani.

Kutatásom során a GPT-2 modellel kísérleteztem.

### 3. Kísérletek

A kutatásom során egy kísérleti GPT-2 modellt tanítottam elő, majd azt tovább finomhangoltam négy különböző feladatra.

#### 3.1. Előtanítás

A GPT előtanítását egy TDS cikk<sup>7</sup> alapján végeztem el, a cikkben megadott beállításokkal. Egy GPT-2 kísérleti modell került betanításra, amelyhez bekezdésekre bontott magyar Wikipédiát (Nemeskey, 2020) (méretei az 1. táblázatban láthatóak) használtam. A tanításhoz használt hiperparaméterek: blokk méret:

<sup>6</sup> <https://commoncrawl.org>

<sup>7</sup> <https://towardsdatascience.com/train-gpt-2-in-your-own-language-fc6ad4d60171>



100; batch méret: 12; buffer méret: 1000; tanulási ráta: 3e-5; epoch: 10; szótár méret: 33.000. A tanításhoz 4 darab GeForce GTX 1080Ti GPU kártyát használtam. A betanítás időtartama körülbelül 3 nap volt.

### 3.2. Finomhangolás

Az előtanított GPT-2 modellem teszteléséhez az alábbi finomhangolási feladatokat végeztem el:

- Mondatszintű szövegosztályozás: szentiment analízis 2 osztályos (2-CLASS) és 5 osztályos (5-CLASS)
- Tokenszintű szövegosztályozás: névelemfelismerés (NER) és főnévi csoport felismerés (NP)
- Absztraktív összefoglaló generálás (SZUM)
- Szöveggenerálás: hírgenerálás (HÍR) és versgenerálás (VERS)

A modellek tanításához az alábbi korpuszokat használtam fel, a korpuszok méretei az 1. táblázatban láthatóak:

- **SZENT**: A mondatszintű szentiment analízis feladatra a PrecognoX Kft.<sup>8</sup> által készített Hungarian Twitter Sentiment Corpus<sup>9</sup>-t használtam fel, amelyet két részkorpuszra bontottam fel:
  - 2-CLASS: bináris osztályozású részkorpusz. Az értékeket a következőképpen konvertáltam: 1-es és 2-es értékeket 0-ás értékre, mint negatív értékek, 4-es és 5-ös értékeket 1-es értékre, mint pozitív értékek. A 3-as értéket figyelmen kívül hagytam, hogy elkerüljem a félreérthető eseteket.
  - 5-CLASS: Az eredeti 5 osztályos korpusz. 1: nagyon negatív, 2: negatív, 3: semleges, 4: pozitív, 5: nagyon pozitív.
- **NER**: A névelemfelismerés feladathoz a NYTK-NerKor (NerKor) (Simon és Vadász, 2021) és a Szeged NER (SzNer) korpuszt (Szarvas és mtsai, 2006) használtam.
- **NP**: A főnévi névcsoport felismeréshez a Szeged Treebank korpuszt (Csendes és mtsai, 2005) használtam.
- **SZUM**: Az absztraktív összefoglaló feladathoz a H+I korpuszt használtam, amelyet Yang és mtsai (2021) korábbi kutatásaik során is alkalmaztak (Yang és mtsai, 2021). A választásom azért erre esett, hogy össze tudjam hasonlítani az eredményemet a kutatásukkal.
- **HÍR**: A hírgenerálás feladatához a HVG (online és nyomtatott) és NOL (Népszabadság Online - NOL.hu) korpusz (H+N) cikkeit használtam fel. A választás azért esett ezekre a korpuszokra, mert bekezdésekre vannak bontva, amelyeket fel tudtam használni a tanításhoz.
- **VERS**: A versgenerálás feladatához a Petőfi Sándor összes költeményei című művet használtam, amelyet a Magyar Elektronikus Könyvtár oldaláról<sup>10</sup> töltöttem le.

<sup>8</sup> <https://www.precognoX.hu>

<sup>9</sup> <http://opendata.hu/dataset/hungarian-twitter-sentiment-corpus>

<sup>10</sup> <https://mek.oszk.hu/01000/01006/>

	Tanító	Validáció	Teszt
Magyar Wikipédia (bekezdés #)	2.020.956	-	3.000
2-CLASS (szegmens #)	2.468	-	269
5-CLASS (szegmens #)	3.600	-	400
NerKor (mondat #)	1.633.952	211.392	210.884
SzNER (mondat #)	8.484	514	932
NP (mondat #)	65.679	8.209	8.209
SZUM (cikk+lead pár #)	559.162	3.000	3.000
HÍR (cikk #)	589.914	-	65.545
VERS (vers #)	735	-	80

1. táblázat. Felhasznált korpuszok méretei.

**A mondatszintű szentiment analízis** esetében, a Huggingface transformers által rendelkezésre bocsátott „text classification” könyvtárát<sup>11</sup> használtam, a következő módosított paraméterekkel: tanulási ráta: 2e-5; batch méret: 32; maximális szekvenciahossz: 128; epoch: 10. Az eredmény táblázatba (lásd 2. táblázat) a 10 epochból a legjobbat írtam ki. A kiértékeléshez a pontosság (accuracy) metrikát alkalmaztam.

**A NER és NP** feladatokra történő finomhangolásnál a Huggingface transformers „token classification” könyvtárát<sup>12</sup> használtam, a következő módosított paraméterekkel: tanulási ráta: 5e-5; batch méret: 4; maximális szekvenciahossz: 128; epoch: 10. Az eredmény táblázatba (lásd 2) a 10 epochból a legjobbat írtam ki. A kiértékeléshez az F-mérték metrikát alkalmaztam.

**Az absztraktív összefoglalás** finomhangolásához a GPT-2 cikkben (Radford és mtsai, 2019) alkalmazott módszert használtam. A hírek szövegeit és a hozzájuk tartozó leadeket az alábbi formátumba alakítottam át:

– 1 sor: [hír szövege] TL;DR: [lead szövege]

Majd ezt a korpuszt használtam fel egy nyelvi modell finomhangolásához az előtanított GPT-2 modell segítségével. A nyelvi modell finomhangolásához a Huggingface transformers „language modeling” könyvtárát<sup>13</sup> használtam, a következő módosított paraméterekkel: tanulási ráta: 5e-5; batch méret: 4; blokk méret: 512; epoch: 10.

**A hírgenerálás** feladatához H+N korpusz híreinek bekezdéseit használtam, egy sor egy hír, a bekezdések <n> szeparátorral vannak elválasztva. A zajok kiszűréséhez kivettem a tíz és annál több bekezdésből álló cikkeket, valamint a kettőnél kevesebb bekezdésből álló cikkeket. Továbbá végeztem tisztításokat: kivettem azokat a cikkeket, amelyek hivatkozásokat és szkripteket tartalmaztak.

<sup>11</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification>

<sup>12</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/token-classification>

<sup>13</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/language-modeling>

A tanításhoz ugyanazt a nyelvi modell finomhangoló kódot használtam mint, amelyet az absztraktív összefoglaló feladatnál használtam. A finomhangolást a következő módosított paraméterekkel végeztem el: tanulási ráta: 5e-5; batch méret: 5; blokk méret: 512; epoch: 10.

A **versgenerálás** feladatához ugyanazt a könyvtárat használtam, mint a hírgenerálás feladatában, a következő módosított paraméterekkel: tanulási ráta: 5e-5; batch méret: 5; blokk méret: 512; epoch: 10.

### 3.3. Eredmények

A 2-es táblázatban láthatóak az osztályozási (mondat- és tokenszintű) feladatok eredményei. Az összefoglalt eredmények alapján elmondható, hogy a GPT-2 modellek nem tudják felülmúlni a jelenlegi state-of-the-art huBERT modellt. Ez az eredmény várható volt, mivel az általam használt modellek számos jellemzőjük tekintetében gyengébbek (kevesebb tanítóadat, kisebb batch méret stb.). Ennek ellenére, mégis magas eredményeket érnek el. A szentiment analízis kísérletekben például csak 5-7 százalékkal maradnak alul a huBERT-hez képest.

	NerKor (F1)	SzNER (F1)	NP (F1)	2-class (Pontosság)	5-class (Pontosság)
huBERT	90,18	97,51	96,97	85,92	68,50
GPT-2	69,43	88,06	85,02	80,37	61,00

2. táblázat. NER, NP és Szentiment analízis eredmények.

A GPT-2 modellek erősebb oldala a szövegenerálás. A szövegeneráló GPT-2 modellek kvantitatív kiértékeléséhez a perplexitást választottam, mivel az eredeti GPT-2 kutatásban is ezt alkalmazták. A 3. táblázatban láthatóak a modellek perplexitás értékei. Az összefoglaló generáló modell 19,5 perplexitást ért el, ami kiemelendő, hiszen az eredeti GPT-2 kutatásban 18,4-t értek el.

A hírgeneráló modell 22,06 perplexitást ért el, ami gyengébb, mint ami az eredeti publikációban szerepel. Ez a várakozásomnak megfelel, hiszen kevesebb adaton tanítottam, azonban ez nem is probléma, hiszen nem célozom visszaismételné a híreket, hanem a modellnek egy adott témában kreatívan kell magától előállítania tartalmakat. A hírgeneráló modell megtalálható a Hugging Face oldalon<sup>14</sup>.

A versgeneráló modell perplexitása már jóval gyengébb, 47,5 körüli érték. Ez azonban egy vers esetében nem feltétlenül negatívum, hiszen nem szóról szóra szeretném visszaidézni a szerző verseit, a célozom, hogy nagyobb kreativitást adjak a modellnek. A versgeneráló modell megtalálható a Hugging Face oldalon<sup>15</sup>.

A 4. táblázatban látható az absztraktív összefoglaló generálás eredménye. A generálás folyamán megadtam a tesztanyag cikk szövegét, majd utána hozzáillesztettem a „TL;DR:” kifejezést, jelezve, hogy összefoglalót várok generált

<sup>14</sup> <https://huggingface.co/NYTK/text-generation-news-gpt2-small-hungarian>

<sup>15</sup> <https://huggingface.co/NYTK/text-generation-poem-petofi-gpt2-small-hungarian>

	Perplexitás
SUM	19,52
HÍR	22,06
VERS	47,46

3. táblázat. A szöveggeneráló GPT-2 nyelvi modellek perplexitása.

szöveggént. A modellel ezután 3 szöveget generáltattam, majd mindegyik szöveg első mondatát összekonkatenáltam. Így eredményként 3 mondat alkotja a generált összefoglalást.

Az összehasonlíthatóság végett beillesztettem a Yang és mtsai (2021) kutatásából a huBERT alapú PreSumm modell legjobb eredményét. A nemzetközi kutatásokban az F-mértéket (F) szokás megjeleníteni, azonban mind a PreSumm, mind a GPT-2 megoldása több mondatot is visszaad eredményül. A GPT-2 által javasolt megoldásban konkrétan 3 mondatot. Ilyen szempontból releváns a ROUGE fedés értékek megjelenítése, hiszen ha sok szöveget generál, magasabb lesz a fedés (R) és alacsonyabb a pontosság (P). Az eredmények azt mutatják, hogy a GPT-2 által javasolt összefoglalás generáló módszer (ami nem egy „hagyományos” szövegből szöveg – seq2seq – enkóder-dekóder architektúrán alapuló) versenyképes eredményt képes produkálni. Ha az F-mértéket nézzük, ROUGE-1 esetében még magasabb eredményt is el tudott érni a GPT-2.

	PreSumm	GPT-2
ROUGE-1 R	57.07	40,90
ROUGE-1 P	14.63	17,53
ROUGE-1 F	22.42	<b>23,06</b>
ROUGE-2 R	26.97	11,89
ROUGE-2 P	6.64	4,94
ROUGE-2 F	10.24	6,56
ROUGE-L R	48.28	27,46
ROUGE-L P	12.19	11,35
ROUGE-L F	18.72	15,04

4. táblázat. Absztraktív összefoglaló eredménye.

Az 5. táblázatban látható egy példa összefoglaló generálásra. Közelebbről megvizsgálva a generált eredményt más képet kapunk a kvantitatív eredményhez képest. A példában az látható, hogy releváns mondatokat generált a modell, azonban inkább „hallucinációk”, mint összefoglalás. Ez egy nagy különbség egy seq2seq módszerhez képest. A seq2seq nehezen tud kilépni az adott bemeneti szöveg korlátaiból, míg ezzel a nyelvi modellezés módszerrel való összegzés sokkal „kreatívabb”, ami a jelen feladatban inkább hátrányt jelent, mint előnyt. Összehasonlításképpen a PreSumm által generált összefoglalót is betettem, ennek eredményeként az látható, hogy a PreSumm is „hallucinál”, de sokkal kevesebbet és relevánsabb kifejezéseket. A PreSumm generált egy „autóalkatrész-műhely” kife-

jezést, ami nem volt a cikkben, azonban az eredeti leadben látható, hogy valóban egy autóalkatrész-műhely gyulladt ki. Ez egy kifejezetten precíz „hallucináció”. A GPT-2 ehelyett „ruhaterelő cég” és „családi ház” kifejezéseket „hallucinált”, amelyek egyáltalán nem relevánsak.

---

**Cikk szövege:**

Az Erika utcai kétszáz négyzetméteres műhelyhez az érdi tűzoltókat és a megyei katasztrófavédelem műveleti szolgálatát riasztották . A tűzoltók egy oxigén- és egy disszugálpalackot hoztak ki az ingatlanból , az utóbbit folyamatosan hűtötték , így végül nem volt szükség a kilövésére . A műhelyben keletkezett tűz veszélyeztette a szomszédos száz négyzetméteres irodaépületet is , de három vízsugárral sikerült a lángokat megfékezni . Személyi sérülés nem történt .

---

**Eredeti lead:**

Lángra kapott egy érdi autószerelő-műhelyben álló kisteherautó , a tűz átterjedt a műhely berendezési tárgyaira és tetőszerkezetére is csütörtök hajnalban – közölte a Pest megyei katasztrófavédelem .

---

**GPT-2-SZUM által generált összefoglaló:**

Kétezer liter mérgező gázt próbáltak ki Budapesten az érdi tűzoltók .

Kigyulladt egy százezer forintos ruhaterelő cég telephelyének melléképülete péntek délután a Pest megyei Visegrádon .

Meghalt egy ercsi családi ház tetőszerkezetében két munkás , amikor kigyulladt egy 1000 négyzetméteres műhely .

---

**PreSumm által generált összefoglaló:**

Kigyulladt egy autóalkatrész-műhely Érden , a tűzben senki sem sérült meg - közölte a Pest Megyei Katasztrófavédelmi Igazgatóság szóvivője . A tűzben senki nem sérült meg , de a tüzet sikerült megfékezni . A tüzet sikerült eloltani . Az épületből a szomszédos irodaépületet is ki kellett löni , de az ott keletkezett tüzet sikerült megfékeznie a tűzoltóknak – közölték a katasztrófavédelemtől . Az épületet kiürítették , a tüzet eloltották , senki nem sebesült meg – közölték szerdán . A tüzet a tűzoltók fékezték meg , a tűz a szomszédos irodaépületre is átterjedt .

---

5. táblázat. Példa absztraktív összefoglalás generálásra.

A GPT módszerek legnagyobb erőssége a szöveggenerálás. Az eredeti GPT-2 publikációkban és demókban is elsősorban ezen a területen mutatnak látványos példákat. Ezért kutatásomban ennek próbáltam utánajárni. A 6. és a 7. táblázatban láthatóak a szöveggenerálási példák. A SUM modell esetében a szöveggenerálás hosszúságát 256-ra állítottam. A modell három különböző szöveget generált. Kezdesnek megadtam néhány szót, majd arra generált a modell folytatást. Egyik érdekes dolog, hogy nem írtam vesszőt a kezdőszöveg végére, azt már a gép generálta, illetve a 3. példában tovább toldalékolta az általam megadott szöveget. Manuálisan megvizsgáltam a folytatásokat, ezek a szövegek ilyen formában nem szerepelnek a korpuszban, vagyis a folytatás szövegét „kreatív” módon állítja össze a modell. Továbbá érdekes még, hogy kellő hosszúságú szöveg generálása esetén az összefoglaló TL;DR: részt is legenerálja, ez azonban nem váratlan, hiszen ilyen szövegen tanult.

---

**Kezdő szöveg:**

Szeptember végén zárul a balatoni szezon

---

**1. generált szöveg:**

Szeptember végén zárul a balatoni szezon, a Magyar Fürdőszövetség pedig ősszel még két nagy meghívásos fürdőhotel építését tervezi. A rendezvényszervezők és a Balaton Szépe Bizottság üzleti tervében csaknem 100 millió forintot különítettek el a fürdőfelújításra és korszerűsítésre.

TL;DR:

A balatoni szezonkezdet közeleddével a kormányzati holdingok telítődnek.

---

**2. generált szöveg:**

Szeptember végén zárul a balatoni szezon, a rendezvény honlapja szerint a Balaton északi partján két kemping, a Balatonfüred-Balatonkenese és a balatonfüredi Tóni kemping működik, a kisebbeknek is három kempingje van. A szállást-köröket 650-550-en választják a Balaton partján, nyáron 500-1000 látogató is lehet.

A júniusban átadandó kenesei Siotour 2012 a Balaton partján a négy magyar nemzet üdülőhelyeinek, egy-egy kempingnek ad otthont.

TL;DR:

Balatonakarattya egykori üdülőfalujának termálvizes strandját bővítette a Magyar Turisztikai Ügynökség.

---

**3. generált szöveg:**

Szeptember végén zárul a balatoni szezonja. Csonthegynek ki is gyulladt a Velencei-tó csabai pihenőhelye, ami miatt a helyiek nem tudnak felkészülni a csütörtök délutáni kéthetes kempingezésre.

---

6. táblázat. Hírgenerálás példa a SUM modellel.

A HÍR modell esetében 512 hosszúságúra állítottam a generált szöveg hosszát és megállási feltételként a „\n” tokent adtam meg, hiszen a modell ezen token után új témában kezd el hírt generálni. A 7. táblázat példájában az látható, hogy az első szöveg rövidebb, míg a második és a harmadik szövegek kicsit hosszabbak. A bekezdéseket elválasztó tokent (<n>) is a gép magától generálta. A harmadik példa témájában már nagyon eltér az előző kettő és az eredeti mondattól. Sőt megfogalmazásában rendkívül szokatlan mondatokat generált a modell.

A generált szövegek általánosságban nézve nyelvtanilag helyesek és könnyen érthetőek. A mondatok viszonylag logikusan vannak felépítve (idő hiányában manuálisan történt a vizsgálat, kettő ember által). Azonban szemantikai szempontból szokatlan szövegeket kapunk, kevésbé összetartozó kifejezéseket láthatunk egy mondaton belül.

Végezetül, a 8. táblázatban látható a versgenerálás eredménye. A példában megadtam egy kezdő sort írásjelekkel, majd a modell erre generált folytatást. Ebben az esetben szintén 256-ra állítottam a kimenet méretét, azonban néhány sor után ismételni kezdte azokat, ezért levágtam belőle, az első 10 sor látható a példában. Ellenőriztem, ezek a sorok nem szerepelnek az eredeti művekben, a modell maga rakta össze ilyen formában őket. Külön érdekesség, hogy a sortöréseket is a modell generálta.

A modellek kipróbálhatóak a demó oldalunkon, ahol további példák is találhatóak: <https://juniper.nytud.hu/demo>.

---

**Kezdő szöveg:**

Szeptember végén zárul a balatoni szezon

---

**1. generált szöveg:**

Szeptember végén zárul a balatoni szezon. Csütörtökön a hagyományos kempingstrandot, a Festival Balett Singers Travel rendezvényét és a Mátyás-templom barokk orgonaeépítő üzemet alakítanak ki. A lehetőségekre a [www.volstadt.hu](http://www.volstadt.hu) honlapon várják az érdeklődőket.

---

**2. generált szöveg:**

Szeptember végén zárul a balatoni szezon. <n>

Az Egy Balaton Nap célja, hogy a fesztiválozók az idén is felcsendülhessenek kedvenc dalaikból. <n>

Az önkormányzat is részt vesz a fesztivál "témacsúcsán", és a látogatókat szeretettel várják a belvárosi strandokra. <n>

A település lakói idén júniusban kilenc helyszínen koccinhattak a

Balaton Soundon, idén a négynapos fesztiválon összesen több mint 250 ezer ember volt kíváncsi a legnépszerűbb műsorszámokra. <n>

A Strand Fesztivál szakmai partnerei a Balaton Soundon fellépett előadók, akik a Magyarország legnagyobb újdonságai - sportfesztiválok, ismert előadók és zenekarok - mellett rendhagyó, különleges kísérőprogramok, színházi események is várták a résztvevőket. <n>

Mint elmondta, a főváros strandjai ezen a nyáron is bővelkednek, hiszen augusztus 20-tól kezdve október 31-ig többször, több helyen lesz lehetőségük élő bejelentkezéseket tenni a fesztiválra. <n>

"A három nap minden más sportághoz hasonlóan a hazai vízilabda-válogatotté, az összeállításunk egyértelműen a magyar csapaté, ráadásul - akár négyéves lemaradással is - alapvetően van jelen a magyar vízilabda-válogatott: a tavaszi hónapokban eddig még csak egyszer szerepelt magyar csapat, amely Európa-bajnokságon szerepelt és azóta mindössze egyszer marad a döntőben.

A hazai úszósportban azonban valószínűleg nem lesz olyan, mint egy átlagos nyári napon, hiszen a hazai válogatott kiesése esetén sokáig nem áll szándékában a felnőtt nemzeti együttessel is kikerülni" - mondta a polgármester.

---

**3. generált szöveg:**

Szeptember végén zárul a balatoni szezon, az első fészek hamar ki is nyílik majd. <n>

A gyermek egyéves koráig anyatejet tartalmazó tápszert ad a borjúnak. A vádlott két éven át az erszényt, egy gyermekbántalmazási programot is megnyert, amelyből táplálkozni is tudott, de nem tudott uralkodni magzatokon. <n>

Másnap a másodrendű vádlott fiatal, büntetett előéletű testvérel és a nő egyik barátjával vizlatta a reggeli szekrényt, és a teljes adagot elfogyasztotta. A gyerekeket éjszaka, munkaidőben végzett tevékenységgel foglalkoztatta, éjszaka a mosott borral többször megivott, majd megvendégtelt és a magára hagyott pénzt kivette a biztonsági őrök kezéből. <n>

A gyerekeket szórakozás céljából kivette a tejesebdozból, hogy elrejtse az ablaküvegre szerelt, hideg falú ajtó rácsait, majd távoztak. Az ablakon keresztül odalett az a hét alsó tagozatos diák, akik éjjel - a szülők halálával egy időben - megpróbálták bezárni az ajtót. <n>

A gyerekek ki tudtak szabadulni az ajtókról, illetve elhajtottak a garázsban, a házba bejutni azonban már nem tudtak. A kislány a rendőrségen végzett beismerő vallomásaiban tagadta, hogy ő robbantotta volna fel a falat, és a körözés alatt is tagadta a bűncselekményt. Az ügyészség a vádiratban valamennyi gyilkosságot felfüggesztett börtönbüntetéssel, míg társait életfogytig tartó szabadságvesztéssel, illetve közügyektől eltiltással sújtotta. <n>

Az elsőrendű vádlott mindvégig tagadta bűnösségét, vallomását részben valótlanak vallotta.

---

7. táblázat. Hírgenerálás példa a HÍR modellel.

---

**Kezdő szöveg:**

Szegeden, január végén,

---

Szegeden, január végén, őszapófa.

Hogy mulatságos ünnepe vagy multság, az már rosz kéz,

Menyasszonynak végletes reménye?

Mégis el vannak választva.

Elbeszélve... szégyenszem.

Ne hagyj szemünknek lobogója,

S ím a szivárvány,

S, mi van belőlem... harsogó csonka szem.

„És ha e sugarat áldozánk,

Kínálok én is bokrétával!”

---

Szólj, hogy lesz az, hogy itt néhány perc.

8. táblázat. Versgenerálás eredménye.

---

## 4. Összegzés

Kutatásomban egy GPT-2 modellt tanítottam magyar nyelvre, amelyet különböző nyelvtechnológiai feladatokra finomhangoltam. A kísérleteim során a betanított GPT-2 modellt szövegosztályozás, absztraktív összefoglalás és szöveggenerálás feladatain tettem próbára. Az eredmények azt mutatták, hogy a GPT-2 modellem, mint autoregresszív modell, elsősorban szöveggenerálás feladataira teljesít jól, azon belül is a hír- és versgenerálás feladatában produkálnak figyelemre méltó eredményeket. Érdeemes őket kísérleti modelleknek tekinteni, hiszen az eredeti publikációkban a szerzők óriási mennyiségű adaton tanították modelleiket, ezzel ellentétben a modelljeimet csak a magyar Wikipédián és pár százezer hírrel tanítottam be.

## Hivatkozások

- Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Bajgar, O., Kadlec, R., Kleindienst, J.: Embracing data abundance: Booktest dataset for reading comprehension. In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017). Palais des Congrès Neptune, Toulon, France (2017)
- Black, S., Leo, G., Wang, P., Leahy, C., Biderman, S.: GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow (Mar 2021), <https://doi.org/10.5281/zenodo.5297715>, If you use this software, please cite it using these metadata.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A.,



- Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (szerk.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12(null), 2493–2537 (Nov 2011)
- Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (szerk.) *Text, Speech and Dialogue*. pp. 123–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
- Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., Łukasz Kaiser: Universal transformers. In: *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. New Orleans, LA, USA (2019)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Graff, D., Kong, J., Chen, K., Maeda, K.: English gigaword. *Linguistic Data Consortium, Philadelphia* 4(1), 34 (2003)
- Graves, A.: Adaptive computation time for recurrent neural networks (2017)
- Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M.M.A., Yang, Y., Zhou, Y.: Deep learning scaling is predictable, empirically (2017)
- Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling (2016)
- Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks. In: *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*. Caribe Hilton, San Juan, Puerto Rico (2016)
- Liu, P.J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N.: Generating wikipedia by summarizing long sequences. In: *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. Vancouver Convention Center, Vancouver, BC, Canada (2018)
- Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models (2017)
- Nemeskey, D.M.: *Natural Language Processing Methods for Language Modeling*. Ph.D.-értekezés, Eötvös Loránd University (2020)
- Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified

- text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020), <http://jmlr.org/papers/v21/20-074.html>
- Rei, M.: Semi-supervised multitask learning for sequence labeling. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 2121–2130. Association for Computational Linguistics, Vancouver, Canada (Jul 2017)
- Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B.: *Megatron-lm: Training multi-billion parameter language models using model parallelism* (2020)
- Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) *Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science*, vol. 12848, pp. 222–234. Springer (2021)
- Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. In: Todorovski, L., Lavrač, N., Jantke, K.P. (szerk.) *Discovery Science*. pp. 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
- Yang, Z.G., Agócs, Á., Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (2021)
- Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y.: Defending against neural fake news. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems*. vol. 32. Curran Associates, Inc. (2019)

# Lehet-e automatikus családfaépítő programot készíteni?

Naszódi Mátyás

MorphoLogic KFT., e-mail: naszodim@morphologic.hu

## Kivonat

Ha WEB-es tartalmak alapján akarunk családfát készíteni, számos buktató kerül az utunkba: a források megbízhatatlansága, az OCR bizonytalansága, a nevek változatossága, azonos nevű különböző személyek elkülönítése. Ezeket a gondokat járja körbe a cikk, és keres megoldásokat, továbbá ezeknek a hibáknak csökkentésére tesz kísérletet a családfa építésének során. Módszerében felhasználja a szokásos szövegkorrekciót, a szintakszist, adatbázisok elemeinek összevetését. A konkrét példák specifikusak, de metodika általános és hatékony a névelém témakörében. Genealógiai adatbázis segít a források korrekciójában, és megfordítva, a források javítása segíti a családkutatást.

**Kulcsszavak:** karakterfelismerés, szövegkorrekció, nyelvtan, névelém-egyértelműsítés, helyesírás-ellenőrző, genealógia

## 1. Családfakutatás

### 1.1. Névelemek és egyedek

Személyeket a tulajdonnevük szerint különböztetjük meg, de a név nem azonosítja a személyt (Gulás és mtsai, 2021). Fontosabb a családi kapcsolatok hálózata, vagyis, hogy el tudjam helyezni a származási hálóban. Nem nevezem családfának, mert a kapcsolatrendszer gráfelméleti szempontból nem fastruktúrájú.<sup>1</sup>

### 1.2. Személyek unifikációja

A különböző adatbázisokban azonos személyek szerepelnek. Átlalában ezek egyértelműsítésére törekednek, de jelen esetben ennél többre van szükség, azonosításra, vagyis egy adatbázis elemére való egyértelmű hivatkozásra. (Nguyen és mtsai, 2016) Az unifikáció jelen esetben azt jelenti, hogy egy egyedként regisztrálhatjuk a genealógiai hálózatban. Ez a háló úgy azonosít, mint a szavak elhelyezése egy szemantikai hálóban. Az egyértelműsítést a néven kívül egyéb adatok segítik: szülei neve, gyermekei, születési, halálozási adatai.

<sup>1</sup> Azért nem fa, mert létezik rokonházasság. Emiatt két ember közt több rokoni szál is lehetséges. A genealógiai struktúra egy ciklusmentes irányított páros gráf, ahol az egyik csoport a családokat, a másik az egyéneket tartalmazza. Az egyéntől akkor irányul el a családhoz, ha szülőről, míg családtól az egyénhez akkor mutat el, ha gyermekről van szó. Csak akkor lehetne ebben ciklus, ha valaki saját maga elődje: *Oidipusz esete*.

### 1.3. Genealógiai adatbázisok, standardizálásuk

A genealógiai adatbázisoknak kialakult egy formája. A struktúrában kétfajta objektum van: *személyek* és *családok*. A személyeknek tulajdonságai lehetnek: *nem, családi név, keresztnév, becenév, születési hely, születés ideje...* Ezek elég rugalmas mezők a leírásban, de vannak olyan korlátok, melyek ma már elavulhattak, de az általam vizsgált korban és területen nem okoznak gondot. Ilyen a többnejűség mohamedánoknál, az azonos nemű pár, a nemváltás... Az adatbázisok leírására kialakult egy (nem jól szabványosított) nyelv (LDS, 1984), emiatt aki kommunikálni akar ebben a témában, ehhez alkalmazkodik.

### 1.4. A források sokfélesége

A weben hozzáférhető forrásanyagok között azok jönnek számításba, melyek nagy mennyiségű genealógiai adatot tartalmaznak. A következők hasznosak:

- **Családregények, memoárok:** Automatikus feldolgozásuk nehézkes, mivel a lényegi információ egy természetes nyelvű szövegben van elszórva.
- **Családfák:** A konkrét családra vonatkozóan általában pontos adatokat tartalmaznak. Automatikus feldolgozásuk könnyű, de minden családfának egyedi a formája.
- **Lexikon jellegű források:** Neves embereknel hatékony. A megbízhatósága lexikonfüggő. Automatikus feldolgozásuk könnyű, a lényeges genealógiai adatok jól szeparálhatók.
- **Szabad szövegű adathalmazok:** A legjellemzőbb erre a kategóriára a gyászjelentések, nekrológok. Ezek könnyen találhatóak és szűrhetőek folyóiratokból. Nagy mennyiségben találhatóak az OSzK gyűjteményében. (OSzK, 2015) A szövegek aránylag könnyen kezelhetőek lennének, ha az OCR minősége megfelelné.
- **Kötött formátumú adathalmazok:** Ilyenek a digitalizált telefonkönyvek, lakcím-nyilvántartások. A kötött formájú adatok könnyen feldolgozhatóak.
- **Mezőkbe rendezett adatbázisok:** Ezek feldolgozása lenne a legkönnyebb. Legnagyobb a Salt Lake City-ben (LDS, 2000) összegyűjtött adathalmaz (anyakönyvi kivonatok). Az adott nyelvet nem ismerők kódolták, ezért tele van elírással. A hibák öröklődnek, mert erre hivatkozik számos genealógiai szolgáltatás: a GENI (GEN, 2006), a MyHeritage (MyH, 2003), a JewishGen (Jew, 1987) ...

### 1.5. Adatok egysége formája

A feldolgozás érdekében minden forrásanyagot GEDCOM formátumúvá alakítom:

**özv. Wenzel Mihályné szül. Ulber Rozina asszony**

magánzónő,

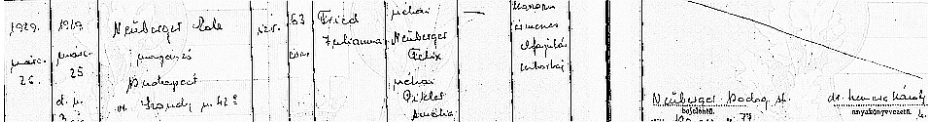
f. évi szeptember hó 15-én reggel 1;7 órakor rövid szenvedés után életének 86. évében történt gyászos elhunytát.

A boldogult földi maradványai t. hó 17-én délután 3 órakor fog-  
nak az oroszvári evang. temetőben örök nyugalomra helyeztetni.

Oroszvár, 1913. évi szeptember hó 15-én.

```

0 @I1@ INDI
1 NAME Mihály /Venzel/
1 SEX M
1 FAMS @F1@
1 DEAT
2 DATE BEF 1913
0 @I2@ INDI
1 NAME Rozina /Ulber/
1 SEX F
1 BIRT
2 DATE ABT 1827
1 DEAT
2 DATE 15 Sep 1913
2 PLAC Oroszvár
0 @F1@ FAM
1 HUSB @I1@
1 WIFE @I2@
    
```



```

0 @I3@ INDI
1 NAME Ede /Neüberger/
1 SEX M
1 FAMS @F2@
1 FANC @F1@
1 BIRT
2 DATE ABT 1866
1 DEAT
2 DATE 25 MAR 1929
2 CAUS koszoru ér meszes elfajulás cukor baj
0 @I4@ INDI
1 NAME Julianna /Fried/
1 SEX F
1 FAMS @F2@
1 DEAT
2 DATE AFT 1929
0 @I2@ INDI
1 NAME Amália /Pickler/
1 SEX F
1 FAMS @F1@
1 DEAT
2 DATE BEF 1929
0 @I1@ INDI
1 NAME Felix /Neuberger/
1 SEX M
1 FAMS @F1@
1 DEAT
2 DATE BEF 1929
0 @F1@ FAM
1 HUSB @I1@
1 WIFE @I2@
1 CHIL @I3@
0 @F2@ FAM
1 HUSB @I3@
1 WIFE @I4@
    
```

## 1.6. Ideális eset

Tételezzük fel, hogy a forrásainkban az adatok pontosak, egyértelműek.

### 1.6.1. Adat (rekord) információtartalma

Shannon értelmezésében egy adathalmaz információtartalma nem más, mint az adategyüttes valószínűségének reciproka (illetve ennek logaritmus). Ha a részadatok függetlenek, akkor a teljes információtartalom a részadatok információtartalmának összege (szorzata vagy összege, aszerint, hogy milyen skálázást alkalmazunk). A személyi adatok nem függetlenek, a valószínűségek nem ismertek, de relatív gyakorisággal jól közelíthetők. Egy név önmagában csak akkor döntő, ha nagyon ritka. Egy születési évnek 365-öde az információtartalma, mint egy napra pontos dátumnak. A független részek információtartalma összeszorozódik.<sup>2</sup>

### 1.6.2. Forrásanyag információtartalma

Egy adat információtartalma becsülhető az elemei relatív gyakoriságának reciprokával. Hiányzó mező esetén 1 a relatív gyakoriság, mivel a hiány bármit jelenthet.

$$Inf(Record) = \prod_{Field \in Record} 1/Q_{Field} \quad (1)$$

<sup>2</sup> Ha Shannon-ban számolnánk, akkor ennek logaritmus lenne, de akkor nem szorzódnak, hanem összeadódnak az információ-részmennyiségek.

*Név (családi, kereszt-, becenév. . .)* esetén a név relatív gyakoriságának reciproka az adott korban. *Születési dátum* esetén az adott időszakban született emberek számának reciproka. Ha annyit tudunk, hogy *Nagy Pista 1837-ben született Nagyatádon*, akkor az összes *1837-ben Nagyatádon születettek* számát kell elosztani az *1837-ben Nagyatádon született Nagy Pisták* számával.

### 1.6.3. Új adat bizonyossága

Azt vetjük össze, hogy az új adatok mennyire illeszkednek a már meglévőkhöz.<sup>3</sup> A bizonyosság mértéke számítható az összevethető adatelemek információtartama alapján.

$$\text{Cert}(\text{Unit}) = \prod_{\text{Field}} \text{Inf}(\text{Field}) \quad (2)$$

### 1.6.4. Új adat felvétele, ütközések

Ha egy forrásadat bizonyossága elér egy mértéket, felvehetjük a hálózatunkba. Ha bármi ütközik a már felvett adatokkal, meg kell vizsgálni, hogy a regisztrált adatnak vagy a felveendőnek nagyobb a bizonyossága, esetleg adatok alternatívájáról van szó. Ha pontosak az adatok, az ütközések kizárják a beépítést.

### 1.6.5. Munkamenet

Mindezek alapján a családfaépítés menete hasonló egy puzzle játékhoz:

1. **Kiindulási családfa felvétele:** Ez lehet akár egy személy, de lehet egy ismert család, netán egy már ismert család adathalmaza.
2. **Jelöltkeresés:** Keresünk a forrásanyagban egy elemet. Mivel a forrásanyag kimeríthetetlen, érdemes szűrőket alkalmazni, például névre rákeresni.
3. **A forrásanyag formalizálása:** A rekord átalakítása GEDCOM adattá.<sup>4</sup>
4. **Bizonyosságszámítás:** Megnézzük, az adatok alapján hogyan lehet a legnagyobb bizonyossággal beépíteni az új elem(ek)et, illetve több jelölt esetén melyiknek nagyobb a bizonyossága.
5. **Ütközésfeloldás, beépítés:** Ha adatütközés van, fel kell oldani azt: vagy a régi adat módosításával, egy korábbi elem törlésével, vagy az új elvetésével.
6. **Iteráció:** Folytathatjuk a 2. lépéstől.

Az algoritmusban lényeges, hogy a forrás általában nem zárt halmaz abban az értelemben, hogy nem ismerjük teljes terjedelmében vagy azért, mert túl nagy, vagy azért, mert a hozzáférés csak lekérdezéseken keresztül lehetséges.

<sup>3</sup> Ha két személy apját, anyját ugyanúgy hívják, akkor nem gyakori nevek esetén feltételezhetjük, hogy testvérek.

<sup>4</sup> A formalizálásra talán a legjobb eszköz a MorphoLogic Markup Wizard-ja lenne alkalmas, de ez a program hibás adatok kezelésére nem megfelelő.

## 1.7. Nem ideális eset

Eddig feltételeztem, hogy a források megbízhatóak. A gyakorlatban azonos személy azonos jellegű adatai forrásonként eltérhetnek. Ennek több oka van.

### 1.7.1. Forrás adata nem egyértelmű

Még anyakönyvi kivonatokban, sírfeliratokon is találhatók eltérések. Erre példa lehet az utólagos anyakönyvezés. Elírások, változatok gyászjelentésekben is megjelennek, de a sírfeliraton is bizonytalanság, ha a születési évet becsüljük: *élt x évet* vagy *életének x-edik évében*. Ha a halál ideje napra pontos, ez egy év intervallumnak számít, de ha csak az év volt feltüntetve, ez kétesztendő intervallumot jelent.

Nem csak személynév, hanem helységnév is lehet többértelmű.<sup>5</sup>

Egy ember neve változhat élete során, másrészt a neveknek változatai léteznek. A névváltozatok kezelésének egy részére a GEDCOM lehetőséget ad. Jelölhető, hogy a *Bódog*, a *Felix*, a *Felice* ugyanannak a magyar, német, illetve olasz változata. A keresztnemek halmaza zárt, emiatt bokrokba rendezhetem őket, mint a szinonimaszótárakban. Praktikus, ha a nevek között egy távolságot számítunk.<sup>6</sup> A családnevek halmaza nem feltérképezhető, emiatt célravezető a helyesírás-ellenőrzőknél használt, de a célra hangolt távolságszámítás.

A mezők távolságával módosítható a bizonyosság számítása:

$$\text{Cert}(\text{Unit}) = \prod_{\text{Field}} \frac{\text{Inf}(\text{Field}_{\text{source}})}{|\text{Field}_{\text{data}} \text{Field}_{\text{source}}|} \quad (3)$$

### 1.7.2. A forrás megbízhatósága

A forrás minden átírási lépésnél torzulhat. Ezt a **2.** fejezetben taglalom. A forrásállomány megbízhatóságával súlyozhatom az adat beépíthetőségét. (A mezők távolsága a változás valószínűségének reciproka.)

$$\text{Cert}(\text{Unit}) = \prod_{\text{Field}} \frac{\text{Cert}(\text{Field}_{\text{source}}) * \text{Inf}(\text{Field})}{|\text{Field}_{\text{data}} \text{Field}_{\text{source}}|} \quad (4)$$

### 1.7.3. A forrásállomány javítása

A hibák pár százalékos csökkentése a családfaépítést nagyságrenddel javíthatja, mivel a rokoni kapcsolatok távolsága 10-nél is nagyobb is lehet.

A Hadifoglyok állományban a családi nevek nagy százaléka sérült (Sass és mtsai, 2021). Ha az egyéb adatok megegyeznek a rekordban, akkor korrigálhatjuk a forrásanyagot.

<sup>5</sup> Nagyapámról tudtam, hogy *Telcsen* született, de csak további kutatásból derült ki, hogy nem az ismertebb cseh *Telč*-en, hanem a román területen levő *Telciu*-n.

<sup>6</sup> A *Katalin*, *Kata*, *Katica*, *Katerina*... közel vannak. Hasonlóan a *Telcs*, *Tetsch*, *Telch* családi nevek is.

Лагерь № \_\_\_\_\_ спецгоспиталь № \_\_\_\_\_ ОРБ № \_\_\_\_\_

Национальность Венгерец В какой армии противника состоял Венгерской

1. Фамилия Др. Фени

2. Имя Тибор 3. Отчество Золтан

4. Год и место рождения 1912 г. Венгрия г. Будапешт  
ул. Герескерей № 5

5. Адрес до призыва тот же

6. Подданство или гражданство Венгерское

7. Партийность н/п 8. Вероисповедание католический

9. Образование:

а) общее Чл. народ школы

б) специальное Всп. гимназия

в) военное Чл. университета

10. Профессия Доктор - юрист

Учетное дело № 18637

Арх. № 01167498

Тип. Мингаздушья, в. 3277. т. 600000

	Orosz szöveg:	Az adatbázisban:	A valóságban:
Vezetéknév	Др. Феник	Fenyik	Dr. Fenyő
Utónév	Тибор	Tibor	Tibor
Apai utónév	Золтан	Zoltán	Zoltán
Állampolgárság	Венгерское	magyar	magyar
Születési hely	Будапешт	Budapest	Budapest
Születési év	1912	1912	1912

## 2. Forráskorrekció

A másodlagos (digitálisan kereshető) források esetén az adatok 10-30 százalékban hibásak, mely erősen csökkenti a megbízhatóságot.

Vannak olyan eszközök, melyekkel jobb eredmény érhető el, mint általános célú nyelvi korrekciókkal vagy akár neuronhálós módszerekkel (Gulás és mtsai, 2021). Három korpuszt vizsgálok részletesebben:

	Hadifoglyok <sup>7</sup>	Öröklét <sup>8</sup>	Gyászjelentések <sup>9</sup>
szerkezet	record/mező	record/mező	folyó szöveg
méret	~700 000 rekord	~260 000 rekord	~600 000 rekord
kép	elérhető	nem elérhető	elérhető

<sup>7</sup> Az orosz levéltárban őrzött II. világháborús hadifogoly, illetve "málenkij robot" nyilvántartása. Kézzel, cirill betűkkel írt rekordok. (HAD, 2021)

<sup>8</sup> Boross Lajos projektje, mellyel igyekezett kereshetővé tenni magyarországi zsidó sírokat. (Boross, 2010)

<sup>9</sup> Eredetileg a mormonok a II. Világháborút követő projektjének magyarországi lecsapódása az OSzK gondozásában. (OSzK, 2015)



## 2.1. Az előfeldolgozás

A korpuszokat vagy egy humán lejegyzéssel, vagy OCR-rel digitalizálták. Ezt követően számos előfeldolgozásra lehet szükség: kódkonverzió, transzkripció, helyesírás-ellenőrzés, humán vagy gépi lektorálás, formai korrekció. . . , melyek egy részét már elvégezték, mire szabadon hozzáférhető anyaggá vált.

- A **Hadifoglyok** korpusza annyiban inhomogén, hogy különböző képességű emberek vették fel az adatokat, és különböző minőségben kódolták az egyes fázisokban. A magyar feldolgozás a digitalizált cirill betűs szövegeknél kezdődött, melynek kódolása egyértelmű.
- A **Gyászjelentések** nyomtatott betűkkel írt korpuszát OCR programmal digitalizálták. Mind nyelve (magyar, német, szlovák. . . ), mind betűtípusa változó.
- Ezekből a lényegesen különbözik az **Öröklét** honlap adatbázisa, ahol a származás kézzel írt nyilvántartás, de van sírfelirat-feldolgozás is. Nyelve többnyire magyar, de sok német, esetenként horvát, spanyol. . . Egy részét OCR programmal digitalizálták, más részét humán erőforrással vitték gépre. Nem láthatjuk az eredeti képet sem.

Más szabadon elérhető forrásokat is vizsgáltam: lexikonokat, régi telefonkönyveket. . . , de a tanulság szempontjából a példaim elegendőek.

## 2.2. A javítás elve

Adott egy erősen rongált  $d$  kódsorozat, melyet a lehetséges de nem hozzáférhető helyes  $S$  kódsorozatokkal kell összevetni, és amelyikhez legközelebb áll, azt kell elfogadni eredménynek.  $s : s \in S | \min(|s - d|)$

Ha figyelembe vesszük a helyes szövegek valószínűségét:

$$s : s \in S | \min(p(s) * |s - d|)$$

A legegyszerűbb esetben Hamming távolságot vesznek. Ez nem alkalmazható, ha a transzformációk között kódkonverzió is volt. A távolság számítása pontosabb, ha annak a valószínűségét nézzük, milyen karakterekből milyen valószínűséggel milyen karakter keletkezhet a transzformációk során. Így egyrészt nem kell a két karaktorsorozatnak azonos kódban íródnia, másrészt könnyebben lehet alternatívákat elfogadni:

$$s : s \in S | (p(s) * \prod_i |s_i - d_i|) < K \quad (5)$$

A helyesírás-ellenőrzők a legközelebbi találatokat választják ki. A végső döntés a felhasználóé. Az optikai és a beszédfelismerők mindig elfogadják a vélt megoldást. Most ez nem járható, mert nagy a hibaszázalék. Az (5) képlet alapján dolgoznak a Levenstein algoritmus javított változatai, pl. Viterbi algoritmus, amely a forrással lineáris sebességű algoritmus. A karaktértávolság alkalmazásonként eltér, mert figyelembe kell venni, mi mivé torzulhat a processzálas alatt. Az is gond, hogy az  $S$  forrástér meghatározása nehézségbe ütközik.

## 2.3. Karakterszintű távolság

Az említett források alkalmatlanok arra, hogy statisztikai alapon (öntanulással) határozzuk meg a karakterek távolságát, mert kicsi a korpusz ahhoz, hogy be-tanítsuk. Emiatt érdemes elemezni, mi mivé torzulhatott a forrásanyag kiala-kulása közben. Figyelembe kell venni a forrás kialakulásának lépéseit, és ehhez alkalmazkodó javítómintákat kell építeni. Erre kiváló példa a Hadifoglyok pro-jekt és az Öröklét adatbázis.

### 2.3.1. Karakterszintű távolság idegen karakterek esetén

Ha a forrás keletkezésénél transzkripció is szerepel, akkor a transzkripció előtti és utáni karakterek viszonyát kell leírni. Ez történt a Hadifoglyok projektben is. A latin-cirill átírás inverze adja meg az elsődleges közelséget. Maga az átkódolás nem egyértelmű. Ezt részletesen tárgyalja az erről szóló cikk. (Sass és mtsai, 2021)

### 2.3.2. Kódnormalizálás

Az Öröklét adatbázisban vegyesen szerepel különböző kódkészlet. A kilógó (nem helyes kódban használt) karakterek többségét automatikusan helyre lehet állítani, esetleges ligatúrákat kifejteni, de ahol ez nem egyértelmű, ott az alternatívák jelentik a közeli karaktereket.<sup>10</sup> Bár a feloldás semmiben sem különbözik a helyesírás-ellenőrzőknél használtaktól, érdemes ezeket először korrigálni. Egyszerű karakterstatisztikával ki lehetett szűrni az idegen elemeket. A korpusz-ban a következő kérdéses kódok szerepeltek a hozzájuk tartozó előfordulással:

char	freq	mi helyett
_	10008	Nem felismert betű (ált. ü vagy más ékezetes magánhangzó)
µ	2678	a legtöbb esetben á (852-es kód)
£	509	a legtöbb esetben ú (852-es kód)
˘	330	a legtöbb esetben í (852-es kód)
™	59	valószínűleg egy EÜ ligatúra (nincs UNICODE-ja)
;	16	é (a billentyűt elfelejtették angolról magyarra állítani)
=	15	ált. ó (a billentyűt elfelejtették magyarra állítani)
à	9	a legtöbb esetben Ő (852-es kód)
Ű	8	többnyire Ő
š	3	a legtöbb esetben ű vagy ü (852-es kód)
ä	3	ö (tipikus OCR hiba, ha magyarra van állítva)
*	3	-
%,	2	ä
[	2	ö (a billentyűt elfelejtették magyarra állítani)
Š	1	Ű (852-es kód)
}	1	Ü (a billentyűt elfelejtették magyarra állítani)
#	1	p

<sup>10</sup> Az Öröklét korpusz általában 1250-es kódkészlettel íródott, de volt ettől eltérő kódolás is. Ezeket fel kellett ismerni, átkonvertálni. Ennél cifrább esetekkel is találkoztam. PDF anyagokban egyes ékezetes karaktereket lokálisan olyan kódra konvertálnak, melyek előre nem becsülhetők. Más esetekben lebegő (prefix) ékezeteket használnak, és gyakoriak a ligatúrák (karakterösszeolvadások). Néhány OCR program a bizonytalan felismeréseket speciálisan kódolja.



#### **2.4.1. Ha a forrás mezőkre osztott rekordokból áll**

A temetői adatbázis és a Hadifoglyok adatbázis ilyen. Az egyszerűbb eset a Hadifoglyok adathalmaza: vezetéknev, családnév, foglalkozás... mind olyan kategória, melyek lehetséges elemei felsorolható halmazt jelentenek. A laccím ennél kicsit összetettebb, de ez sem jelent lényeges gondot. Ami viszont igazi probléma, a családi nevek halmaza.

Az Öröklét adatbázisa egy fokkal nehezebb. A személynevek mezőben nincs elválasztva a családi és keresztnév, sőt, a nevek írásmódja változatos: *Tarr Gézáné, Márkus Ildikó – Tarrné Márkus Ildikó*... Ennek ellenére a mezőkben szerepelő lehetséges szavak jellegzetesek és elvileg összegyűjthetőek.

Ha mezőnként egy-egy külön szókészletből álló halmazt készítünk, mezőnként a forrás lehetséges terét lényegesen lecsökkentjük ahhoz, hogy a szavak távolsága nagyobb legyen, a javítási algoritmus hatékonyra váljék.

#### **2.4.2. Ahol némi szintakszisra is szükség lehet**

Ha összetettebb egy-egy mező, akkor egyszerű reguláris kifejezéssel jól le lehet fedni a szintakszist. Ilyen a laccím. Lefedésen azt értem, hogy funkcionálisan megtalálják a részeket, de ezeknek nem kell karakterpontosan helyesnek lenniük. Ide tartozik az is, amikor mezőket tévesztenek a bejegyzéseknél: A Hadifoglyok adatbázis esetén a felekezeti hovatartozás és a nemzetiség kitöltése időnként elcsúszik. Az Öröklét esetén adatok át-átcsúsznak a következő mezőbe.

Ha a mezők a helyükön vannak, akkor a szavankénti ellenőrzés az előzőek alapján már lehetséges.

#### **2.4.3. Ahol komolyabb szintakszisra van szükség**

A Gyászejelentések természetes nyelvű korpusz. A javítás szempontjából itt is fontos szűkíteni, hogy egyes pozíciókban milyen szó jöhet számításba. A szintakszis a nevek helyének (és minősítésének) felismerését határozza meg, de a névelemek felismerésén kívül a mondatok szerkezete támpontot ad a családi kapcsolatokra. Családrégiyek, memoárok kiértékeléséhez általános célú elemzőre lenne szükség, de ezzel nem foglalkozom, ez talán egy másik projekt lesz.

### **2.5. Mondatszint a Gyászejelentések feldolgozásában**

Ha eltekintünk a szótári bejegyzésektől, akkor pár száz szabállyal olyan nyelvtant kreáltam, mellyel a korpusz mondatai elemezhetőek, és a nyelvtan alapján generált mondatok nagyobb hányada helyes.<sup>11</sup>

A szintakszisban részletesebb szótani kategóriákat használok a szokásosnál, és a különböző helyeken csak az alkategóriának megfelelő szót engedem meg. Helységnevek és személynevek klasszikusan a tulajdonnevek kategóriájába tartoznak, de a szintaktikai szerepük eltér. Ebből a szempontból lehet külön szófaj az orvosi

<sup>11</sup> A mondatelemzők túlgenerálnak, mert elemzésre készülnek. Még a helyesírás-ellenőrző mondatszintű megvalósítása is túlgenerál abban az értelemben, hogy sok hibát nem vesz észre, illetve szemantikailag lehetetlen mondatokra ritkán riaszt.

szövegekben a *testrészt*, *orvosság*. A mi esetünkben a *keresztnév*, a *vezetéknev*, az *asszonynev* mind más, más szófaj.

### 2.5.1. A szintakszis

A gyászjelentésekben a mondatok korlátos sémáknak felelnek meg. A mondatokat funkciójuk szerint külön kategóriákba sorolom. A gyászjelentés fő mondatának, az **enounce**-nak a szabálya AGFL-ben (Koster, 1991):

```
RULE enounce(LANG): [SUBJ(LANG)], [INTROM(LANG)], # Enouncer(s)
                    [[SORROWLY(LANG)], [BUT(LANG)], [sorrowcontr(LANG)]]],
                    INFORM(LANG), [OPTTHAT(LANG)], # verbal part
                    mainperson(LANG, GENDER, CASE), # Dead
                    circumstances(LANG), # Time, place, ...
                    died(LANG, CASE) . # verbal part
```

A fő mondat kötelező része a gyászjelentésnek, de a bejelentő személye opcionális. A mondat fő igéje az **INFORM** részben van, míg a mellékmondat igéjét a **DIED** egység foglalja magába.

A **circumstances** kötelező:

```
RULE circumstances(LANG): [town(LANG, CASELOC), ["", ""], date(LANG), [{"", ""}, time(LANG)], #of death
                        [age(LANG)], # in the time of dead
                        [fage(LANG)], # years in marriage
                        [CAUSE(LANG)] . # of death
```

de ebben csak a halál időpontja, ami nem hiányozhat a szövegből.

### 2.5.2. ... és mi van a mondaton túl?

A mondat csak szintakszison belüli kategória. A mondatokra való tagolás beleolvad a dokumentum formális megfogalmazásába. A teljes gyászjelentés a következő szabállyal adható meg:

```
RULE abitoary: [cite(LANG)], # some lyric pe. from the Bible
              enounce(LANG), # Enouncement of the person's deth
              [burrial(LANG)], # Burrial
              [misa(LANG)], # Celebration
              DATE(LANG), # date of enouncement
              [FARWELL(LANG)], # Farwell words
              [personlist(LANG)], # list of family members
              [address(LANG)], # adress to reply
              [publish(LANG)], [PRESS(LANG)] # Publisher and print
```

Németre, angolra is megalkottam a szabályokat. Az elemzés szerkezete nem tér el a magyartól. A felhasznált eszköz alkalmas keverni a robusztus elemzést a pontos találatokkal, így lexikális hibák esetén is kiadhatja az elemzést feltüntetve a kérdéses szavakat. Ez nagy segítség a nem ismert tulajdonnevek keresésében és a korrekciók helyének megtalálásánál.

### 2.5.3. A javítás menete

Több ellenőrző eszköz van, amellyel hibák korrigálhatók. Minden szűrés után javul a forrás, módosul a szótár, esetleg a szintaktikus leírás, emiatt a korábbi fázisokat újra lehet futtatni. Az ellenőrzési szintek a következők:

- Idegen karakterek szűrése
- Kódnormalizálás
- Jellegzetes karakterhibák javítása
- Szószintű ellenőrzés
- Szintaktikai ellenőrzés / mezőkre bontás
- Kivonatolás / konvertálás GEDCOM formátumra

### 3. Tapasztalatok

A rendszer nincs kész, de az elért eredmények választ adnak a címben feltett kérdésre. Hiányzik az adatbázisok lekérdezőrendszerének automatizálása. Ezt kézzel pótoltam, alkalmazkodva a forrás elérési lehetőségeihez.

A kidolgozott eszköz jelenleg nem helyettesíti az emberi munkát, de sokban segít. *Telcs Edének* 9 generációs ismert családfájából (690 személy) egy-két generációs szigeteket tártam fel így. Viszont a család által összerakott fában is letem hibára, amit a gépi módszer tisztázott. Más neveknél (*Korányi, Neuberger*) ennél jobb eredményt értem el.

Kísérletképpen kiindultam Kornai János gyászíréből. A standard források biztos adatai alapján 40 családtagot szedtem össze. Enyhítve a megbízhatóságon 100 családtagra derített fényt nagy megbízhatósággal az algoritmus. Kicsit lejjebb véve a megbízhatósági paramétert újabb 70 személyt helyeztem el a hálóban, de az így kapott családfában előfordulhat tévedés is.

A családkutatás a jelenlegi forrásállománynál gyakori nevek esetén eleve kudarcra van ítélve, ha nincs mellettük jobban kereshető rokon. Az eredményesség attól is függ, hogy milyen nagy választékban találunk forrást. A XIX. századot megelőzően többnyire csak nemesekre lehet lelni. A nyomtatás elterjedése és a kötelező regisztrációk miatt a XIX. század közepétől várhatóak jó eredmények. Amennyiben valóban megvalósul Magyar Nemzeti Levéltár 2023-ra tervezett digitalizálása (Biszak és mtsai, 2017), a teljes automatizálásra is sor kerülhet.

A források korrigálása általában 10-20 százalékkal növeli a fellelhető személyek számát. Emiatt valóban érdemes a források javításával foglalkozni.

Pontosabb számszerű kiértékelésre nem ad módot a jelenlegi munka, de egyértelmű a haszna.

### 4. Tervek

Mindenképpen megérné, ha a karakterfelismerő és a nyelvi modul egyszerre venne részt a döntésben. (Naszódi, 2000)

Kísérleteim azt is igazolják, hogy a kép előfeldolgozása (szürkeség-, ferdeségkorrekció, a valós szöveg behatárolása, inicálé felismerése) javít az OCR minőségén. A gyászjelentések hasábeosztása hiányos, ezért a szórend sérül, ami a mondat szintű elemzést rontja el. Az OCR programok szokásos hasábolását, képfelismerését újságokra optimálták, pedig ez is lehetne témakörfüggő.

Ha a forrásban valamilyen rendezésben szerepelnek a rekordok, a rendezési elv is adhat támpontot a szövegek korrigálására. Ha a nevek szerint vannak listába szedve (telefonkönyv, összeírás), akkor a sorrendtől eltérő bejegyzésnél könnyen felismerhető a hiba. Ha anyakönyvi kivonat, akkor az esemény ideje adja meg a sorrendezést.

További lehetőség, ha hasonló, de más időben készült regisztrációkat vetünk össze. Erre tipikus példa a telefonkönyv, melynek egymás utáni kiadványai nem sokban különböznek. Ha az egyikben nem olvasható jól egy mező, de hasonlít a másikban levőhöz, triviális a korrekció, pontosabban több bizonytalan bejegyzés egymást korrigálhatja, kiegészítheti.

## Hivatkozások

- JewishGen: Zsidó történelmi és genealógiai adatok rendszerezett, kereshető tára. WWW (1987), <https://en.wikipedia.org/wiki/JewishGen>
- MyHeritage: Családfaépítő és megosztó online szolgáltatás. WWW (2003), <https://en.wikipedia.org/wiki/MyHeritage>
- GENI: Általános családfaépítő és megosztó szolgáltatás. WWW (2006), <https://en.wikipedia.org/wiki/Geni.com>
- Szovjetunióba elhurcoltak. AOL (2021), <https://adatbazisokonline.hu/gujtemeny/szovjetunioba-elhurcoltak>
- Biszak, S., Lakatos, A., Ádám, V.: Az egyházi anyakönyvek digitalizálásának lehetőségei - Módszertani tanulmány (2017), [https://archivum.asztrik.hu/sites/default/files/letoltesek/akvi\\_tanulmany.pdf](https://archivum.asztrik.hu/sites/default/files/letoltesek/akvi_tanulmany.pdf)
- Boross, L.: Öröklét (2010), <http://www.oroklet.hu/>
- Gulás, M., Yang Zijian, G., Dömötör, A., Laki, L.J.: Automatikus hibajavítás statikus szövegeken. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia előadásai. p. 243-251. Szegedi Tudományegyetem (2021), [http://nlp.g.itk.ppke.hu/sites/default/files/publications/MSZNY2021\\_paper\\_29.pdf](http://nlp.g.itk.ppke.hu/sites/default/files/publications/MSZNY2021_paper_29.pdf)
- Koster, C.H.: Affix Grammars for Natural Languages. In: Melichar, B. (szerk.) Summer School on Attribute Grammars, Applications and Systems (1991), [https://link.springer.com/chapter/10.1007%2F3-540-54572-7\\_19](https://link.springer.com/chapter/10.1007%2F3-540-54572-7_19)
- LDS: GENEALOGICAL DATA COMMUNICATION. WWW (1984), <https://en.wikipedia.org/wiki/GEDCOM>
- LDS: FamilySearch (2000), <https://www.familysearch.org/hu/>
- Naszódi, M.: Nyelvi visszacsatolás karakter-, kézírás- és beszédfelismerő rendszerek számára. Tech. rep. (2000), <http://www.nefmi.gov.hu/ikta/projektek/ikta3/20010228/i063/ossze1.html>
- Naszódi, M.: A magyar helyesírás-ellenőrzők mai állása. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia. p. 347-354. Szegedi Tudományegyetem (2017), <http://www.cs.bme.hu/~naso/langeng/SpellsSate20016.pdf>
- Nguyen, D.B., Theobald, M., Weikum, G.: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. In: Transactions of the Association for Computational Linguistics, Volume 4. p. 215-2229 (2016), <https://aclanthology.org/Q16-1016>
- OSzK: Gyászjelentések. Pannon Digitális Egyesített Archívum (2015), <https://dspace.oszk.hu/handle/20.500.12346/663648>
- Prószéky, G., Naszódi, M., Kis, B.: Recognition Assistance. In: Conference on Computational Linguistics. Taipei, Tajvan. p. 1263-1267 (2002), <http://www.cs.bme.hu/~naso/langeng/C02-2014.pdf>
- Sass, B., Mittelholcz, I., Halász, D., Lipp, V., Kalivoda, A.: Magyar hadifoglyok adatainak orosz-magyar átirása és helyreállítása, és a szabadszöveges adatbázisok tulajdonságai. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia előadásai. p. 39-51. Szegedi Tudományegyetem (2021), [http://www.nytud.hu/oszt/korpusz/resources/sb\\_et\\_al\\_hadifoglyok.pdf](http://www.nytud.hu/oszt/korpusz/resources/sb_et_al_hadifoglyok.pdf)





## A CLARIN ParlaMint magyar korpusza

Üveges István<sup>1,2</sup>, Ring Orsolya<sup>2</sup>

<sup>1</sup>Szegedi Tudományegyetem, Nyelvtudományi Doktori Iskola

<sup>2</sup>Társadalomtudományi Kutatóközpont, Politikatudományi Intézet  
uvegesistvan898@gmail.com  
ring.orsolya@tk.hu

**Kivonat** Cikkünkben bemutatjuk CLARIN ParlaMint projekt keretében 2020 novembere és 2021 májusa között készült, a Covid19-járvány kommunikációjának vizsgálatára is alkalmas, egységes morfológiai és szintaktikai annotációt tartalmazó korpuszok között helyet kapó magyar nyelvű korpuszt, amely a magyar Parlamentben 2014 júniusa és 2020 decembere között elhangzott interpellációk és azonnali kérdések leiratait tartalmazza. Az eredeti leiratok a Magyar Országgyűlés honlapján<sup>1</sup> érhetőek el. Röviden ismertetjük a korpusz főbb szófaji statisztikáit, az alkalmazott (gépi) annotációs rétegeket, illetve bemutatunk néhány lehetséges alkorpuszokra való felbontást.

**Kulcsszavak:** parlamenti korpusz, clarin, msd, xml, Covid19

### 1. Bevezetés

A parlamentek a politikai kommunikáció fontos helyszínei, ahol a választott képviselők megvitatják a benyújtott törvényjavaslatokat és más országos jelentőséggel bíró ügyeket. Az itt elhangzó beszédek általában előre megtervezett beszédaktusok, mivel a képviselők kiemelt célja, hogy meggyőzzék a hallgatóságot és megszerezzék támogatásukat. A parlamenti viták jegyzőkönyveinek egyedi tartalma, szerkezete és nyelvezete fontos forrásai a társadalomtudományi és nyelvészeti kutatásoknak. A politikai kommunikáció korpuszokon és NLP módszereken alapuló kutatása az elmúlt időszakban kiemelt jelentőséget kapott, de megjelent tanulmányok legtöbbször a politikusok médiában és közösségi médiában megjelenő megnyilatkozásait elemzik (Gollust és mtsai, 2020; Mariani és mtsai, 2020; Aparicio és mtsai, 2021; Wang és mtsai, 2021; Rufai és Bunce, 2020).

A parlamenti viták leiratai lényegében a beszélt nyelv ellenőrzött és szabályozott körülmények között készült átiratai, melyek szabadon elérhetőek, mivel az információszabadságról szóló törvény alapján nem vonatkoznak rájuk a szerzői jogi vagy a személyes adatok védelmére vonatkozó jogszabályok. Éppen ezért az utóbbi években több nemzetközi projekt keretében készült és készül korpusz parlamenti felszólalásokból<sup>2</sup>.

<sup>1</sup> <https://www.parlament.hu/>

<sup>2</sup> Ilyen például a CLARIN <https://www.clarin.eu/>, a Comparative Agendas <https://www.comparativeagendas.net/> vagy az OPTED <https://opted.eu/> projekt

A CLARIN kutatási infrastruktúra keretében lezajlott ParlaMint projekt<sup>3</sup> célja egységesen kódolt, ezáltal jól összevethető többnyelvű, nyelvészeti annotációval ellátott korpuszok létrehozása volt. A projekt keretében 17 ország parlamenti felszólalásai kerültek feldolgozásra, összesen mintegy 500 millió token terjedelemben, amelyből a magyar korpusz mintegy 1,019,576 token.

A Magyar Országgyűlésben elhangzott interpellációkból és azonnali kérdésekből politikatudományi felhasználásra már készült korpusz a Hungarian Comparative Agendas Project keretében<sup>4</sup> amely ugyan nem tartalmaz nyelvészeti és szintaktikai annotációt, azonban minden tekintetben alkalmas volt arra, hogy a szükséges nyelvészeti és szintaktikai elemzésekkel és minimális metaadat kiegészítésekkel a nemzetközi korpusz részévé váljon, ezzel kapcsolódási lehetőséget teremtve a politikatudományi és nyelvtudományi célra épült korpuszok között.

A ParlaMint projekt során létrejött korpuszok időbeli eloszlása és nagysága is különböző. Néhány alapelvtől eltekintve a résztvevő kutatócsoportok döntésén alapult, hogy mely parlamenti beszéd típusokat, milyen időintervallumban dolgoznak fel. A létrehozott korpuszokban a 2019 novembere után keletkezett szövegek a Covid19-korpuszba, míg a korábbi szövegek a referenciakorpuszba kerültek. A referenciakorpuszok időhatára alkalmazkodhatott az egyes országok parlamenti ciklusaihoz, de a kezdődátuma nem lehetett 2015 utáni. A korpuszok CLARIN TEI XML séma<sup>5</sup> szerint készültek, emellett egységes szemléletű nyelvészeti és szintaktikai feldolgozáson esetek át.

Mivel a parlamenti beszédek leiratainak egyik fontos jellemzője, hogy közvetlenül reagálnak a bekövetkező eseményekre, így például a jelenlegi Covid19-világjárványra, a 17 nyelven létrehozott korpuszok az adatok szinkron és diakronikus összehasonlításán keresztül alkalmasak a járványhoz kötődő kommunikáció többnyelvű kontextusban történő vizsgálatára.

A korpuszok kiterjedt metaadat-struktúrával rendelkeznek a felszólalókról (név, nem, pártállás, képviselői státusz) és a parlamenti ülésekről, emellett minden beszéd mellett megtalálható előadójának aktuális szerepe (elnök, rendes előadó) is. A beszédek emellett tartalmazznak az elhangzott szövegekre vonatkozó olyan megjegyzéseket is mint például a közbeszólások, bekiabálások vagy a taps. A korpuszok letölthetőek a CLARIN.SI repozitóriumból<sup>6</sup> és elérhetőek noSketchEngine-en keresztül<sup>7</sup>. A repozitóriumban elérhetőek a korpuszvalidáláshoz használt XLST és Perl állományok, amelyek hasznosak lehetnek a TEI XML fájlok tovább alakítása esetén.

A tanulmány a következők szerint épül fel; a 2. fejezet a vizsgált szöveg típusokat ismerteti röviden, majd a 3. fejezetben a magyar korpusz főbb jellemzőit mutatjuk be, míg a 4. fejezet az XML sémában elhelyezett nyelvészeti annotációt, és az ennek elkészítéséhez használt eszközöket tárgyalja. A 5. fejezet-

<sup>3</sup> <https://www.clarin.eu/content/parlamint-towards-comparable-parliamentary-corpora>

<sup>4</sup> <https://cap.tk.hu/hu>

<sup>5</sup> <https://github.com/clarinsi/TEI-schema>

<sup>6</sup> <https://www.clarin.si/info/about-repository/>

<sup>7</sup> <https://www.clarin.si/noske/parlamint.cgi/>

ben részletesen kitérünk a szövegek nyelvészeti annotációinak néhány fontosabb TEI XML specifikus jellemzőjének ismertetésére. A tanulmányt ezt követően rövid konklúzió zárja.

## 2. Az interpelláció és azonnali kérdés, mint a parlamenti ellenőrzés eszközei

Az interpelláció és az azonnali kérdés a képviselők által gyakorolható hagyományos parlamenti ellenőrzési eszköz. Különbség közöttük a címzettek körében és a tárgyalási rendjükben van.

Az Alaptörvény 7 cikk (2) bekezdése szerint az országgyűlési képviselők joga, hogy interpellációt intézzenek a Kormányhoz és a Kormány tagjához a feladatkörükbe tartozó bármely ügyben<sup>8</sup>. Az interpelláció során a képviselő szóban ismerteti az interpelláció szövegét, majd a válasz és a képviselői viszontválasz következik. Végül a plenáris ülés szavaz arról, hogy elfogadja-e a választ, avagy elutasítva azt a kérdésről egy bizottsággal jelentést készítet. Az interpelláció címzettje csak a kormány vagy annak valamely tagja lehet. Az interpelláció szövegét napokkal elhangzása előtt be kell nyújtani (Magyar, 2018).

Az 1994-ben bevezetett azonnali kérdéseket a frakcióvezetők terjesztik be, majd képviselők mondják el. Minden héten legalább hatvan perc áll rendelkezésre az azonnali kérdésekre, és minden képviselőcsoportot megilleti a jog ezalatt legalább egy azonnali kérdés ismertetésére, melyeket legalább az ülés megkezdése előtt hatvan perccel be kell nyújtani.

## 3. A magyar korpusz jellemzői

Az elkészült magyar korpusz egy lezárt (2014-2018) és a jelenleg is folyamatban lévő (2018-) parlamenti ciklusban elhangzott valamennyi interpelláció és azonnali kérdés szövegét tartalmazza. A szövegeket web scraping segítségével kerültek letöltésre a Magyar Országgyűlés honlapjáról, az alapvető metaadatokkal együtt, a Hungarian Comparative Agendas Project keretében.

Az interpellációkból és az azonnali kérdésekből így létrehozott adatbázisban szereplő legfontosabb változók az alábbiakra terjednek ki: az interpelláció címe, az interpelláció betervezőjének neve, az interpelláció betervezésének időpontja, az interpellációk közpolitikai tartalma, a válaszadó neve és az Országgyűlés döntése a miniszteri válasz elfogadásáról. Az azonnali kérdések esetében pedig azok címe, a betervező neve, az azonnali kérdés közpolitikai tartalma, az azonnali kérdés címzettjének neve, valamint a betervezés időpontja. A CLARIN ParlaMint projekt során ezen adatbázisokat és a hozzá tartozó szövegállományt alakítottuk CLARIN TEI XML formátumú korpusszá.

<sup>8</sup> <https://njt.hu/jogszabaly/2011-4301-02-00.11>

### 3.1. Kereshetőség

Az online keresőfelület<sup>9</sup> lehetővé teszi többek között:

- konkordancia készítését lemma, frázis, szóalak, karakter vagy CQL alapon,
- szűrés kontextusra (+/- 15 token távolságig),
- a korpusz többféle felosztását, például parlamenti ciklusok vagy a koronavírus járvány kitörését megelőző, és az azt követő időszak felszólalásaira,
- keresést adott frakció hozzászólásaiban illetve nemek szerint is (a részletesebb beállítási lehetőségeket az 1. ábra szemlélteti).

The image shows a web interface titled "Text types" for searching through a corpus. At the top right, there is a "Subcorpus: create new" link. The interface is organized into several filter panels, each with a "Select All" button:

- SPEECH.SUBCORPUS:** Includes checkboxes for "COVID" and "Reference".
- SPEECH.FROM** and **SPEECH.TO:** Two input fields for date or range selection.
- SPEECH.TERM:** Includes checkboxes for "7" and "8".
- SPEECH.SPEAKER\_TYPE:** Includes a checkbox for "MP".
- SPEECH.SPEAKER\_ROLE:** Includes a checkbox for "Regular".
- SPEECH.SPEAKER\_PARTY:** A list of checkboxes for various political parties: DK, Fidesz, Jobbik, KDNP, LMP, MLP, MSZP, Párbeszéd, and független.
- SPEECH.SPEAKER\_PARTY\_NAME:** A list of checkboxes for specific party names: Demokratikus Koalíció, Fidesz - Magyar Polgári Szövetség, Jobbik Magyarországért Mozgalom, Kereszténydemokrata Néppárt, Lehet Más a Politika, Magyar Liberális Párt, Magyar Szocialista Párt, Párbeszéd Magyarországért, and independent.
- SPEECH.SPEAKER\_NAME:** An input field for the speaker's name.
- SPEECH.SPEAKER\_GENDER:** Includes checkboxes for "F" (female) and "M" (male).
- SPEECH.SPEAKER\_BIRTH:** An input field for the speaker's birth date.

At the bottom of the interface, there are two buttons: "Make Concordance" and "Clear All".

1. ábra. A korpusz keresőfelületének néhány beállítási lehetősége.

<sup>9</sup> <https://www.clarin.si/noske/parlamint.cgi/first.form?corpname=parlamint21.hu;align=>

Ahogy korábban kifejtettük a ParlaMint projekt célkitűzése szerint a létrejövő korpusz fő fókuszában az állt, hogy a Covid19 járvány megjelenését és hatásait a nemzeti parlamentekben elhangzó felszólalásokban egyszerűen követhetővé és vizsgálhatóvá tegye, ennek kapcsán tehát a felület natívan kezeli a korpusz illetően felosztását. Hasonlóan egyszerűen elvégezhető például az ellenzék - kormánypártok felosztás (a megfelelő pártok együttes kijelölésével), vagy akár a fentiek kombinálása a megfelelő parlamenti ciklus kijelölésével. Mindezeknek köszönhetően a szövegek alkalmasak lehetnek például az egyes pártoknak a koronavírussal összefüggő kommunikációja vizsgálatára, vagy akár konkrét képviselők felszólalásainak összevetésére is.

### 3.2. Leíró statisztikák

Ahogy már említettük, a magyar CLARIN ParlaMint korpuszt hozzávetőlegesen 1 millió tokenes szövegállomány alkotja. Az 1. táblázat néhány lehetséges felbontás szerint mutatja be a képezhető részkorpuszok szófaji statisztikáit.

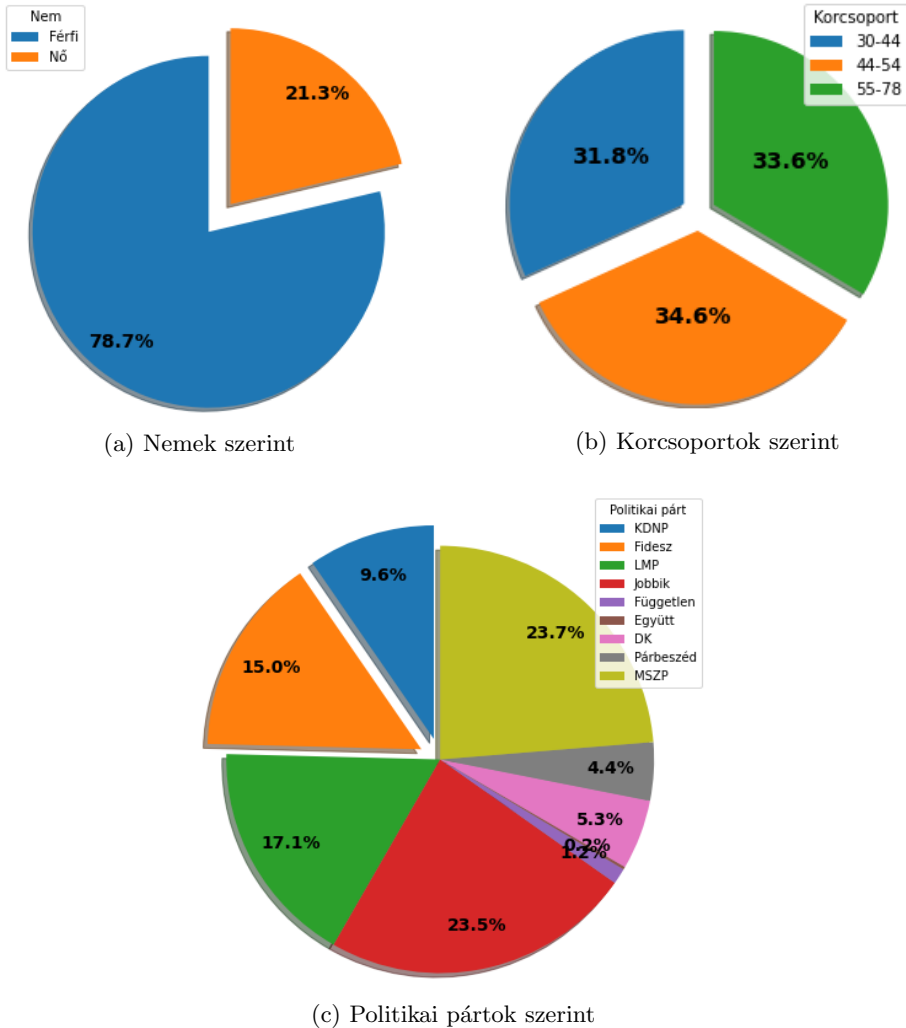
	post-covid	pre-covid	1.ciklus	2.ciklus	ellenzék	kormány
token	198.930	820.646	635.791	383.785	798.449	221.127
mondat	11.030	43.968	33.995	20.983	44.189	13.654
NOUN	43.923	187.066	145.129	85.850	178.393	57.349
ADJ	21.585	92.898	71.504	42.962	81.859	30.612
PRON	14.492	56.721	44.013	27.218	59.122	10.608
CONJ	8.895	36.664	28.223	17.325	35.034	9.739
NUM	4.123	21.262	17.254	8.143	19.427	5.543
VERB	22.354	86.680	66.888	42.149	87.269	20.257
ADV	15.635	62.682	48.497	29.899	64.169	12.696
PROP	5.958	24.943	18.964	11.930	23.570	8.221
ADP	2599	11.645	9.031	5.221	10.524	3.401
AUX	1	1	1	1	1	1
DET	20.420	84.150	65.221	39.324	82.164	24.341
INTJ	273	873	670	475	969	170
PART	761	2.769	2.110	1.423	2.858	643
PUNCT	31.734	128.756	100.218	60.213	128.624	32.892
SCONJ	6.158	23.454	17.999	11.620	24.384	4.627
SYM	0	5	5	0	4	1
X	18	76	62	31	72	25

1. táblázat. Különböző szófaji címkék számossága a CLARIN ParlaMint korpusz néhány felbontása esetén.

Tekintettel arra, hogy a korpuszba kerülő szövegek időarányosan kerültek kiválogatásra, így a Covid19 kitöréséhez képest kialakítható részkorpuszok aszimmetrikusan alakulnak a járványt megelőző időszak javára. Az ellenzék - kormány felosztás hasonló mértékű aszimmetriája (a részkorpuszok tokenszámát tekintve) az interpellációk / azonnali kérdések természetének tudható be; itt ellenzéki

pártok intéznek kérdést a kormányhoz, majd a válasz után szintén az ő viszontválaszuk következnek.

A 2. ábra az XML-ben kódolt metaadatok alapján kiválogatott felszólalások számarányát mutatja három lehetséges bontásban.



2. ábra. Felszólalások arányai a CLARIN ParlaMint magyar korpuszában.

Nemek szerinti csoportosítva a felszólalásokat azt látjuk, hogy jelentős túlsúly mutatkozik a férfi képviselők javára a nőkkel szemben (78,7% a 21,3%-kal szemben). Érdeemes megemlíteni, hogy a női képviselők aránya a parlamentben az első

(2014-2018-ig tartó) parlamenti ciklusban, amelyet a korpusz tartalmaz 10.1% körül alakult, míg a második ciklus (2018-) esetében ez az arány 12,6% körül alakul<sup>10</sup>.

Habár ez a 3. legalacsonyabb arány Európában (az Európai Unió átlaga nagyjából 30% körül mozog), a női képviselők hozzászólásainak a nők számarányához mért közel kétszeres aránya arra enged következtetni, hogy igen aktívan részt vesznek a parlamenti üléseken zajló politikai diskurzusban.

Korcsoportok szerint osztályozva a képviselőket a két ciklus átlagában jól kirajzolódik egy öregedő korfa; a jelen ciklusban mindösszesen 2 fő 30 év alatti képviselő rendelkezik mandátummal, és ez a szám a megelőző ciklusban is mindössze 4 fő volt. A leginkább jellemző a felszólalók között az 50 év körüli életkor volt.

Politikai pártok tekintetében a legaktívabbnak egyértelműen az MSZP képviselői tekinthetők; az 1. ciklusban a képviselői mandátumok 14,5%-a, a 2. ciklusban pedig 7,5% -a volt a párt birtokában, ezzel szemben ők adták az összes felszólalás mintegy 23,5% -át. A legkevésbé aktív ezzel szemben egyértelműen a Fidesz volt, akik 58,2% illetve 58,8%-nyi mandátumukhoz a hozzászólások 15%-ával rendelkeznek a két ciklus összesítésében. Kormánypárt - ellenzék szerint polarizálva a számosságokat 24,6% adódik a 75,4% ellenében, amely a mandátumok eloszlásának (66,83% a 33,17% ellenében) közel fordítottja. Ezek az arányok mind az MSZP (ellenzék) - Fidesz (kormány), mind az ellenzék - kormány viszonylatban egyértelműen a beszéd típusok bevezetőben említett jellegzetességével magyarázhatóak.

### 3.3. Lexikai alapú doménhasonlóság

Annak érdekében, hogy képet kaphassunk a felépített korpusz hasonlóságáról más domének szövegeihez viszonyítva, a korpusz szövegét a Jaccard-távolság metrika felhasználásával összevetettük a Szeged Treebank (Vincze és mtsai, 2010) 6 részkorpuszával, amelyek tartalma;

- iskolai fogalmazások,
- szépirodalmi szövegek,
- számítástechnikai szövegek,
- újsághírek,
- jogi szövegek,
- valamit üzleti rövidhírek

közül került ki. A Jaccard-távolság alapját a vizsgált szövegek szókészlete adja; arról ad visszajelzést, hogy az összevetett szövegek esetében mekkora arányú a közös szókincs, értéke 0 és 1 között változik, ahol 1 a tökéletes egyezést jelenti, 0 pedig azt, hogy a két mért szöveg szókincse diszjunkt halmazt alkot<sup>11</sup>. A mért távolságokat a 2. táblázat mutatja be.

<sup>10</sup> Forrás: Eurostat ([https://ec.europa.eu/eurostat/databrowser/view/sdg\\_05\\_50/default/-table?lang=en](https://ec.europa.eu/eurostat/databrowser/view/sdg_05_50/default/-table?lang=en))

<sup>11</sup>  $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ , ahol A és B a két szöveg szavaiból (pl. lemmák) képzett halmazok.

	Jaccard távolság
Üzleti rövidhírek	0,8390
Számítástechnika	0,8386
Szépirodalom	0,8450
Újsághírek	<b>0,7551</b>
Iskolai fogalmazások	0,8354
Jogi szövegek	<b>0,8701</b>

2. táblázat. A CLARIN ParlaMint korpusz Jaccard-távolsága a Szeged Treebank egyes részkorpuszaitól.

Ez alapján az elkészült korpusz legtávolabb az újsághírek szókincsétől helyezkedik el, míg a legnagyobb átfedést a Szeged Korpuszban a jogi doménbe sorolt szövegekkel mutatja. A távolságok megoszlásában az újsághírek 0.75-ös értéke egyértelműen szélsőségesnek számít, de kiugrónak tekinthető a legközelebbi részkorpusz (jogi szövegek) 0.87-es értéke is, tekintettel arra, hogy a fennmaradó 4 részkorpusz távolsága meglehetősen homogén (rendre 0.83 - 0.84 körül ingadozik).

A jogi szövegekkel vett legnagyobb hasonlóság várható volt, tekintettel arra, hogy a parlamenti felszólalások témája sok esetben a jogalkotási folyamathoz kötődik, ami eszerint tehát világosan leképeződik a használt szókincsében is. Az újsághírektől vett (a többi részkorpuszhoz képest) kiugróan nagy távolság feltehetőleg azok témaválasztásbeli változatosságával magyarázható; a kevésbé egységes topikok szétartóbb szókincsét eredményezhetnek.

#### 4. Annotációs rétegek

Ahogy említettük, annak érdekében, hogy a korpuszban helyet kapó valamennyi nyelvű leiratok összevethetőek maradjanak, azokat egységes nyelvészeti annotációval kellett ellátni. A korpuszból a munka során két változat készült, amelyek közül a nyelvészeti elemzett korpuszvariánsnak az alábbi annotációkkal kellett rendelkeznie:

- Univerzális Dependencia (UD) szerinti szintaktikai elemzés (Zeman és mtsai, 2020)
- az egyes tokenekhez a megfelelő MSD kód hozzárendelése (Erjavec, 2012)
- a mondatokban szereplő névelemek taggelése.

Tekintettel arra, hogy magyar nyelvre egyben egyetlen elemző sem biztosítja mind a három fenti standard szerinti kimenetet, ezért az előelemzés három különböző eszköz kimenetének egyesítésével volt csak megoldható. Az Univerzális Dependencia szerinti függőségi nyelvtani címkézést a UDPipe 2.0 elemző (Straka, 2018) REST API -ként elérhető szolgáltatásával valósítottuk meg, az MSD kódolást a magyarlanc régebbi, 2.0-ás változata (Zsibrita és mtsai, 2013), a



névelemek azonosítását és csoportokra bontását pedig a Szegedi Tudományegyetem Mesterséges Intelligencia Kutatócsoportjában fejlesztett névelem-felismerő (Szarvas és mtsai, 2006) biztosította.

A fenti eszközök mindegyike az előfeldolgozás lépéseként tokenizálja és mondatokra szegmentálja a kapott szöveget, azonban ezek a felbontások az egyes elemzők esetében nem feltétlenül esnek egybe. Ennek következtében az egyes kimenetek egyesítése során szükséges volt kiválasztani egy "etalont", amelybe a többi címkékészlet elemeit integráljuk. Erre a célra (lévén mind közül ez a legkorszerűbb) a UDPipe elemző kimenetét választottuk, más szóval az ezáltal előállított tokenekhez kerestünk a másik két elemző kimenetében megfelelő címkézést. A címkéket akkor tekintettük megfeleltethetőnek, és egyesítettük egy közös formátumba, amikor a tokenizálás azonos eredményt hozott valamennyi eszköz esetében.

A kimenetek összeillesztése automatikusan történt. Azokban az esetekben, amikor a magyarlanc 2.0 vagy a névelem-felismerő címkézése az eltérő tokenizálás miatt nem volt konzisztens a UDPipe kimenetével, az ilyen módon hiányzó címkék helyére technikai adatokat helyeztünk el, indikálva, hogy a megfelelő kimenetek nem voltak egyesíthetők. Az eredeti teljes szövegmennyiség egy kisebb részhalmazán végzett kézi ellenőrzés alapján ilyen hibák az összes elemzett szövegnek mintegy néhány százalékát érintették.

## 5. TEI XML integráció

A fenti lépések során kinyert morfológiai és szintaktikai információkat a munka következő fázisában a projekt alapját képező Parlamint TEI XML sémába<sup>12</sup> illesztettük. A séma teljes leírása messze meghaladná a jelen tanulmány kereteit, így itt csak néhány fontosabb elem ismertetésére térünk ki.

Amennyiben minden fázis sikeresen végbemenet, a morfológiai annotációt egy `<w>`-tag zárta egységbe, amelynek attribútumai és értéke a következők szerint alakult:

```
(1) <w pos="Pd3-sn"
      lemma="olyan"
      msd="UPosTag=ADJ | Case=Nom | Degree=Pos | Number=Sing"
      xml:id="IC7_157_2.2.2.1">
      Olyan
</w>
```

Fontos kiemelni, hogy a használt XML séma elvárásainak megfelelően az MSD kód a `pos` attribútum értékeként, a UDPipe által meghatározott morfológiai jegyhalmaz pedig az `msd` attribútum értékeként jelent meg. Az XML tag-ben ezen felül még az adott tokenhez a UDPipe által rendelt `lemma` szerepelt a neki megfelelő attribútum értékeként, az `xml:id` pedig a tokennek az

<sup>12</sup> <https://clarin-eric.github.io/parla-clarin/>

adott nap parlamenti felszólalásai között elfoglalt helyét jelölte ki (a fájl azonosítója, pl.: IC\_157\_2, majd pontokkal elválasztva a fájlban belüli felszólalás sorszáma, azon belül a mondat és a mondaton belül a konkrét token sorszáma). A névelemek jelzésére a <name> tag szolgált, amely több tokenes névelemek esetén magában foglalta valamennyi tokent (<w>), és amelynek **type** attribútuma a névelem típusát jelölte (ORG - organization, PER - Person, LOC - Location vagy MISC - Miscellaneous):

```
(2) <name type="ORG">
      <w lemma="kuria"
          msd="UPosTag=PROPN|Case=Sub|Number=Sing"
          xml:id="IC7_165_2.1.4.3">
          Kuriara
      </w>
</name>
```

Az egyes mondatok tokenenként kódolt morfológiai információi után a szintaktikai elemzés **linkGrp** tagen belül foglalt helyet; minden token egy önálló **link** taget kapott, amelynek **ana** attribútuma adja meg az UD szintaktikai élcímét, a **target** attribútum pedig az él kiindulását és érkezését:

```
(3) <link ana="ud-syn:det"
      target="#IC7_164_2.1.3.5 -#IC7_164_2.1.3.1" />
```

A fenti példában a #IC7\_164\_2.1.3.1 jelenti az aktuális token számát; ez lesz az él kiindulása, a #IC7\_164\_2.1.3.5 pedig az él érkezési tokenjét kódolja, így ezek végig követhetők a teljes szintaktikai fa visszafejthető. Az UD elemzésben a mondat fejének tekintett **root** komponens annyiban speciális, hogy őseként a mondat azonosítója van megjelölve, tokenszám nélkül (a fenti példa esetében: #IC7\_164\_2.1.3).

A korpuszban emellett jelölve lettek a parlamenti leiratozók által feltüntetett különféle hanghatások (mint például taps, csengetés). Ezeket a **kinesic** tagek hivatottak kódolni az egyes megszólalások teljes szövege után (vagyis jelzésük nem a felszólalás alatti elhangzás valós ideje szerint történt), pl.:

```
(4) <kinesic type="vocal">
      <desc>(Zaj. – Az elnök csenget.)</desc>
</kinesic>
```

A korábban már említett központi **Git** repozitóriumban a korpusznak két xml variánsa található meg. Ezek között a fő különbség, hogy míg az egyik a nyelvészeti annotált, a fentieknek megfelelő tag-eket magukban foglaló fájlokat tartalmazza (.ana.xml kiterjesztéssel), addig a másik változatban a hozzászólások hagyományos szöveges formában, néhány hozzájuk rendelt metaadattal szerepelnek, felszólalások szerint bontva.

Ezek esetében a metaadatok közül közvetlenül a felszólaló neve érhető el (pl.: `<note>DR. TÓTH BERTALAN (MSZP):</note>`), a többi metaadat a felszólalás azonosítójához rendelve érhető el a ParlaMint-HU.xml fájlba szervezve. Egy-egy képviselőhöz például az 1. fejezetben említett adatok a következők szerint kereshetők:

```
(5) <person xml:id="TiborBana">
      <persName>
        <forename>Tibor</forename>
        <surname>Bana</surname>
      </persName>
      <sex value="M">Ferfi</sex>
      <birth when="1985">1985</birth>
      <affiliation role="member" ref="#party.FUGGETLEN" />
      <affiliation role="MP" />
    </person>
```

A `<teiHeader>` mindkét esetben tartalmazza például az egyes fájlok további metaadatait (mint amilyen az alkalmazott tag-ek száma), így azok különösen hasznosak lehetnek leíró statisztikák készítéséhez. Külön fájlba szervezve, szintén a `<teiHeader>` tag tartalmazza a teljes korpuszra és metaadatokra vonatkozó összesített információkat is.

## 6. Összegzés

Cikkünkben röviden bemutatottuk a CLARIN kutatási infrastruktúra ParlaMint projekt keretében készült, magyar nyelvű parlamenti felszólalásokat tartalmazó korpuszát. A korpuszban foglalt szöveganyag, illetve az elkészült morfológiai és szintaktikai annotáció, illetve a korpusz metaadatai lehetőséget teremtenek például a különböző parlamenti frakciók kommunikációjának elemzésére, a képviselők megnyilatkozásainak vizsgálatára a Covid19-et megelőző és az azt követő időszakban, vagy éppen a képviselők felszólalásainak kor-, nem- és pártállás szerinti bontásban történő elemzésére is.

A jövőben a korpusz kiegészítését tervezzük más parlamenti beszéd típusokkal, valamint további parlamenti ciklusok felszólalásainak szöveganyagával, ezáltal teret biztosítva széles körű nyelvészeti és társadalomtudományi vizsgálatoknak.

## Köszönetnyilvánítás

A publikációban szereplő kutatást, amelyet a Társadalomtudományi Kutatóközpont valósított meg, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia

Nemzeti Laboratórium keretében. Külön köszönet illeti a Társadalomtudományi Kutatóközpont Comparative Agendas Project kutatócsoportjának tagjait és gyakornokait a felhasznált korpuszok előkészítéséért.

The research was supported by the European Union’s Horizon 2020 research & innovation programme under Grant Agreement no. 951832.

The research was supported by CLARIN ERIC ParlaMint Project.

## Hivatkozások

- Aparicio, J.T., de Sequeira, J.S., Costa, C.J.: Emotion analysis of portuguese political parties communication over the covid-19 pandemic. In: 2021 16th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6. IEEE (2021)
- Erjavec, T.: Multext-east: morphosyntactic resources for central and eastern european languages. *Language Resources and Evaluation* 46(1), 131–142 (2012), <http://www.jstor.org/stable/41486069>
- Gollust, S.E., Nagler, R.H., Fowler, E.F.: The emergence of covid-19 in the us: a public health and political communication crisis. *Journal of health politics, policy and law* 45(6), 967–981 (2020)
- Magyar, Z.: A parlamenti ellenőrzés eszközei az országgyűlés gyakorlatában. *Parlamenti Szemle* 2, 125–150 (2018)
- Mariani, L.A., Gagete-Miranda, J., Retti, P.: Words can hurt: How political communication can change the pace of an epidemic. *Covid Economics* 12, 104–137 (2020)
- Rufai, S.R., Bunce, C.: World leaders’ usage of twitter in response to the covid-19 pandemic: a content analysis. *Journal of public health* 42(3), 510–516 (2020)
- Straka, M.: UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 197–207. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), <https://aclanthology.org/K18-2020>
- Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. In: Todorovski, L., Lavrac, N., Jantke, K.P. (szerk.) *Discovery Science*. pp. 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
- Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., Csirik, J.: Hungarian dependency treebank. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10). European Language Resources Association (ELRA), Valletta, Malta (May 2010), [http://www.lrec-conf.org/proceedings/lrec2010/pdf/465\\_paper.pdf](http://www.lrec-conf.org/proceedings/lrec2010/pdf/465_paper.pdf)
- Wang, Y., Croucher, S.M., Pearson, E.: National leaders’ usage of twitter in response to covid-19: A sentiment analysis. *Frontiers in Communication* p. 183 (2021)

Zeman, D., Nivre, J., Abrams, M.: Universal dependencies 2.6 (2020), <http://hdl.handle.net/11234/1-3226>, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A tool for morphological and dependency parsing of hungarian. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013. pp. 763–771. INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria (9 2013), <https://aclanthology.org/R13-1099>



# Towards abstractive summarization in Hungarian

Márton Makrai<sup>1,2</sup>, Ákos Máté Tündik<sup>1</sup>, Balázs Indig<sup>3,4</sup>, György Szaszák<sup>1</sup>

<sup>1</sup> BME Department of Telecommunications and Media Informatics  
`{tundik,szaszak}@tmit.bme.hu`

<sup>2</sup> MTA Research Institute for Natural Sciences  
`makrai.marton@ttk.hu`

<sup>3</sup> Eötvös Loránd University Department of Digital Humanities  
`indig.balazs@btk.elte.hu`

<sup>4</sup> Digital Heritage National Laboratory

**Abstract.** We publish an abstractive summarizer for Hungarian, an encoder-decoder model initialized with huBERT, and fine-tuned on the ELTE.DH corpus of former Hungarian news portals. The model produces fluent output in the correct topic, but it hallucinates frequently. Our quantitative evaluation on automatic and human transcripts of news (with automatic and human-made punctuation) shows that the model is robust with respect to errors in either automatic speech recognition or automatic punctuation restoration.

**Keywords:** summarization, pre-trained model, automatic speech recognition, punctuation

## 1 Introduction

Automatic text summarization requires several complex language abilities: understanding the text, discriminating what is relevant, and writing a short synthesis (most commonly a couple of sentences). *Extractive* systems select sentences or words from the input document, while *abstractive* models are supposed to paraphrase the content. Extractive methods suffer from some limitations, including weak coherence between sentences, inability to simplify complex and long sentences, and unintended repetition (Hasan et al., 2021). In the past few years, pre-trained deep language models have achieved great advancements in both extractive and abstractive summarization (Edunov et al., 2019; Liu and Lapata, 2019; Rothe et al., 2020), but these models are heavily data-driven, and multilingual abstractive summarization datasets (Giannakopoulos et al., 2015; Scialom et al., 2020; Ladhak et al., 2020; Hasan et al., 2021) miss Hungarian.

We applied the pre-training-based method to Hungarian by fine-tuning an encoder-decoder model. The model has been initialized with huBERT (Nemeskey, 2020), the freely available Hungarian deep language model, as both the encoder and the decoder. The fine-tuning corpus consisted of news from former Hungarian portals. These articles have been crawled by the Hungarian National laboratory of Digital Heritage (Indig et al., 2020). Then the model is also evaluated on a Hungarian TV broadcast database Varga et al. (2015).

Qualitative analysis shows that our model produces fluent texts whose topic closely match that of the input, but the summaries contain much hallucination. The name of the model, `fosztogatnak2osztogatnak` refers to this phenomenon, more specifically an old Hungarian pun, which can be roughly translated as follows: People ask the Yerevan radio if it is true that Moskvitches are being handed out for free in Yerevan. Reply: “The news is true. However, not in Yerevan, but in Tbilisi. Not Moskvitches, but Volgas. And they are not handed out (Hungarian: *osztogatnak*) but plundered (*fosztogatnak*). We share our model at <https://huggingface.co/BME-TMIT/foszt2oszt>.

## 2 Related work

### 2.1 Evaluation methods

One of the greatest problems of summarization research resides in evaluation. A common metric to automatically evaluate summaries is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004), which counts the overlap of words or word units. Variants of this originally recall-oriented metric remain a strong baseline.

Fabbri et al. (2021) demonstrate the usefulness of some simple data statistics. The first three are *extractiveness* measures introduced by Grusky et al. (2018): *extractive fragment coverage* is the percentage of words in the summary that are from the source article, measuring the extent to which a summary is a derivative of a text; *density* is the average length of the extractive fragment to which each summary word belongs; and *compression ratio* is defined as the word ratio between the articles and the summary. Fabbri et al. also include the percentage of n-grams in the summary not found in the input document as a *novelty score*, and that of repeated n-grams in the summary as a *redundancy score*. Human evaluations show that summaries generated by recent models such as Pegasus (Zhang et al., 2019), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020) score higher in some respect than reference summaries.

### 2.2 Factual faithfulness and hallucination

Huang et al. (2021) review experiments for solving the problem with neural encoder-decoder models pioneered by the Seq2Seq framework that while they produce summaries that are more abstractive, more readable, closer to human-edited summaries, they distort the article or generate fabrication of factual information in it. Previous evaluation methods of text summarization are not suitable for detecting this. The current research in response, unfortunately yet limited to English, is predominantly divided into fact-aware evaluation metrics to select outputs without factual inconsistency and new summarization systems optimized towards factual consistency.

Maynez et al. (2020) conduct a large scale human evaluation of neural extreme abstractive summarization models (extremity means that the summary is



a single sentence) and the types of hallucinations they produce, and find substantial amounts of hallucinated content in all model generated summaries.

They are interested in how frequently abstractive summarizers hallucinate content; whether models hallucinate by manipulating the information present in the input document (intrinsic hallucination) or by adding information not directly inferable from the input document (extrinsic); how much hallucinated content is factual, even when unfaithful; and whether there are automatic means of measuring these hallucinations (at least for English).

Their experiments suggest that intrinsic and extrinsic hallucinations happen in more than 70% of 1-sentence summaries; the majority of hallucinations are extrinsic, which potentially could be valid abstractions that use background knowledge, but over 90% of them are erroneous; models initialized with pre-trained parameters perform best both on automatic metrics and human judgments of faithfulness/factuality: they have the highest percentage of factual within extrinsic hallucinations, at least on in-domain summarization. Textual entailment measures better correlate with faithfulness than standard ones, potentially leading the way to automatic evaluation metrics as well as training and decoding criteria for English. For Hungarian, these high level models are unfortunately yet unavailable.

The same limitation makes the findings of Gabriel et al. (2021) less relevant for Hungarian. They introduce a meta-evaluation framework for evaluating factuality evaluation metrics. They define conditions to evaluate factuality metrics on diagnostic factuality data across three summarization tasks. They find that question-answering metrics improve over standard metrics that measure the factuality of English summaries across domains, but their performance is highly dependent on the way in which questions are generated.

A survey by Huang et al. (2020) find that under similar settings, extractive summarizers are in general better than their abstractive counterparts thanks to strength in faithfulness and factual-consistency; milestone techniques (See et al., 2017) such as copy, coverage, and hybrid extractive/abstractive methods bring specific improvements but also show limitations; pre-training techniques, and in particular sequence-to-sequence pre-training, are highly effective for improving text summarization, with BART giving the best results.

### 2.3 Multilingual and medium-resource summarization

We follow the footsteps of MLSum (Scialom et al., 2020), a multilingual summarization dataset obtained from online newspapers. MLSum contains 1.5M+ article-summary pairs in 5 languages (French, German, Spanish, Russian, and Turkish) besides English. Scialom et al. conduct cross-lingual comparative analyses based on state-of-the-art systems. They offer strong baselines from multilingual abstractive text generation models, and distinguish between two theoretically independent factors to explain differences of results in cross-lingual summarization with different approaches: data (e.g. structure of the article, the abstractiveness of the summaries, quantity) and language (e.g. metric biases due to a different morphological type). The former has more to do with domain

adaptation, while the latter motivates further the development of multilingual datasets, since they are the only means to study such phenomena.

Giannakopoulos et al. (2015) and Ladhak et al. (2020) publish multilingual summarization benchmarks. Hasan et al. (2021) publish a dataset, a crawling curation tool, and summarization model checkpoints for multilingual summarization. All the of these sources miss Hungarian.

Most similar to us, Yang Zijian et al. (2021) report extractive and abstractive summarization systems for Hungarian. Their best results are obtained with huBERT, the same model we utilize. Another similarity is that their model is fine-tuned on news (HVG and index.hu), besides the MARCELL legislative corpus (Váradi et al., 2020). They mention the problem of using lead as summarization: Often the function of the lead is to attract attention and not to summarize, e.g. one of their articles is about damages caused by storms and the payments by the insurers. The lead is only about the insurers, it does not even mention the storm and the damages. Unfortunately, their model is not freely available.

## 2.4 Neural text generation

In the encoder-decoder setup, models are trained (or fine-tuned) to maximize the probability of the document-summary pairs provided in the training data, and at inference time, the output of the encoder is used for generating the summary. While fine-tuning is relatively computation intensive, and we experimented only with a couple of promising settings, we experimented with some methods to reduce hallucination – with limited success. Zarrieß and Schlangen (2018) give a survey of decoding methods in neural language generation grouped by their objectives: sequence-likelihood, diversity, and task-specific linguistic constraints or goals. The most well-known and de facto standard decoding procedure remains beam search (Lowerre, 1976). Our experiments with some of the recent methods can be found in Section 3.2.

## 2.5 Spoken document summarization

A challenging task is spoken document summarization, where additional factors can make the problem a bit more complex. In Tündik et al. (2019), the authors were analysing what additional distortion effects can arise during summarization in the case the document source is audio, and an automatic speech recognizer (ASR) is used to obtain the transcripts. These transcripts are then used as bases for the same summarization algorithm which is used for text documents. The three main differences and challenges are the following:

- ASR errors can propagate further into the processing pipeline;
- Due to missing punctuation marks and capitalization, tokenization (crucial in extractive summarization) lacks the necessary cues;
- Spoken documents may follow a different structure and can grammatically be different from written documents, hence there is a mismatch with models trained on written documents.

In Tündik and Szaszák (2019) the authors evaluated punctuation efforts on Hungarian and English ASR output. In Tündik et al. (2019) a similar punctuation model was applied to ASR output and assessed for distortions in the generated summaries. In the case of extractive summarization, punctuation errors were found to be slightly more critical than ASR errors, although this finding is limited by the summarization and its evaluation approaches, as in extractive summary, misplaced sentence boundaries will result in different N-gram sequences and hence most likely different ROUGE scores. With abstractive summarization, we hypothesize that punctuation errors become less impactful, but the difference between spoken and written documents becomes more relevant, beside the impact of ASR errors in both cases. Following Tündik et al. (2019) where extractive summarization was the target, here we focus on the effect of word and/or punctuation errors in case of abstractive summarization task. It is especially interesting whether the mismatch between spoken and written document styles and the ASR errors lead to more hallucinations than in the written based summarization.

### 3 Experimental Results

#### 3.1 The ELTE.DH corpus: former Hungarian news portals

The used corpus contains a subset of the continuously growing *ELTE.DH corpus* (Indig et al., 2020). The ELTE.DH corpus is created by crawling the archives of Hungarian news portals into the ISO standard *WARC (Web ARChive)* format. The downloaded material was converted into standard TEI XML format (Schreibman et al., 2008) by carefully extracting the metadata and the text content and the result was deposited at Zenodo.org repository where it is available for research purposes upon request for the sake of reproducibility. Compared to the common crawling methods, these XML files not just have paragraphs, but have all the textual data (and no boilerplate) with the formatting kept. All available metadata for each article and the typological formatting is normalized to eliminate portal specific markup. The standard TEI XML format is widely used on the field of digital humanities as long-term archiving format for textual data which ensures its usability for various tasks.

For most of the portals, the special first paragraph of each article, which we will call the *lead*, is commonly used to summarize the content and attract the reader to read further for details. This intention, if it is marked by some kind of emphasis, is identified and noted in the XML files, and can be used for the summarisation task. For the purposes of *fosztogatnak2osztogatnak*, we recorded the site of origin, the date, the title, the lead, and the body text of each article.

Table 1 shows how many articles and leads we have after a rather generous length-filtering. (Unlike Straka et al. (2018), we did not drop articles with a low text-to-abstract ratio. Future work may investigate this direction.) The train/validation/test split has been designated on a chronological basis, similarly to Scialom et al. (2020). We ensure a ratio of 8:1:1 by assigning articles

site	# articles (>50 char)	has lead (>20 char)
Magyar idők	163 609	82 %
válasz	84 714	86 %
vs	51 302	93 %
abcúg	2 798	94 %
mosthallottam	389	80 %

**Table 1.** The number of articles (consisting of at least 50 characters) crawled from each former news site in our corpus, and the ratio of the articles that have a lead consisting of at least 20 characters.

before 2017 October 9 to the training set, those after 2018 May 30 to the test set, and those in between to the development set. The rationale of a chronological split is that new topics appear over time, and this method prevents asking the model to extract an article about an event that is present in the training data from another portal. Our pilot results reported in the next section are based on one percent of the validation set. The test set is not used for anything, but it is delineated for future work.

### 3.2 Fine-tuning and results on ELTE.DH corpus

In fine-tuning and inference, we followed a jupyter notebook<sup>1</sup> by Patrick von Platen. Most hyper-parameters are the same as those by von Platen, but we found it advantageous to change the minimum length of the summary to 8 word-pieces (instead of 56), and the number of beams in beam search to 5 (instead of 4). We experimented with other parameter settings on one percent of the validation set, but they led to inferior ROUGE-scores (F1-score of stemmed bigrams), as shown in Table 2. The stemmed score is more informative, because when the unstemmed version returns 0, we can still learn something from the stemmed metric. Stemming in ROUGE has originally been motivated with the more accurate measurement of semantic compliance. One can argue that nowadays the models can achieve such semantic quality that a fluidity/fluency becomes also a key evaluation criterion. However, our model has difficulties with factual faithfulness, rather than fluency, so stemmed ROUGE seem more relevant.

ROUGE has three hyper-parameters: whether we apply morphological stemming to summaries (both the gold and the generated one),  $n$  (1, 2 or “longest”), and the direction of the comparison (precision, recall, or F-score). We used spaCy<sup>2</sup> for stemming, and the `rouge` Python package for computing ROUGE. Comparing these  $2 \times 3 \times 3$  settings, scores obtained in all pairs of settings correlate well (Pearson  $\geq 0.47$  on the output of our model).

<sup>1</sup> [https://github.com/patrickvonplaten/notebooks/blob/master/BERT2BERT\\_for\\_CNN\\_Dailymail.ipynb](https://github.com/patrickvonplaten/notebooks/blob/master/BERT2BERT_for_CNN_Dailymail.ipynb)

<sup>2</sup> <https://github.com/spacy-hu/spacy-hungarian-models>

<sup>3</sup> `num_beam_groups=5`, (Vijayakumar et al., 2016)

				stemmed		
	rouge-1	rouge-2	rouge-l	rouge-1	rouge-2	rouge-l
fosztozatnak2osztozatnak	<b>19.85</b>	<b>06.71</b>	<b>17.15</b>	<b>26.95</b>	<b>10.80</b>	<b>22.57</b>
num_beams=4	19.46	06.32	16.81	26.76	10.44	22.35
num_beams=6	19.51	06.18	16.79	26.61	10.32	22.26
top_k=50 (Fan et al., 2018)	19.08	05.97	16.83	25.75	09.67	21.94
diversity_penalty=0.5 <sup>3</sup>	18.60	05.95	16.41	25.70	09.43	21.76
von Platen	18.47	05.53	16.02	25.83	09.24	21.69
top_p=0.9 (Holtzman et al., 2020)	18.44	05.65	16.11	25.27	09.24	21.33
temperature=0.7 (Ackley et al., 1985)	17.80	05.63	15.55	23.64	09.20	19.87

**Table 2.** ROUGE F-scores obtained with alternative generation strategies.

Tables 3 to 5 illustrate the text generation on the ELTE.DH corpus with some examples.

<b>Title</b>	Lakóháznak ütközött egy kisrepülőgép San Diegóban
<b>Body</b>	A híradás szerint a baleset szombaton történt egy Beechcraft Bonanza típusú kisrepülőgéppel, amelynek a fedélzetén négyen voltak. Brian Fennessy, a San Diegó-i tűzoltóság parancsnoka elmondta, hogy a gép pilótája súlyos hajtóműhibáról értesítette az irányítótornyot röviddel azután, hogy felszállt a Mongtomery-Gibbs repülőtérről...
<b>Summary</b>	Lezuhant egy kisrepülőgép az egyesült államokbeli San Diegóban, a balesetben négyen meghaltak – közölte a helyi tűzoltóság.
<b>Phenomena</b>	Correctly recognized topic (flight accident), extra information about the country is also correct: San Diego is in the US.

**Table 3.** Text generation on the ELTE.DH corpus - Example 1.

<b>Title</b>	Civil szakértők bírálják a MÁV-Startot
<b>Body</b>	... a vasúttársaság 2015 végén jelentősen ritkította a legforgalmasabb fővonalak gyorsvonatait... Az év végével megszüntetik a Budapest–Berlin éjszakai vonatot... MÁV-Start ... A vasúttársaság lapunk kérdésére azt közölte,
<b>Summary</b>	A Magyar Államvasutak Zrt. és a MÁV - START Zrt. megállapodott abban, hogy megszüntetik a Budapest – Bécs vasútvonalat – közölte a vasúttársaság szerdán az MTI - vel.
<b>Phenomena</b>	Correctly mentioned topic related to Hungarian railway companies, but wrong information about discontinuing Budapest-Wien railway line operation. Hungarian telegraph office and the time are referenced incorrectly.

**Table 4.** Text generation on the ELTE.DH corpus - Example 2.

<b>Title</b>	Útravaló mindenkinek
<b>Body</b>	...fogalmazott Prőhle Gergely, a Petőfi Irodalmi Múzeum főigazgatója a sajtótájékoztatón a Márvány utcai iskolában, amelynek üres tűzfala Arany János hatalmas portróját kapta... a költő születésének 200. évfordulójára rendezett emlékv tavaly lezárult...
<b>Summary</b>	Emléktáblát avattak az Arany János - emlékv keretében a Petőfi Irodalmi Múzeumban ( PIM ) a Kossuth - és József Attila - díjas költő tiszteletére. A köztéri alkotás a költészet napja alkalmából készült.
<b>Phenomena</b>	The topic is correct: the (200th) anniversary of Poet János Arany's birth, but instead of a portrait painting of him on a firewall, a memorial plaque is mentioned with a wrong place and date. He did not receive Kossuth- and József Attila-prizes either.

**Table 5.** Text generation on the ELTE.DH corpus - Example 3.

### 3.3 Robustness against speech recognition and punctuation errors

**Scenarios** Now we turn to the assessment of semantic bias introduced by the presence of ASR and/or punctuation errors in case of abstractive summaries. We create therefore 4 kinds of transcripts to be compared:

- **MT-MP**: Manual Transcript with Manual Punctuation: this is a simple human made gold transcript, which includes punctuation for { . , ? ! };
- **AT-MP**: ASR Transcript with Manual Punctuation: we use an ASR transcript, but restore punctuation from the gold transcripts (based on their timestamps, followed by human check);
- **MT-AP**: Manual Transcript with Automatic Punctuation: we remove punctuation from the gold transcript, and predict punctuation automatically;
- **AT-AP**: ASR Transcript with Automatic Punctuation: ASR transcripts are punctuated with the model described in Tündik et al. (2018).

**Dataset** We use 10 snippets (blocks) from a Hungarian TV broadcast database covering various genres, such as sport news, weather forecasts and news Varga et al. (2015) with its corresponding ASR solution gently provided by SpeechTex Ltd for the experiments. We have overall 500 sentences and 8k word tokens in total. We use the Kaldi version of the ASR in Varga et al. (2015) (with Kaldi decoder) by 6.8%, 10.1%, and 21.4% Word Error Rates (WER) on weather forecasts, news and sport news, respectively. For AP (automatic punctuation) we use the model from Tündik et al. (2018) and obtain F1-measures in the range of 60-70% on MT (manual transcript) and 45-50% on AT (ASR transcript).

We use human made summaries prepared by 3 independent annotators based on the MT-MP scenario transcripts.

Table 6 and Fig 1 show the results (F1-scores of different ROUGE-metrics).

The tendency is different compared to what the authors of Tündik et al. (2019) experienced in case of extractive summarization; categories with erroneous texts (mistakes are coming either from ASR or punctuation) are often

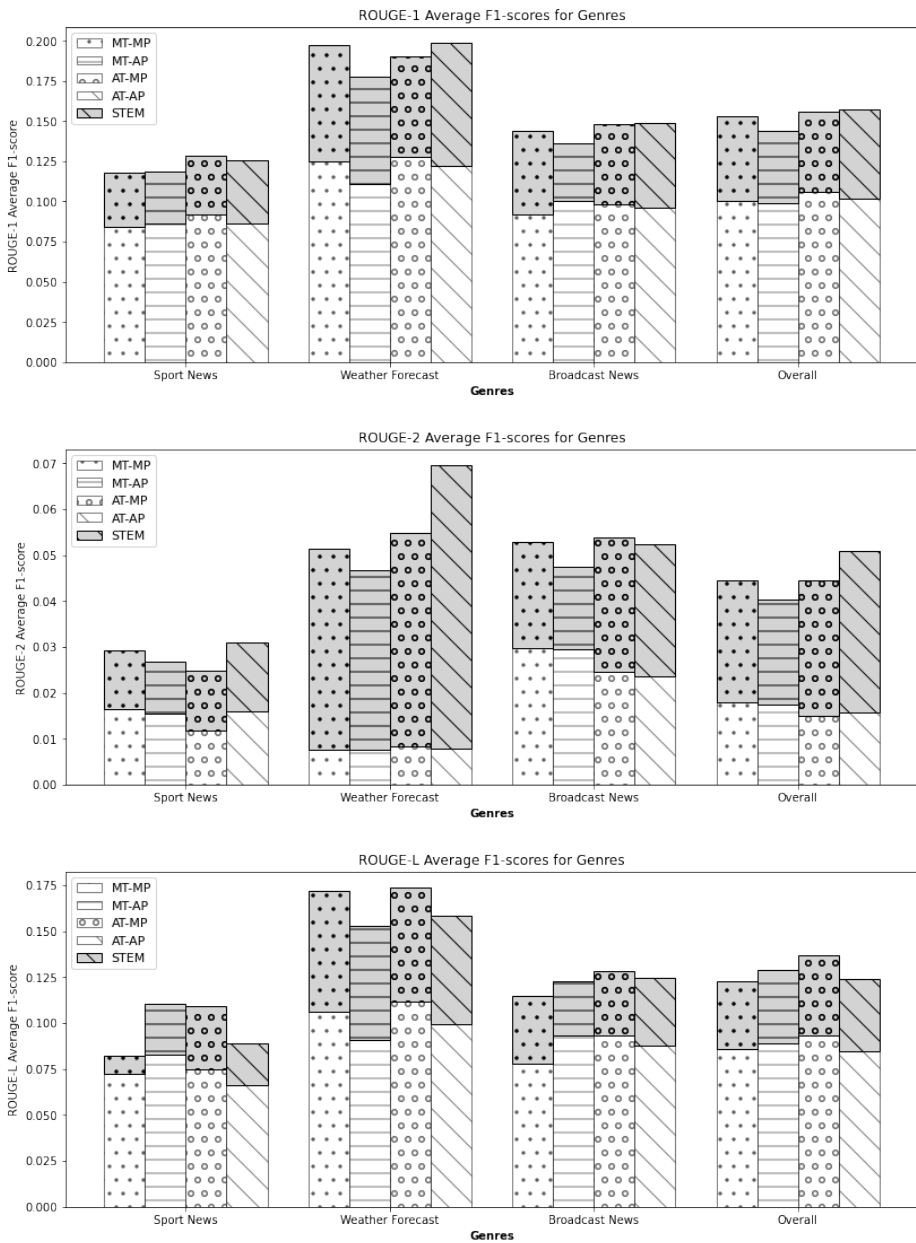


Fig. 1: Genre-based ROUGE-scores (with and without stemming)

Transcript Type	No stemming			Stemming		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
MT-MP	10.02	<b>1.79</b>	8.55	15.32	4.45	12.30
MT-AP	9.90	1.75	8.88	14.40	4.03	12.86
AT-MP	<b>10.60</b>	1.49	<b>9.33</b>	15.56	4.44	<b>13.70</b>
AT-AP	10.15	1.58	8.46	<b>15.76</b>	<b>5.1</b>	12.41

**Table 6.** ROUGE-scores obtained on TV Broadcast Dataset.

close to the performance with gold transcripts in ROUGE, or even better. Moreover, significant increase is revealed using stemming algorithm. Although it is not a typical step to involve, this difference is subjected to the morphologically rich Hungarian language, e.g. a relevant noun (focus) in a singular vs. in a plural form. Surprisingly, in case of stemming, AT-AP transcript - based abstractive summaries provide the highest ROUGE-1 and ROUGE-2 F1-scores. The genre-level investigations uniformly confirm the positive role of stemming, the increases are shown with grey.

Summaries of Weather Forecast have the highest scores For ROUGE-1 and ROUGE-L, showing that the length of the summary is the most appropriate for this category. The length of the abstractive summaries shall be fine-tuned in the future, because there are various topics in the sport news and the broadcast news typically, which cannot be compressed; usually one main topic is emphasized by the generator, or smaller information blocks were combined like e.g. *nagyerdei stadion* (the name of a football stadium) + *gazdagréti uszoda* ('swimming pool in Gazdagrét') → *nagyerdei uszoda* ('Nagyerdei Swimming Pool'). Sometimes the polarity of the utterance changes, e.g. negation is dropped. Unfortunately, since ROUGE-scores are just simple numerical approximations, they hide these hallucination errors, so in case of abstractive summaries, subjective evaluation is also needed in the future.

Tables 7 to 10 illustrate the text generation on the Broadcast Dataset with some examples (possibly containing ASR and/or punctuation errors).

<b>Sport News</b>	"...a Veszprém férfi kézilabdacsapata. Császár Gábor együttesével a Paris Saint-Germainnel találkozik a férfi kézilabda Bajnokok Ligájában..."
<b>Summary</b>	"...A magyar férfi kézilabda - válogatott a franciaországi Európa - bajnokságra készülődött, és a csoportkörbe jutásért játszik. A magyar válogatott a margitszigeti Eb - selejtezőn a franciákkal találkozik, és az Eb - n is ott lesz. A franciaországi Eb - re készülő magyar válogatott az utolsó pillanatban került be a nyolcaddöntőbe..."
<b>Phenomena</b>	The topic is hit by the algorithm (handball), but it mentions countries (Hungary and France) instead of specific clubs (Veszprém and Paris Saint-Germain). There is a mismatch in the competition (Champions League vs. European Championship), mentioning different stages (e.g. eight-final) from the article, and an unreferenced location ('margitszigeti')

**Table 7.** Text generation on the Broadcast Dataset - Example 1.



<b>Sport News</b>	"...ma esti Manchester United-Bayern München BL-negyeddöntő..." "...a Barcelona eddig százöttször játszott az Atlético ellen..."
<b>Summary</b>	"A Manchester United és az Atlético Madrid összecsapása lesz a labdarúgó Bajnokok Ligája negyeddöntőjének első felvonása. A Barcelona és a Bayern München összecsapása az elődöntő első felvonásán, a Manchester City és a Barcelona pedig az elődöntőben találkozik a fináléban. A Manchester City a címvédő Real Madridot fogadja, a Barcelona a Barcelonát fogadja. A másik ágon a Barcelona és az FC Barcelona találkozik."
<b>Phenomena</b>	Unreferenced teams came up (Real Madrid, Manchester City), Repetitions for Barcelona, New/swapped pairings (MU - Atlético, Barcelona - Bayern), Quarter-final and semi-final in the same sentence

Table 8. Text generation on the Broadcast Dataset - Example 2.

<b>Weather Forecast</b>	"...a keleti országrészben pedig helyi záporok, egy-két helyen zivatar kialakulása...." "...abban a térségben nem lesz ritka harminc fok feletti hőmérséklet, meg sem. Ezzel. Európa legmelegebb pontja lesz ma az a térség..." "...a jövő hét közepétől kezdve helyenként a hőmérséklet még a harminc fokot is megközelítheti..."
<b>Summary</b>	"A jövő héten is folytatódik a változékony időjárás, a napsütés mellett egyre többfelé várható zápor, zivatar. A hét második felében akár 30 Celsius - fokig is emelkedhet a hőmérséklet, a hétvégén viszont akár 30 fok is lehet - derül ki az Országos Meteorológiai Szolgálat előrejelzéséből."
<b>Phenomena</b>	The same temperature is mentioned but with digits and text form (30 vs. thirty), even the types of rainfall are correct, and the unreferenced Hungarian Meteorological Service (OMSZ) is nicely included in the sentence.

Table 9. Text generation on the Broadcast Dataset - Example 3.

<b>Broadcast News</b>	"...a fogyatékossgal élők munkába állása, valamint a női foglalkoztatás eredményei is nagyban hozzájárultak az országos foglalkoztatási adatok javulásához - jelentette ki a szociális és családügyért felelős államtitkár. Soltész Miklós hangsúlyozta, a megváltozott munkaképességű nyilvántartott álláskeresők száma az elmúlt kormányzati ciklusban tizenhét és fél ezerre csökkent."
<b>Summary</b>	"A kormány elkötelezett a fogyatékkal élők foglalkoztatásának javítása mellett, a kormány a következő években is folytatja a munkaerőpiaci integrációt – mondta a szociális ügyekért és társadalmi felzárkózásért felelős államtitkár. Soltáry - Siklósi Anita kiemelte : a kormány célja, hogy minél több embert vonjanak be a munkaerőpiacra, és a jövőben is legyen lehetőség a pályakezdekre."
<b>Phenomena</b>	A hard topic is recognized (integration of people with disabilities into the job market), but a wrong (non-existent!) person is mentioned.

Table 10. Text generation on the Broadcast Dataset - Example 4.

## 4 Conclusions

In this paper we were investigating encoder-decoder based abstractive summarization for Hungarian language. Beside fine-tuning the baseline huBERT based model, we were also looking into aspects of using the summarizer on spoken documents transcribed by ASR. We evaluated our model with objective scores, i.e. ROUGE. As the fine-tuning and evaluation were performed with corpora using news lead as target summary, this evaluation is to some extent biased, but with lacking human reference this is still the best evaluation scenario we could think of. In terms of spoken documents, the ASR can introduce two types of errors, word errors and punctuation errors. A bit surprisingly these errors did not significantly impact summarization performance, although the before mentioned mismatch resulting from using the leads as targets limits the applicability of these findings. In the future we would like to focus both on the improvement w.r.t. ROUGE-scores, e.g. deeper investigation of the min/max length parameter of the decoder part, and/or also changing the encoder part to fit better to long broadcast data, and on the subjective evaluation of our summarizer. Eventually a finer method designed for semi-supervised or unsupervised target extraction is also in our interest, so that we can use more accurate references than the leads, which are not always true summaries of the following document.

## Acknowledgement

Our model was fine-tuned on a server of the SZTAKI HLT group, which kindly provided access to it. We are grateful to SpeechTex Ltd. for letting us use their ASR engine on some spoken documents to simulate spoken document summarization with an ASR + Summarizer cascade. We thank Dávid Nemeskey for help in uploading the model to Hugging Face. We thank NKFIH for the financial support of the experiments and project (under contract FK-124413).

## Bibliography

- Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for boltzmann machines. *Cognitive science* 9(1), 147–169 (1985)
- Edunov, S., Baevski, A., Auli, M.: Pre-trained language model representations for language generation. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4052–4059. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), <https://aclanthology.org/N19-1409>
- Fabbri, A.R., Kryściński, W., McCann, B., Xiong, C., Socher, R., Radev, D.: Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9, 391–409 (2021)

- Fan, A., Grangier, D., Auli, M.: Controllable abstractive summarization. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation. pp. 45–54. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), <https://aclanthology.org/W18-2706>
- Gabriel, S., Celikyilmaz, A., Jha, R., Choi, Y., Gao, J.: GO FIGURE: A meta evaluation of factuality in summarization. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 478–487. Association for Computational Linguistics, Online (Aug 2021), <https://aclanthology.org/2021.findings-acl.42>
- Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., Poesio, M.: MultiLing 2015: Multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 270–274. Association for Computational Linguistics, Prague, Czech Republic (Sep 2015), <https://aclanthology.org/W15-4638>
- Grusky, M., Naaman, M., Artzi, Y.: Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 708–719. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), <https://aclanthology.org/N18-1065>
- Hasan, T., Bhattacharjee, A., Islam, M.S., Mubasshir, K., Li, Y.F., Kang, Y.B., Rahman, M.S., Shahriyar, R.: XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4693–4703. Association for Computational Linguistics, Online (Aug 2021), <https://aclanthology.org/2021.findings-acl.413>
- Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2020), <https://openreview.net/forum?id=rygGQyrFvH>
- Huang, D., Cui, L., Yang, S., Bao, G., Wang, K., Xie, J., Zhang, Y.: What have we achieved on text summarization? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 446–469. Association for Computational Linguistics, Online (Nov 2020), <https://aclanthology.org/2020.emnlp-main.33>
- Huang, Y., Feng, X., Feng, X., Qin, B.: The factual inconsistency problem in abstractive text summarization: A survey (2021)
- Indig, B., Knap, Á., Sárközi-Lindner, Z., Timári, M., Palkó, G.: The ELTE.DH pilot corpus – creating a handcrafted Gigaword web corpus with metadata. In: Proceedings of the 12th Web as Corpus Workshop. pp. 33–41. European Language Resources Association, Marseille, France (May 2020), <https://aclanthology.org/2020.wac-1.5>
- Ladhak, F., Durmus, E., Cardie, C., McKeown, K.: WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 4034–

4048. Association for Computational Linguistics, Online (Nov 2020), <https://aclanthology.org/2020.findings-emnlp.360>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020), <https://aclanthology.org/2020.acl-main.703>
- Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), <https://aclanthology.org/W04-1013>
- Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (Nov 2019), <https://aclanthology.org/D19-1387>
- Lowerre, B.: The HARPY Speech Recognition System. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PN, USA (1976)
- Maynez, J., Narayan, S., Bohnet, B., McDonald, R.: On faithfulness and factuality in abstractive summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1906–1919. Association for Computational Linguistics, Online (Jul 2020), <https://aclanthology.org/2020.acl-main.173>
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140), 1–67 (2020), <http://jmlr.org/papers/v21/20-074.html>
- Rothe, S., Narayan, S., Severyn, A.: Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics* 8, 264–280 (2020), <https://aclanthology.org/2020.tacl-1.18>
- Schreibman, S., Siemens, R., Unsworth, J.: *A companion to digital humanities*. John Wiley & Sons (2008)
- Scialom, T., Dray, P.A., Lamprier, S., Piwowarski, B., Staiano, J.: MLSUM: The multilingual summarization corpus. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8051–8067. Association for Computational Linguistics, Online (Nov 2020), <https://aclanthology.org/2020.emnlp-main.647>
- See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), <https://aclanthology.org/P17-1099>

- Straka, M., Mediankin, N., Kocmi, T., Žabokrtský, Z., Hudeček, V., Hajič, J.: SumeCzech: Large Czech news-based summarization dataset. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), <https://aclanthology.org/L18-1551>
- Tündik, M.A., Kaszás, V., Szaszák, G.: Assessing the semantic space bias caused by asr error propagation and its effect on spoken document summarization. In: Proc. Interspeech (2019)
- Tündik, M.A., Szaszák, G.: Leveraging a character, word and prosody triplet for an asr error robust and agglutination friendly punctuation approach. In: Proc. Interspeech (2019)
- Tündik, M.A., Szaszák, G., Gosztolya, G., Beke, A.: User-centric Evaluation of Automatic Punctuation in ASR Closed Captioning. In: Proc. Interspeech 2018. pp. 2628–2632 (2018)
- Váradi, T., Koeva, S., Yamalov, M., Tadić, M., Sass, B., Nitoń, B., Ogrodniczuk, M., Pezik, P., Barbu Mititelu, V., Ion, R., Irimia, E., Mitrofan, M., Păiș, V., Tufiș, D., Garabík, R., Krek, S., Repar, A., Rihtar, M., Brank, J.: The MARCELL legislative corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3761–3768. European Language Resources Association, Marseille, France (May 2020), <https://aclanthology.org/2020.lrec-1.464>
- Varga, Á., Tarján, B., Tobler, Z., Szaszák, G., Fegyó, T., Bordás, C., Mihajlik, P.: Automatic Close Captioning for Live Hungarian Television Broadcast Speech: A Fast and Resource-Efficient Approach. In: Proceedings of SPECOM. pp. 105–112. Springer (2015)
- Vijayakumar, A.K., Cogswell, M., Selvaraju, R.R., Sun, Q., Lee, S., Crandall, D.J., Batra, D.: Diverse beam search: Decoding diverse solutions from neural sequence models. CoRR abs/1610.02424 (2016), <http://arxiv.org/abs/1610.02424>
- Yang Zijian, G., Agócs, Á., Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (5 2021), selected papers of the 2020 Conference on Information Technology
- Zarrieß, S., Schlangen, D.: Decoding strategies for neural referring expression generation. In: Proceedings of the 11th International Conference on Natural Language Generation. pp. 503–512. Association for Computational Linguistics, Tilburg University, The Netherlands (Nov 2018), <https://aclanthology.org/W18-6563>
- Zhang, J., Zhao, Y., Saleh, M., Liu, P.J.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. CoRR abs/1912.08777 (2019), <http://arxiv.org/abs/1912.08777>



## Szövegaugmentálási módszerek összehasonlítása politikai szövegek szentimentanalízise során

Üveges István<sup>1,2</sup>, Csányi Gergely Márk<sup>1,3</sup>, Ring Orsolya<sup>2</sup>, Orosz Tamás<sup>1</sup>

<sup>1</sup>MONTANA Tudásmenedzsment Kft.

<sup>2</sup>Társadalomtudományi Kutatóközpont, Politikatudományi Intézet

<sup>3</sup>Budapest Műszaki és Gazdaságtudományi Egyetem

{uvegesi,csanyi.gergely,orosz.tamas}@montana.hu

ring.orsolya@tk.hu

**Kivonat** Cikkünkben bemutatjuk a gépi tanítási feladatokban gyakran előforduló kiegyensúlyozatlan tanítóhalmaz probléma egy lehetséges megoldását az alacsony elemszámú kategóriák szöveg-augmentálásával. Az összevethetőség érdekében egyszerű szövegaugmentálási technikákkal (EDA) és egy szövektor alapú módszerrel is kísérletet teszünk. A módszerek hatékonyságát politikai doménbe tartozó szövegek szentimentelemzési feladatán teszteljük, amihez a TK-MILAB szentiment korpusz egy kisebb szeletét használjuk. Az alulreprezentált kategória bővítésével elért eredményeket a kiváltott F-érték változás függvényében értékeljük.

**Kulcsszavak:** Easy Data Augmentation, kiegyensúlyozatlan tanítóhalmaz, emóció elemzés

### 1. Bevezetés

A gépi tanuláson alapuló feladatok esetében, amelyek például szövegosztályozást kísérnelnek megvalósítani, gyakran előforduló probléma az adatok kiegyensúlyozatlansága a tanítóhalmazban (Kubat és mtsai, 1997; Menardi és Torelli, 2014). Ilyen esetben a különböző címkék aránya erősen aszimmetrikus, egyes osztályok jelentősen alulreprezentáltak, ami megnehezíti az ilyen osztályok jó hatásfokú predikcióját.

Ez a probléma felmerül magyar nyelvű politikai szövegek felügyelt gépi tanuláson alapuló klasszifikálása során is. Az Comparative Agendas Project<sup>1</sup> keretében zajló klasszifikálás során az egyes megfigyeléseket szokásosan 21 közpolitikai osztályba sorolják. A kézzel képzett tanítóhalmaz minden esetben kiegyensúlyozatlan, aminek a gépi osztályozás során (Sebők és Kacsuk, 2021) egy lehetséges megoldása például az ún. bináris hólabda megközelítés (*binary snowball approach*) alkalmazása, melynek során bináris választások sorozatává egyszerűsítjük a többsztályos osztályozást.

Az ilyen kiegyensúlyozatlan adatbázis-struktúrák általában is jellemzik az összehasonlító politikatudományi elemzésekre használt korpuszokat. Ezekben az

<sup>1</sup> <https://cap.tk.hu/>

esetekben a legnagyobb problémát az osztályozás során a nagy pontosság mellett a magas recall arány elérése jelenti (Kumar és Gopal, 2010) különösen szupport vektor gép (Support Vector Machine, SVM) használatakor. A probléma kezelésének egyik módja, ha teljesen figyelmen kívül hagyjuk az alulreprezentált kategóriát, ezzel azonban a kutatás szempontjából értékes megfigyeléseket veszíthetünk. Hogy ezt elkerüljük, az egyik lehetőség a véletlenszerű mintavételezés (*random sampling*) túl- vagy az alulmintavételezéssel, ezzel kompenzálva a korpusz belső egyensúlytalanságát. A véletlenszerű mintavételezés hátránya azonban, hogy megnöveli a túlillesztés esélyét (Lango és Stefanowski, 2018; Nguyen és mtsai, 2011). A nemzetközi politikatudományban emellett az elmúlt években jelentősen megnőtt a nagyméretű (például parlamenti szövegekből, jogszabályokból vagy politikai hírekből álló) korpuszok elemzésén alapuló kutatások száma is. Mindez magával hozta az igényt a különböző gépi klasszifikálási feladatok hatékonyságának növelésére, ami szükségessé teszi a kiegyensúlyozatlan tanulóhalmazok problémájának megoldását (Hillard és mtsai, 2008; Breeman és mtsai, 2009; Burscher és mtsai, 2015).

Napjainkban ugyancsak kiemelkedő jelentőségű kutatási feladat a különböző szövegek gépi tanuláson alapuló szentiment- illetve emóció-klasszifikálása (Van Atteveltdt és mtsai, 2008; Bhowmick és mtsai, 2009; Jia és mtsai, 2009; Young és Soroka, 2012; Dadgar és mtsai, 2016). Mivel a politikai doménbe tartozó szövegek emóciótartalma ugyancsak különösen kiegyenlítetlen, ezen a területen kiemelkedően fontos a tanítóhalmaz kiegyensúlyozása.

A már említetteken kívül az adatok kiegyensúlyozatlanságára megoldást jelenthet a rendelkezésre álló adatok augmentálása is, ami a meglévő példák alapján új példányok készítését jelenti a tanítóhalmazba. Ez a technika a gépi látás területén már régóta bevett eljárásnak számít (Zhang és mtsai, 2015; Fawzi és mtsai, 2016; Taylor és Nitschke, 2018), később néhány alapvető ötletet is innen merített az NLP területe (Fadaee és mtsai, 2017; Wei és Zou, 2019; Csányi és Orosz, 2021).

Tanulmányunk célja, hogy különböző szöveg augmentálási technikákat próbáljunk ki politikai doménbe tartozó mondat szintű szövegek szentiment osztályozhatóságának javítása érdekében. Tekintettel a kutatás pilot jellegére, ezt két kiválasztott szentiment osztály példáján keresztül mutatjuk be. Véleményünk szerint ugyanakkor a bemutatott eredmények hasznos tanulságokkal szolgálhatnak bármely kiegyensúlyozatlan tanítóhalmazzal rendelkező gépi tanítási feladat esetében. A különböző augmentálási technikák hatékonyságát háromféleképpen előfeldolgozott szövegen hasonlítottuk össze. Az itt bemutatott módszereket a *digital-twin-distiller*<sup>2</sup> keretrendszerbe integráltuk. A cikkben bemutatott eszközök és a szentimentanalízis során betanított modellekből készített applikáció is megtalálható a GitHubon<sup>3</sup>.

<sup>2</sup> *digital-twin-distiller* projekt elérhetősége a GitHubon: <https://github.com/montana-knowledge-management/digital-twin-distiller>

<sup>3</sup> Politikai témájú szövegek mondat szintű szentiment analízise projekt keretében készült szentiment felismerő eszközök: <https://github.com/montana-knowledge-management/hungarian-political-sentiment-analysis>



## 2. TK-MILAB szentiment korpusz politikai doménre

### 2.1. Annotálási elvek

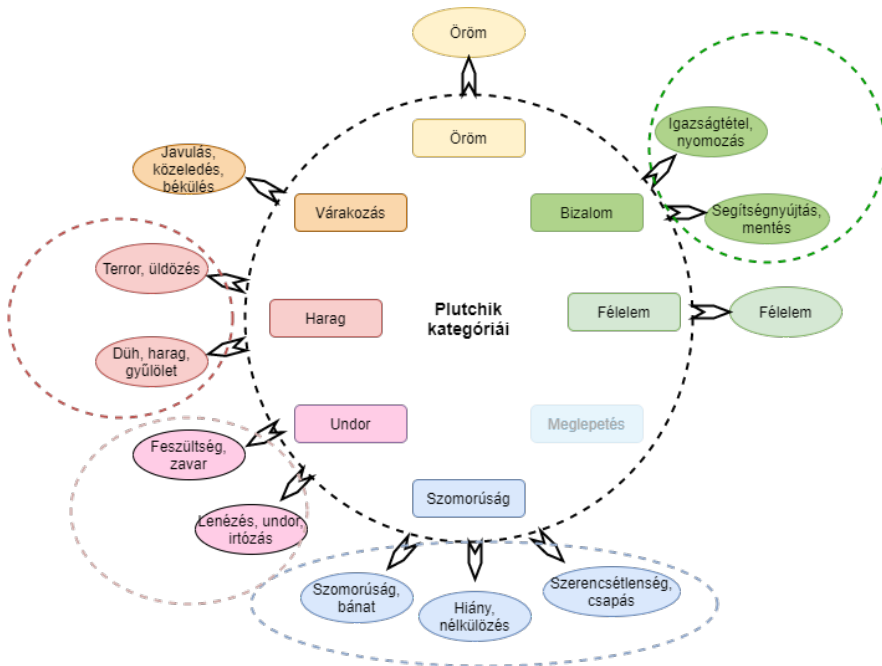
Vizsgálatunk kezdetekor a TK-MILAB „Doménspecifikus szentimentelemzési eljárás kidolgozása magyar nyelvű szövegek elemzésére” részprojektje keretében készülő<sup>4</sup> korpuszban mintegy 5700 mondatnyi kézzel annotált szöveg volt elérhető. A korpusz 12 emóció-kategória szerint osztályozott mondatokat tartalmaz, és benne minden mondat pontosan egy emóció kategóriára jellemző címkével van ellátva. A kategorizálás során először induktív módon a szövegből kiindulva kerültek meghatározásra az egyes emóciókategoróriák, melyek szükség esetén aggregálhatók Plutchik (Kellerman és Plutchik, 1968) emóció-kategória rendszerére, amely nyolc osztályt különböztet meg (lásd 1. ábra).

Erre a kibővített rendszerre azért volt szükség, mert a politikai hírszövegekben található mondatok egyébként nem vagy csak rendkívül alacsony annotátori egyetértés mellett voltak besorolhatóak a Plutchik-féle („hagyományos”) kategóriákba. Egy jó példa lehet erre a következő mondat: „Egy ember összeesett az utcán, de a járókelők megmentették az életét.”. Itt az első tagmondatnak a „Szomorúság”, míg a másodiknak a „Bizalom” az alapérzelme. Az egész mondatot azonban nem lehet egyértelműen besorolni az egyik vagy a másik kategóriába az eredeti, Plutchik-féle érzelmerék alapján. A bővített, TK-MILAB projekt keretében kidolgozott rendszer segítségével az előbbi mondat egyértelműen besorolható a „Segítségnyújtás, mentés” kategóriába, mely azután Plutchik-féle rendszer „Bizalom” kategóriájára aggregálható. A bővített rendszer segítségével a korpusz magas kódolók közötti egyetértéssel volt annotálható (Ring és mtsai, 2021).

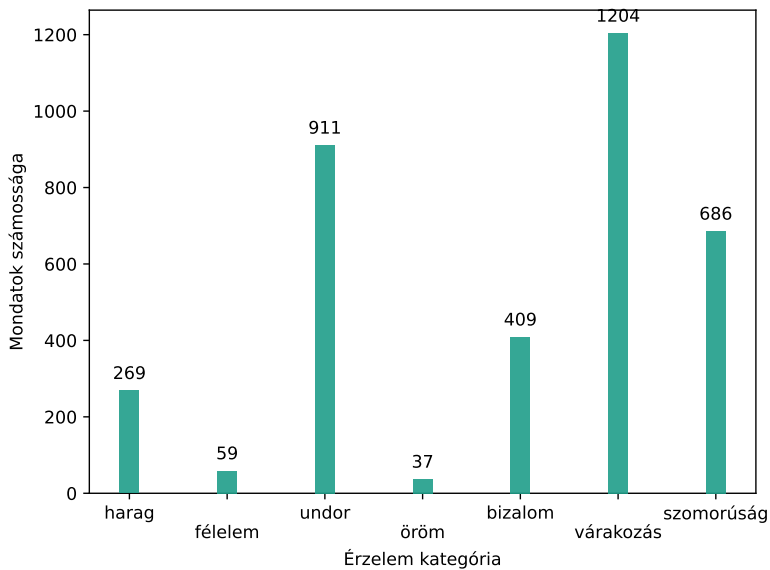
### 2.2. Annotálási eljárás

A mondatok címkézését három annotátor végezte. Elsőként két független annotátor egy-egy címkét rendelhetett minden mondathoz, majd ezt követően egy szakértő kiválasztotta, illetve validálta a korpuszba kerülő címkét. A kész korpuszban tehát minden egyes mondat pontosan egy emóció kategóriához tartozhatott. Minden érzelmi kategória leírható egy fő -, illetve egy, a TK-MILAB korpuszban feltüntetett mellékkategóriával. Az iménti példamondat eszerint tehát a „Bizalom”, azon belül pedig a „Segítségnyújtás/mentés” kategóriába sorolható. A korpuszban szereplő mondatok közül csak azokat használtuk fel az augmentáláshoz, amelyek esetében a két annotátor egyetértésben adta ugyanazt a címkét. Ennek célja az volt, hogy a gépi tanításhoz olyan adatokat használjunk fel, amelyek emberi megítélés szerint a lehető leginkább mentesek a bizonytalanságtól. Az így előállt kategóriacsoportok mérete nagyon eltérően alakult (lásd 2. ábra). A „Várakozás” kategóriába került a legtöbb mondat, szám szerint 1204, míg az „Öröm” -be a legkevesebb, összesen 37 darab.

<sup>4</sup> TK-MILAB projekt elérhetősége: <https://milab.tk.hu/domenspecifikus-szentimentelemzesi-eljaras-kidolgozasa-magyar-nyelvu-szovegek-elemzesere>



1. ábra. Plutchik kategóriái és a belőlük képzett emóció kategóriák a TK-MILAB korpuszban.



2. ábra. Plutchik kategóriáira aggregált címkéjű mondatok számossága a TK-MILAB korpuszban.

### 2.3. Összehasonlításához használt tanítóadat-halmazok

A szövegaugmentálási technikák segítséget nyújthatnak ahhoz, hogy a kis elemszámmal rendelkező kategóriákat a meglévő mondatokból előállított szintetikus mondatokkal felbővítsük. Ahhoz, hogy a különböző technikákat tesztelni tudjuk, a különböző módszerek összehasonlításához a rendelkezésre álló kategóriák közül azt a kettőt választottuk ki, amelyek a legnagyobb számossággal bírtak a leválogatott korpuszban („Várákozás” és „Undor”). A vizsgálatokhoz bináris osztályozókat készítettünk különböző méretű tanítóadat-halmazokon. Ezek a bináris osztályozók „Várákozás” vagy „Undor” értékeket vehették fel, hiszen csak az ezeknek megfelelő mondatokat tartalmazta a tanításhoz használt korpusz. A célunk ezzel az volt, hogy megvizsgáljuk, hogy az eredeti adatok felhasználásával milyen hatékonyságú osztályozás érhető el. Az így kapott értékeket a későbbiekben kevesebb eredeti adaton, adott mennyiségű augmentált adat hozzáadása mellett készült modellek értékeivel vetettük össze.

Ehhez ugyanazt a modellt különböző méretű tanítóadat-halmazokon tanítottuk föl. Elsőként a két kiválasztott kategória 900 - 900 mondatából kialakított születén hajtottunk végre gépi tanítást, a `scikit-learn`<sup>5</sup> lineáris kernelű szupport vektor gép modelljével. A tanításokat - a kiegyensúlyozatlan minták hatását csökkentve - a `class_weight="balanced"` beállítással végeztük, a többi paramétert alapértelmezettként hagytuk. A tanítást minden esetben tf-idf vektorokon (Luhn, 1957; Jones, 1972) végeztük el, uni- és bigramokat is figyelembe véve a vektorokban, a többi paramétert szintén alapértelmezetten hagytuk.

Az „Undor” kategória mondatait használtuk később augmentálásra. A választott részhalmazok számosságai a következők szerint alakultak: 10, 25, 50, 100, 250, 500 (ezekre az értékekre a későbbiekben az augmentálás *bázisa*-ként hivatkozunk). A „Várákozás” kategória minden tanítás esetén 720 mondatnyi tanító adatot tartalmazott. A referenciaként betanított modellek ennek megfelelően nem kiegyensúlyozott tanító adatot kaptak bemenetként (pl. 10 db „Undor”, 720 db „Várákozás” mondatot). Az augmentáláshoz használt modellek vizsgálata során az „Undor”-ban lévő mondatok számosságát mindig 720-ra augmentáltuk. Az augmentálás arányát ( $\frac{n_{aug}}{n}$ , ahol  $n$  az augmentálás bázisa,  $n_{aug}$  pedig az augmentált adat számossága), valamint a kapott korpuszokba bekerülő augmentált mondatok számosságait az 1. táblázat szemlélteti. Az ilyen módon kiválasztott (összesen minden esetben 720 mondatnyi) „Undor” címkével annotált mondat mellett konstans 720 mondatnyi „Várákozás” címkéjű mondatot választottunk ellenpéldaként. Így kiegyenlített tanítóhalmazon történt a modellek tanítása. A kiértékelést minden modell esetében emóciókategóriánként 180 (összesen tehát 360) mondaton végeztük el. Az azonos méretű tanítóhalmazon augmentált adatok  $F_1$  értékeinek javulását hasonlítottuk az azonos méretű, kiindulási referencia korpuszhoz viszonyítva.

<sup>5</sup> <https://scikit-learn.org/stable/>

$n$	$n_{aug}$	$\frac{n_{aug}}{n}$	$\frac{n_{aug}}{N}$
[db]	[db]	[-]	[%]
10	710	71	98,6
25	695	27,8	96,53
50	670	13,4	93,06
100	620	6,2	86,11
250	470	1,88	65,28
500	220	0,44	30,56

1. táblázat

### 3. Szöveg augmentálás

#### 3.1. Alkalmazott módszerek

A szövegek augmentálásához a `digital-twin-distiller` keretrendszeren belül elérhető augmentáló algoritmusokat (Csányi és Orosz, 2021) alkalmaztuk. Ezeknek két nagyobb csoportja különíthető el, melyek közül az elsőbe az *egyszerű augmentálási módok* tartoztak (*Easy Data Augmentation - EDA*, Wei és Zou (2019)), a másikba pedig a szóbeágyazáson alapuló módszer (Csányi és mtsai (2021)).

Az összehasonlítás során az EDA-ba tartozó technikák közül a következő négy augmentálási megoldást alkalmaztuk:

- Szinonima helyettesítés (Synonym Replacement - SR): adott számú szót egy random választott szinonimájával helyettesít.
- Random beszúrás (Random Insertion - RI): kiválaszt szavakat a mondatból, és azok szinonimáit random pozíciókra helyezi el a mondaton belül.
- Random csere (Random Swap - RS): adott mennyiségű szópár pozíciójának cseréje a mondaton belül.
- Random törlés (Random Deletion - RD): szavak törlése az augmentált szövegből adott valószínűséggel.

A szóbeágyazási módszer esetében 100, illetve 300 dimenziós `fastText`<sup>6</sup> (Bojanowski és mtsai (2017)) modelleket alkalmaztunk. A modellek alapját az augmentálásra kiválasztott mondatok adták, augmentálás során pedig a modellben leginkább hasonló 10 szó közül véletlenszerűen választottunk egyet, amellyel az eredeti tokent az algoritmus helyettesítette. Azt, hogy az augmentálás milyen arányban történjen, az  $\alpha$  paraméter változtatásával lehetett beállítani.

#### 3.2. Preprocesszálás

A szövegek normalizálása során közös lépés volt a mondatok stopszó szűrése, a számok és az írásjelek eltávolítása, valamint a kisbetűsítés. A különböző ragozott és képzett alakok normalizálását emellett három különböző eszközzel is elvégeztük:

<sup>6</sup> <https://fasttext.cc/docs/en/crawl-vectors.html>

- a spaCy 2.0 verziójához (Honnibal és Montani, 2017) elérhető, Orosz György által készített nyelvmódellet<sup>7</sup> használva készítettünk a mondatokból egy lemmatizált változatot. A továbbiakban az ilyen módon előkészített szövegekre mint lemmatizált korpuszvariánsra hivatkozunk,
- kétféle stemmer segítségével szótövezést végeztünk; az egyik a Hunspell programcsomag beépített szótövezője volt (a továbbiakban: hunspell<sup>8</sup>), a másik eszköz pedig a Hunspell keretrendszerhez bővítményként vagy önálló python csomagként is elérhető hungarian-stemmer<sup>9</sup> csomag volt, amely egy kevésbé agresszív szótövező.

A preprocesszált szövegeken betanított modellekkel kapott eredményeket a 2. táblázat ismerteti. Az első két sor jelenti az adott kategóriából a modell tanításához felhasznált tanítóadatok számosságát. A táblázat 100 futtatás átlagát tartalmazza, zárójelben a kapott szórások láthatók. A kapott értékeket jól jellemzi, hogy közeledve a két korpusz kiegyensúlyozott arányához az  $F_1$ -értékek minden esetben jelentősen javulnak. Azon három esetben, amikor mindkét kategória teljes szöveganyaga felhasználásra került, a legjobb értéket a Hunspelllel történt normalizálás mellett értük el.

Kategória	Undor	10	25	50	100	250	500	720
	Várakozás	720	720	720	720	720	720	720
spaCy	P	6 (23,75)	22 (41,42)	55,83 (48,44)	84,4 (12,43)	83,18 (3,15)	75,55 (1,51)	70,97 (0)
	R	0,03 (0,13)	0,13 (0,25)	0,48 (0,52)	4,28 (1,47)	35,63 (2,71)	63,12 (1,82)	73,33 (0)
	$F_1$	0,07 (0,26)	0,25 (0,49)	0,95 (1,02)	8,11 (2,68)	49,83 (2,9)	68,76 (1,35)	73,13 (0)
Hunspell	P	6,5 (24,14)	26,5 (43,28)	67,63 (43,49)	82,45 (12,4)	82,65 (3,11)	76,93 (1,42)	73,51 (0)
	R	0,04 (0,14)	0,17 (0,3)	0,69 (0,63)	5,04 (1,88)	35,05 (2,69)	62,97 (1,98)	75,56 (0)
	$F_1$	0,08 (0,28)	0,34 (0,6)	1,36 (1,23)	9,44 (3,36)	49,17 (2,94)	69,24 (1,42)	74,52 (0)
hungarian-stemmer	P	8 (27,13)	28,75 (45,05)	76,77 (38,29)	88,13 (8,34)	82,2 (3,23)	74,17 (1,39)	69,15 (0)
	R	0,04 (0,15)	0,21 (0,37)	0,96 (0,78)	6,44 (1,9)	37,99 (2,7)	64,86 (1,66)	72,22 (0)
	$F_1$	0,09 (0,3)	0,42 (0,73)	1,89 (1,51)	11,95 (3,33)	51,92 (2,82)	69,19 (1,31)	70,65 (0)

2. táblázat. Különböző tanítóadat számosságok mellett mért pontosság (P), fedés (R) és  $F_1$ -érték ( $F_1$ ) százalékosan kifejezve (zárójelben: a 100 futtatás után mért szórás).

Hasonló a tendencia a köztes számosságok esetén is; a mért  $F_1$  értékek minden tanítóadat összeállítás esetén az azonos számosságú esetekben hasonlóan alakulnak. Az ezen modellekkel kapott eredményeket használtuk a továbbiakban referenciaként a szintetikus mondatokkal felbővített halmazon betanított modellekkel való összehasonlításához.

### 3.3. Az augmentálás folyamata

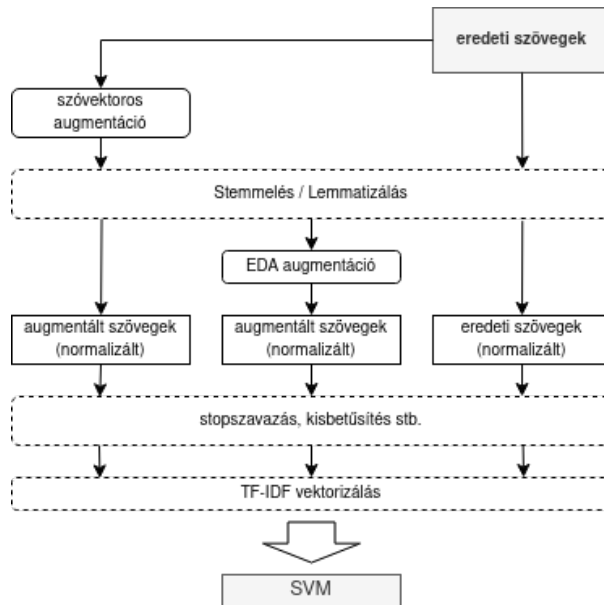
Az egyes augmentálási módszereket az optimális eredmény elérése érdekében különbözőképpen és az előfeldolgozási lánc eltérő pontjain alkalmaztuk. Az EDA

<sup>7</sup> <https://github.com/spacy-hu/spacy-hungarian-models>

<sup>8</sup> Hunspell: <http://hunspell.github.io/>

<sup>9</sup> <https://github.com/montana-knowledge-management/hungarian-stemmer>

módszereket (SR, RI, RS, RD) a lemmatizált vagy szótővezett szövegen futtatuk, azonban még a stopszó-szűrés és a kisbetűsítés előtt. A szövektor alapú módszereket ezzel szemben a lemmatizálást / szótővezést megelőzően alkalmaztuk a tanítóadatban szereplő szövegeken. Ennek az volt az oka, hogy a **fastText** modell segítségével kapott leghasonlóbb szavak többször tartalmaztak valamilyen írásjelet vagy nagybetűs szót, és el akartuk kerülni, hogy a preprocessálás során többször is szükség legyen az írásjelek és számok szűrésére. A folyamatot részletesen a 3. ábra szemlélteti.



3. ábra. Az augmentálási módszerek helye a teljes előfeldolgozási láncban.

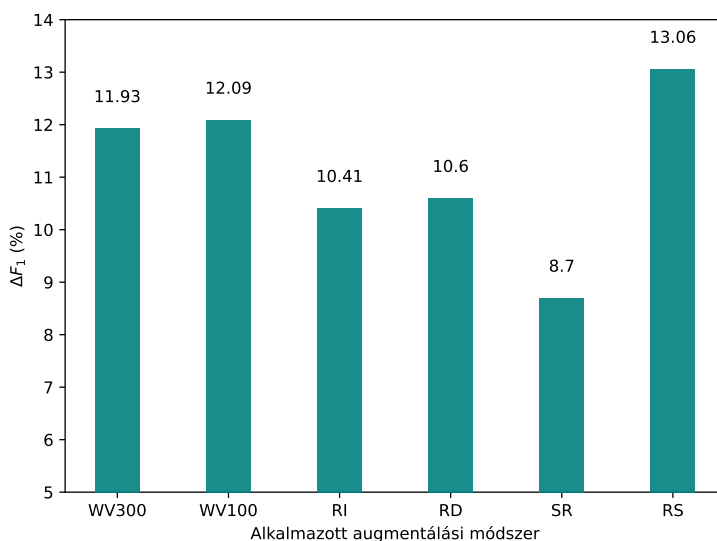
Valamennyi módszer esetében lehetőség van az augmentálást végző algoritmusok további finomhangolására. Az egyik legfontosabb ilyen technika ún. *védett szavak* megadása (ezek esetében a listán szereplő szavakat az augmentálási módszerek nem módosítják). Ez különösen hasznos lehet például szaknyelvek esetében, ahol két köznyelvilag szinonim kifejezés eltérő jelentéssel bír (pl.: „garázdaság” vs. „rongálás” jogi szövegekben). A védett szavak használata további módosításokat nem jelent a kódolás során, alkalmazásuk egy lista megadásával lehetséges, amelynek az összeállításához viszont doménspecifikus szaktudásra van szükség. A vizsgálat során a jelen korpusz egyes kategóriáihoz nem állt rendelkezésre ilyen lista, így ezt a funkciót nem használtuk ki. Az augmentáció vizsgálata során az úgynevezett  $\alpha$ -paraméter megválasztásának hatását is vizsgáltuk, amely a szövegben megváltoztatott szavak arányát szabályozza. Vizsgálatunk során ezt 0,1 és 0,5 (tehát 10% és 50%) között változtattuk minden augmentálási mód

esetében, 0,1-es lépésközzel. Az eddig leírtak összes kombinációjaként összesen 540 különböző SVM modellt tanítottunk be, majd az ezekből visszakapott metrikákat értékeltük ki.

#### 4. Eredmények és következtetések

A tanítás során a tesztadatok minden esetben az augmentálásra fel nem használt mondatok közül kerültek ki és a 900 + 900 mondatos korpusz 20 - 20% -át adták.

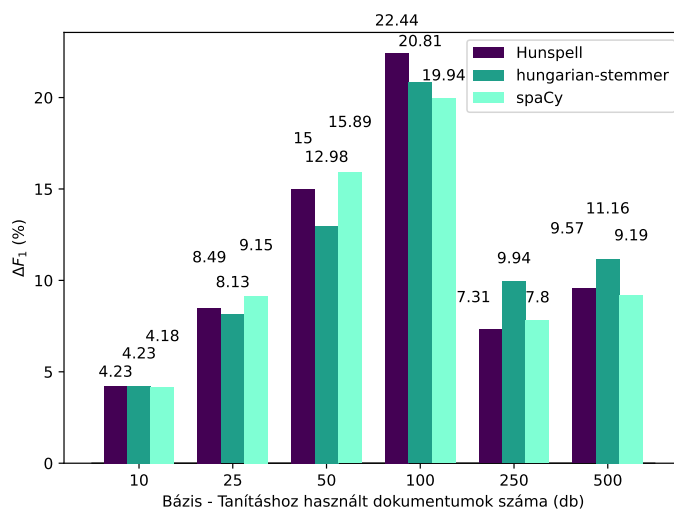
Az összképet tekintve, az augmentálási módszerek szerint összegeztük majd átlagoltuk az eredményeket. Ez minden esetben 90 db különböző modell eredményeinek az átlagolását jelenti. A 4. ábrán az egyes módszerekkel elért átlagos  $F_1$  érték változást tüntettük fel az összes bázis átlagára nézve, az „Undor” kategória felismerése során. Az ábráról leolvasható, hogy az RS algoritmus teljesített a legjobban a vizsgálat során; mintegy 13 %-kal, szignifikánsan növelte az  $F_1$  mérték értékét a referencialamához képest. Ezt nagyjából 1%-kal lemaradva a szóvektor alapú módszerek követették (100, illetve 300 dimenziós `fastText` szóbeágyazás használata mellett).



4. ábra. Az augmentálási módszerek összesített hatékonysága a különböző augmentációs módszerek (WV300 és WV100 - 300 illetve 100 dimenziós `fastText` modellel történt augmentálás, RI - Random Insertion, RD - Random deletion, SR - Synonym replacement, RS - Random Swap) esetén.

Más nézőpontból tekintve az eredményeket, ha bázis és szóalak normalizálásra alkalmazott módszerek szerint végezzük az adatok felbontását, akkor

az 5. ábra szerinti  $F_1$  változások adódnak. A felosztás alapján az látszik, hogy a lemmatizált adatok hatékonysága két kisebb bázis esetében haladta meg a többi módszerrel szótövezett változatokét, míg a két legnagyobb bázis esetében a **hungarian-stemmer** használatával volt elérhető a legnagyobb javulás az értékekben. A **Hunspell** szótövezővel pedig a 100-as bázis esetén lehetett átlagosan a legjobb eredményekhez jutni.



5. ábra. Az augmentálási módszerek hatékonysága a tanításhoz használt dokumentumok darabszáma és szóalak normalizáláshoz használt eszköz szerint.

	spaCy	hunspell	hungarian-stemmer
<b>10</b>	0.04	0.04	0.04
<b>25</b>	0.09	0.08	0.09
<b>50</b>	0.15	0.13	0.16
<b>100</b>	0.24	0.23	0.23
<b>250</b>	0.15	0.20	0.16
<b>500</b>	0.31	0.36	0.30

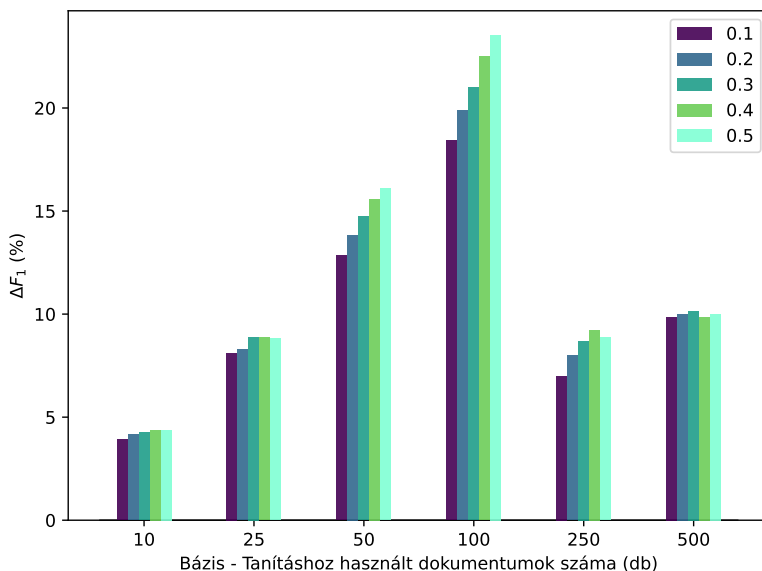
3. táblázat. Átlagos hibacsökkenés aránya az eredeti és az augmentált adattal feljavított halmazok esetében bázisonként.



A 3. táblázat ezzel szemben az átlagos relatív hibacsökkenést mutatja be, azaz azt a mérőszámot, hogy az eredetihez képest az augmentált adattal feljavított halmazok  $F_1$  értékei mennyivel kerültek közelebb a 100%-os értékhez.

Az adatokból is jól kivehető, hogy az augmentálás hozzáadott értéke egy meredek kezdeti emelkedés után egy alacsonyabb szinten stabilizálódik, ahogyan a korpuszban maradó eredeti adatok számossága és ezáltal feltehetőleg változatosága is megfelelően nagy lesz.

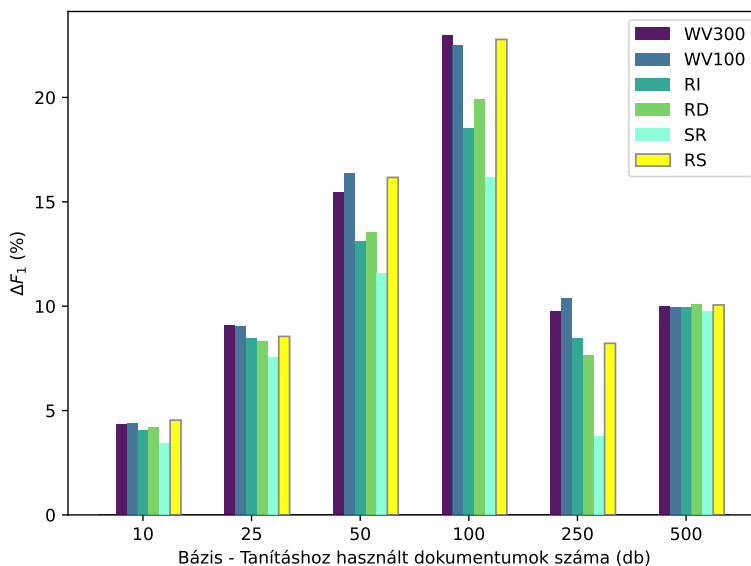
Az  $\alpha$  paraméter változtatásának hatását a háromfajta szóalak normalizálási eljárás esetén a 6. ábra mutatja be. Az ábrázolt értékek a Hunspell normalizálás során kapott konkrét eredményeket mutatják, ugyanakkor a grafikon alakulása tipikusnak volt mondható a másik két normalizálási eljárás esetében is. Az eredményekből látszik, hogy az  $\alpha$  paraméter értékének növekedésével párhuzamosan az augmentált adattal nagyobb növekmény érhető el a legtöbb bázis esetében. Ez a hatás 100-as bázis esetében érvényesül a legmarkánsabban, míg az augmentált adatok arányának csökkenésével a tanítóadatban (a bázis növekedésével párhuzamosan) fokozatosan kiegyenlítődni látszik.



6. ábra. Az  $\alpha$  paraméter változtatásának hatása az egyes bázisok alapján, Hunspell normalizálás mellett.

A 7. ábrán az egyes augmentálási módok átlagos hatását tüntettük fel bázisok szerint bontva. Az ábrán jól kivehető, hogy közepes méretű bázisok esetében (50–250) a szóvektoros augmentálási módok és a RS magasan a többi módszer felett teljesített, míg a bázisok szélső értékeinek esetében (10 és 500) az eredmé-

nyek sokkal inkább kiegyenlítettten alakultak. Minden esetben kivethető, hogy a legalacsonyabb eredményt a SR megoldás érte el.



7. ábra. Augmentálási módszerek átlagos hatékonysága bázisok szerint.

## 5. Összefoglalás

Cikkünkben különböző szöveg augmentációs technikák hatását vizsgáltuk a politikai doménre készülő TK-MILAB szentiment korpuszon. Az augmentáláshoz EDA és szóbeágyazás alapú módszereket alkalmaztunk különböző nagyságú tanítóadaton betanított SVM modellekkel. Az eredmények azt mutatják, hogy az augmentálási módszerek az 50-100-as bázison betanított modellek esetén növelték a legjobban az  $F_1$  értéket. Az összehasonlítás során az EDA csoportba tartozó random csere (RS) produkálta a legjobb eredményt, amelyet szorosan a szövektor alapú augmentálási módszerek követték. Ezeket a módszereket a szótövezett és a lemmatizált szövegen is elvégeztük. Az eredmények összehasonlításából az látszik, hogy a `spaCy`-vel lemmatizált adathalmaz, valamint a `hungarian-stemmer` és a standard `Hunspell`-es szótövezéssel preprocessált szövegen végzett augmentálás váltakozó eredményt mutatott, egyértelmű trend nem volt kimutatható. A `digital-twin-distiller`-ben készített szemantikai elemző modell szabadon letölthető és kipróbálható a projekt GitHub tárolójából<sup>10</sup>.

<sup>10</sup> A projektfájlok és a projekthez tartozó applikáció elérhető a <https://github.com/montana-knowledge-management/hungarian-political-sentiment-analysis> címen.

## Köszönetnyilvánítás

A kutatást az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- Bhowmick, P.K., Basu, A., Mitra, P.: Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Comput. Inf. Sci.* 2(4), 64–74 (2009)
- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146 (2017)
- Breeman, G., Then, H., Kleinnijenhuis, J., van Attevelde, W., Timmermans, A.: Strategies for improving semi-automated topic classification of media and parliamentary documents (2009)
- Burscher, B., Vliegthart, R., De Vreese, C.H.: Using supervised machine learning to code policy issues: Can classifiers generalize across contexts? *The ANNALS of the American Academy of Political and Social Science* 659(1), 122–131 (2015)
- Csányi, G.M., Orosz, T.: Comparison of data augmentation methods for legal document classification. *Acta Technica Jaurinensis* (2021)
- Csányi, G.M., Nagy, D., Vági, R., Vadász, J.P., Orosz, T.: Challenges and open problems of legal document anonymization. *Symmetry* 13, 1490 (2021)
- Dadgar, S.M.H., Araghi, M.S., Farahani, M.M.: A novel text mining approach based on tf-idf and support vector machine for news classification. In: 2016 IEEE International Conference on Engineering and Technology (ICETECH). pp. 112–116. IEEE (2016)
- Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. *CoRR abs/1705.00440* (2017), <http://arxiv.org/abs/1705.00440>
- Fawzi, A., Samulowitz, H., Turaga, D., Frossard, P.: Adaptive data augmentation for image classification. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3688–3692 (2016)
- Hillard, D., Purpura, S., Wilkerson, J.: Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics* 4(4), 31–46 (2008)
- Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
- Jia, Y., Chen, Z., Yu, S.: Reader emotion classification of news headlines. In: 2009 International Conference on Natural Language Processing and Knowledge Engineering. pp. 1–6. IEEE (2009)
- Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* (1972)

- Kellerman, H., Plutchik, R.: Emotion-trait interrelations and the measurement of personality. *Psychological Reports* 23, 1107–1114 (1968)
- Kubat, M., Matwin, S., és mtsai: Addressing the curse of imbalanced training sets: one-sided selection. In: *Icml*. vol. 97, pp. 179–186. Citeseer (1997)
- Kumar, M.A., Gopal, M.: A comparison study on multiple binary-class svm methods for unilabel text categorization. *Pattern Recognition Letters* 31(11), 1437–1444 (2010)
- Lango, M., Stefanowski, J.: Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data. *Journal of Intelligent Information Systems* 50(1), 97–127 (2018)
- Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development* 1(4), 309–317 (1957)
- Menardi, G., Torelli, N.: Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery* 28(1), 92–122 (2014)
- Nguyen, H.M., Cooper, E.W., Kamei, K.: Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms* 3(1), 4–21 (2011)
- Ring, O., Martina Katalin, S., Guba, C., Váradi, B., Üveges, I.: Approaches to sentiment analysis of hungarian political news at sentence level with dictionary-based method and with machine learning. *PLoS One* (2021), megjelenés alatt
- Sebők, M., Kacsuk, Z.: The multiclass classification of newspaper articles with machine learning: The hybrid binary snowball approach. *Political Analysis* 29(2), 236–249 (2021)
- Taylor, L., Nitschke, G.: Improving deep learning with generic data augmentation. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. pp. 1542–1547 (2018)
- Van Atteveldt, W., Kleinnijenhuis, J., Ruigrok, N., Schlobach, S.: Good news or bad news? conducting sentiment analysis on dutch text to distinguish between positive and negative relations. *Journal of Information Technology & Politics* 5(1), 73–94 (2008)
- Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks (2019)
- Young, L., Soroka, S.: Affective news: The automated coding of sentiment in political texts. *Political Communication* 29(2), 205–231 (2012)
- Zhang, C., Zhou, P., Li, C., Liu, L.: A convolutional neural network for leaves recognition using data augmentation. In: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. pp. 2143–2150 (2015)

# Az ige helyhatározói bővítményeinek megkülönböztetése és az argumentumszerkezeti variánsok korpusz alapú szétválasztása

Szécsényi Tibor<sup>1</sup>, Virág Nándor<sup>2</sup>

<sup>1,2</sup> SZTE Általános Nyelvészeti Tanszék

<sup>1</sup> szecsényi@hung.u-szeged.hu

<sup>2</sup> virag.nandor9910@gmail.com

**Kivonat:** Tanulmányunkban Szécsényi (2019) argumentumszerkezeti modelljét vesszük alapul, melyben nincs kategorikus vonzat-szabad bővítmény megkülönböztetés, helyette a különböző argumentumtípusokat az egyes igék melletti megjelenési valószínűségével jellemezve adja meg az ige argumentumszerkezetét. Röviden ismertetjük az argumentumszerkezet valószínűségi vektor alapú modelljét és bemutatjuk, hogyan lehet a modell segítségével az igék helyhatározói bővítményei között megkülönböztetni a valódi vonzatokat és a tematikus vonzatokat a szabad bővítményektől. Ezután a több argumentumszerkezeti variánsal rendelkező igéknél illusztráljuk, hogyan lehet az ige mellett megjelenő több vonzatszerű bővítményt egy vagy több argumentumszerkezeti variánshoz rendelni.

## 1 Bevezetés

Az igék argumentumszerkezetének ismerete nélkülözhetetlen a természetes nyelvi szövegek szintaktikai (lásd pl. Kovács és mtsai, 2016) és szemantikai feldolgozásához (lásd pl. Gildea és Jurafsky, 2002; Palmer és mtsai, 2005). Az argumentumszerkezet vonzatlistaként való megadása nem tükrözi vissza azt a nyelvhasználói viselkedést, hogy az elvileg kötelező vonzatok sem jelennek meg mindig az ige mellett, más szabad bővítmények pedig az egyes igék mellett viszonylag gyakoriak.

Tanulmányunkban Szécsényi (2019) argumentumszerkezeti modelljét vesszük alapul, melyben nincs ilyen kategorikus vonzat-szabad bővítmény megkülönböztetés, helyette a különböző argumentumtípusokat az egyes igék melletti megjelenési valószínűségével jellemezve adja meg az ige argumentumszerkezetét. A 2. szakaszban röviden ismertetjük az argumentumszerkezet valószínűségi vektor alapú modelljét. A 3. szakaszban bemutatjuk, hogyan lehet a modell segítségével az igék helyhatározói bővítményei között megkülönböztetni a valódi vonzatokat és a tematikus vonzatokat a szabad bővítményektől. Végül a 4. szakaszban a több argumentumszerkezeti variánsal rendelkező igéknél illusztráljuk, hogyan lehet az ige mellett megjelenő több vonzatszerű bővítményt egy vagy több argumentumszerkezeti variánshoz rendelni.

A <https://github.com/szecsényi/MSZNY2022-Szecsényi-Virag> githubon elérhetők a kutatáshoz tartozó korpuszadatok.

## 2 Az igei argumentumszerkezetek vektor alapú jellemzése

Az argumentumszerkezetek jellemzése a hagyományos leíró nyelvészeti elméletekben az ige (régens) vonzatainak felsorolásával és azok elvárt tulajdonságainak megadásával történik. De születtek más jellegű modellek is, a magyarban például (Sass, 2018; 2020) duplakocka modellje lehetővé teszi a vonzatok elkülönítését az idiomatikus kifejezésektől, (Kálmán, 2006; 2016) pedig az igei bővítménytípusok több fajtáját is megengedő graduális modellt javasol.

A természetes nyelv feldolgozása során az argumentumszerkezeti leírások használata problémába ütközik, ha az csak az igei vonzatok leírását tartalmazza: a feldolgozandó szövegekben nem csak a vonzatai jelennek meg az igének, hanem más bővítmények is, továbbá az ige vonzatai is sokszor hiányoznak az ige mellől. Máskor azt nehéz eldönteni egy argumentumszerkezet meghatározásakor, hogy egy bővítmény vonzatnak számít-e vagy sem.

Tanulmányunkban a (Szécsényi, 2019) által bemutatott argumentumszerkezet-modellt használjuk, amely nem tesz különbséget a vonzatok és egyéb bővítmények között. A modell az ige mellett megjelenő bővítményeket azok egy jellemző tulajdonsága alapján csoportosítja, esetünkben ez leginkább a bővítmény fejének az esete.

Jelenleg 32 argumentumtípust különböztetünk meg: PV, CP\_cnd, CP\_imp, CP\_ind, HKM, inf, nom, acc, dat, BAN, ON, RA, VAL, UL, BA, RÓL, HOZ, BÓL, TÓL, NÁL, VÁ, IG, ÉRT, KÉNT, KOR, SZOR, NKÉNT, ADP, ADV, FROM, IN, TO. A PV az igekötői bővítmény, CP\_cnd-től inf-ig a különböző mondatbővítmények vannak, nom-tól NKÉNT-ig az esetragok, ADP és ADV a névutós és határozói kifejezések, a FROM, IN és TO pedig az irányhármasságot (is) kifejező esetragok és névutók meta-típusa. Ez utóbbi három típus nincs kiegészítő disztribúcióban az előbbiekkal, de a később tárgyalt tematikus vonzat – valódi vonzat megkülönböztetésnél kulcsszerepet játszhatnak. Az igék argumentumszerkezetét ezen argumentumtípusok megjelenési valószínűségével adjuk meg, vagyis egy 32 dimenziós valószínűségi vektorral. Egy argumentumtípushoz tartozó valószínűségi érték 1, ha az adott ige mellett annak bővítményeként mindig megjelenik az adott típusú kifejezés, 0, ha sohasem. Ez a két szélső érték szinte soha nem jelenik meg az igék argumentumszerkezeti leírásában, ha korpuszból meghatározott valószínűségi értékekkel dolgozunk, mivel a valódi nyelvhasználat során még a valódi vonzatok sem jelennek meg minden esetben egy ige mellett (pl. pro-drop, ellipszis, rövid válasz), viszont nagyon sok bővítménytípus lehet adjunktum is a mondatban. Emiatt jellemzően 0 és 1 közötti értékek figyelhetők meg. A hagyományos vonzat-szabad bővítmény bináris megkülönböztetést egy valószínűségi küszöbérték megadásával kaphatjuk vissza: ha egy argumentumszerkezetben egy argumentumtípus valószínűségi értéke nagyobb, mint a megadott küszöb, akkor az vonzat, egyébként nem. A vonzatsági küszöb argumentumtípusonként változó lehet.

Az argumentumszerkezet valószínűségi vektorát korpuszból határozhatjuk meg. Egy morfológiailag és szintaktikailag annotált korpuszból vesszük azokat a tagmondatokat, amelyek az adott igét tartalmazzák, és megszámláljuk, hogy ezekben a tagmondatokban hány olyan van, amelyben az ige mellett megadott típusú bővítmény szerepel maximális összetevőként: az igét tartalmazó tagmondatok és az igét és a bővítményt tartalmazó tagmondatok aránya adja az argumentum megjelenési valószínűségének az értékét. Jelenleg kétfajta korpusz feldolgozására van elkészített feldol-

gozási láncunk: a kézzel annotált Szeged Dependecia Treebank (SZDT) (Vincze és mtsai, 2010), illetve tetszőleges (de tipikusan MNSz-ből származó) szövegek magyar-lanc segítségével elemzett változata (Oravecz és mtsai, 2014; Zsibrita és mtsai, 2013). A tagmondatok és a tagmondatokat alkotó maximális összetevők meghatározása a mondatok függőségi elemzéséből származnak.

argType	freq	argType	freq	argType	freq	argType	freq
nom	0,559251	BA	0,033539	ÉRT	0,005942	CP ind	0,186934
acc	0,326353	RÓL	0,024550	KÉNT	0,006258	HKM	0,094095
dat	0,055825	HOZ	0,023189	KOR	0,005671	inf	0,104655
BAN	0,115328	BÓL	0,022399	SZOR	0,003663	ADP	0,105482
ON	0,099435	TÓL	0,016555	NKÉNT	0,001271	ADV	0,324841
RA	0,085415	NÁL	0,011734	PV	0,117021	FROM	0,065994
VAL	0,083331	VÁ	0,008620	CP cnd	0,012245	IN	0,261352
UL	0,062121	IG	0,010440	CP imp	0,004107	TO	0,161969

**1. táblázat.** Az argumentumtípusok összesített előfordulási gyakorisága a Szeged Korpuszban.

Az 1. táblázatban látható a Szeged Korpusz magyarlancal elemzett változatában az egyes argumentumtípusok összesített előfordulási gyakorisága, amely a kijelentő módú igét<sup>1</sup> tartalmazó tagmondatok számának (132 951) és ezekben a tagmondatokban előforduló argumentumtípusok előfordulási számának a hányadosa (a legkisebb előfordulási gyakorisághoz is több mint 500 előfordulási szám tartozik). Ezek az adatok általában használhatók az argumentumtípusok egyes igék melletti előfordulásának vizsgálatánál vonzatsági küszöbértéknek: ha egy ige mellett az egyik argumentumtípus előfordulási gyakorisága magasabb a táblázatban megadott értéknél, akkor tekinthetjük vonzatnak.

Az argumentumszerkezet valószínűségi vektorként való értelmezhetőségét egy kisebb argumentumtípus-halmazon mutatjuk be. Kilenc helyhatározói esetrag (BÓL, BAN, BA, RÓL, ON, RA, TÓL, NÁL, HOZ) előfordulási gyakoriságát vizsgáltuk 13 ige mellett: *ad, beszél, fél, hisz, indul, javasol, jön, kap, lát, nevet, rak, teremt, úszik*. Az esettanulmányok megmutatják, hogyan különböztethetjük meg a helyhatározói esetragok három különböző használatát.

### 3 Helyhatározói esetragok eloszlási mintázatai

A kutatás ezen részén azokat a ragos kifejezéseket vizsgáltuk, amelyek irányhármasság szerinti hármassókat alkotnak. A vizsgált ragos kifejezések a BÓL, BAN, BA (belső érintkezéses viszony), RÓL, ON, RA (külső érintkezéses viszony), TÓL, NÁL, HOZ (közelítő viszony) esetekben álltak. Feltételeztük, hogy az ezen ragokkal álló igei bővítményeket lehetséges osztályozni előfordulási gyakoriságuk alapján, de szem

<sup>1</sup> Vizsgálatunk során azért szorítottunk a kijelentő módú igét tartalmazó tagmondatokra, mert a főnévi igeneves szerkezetek esetében a tagmondathatárok nehezen meghatározhatók, illetve a kijelentő módú tagmondatokat tekintettük az igék argumentumszerkezetét legtisztábban megmutató adatoknak.

előtt tartottuk, hogy az így alkotott csoportok nem lesznek egyértelműen, diszkréten elkülöníthetőek. A csoportokat két tengely mentén állítottuk fel a következő módon:

Az első a vonzat-szabad bővítményi tengely, melynek egyik végpontja a teljesen szabad bővítménység, másik végpontja pedig a teljesen vonzati bővítménység. A másik tengely az adott rag kompozicionalitására vonatkozik. A kompozicionalitás elve szerint az összetett nyelvi kifejezések jelentése kiszámítható az őket alkotó kifejezések jelentéséből és kapcsolódási módjukból. Ez értelmezhető a ragos kifejezésekre is, így itt ezt az értelmezést használjuk. Tehát lehetnek a ragos kifejezések kompozicionálisak és nem kompozicionálisak. A csoportok egy-egy példával láthatóak a következő táblázatban.

	Vonzat	Szabad bővítmény
Nem kompozicionális	1. Valódi vonzat pl. <i>bízik Péterben</i>	3. Egyéb szabad határozó pl. <i>kinjában nevet</i>
Kompozicionális	2. Tematikus vonzat pl. <i>Debrecenben/Szegeden lakik</i>	4. Szabad helyhatározó pl. <i>énekel az erdőben</i>

2. táblázat. A bővítmények csoportosítási lehetősége.

A csoportokat értelmezve: a valódi vonzatok csoportját jellemzi, hogy az ilyen bővítménnyel álló igék egy konkrét ragos kifejezést vonzanak, aminek kompozicionális jelentése nem jelenik meg ebben az esetben. Magas gyakorisággal állnak az igei alaptag mellett, sokszor a kívánt jelentés eléréséhez elengedhetetlen a mondatban való megjelenésük.

A tematikus vonzatok csoportjában olyan elemeket találunk, amelyeket az igei alaptag tematikus szerepszerűen vonz, azaz egy irányhármasság szerinti irányt kíván maga mellé, legyen az bármelyik a megfelelő alakokból.

Az egyéb szabad határozók olyan mondatrészek, amelyekben a rag jelentése nem kompozicionális, de csak néha jelennek meg mondatban, nem tekinthetők vonzatnak.

Az utolsó csoport, a szabad helyhatározók csoportja. Ezek esetlegesen jelennek meg, valóban helyviszonyt fejeznek ki. Már itt érdemes megjegyeznünk, hogy az egyéb szabad határozók és a szabad helyhatározók csoportja jelen eszközökkel nem elkülöníthető, így őket egy csoportnak kell kezelnünk, szabad bővítmények néven.

A helyhatározói esztragos bővítményeket 13 ige esetében vizsgáltuk meg: *ad, beszél, fél, hisz, indul, javasol, jön, kap, lát, nevet, rak, terem, úszik*. Az igék kiválasztási szempontja az volt, hogy előreláthatólag legyenek köztük a megkülönböztethető három csoport mindegyikébe tartozók. További szempont volt még, hogy az igéknek ne legyen sok argumentumszerkezeti variánsa.

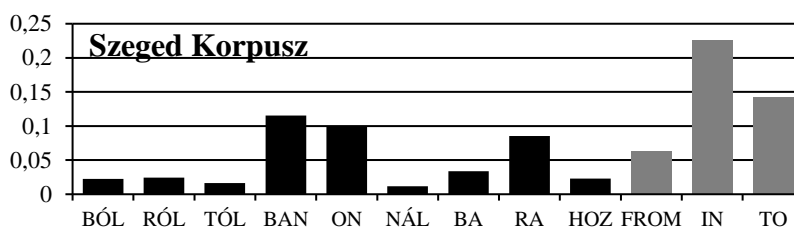
A korpuszpítéshez az adatokat az MNSz-ből (v2.0.5) lekért, igéenként 1500 véletlenszerűen kiválasztott mondat adta. (Gyulai, 2019) alapján a vizsgált igék nem lehetnek igekötősek, mivel az megváltoztatná a vonzatszerkezeteket, ezzel torzítva az adatsort, ezért eleve olyan mondatokat kértünk le a korpuszból, amelyben a vizsgált igék közelében nem volt igekötő. A mondatok elemzését a magyarlanc függőségi elemzővel végeztük el. Ezután további kézi ellenőrzést végeztünk az elemzett mondatlapon, részben az esetlegesen a korpuszba került elváló igekötős igék kiszűrése, részben a rosszul elemzett mondatok kitörlése végett. Rosszul elemzett mondatnak azok számítottak, amelyek eleve nem teljes mondatok voltak, vagy amelyekben a



vizsgált ígét tartalmazó tagmondat határai vagy annak fő összetevői hibásan lettek meghatározva. Ilyen rosszul elemzett mondat az összes mondat 2–3 százaléka volt.

A korpusz mondatainak megszűrése után az előző szakaszban bemutatott elemzési láncot futtatva kaptuk meg a valószínűségi és gyakorisági táblázatokat, amelyeknek adataiból következtetni tudunk.

Ahhoz, hogy egy általános eloszlást érthessünk el a megfigyeléshez, a Szeged Korpusz adatait használtuk. Megfigyelhető, hogy általánosságban a BAN, ON és RA ragos elemek előfordulása a leggyakoribb a vizsgált kifejezések közül, a többi argumentumtípus viszont viszonylag alacsony előfordulást mutatnak. Már itt is észrevehetjük, hogy a közelítő viszonyt kifejező TÓL, NÁL és HOZ fordulnak elő a legkevesebbszer saját terceikben. Az 1. ábrán láthatjuk az így kapott eredményeket, amelyek értékben megegyeznek az 1. táblázatbeli értékekkel. Feketével az egyes ragok előfordulási gyakoriságát láthatjuk (a legkisebb gyakoriságú NÁL is 1560 előfordulási számot takar), szürkével pedig az irányhármasság szerinti csoportok összesített eredményét.

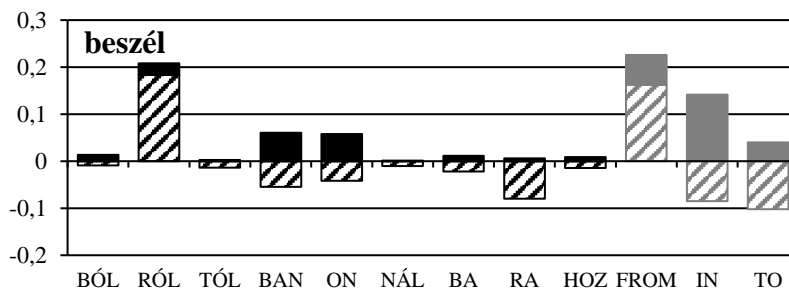


1. ábra: A Szeged Korpuszban található vizsgált ragok előfordulási gyakorisága.

Az eredmények bemutatásaképpen minden csoportból egy-egy ígével illusztráljuk a csoportra jellemző tulajdonságokat.

### 3.1 Valódi vonzattal álló ígék

A *beszél* ige prototipikus valódi vonzattal áll. Vonzata a RÓL, így azt várhattuk el az adatok elemzésénél, hogy ez magas előfordulási gyakoriságot fog mutatni, míg a többi toldalék gyakorisága az átlaghoz mérten csökkenni fog.



2. ábra: A helyhatározói ragok gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival a *beszél* ige esetében.

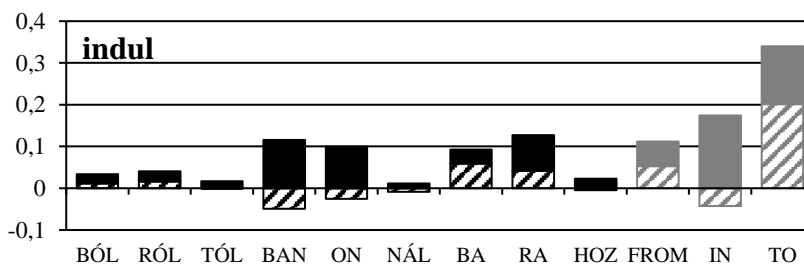
Az eredmények a 2. ábrán láthatóak. Az ábrán az oszlopok 0 fölötti része mutatja az argumentumtípus előfordulási gyakoriságát. Az oszlopok berácsozott részei mutatják a Szeged Korpusz adataihoz mért változást: a 0 fölötti rácskozás az argumentumtípus gyakoriságának a Szeged Korpuszhoz viszonyított növekedésének, a 0 alatti rácskozás pedig a csökkenésének a mértékét mutatja. Az ábrán egyértelműen látszik, hogy minden előfordulás csökkent, csak a RÓL esetében látunk növekedést, és megállapítható, hogy a vizsgált kifejezések közül ez az ige vonzata.

Az adatokat tekintve láthatjuk, hogy az általában leggyakoribb BAN, ON és RA gyakorisága csökkent, a RÓL pedig jelentősen megnövekedett, a vizsgált mondatok 20,80%-ában fordult elő a vonzatnak tekinthető ragos kifejezés. A *beszél* ige 1399 előfordulásában tehát a Szeged Korpusz 2,4%-os átlagos előfordulásához képest majdnem 10-szeres növekedést tapasztalhattunk.

### 3.2 Tematikus vonzattal álló igék

Az *indul* ige érdekes esetet mutat: tematikus vonzattal áll, azaz az adott irányhármasság szerinti toldalékok közül több is megemelkedett előfordulással mutatkozik. Érdekessége abban áll, hogy nem csak egy, hanem kettő irányt is vonz tematikus szerepként, a kiindulópont és a célpont jelentésű ragok gyakorisága is megnövekedett.

Az ige jelentéséből már adódik a vonzatszerkezeti különlegesség, hiszen inherensen tartalmazza azt, hogy az indulási tevékenységnek része ez a két jelentésaspektus. Az adatokra tekintve azt láthatjuk, hogy a belső és külső érintkezéssel viszonyt kifejező BÓL és RÓL, illetve BA és RA jelennek meg megnövekedett számban, azonban a tercek kiegészítő TÓL és HOZ továbbra is alacsony előfordulási gyakorisági adatokat mutat, ezzel is bizonyítva, hogy a közelítő viszonyt kifejező csoport igen ritka. A két különböző tematikus vonzatot kívánó vonzatszerkezet megjelenése lehet az oka annak, hogy az előfordulási számok, bár emelkedtek, nem lettek túl magasak. Ezt azonban jelen eszközökkel nem tudtuk megállapítani, a későbbi, együtt-előfordulással kapcsolatos fejezetben viszont ezzel a jelenséggel foglalkozunk.



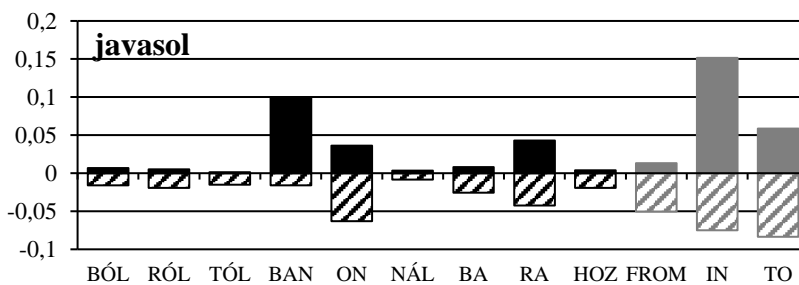
**3. ábra:** A helyhatározói ragok gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival az *indul* ige esetében.

A 3. ábrán láthatjuk, hogy a fentebb említett ragok előfordulása a Szeged Korpuszhoz mérten megemelkedtek. Az érdekesség az utolsó, összesített oszlopokban látható, hiszen a FROM és a TO adatsor is emelkedett. Ezeken megfigyelhető igazán a tematikus vonzati kategória sajátossága: a csoportba tartozó ragok együttesen emelik meg az előfordulások számát.

### 3.3 Szabad bővítménnyel álló igék

A szabad bővítményekkel álló igék csoportjára jellemző, hogy nincsenek kiugró előfordulási gyakorisággal álló adatok. Elvárható a Szeged Korpusz adataihoz hasonló eloszlási mintázat, még ha a pontos számok nem is egyeznek meg az ott mért arányokkal. Azt figyeltük meg, hogy az előfordulások gyakorisága általában csökken. Ennek az lehet az oka, hogy a Szeged Korpusz adatai között szerepeltek olyan igék is, amelyek a vizsgált ragos kifejezéseket vonzzák, így azok megemelik az előfordulási mutatókat.

A *javasol* tipikus ige ebben a kategóriában. Az összes vizsgált argumentumtípusnál csökkenést látunk, csak a BAN éri el nagyjából azt az arányt, ami a Szeged Korpuszban megfigyelhető. A 4. ábrán láthatjuk, hogy az összes oszlopnál kisebb-nagyobb lefelé irányuló sötét oszlop látszik, tehát ez egy jó példa a helyhatározóragos vonzat nélküli igékre.



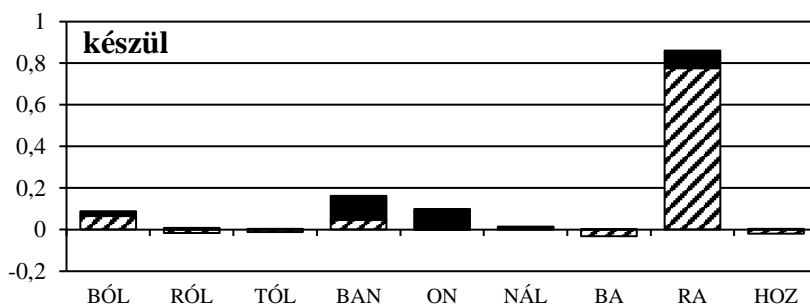
4. ábra: A helyhatározói ragok gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival a *javasol* ige esetében.

### 3.4 Összesített eredmény

A végrehajtott esettanulmányok alapján messzemenő következtetések nem levonhatóak, de tendenciákat megfigyelhetünk, amelyek bizonyítani látszanak a hipotézisünket, miszerint az általunk felállított csoportok az előfordulási gyakoriságok figyelembevételével megállapíthatóak. Több ige és nagyobb esetszám vizsgálata megmutathatja a pontosabb csoporthatárokat és a felismerhető mintázatokat, amelyeket akár egy automatikus elemző is megtalálhat.

## 4 Argumentumszerkezeti variánsok elkülönítése

A tematikus helyhatározói vonzatoknál megfigyelhettük, hogy több, az irányultság tekintetében hasonló argumentumtípus megjelenési gyakorisága is megnőtt a Szeged Korpuszban megfigyeltékhez képest. Ekkor úgy elemeztük az adott igét, hogy van neki vonzata, de nem egy specifikus argumentumtípust vonz, hanem egy meghatározott tematikus szerep betöltésére alkalmas bővítményt. Más igéknél is megfigyelhettünk hasonló jelenséget, vagyis hogy nem egyetlen argumentumtípus gyakorisága nő meg, hanem többé is. Az 5. ábrán például a *készül* ige argumentumszerkezetét és a Szeged Korpuszhoz viszonyított változást láthatjuk:

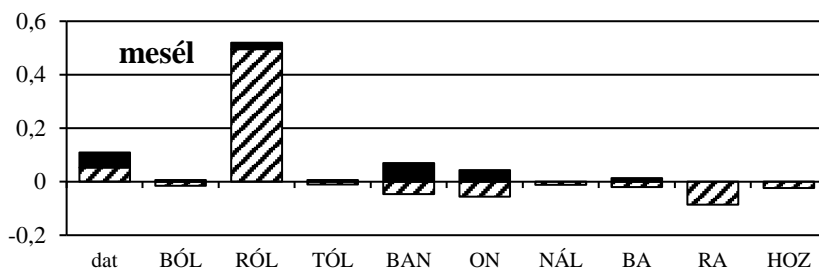


5. ábra: A helyhatározói ragok gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival a *készül* ige esetében.

Látható, hogy bár a BAN megjelenési gyakorisága is megnő kis mértékben, de igazán jelentős növekedést a BÓL és a RA argumentumtípusok esetében figyelhet meg. Ez a két argumentumtípus azonban tematikusan nem sorolható egy csoportba. Itt valójában ugyanannak az igenek két különböző jelentése okozza a több argumentumtípusnál is megfigyelhető gyakoribbá válást: pl. *A cipő bőrBÓL készül*, de *A család születésnapRA készül*. A két argumentumtípus a *készül* ige mellett kiegészítő disztribúcióban figyelhető meg, vagy a BÓL, vagy a RA jelenik meg egy tagmondatban. A tematikus vonzatok esetében szintén kiegészítő disztribúciót találunk, de ott az ige jelentése nem változik az argumentumtípussal: *A vonat DebrecenBE/SzegedRE indul*.

Ha egy ige esetében egynél több argumentumtípus megjelenési gyakorisága is megnövekedik, de az argumentumtípusok kiegészítő disztribúcióban állnak egymással, akkor azt mondjuk, hogy az ige két argumentumszerkezeti variánssal rendelkezik. Ha az argumentumok ugyanabba a tematikus csoportba tartoznak, akkor az ige argumentumszerkezeti variánsainak a jelentése megegyezik, egyébként jellemzően különböző jelentésűek. A különböző jelentésű argumentumszerkezeti variánsok (vagy egyszerűen: különböző variánsok) különböző argumentumszerkezeti vektorral jellemezhetőek. A tematikus vonzattal rendelkező variánsokat tekinthetjük egy variánsnak, amelyben a tematikus vonzatba tartozó argumentumtípusok szabad váltakozást mutatnak, és ezen argumentumtípusok egymáshoz viszonyított gyakorisági eloszlása egy nagyobb mintázathoz illeszkedik.

A 6. ábrán látható a *mesél* ige argumentumszerkezeti vektora és annak változása (a helyhatározói esetragokon kívül a datívuszi argumentumtípussal kiegészítve):



6. ábra: A datívuszi és a helyhatározói ragok gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival a *mesél* ige esetében.

Itt szintén két argumentumtípus megjelenési gyakoriságának a megnövekedését láthatjuk, a datívuszi és a RÓL argumentumtípusét. A *készül* igéhez hasonlóan a két megnövekedett gyakoriságú argumentumtípus itt sem tartozik egy tematikus osztályba. Azonban attól eltérően a datívuszi és a RÓL argumentumtípus nem a *mesél* két különböző variánsánál jelentkezik vonzatként, hanem ugyanannál: *Péter a kirándulásRÓL mesélt a nagymamájáNAK*.

Ha az igék argumentumszerkezetét és argumentumszerkezeti variánsait korpusz alapján szeretnénk meghatározni, akkor csak azt vizsgálhatjuk, hogy az adott igét tartalmazó (tag)mondatban argumentumtípusba tartozó bővítmények jelennek meg, az ige különböző variánsait nem tudjuk elkülöníteni. Ekkor két kérdés adódik:

1. Hogyan lehet (lehet-e?) meghatározni, hogy egy igének több variánsa is létezik?
2. Hogyan lehet (lehet-e?) meghatározni, hogy ha egy igének több variánsa is van, akkor a különböző variánsok milyen argumentumszerkezeti valószínűségi vektorral jellemezhetőek?

Ez utóbbi kérdéshez tartozik az a feladat is, hogy meghatározzuk az ige argumentumszerkezeti variánsainak a megjelenési valószínűségét is.

Jelen tanulmányban az első kérdést próbáljuk megvilágítani. Vizsgálatunkban olyan igéket választottunk, amelyeknek vagy több argumentumszerkezeti variánsa van, és van olyan argumentumtípus, amelyiknek a megjelenését túlnyomórészt csak az egyik variánsnál várjuk (csak az egyik variánsnak vonzata), egy másik argumentumtípust pedig csak a másik variánsnál, vagy pedig olyan egyváltozatos igét, amelynek egynél több vonzata van.

Alaphipotézisünk az, hogy abban az esetben, ha egy variánsnak két vonzata is van: A és B, akkor a két vonzat megjelenésének a valószínűsége független egymástól: ha a két vonzat megjelenési valószínűsége az ige mellett  $P(A)$  és  $P(B)$ , akkor annak a valószínűsége, hogy az ige mellett mindkét vonzat megjelenik,  $P(A) \cdot P(B)$ . Ha viszont a két argumentumtípus az ige más-más variánsánál vonzatok, a két argumentumtípus együttes megjelenésének a valószínűsége  $P(A) \cdot P(B)$ -nél jóval kisebb. Ennek tulajdonképpen nullának kellene lenni, de mivel annál a variánsnál, amelynek az egyik argumentumtípus a vonzata, a másik argumentumtípus is megjelenhet szabad bővítményként, de annak sokkal kisebb a valószínűsége – ezért ebben az esetben is előfordulhatnak együtt, csak sokkal kisebb valószínűséggel.

Vizsgálatunkhoz négy igét választottunk ki. A *bevon* igének két variánsa van, amelyeknek az alanyon és a tárgyon kívül vonzata lehet a BA vagy a VAL: *Péter bevonta a barátját föliáVAL*, ill. *Péter bevonta a barátját a beszélgetésBE*. A már említett *készül* ige két variánsa esetén a BÓL és a RA argumentumtípust vizsgáltuk, a *mesél* egyvariánsos igénél pedig a RÓL és a datívuszi argumentumtípust. A *hív* ige esetében két argumentumszerkezeti variáns vizsgáltunk, az egyiknél csak a tárgyi vonzat kötelező (az alanyon kívül): *Péter hívta a barátját (reggelizni)*; a másik variánsnál pedig a tárgyon kívül datívuszi vonzat is van: *Péter öcskösNEK hívta a barátját*. Célunk az, hogy a korpuszadatok alapján megmutassuk, hogy a *bevon* és a *készül* igének két variánsa van, a *mesél* igének pedig csak egy, illetve lehetőség szerint kimutatni a *hív* ige két variánsát is.

A korpuszvizsgálat során először lekértünk a Magyar Nemzeti Szövegtárból a Mazsola (Sass 2009) segítségével igénként ezer mondatot. A mondatok közül kiszűrtük azokat, amelyek duplikátumok vagy töredékmondatok voltak, és azokat, amelyekben a keresett ige nem kijelentő módban állt. Ezután kézi annotálással bejelöltük, hogy az

igék melyik argumentumszerkezeti variánsa szerepel a mondatban, és csak azokat mondatokat hagytuk meg, amelyekben a vizsgálni kívánt kettő vagy egy variáns volt. Végül a 2. szakaszban említett feldolgozási láncsal kigyűjtöttük tagmondatonként az adott igékre vonatkozó ige-argumentumtípus előfordulási értékeket, amelyekből a vizsgálni kívánt argumentumtípusok előfordulási számát és együtt előfordulási számát, és ezek gyakoriságát is megkaptuk. Ezek a táblázatok igéenként és argumentumszerkezeti variánsokként is rendelkezésre álltak:

<i>bevon</i>	mind	<i>bevon</i> <sub>1</sub>	<i>bevon</i> <sub>2</sub>
ige	289	251	38
acc	258	222	36
VAL	41	5	36
BA	219	218	1
acc&VAL	38	4	34
acc&BA	195	194	1
VAL&BA	6	5	1

<i>mesél</i>	mind
ige	302
acc	77
RÓL	157
dat	33
acc&RÓL	32
acc&dat	9
RÓL&dat	7

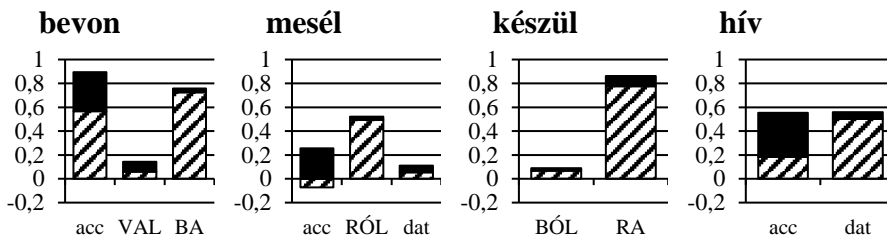
<i>készül</i>	mind	<i>készül</i> <sub>1</sub>	<i>készül</i> <sub>2</sub>
ige	475	46	429
BÓL	42	36	3
RA	409	0	409
BÓL&RA	2	0	2

<i>hív</i>	mind	<i>hív</i> <sub>1</sub>	<i>hív</i> <sub>2</sub>
ige	681	287	394
acc	377	224	153
dat	380	4	376
acc&dat	145	3	142

**3. táblázat.** A *bevon*, *mesél*, *készül* és *hív* igék és vonzataik előfordulási száma, illetve a vonzatok együtt-előfordulási adatai a vizsgált korpuszban.

A 3. táblázatok első oszlopai a vizsgált igékre vonatkozó előfordulási számokat tartalmazzák, majd ugyanezt az ige különböző variánsai esetén. Az első sorban a megfigyelt igék (tagmondatok) számát láthatjuk, alatta az egyes vizsgált argumentumtípusok előfordulási számait, majd az argumentumtípusok páronkénti együttes előfordulási számait. Az előfordulási gyakoriságokat az argumentumtípusok (vagy párok) előfordulási számainak és az ige(variáns) előfordulási számainak hányadosaként kapjuk.

A négy ige mellett megjelenő argumentumtípusok valószínűségi értékei láthatók a 7. ábrán, illetve azok változása a kontrollként használt Szeged Korpuszhoz viszonyítva.



**7. ábra:** A *bevon*, *mesél*, *készül* és *hív* igék vonzatainak gyakorisági adatainak összehasonlítása a Szeged Korpusz adataival.

Az ábrákból, illetve a mögöttük levő gyakorisági adatokból a korábban elmondottaknak megfelelően nem következtethetünk arra, hogy az ábrázolt argumentumtípusok ugyanazon variáns vonzatai-e, vagy különbözőeké.

A korpuszadatokból azonban látszik, hogy a kérdéses argumentumtípusok milyen gyakorisággal fordulnak elő egyszerre az ige környezetében. Ha ugyanazon variánshoz tartoznak, akkor a hipotézisünk szerint az együttes előfordulási gyakoriság az egyes gyakoriságok szorzatához hasonló mértékű, ha viszont külön variánshoz tartoznak, akkor az együttes előfordulási gyakoriságuk jelentősen kisebb ennél. Ezeket az várt és a ténylegesen megfigyelt együtt-előfordulási gyakoriságokat mutatja a 4. táblázat.

<i>bevon</i>		<i>mesél</i>		<i>készül</i>		<i>hív</i>	
p(VAL)	0,142	p(RÓL)	0,520	p(BÓL)	0,088	p(acc)	0,554
p(BA)	0,758	p(dat)	0,109	p(RA)	0,861	p(dat)	0,558
p(VAL)*p(BA)	0,108	p(RÓL)*p(dat)	0,057	p(BÓL)*p(RA)	0,076	p(acc)*p(dat)	0,309
p(VAL&BA)	0,021	p(RÓL&dat)	0,023	p(BÓL&RA)	0,004	p(acc&dat)	0,213

**4. táblázat.** A *bevon*, *mesél*, *készül* és *hív* igék vonzatainak előfordulási gyakorisága, a gyakoriságok szorzata és a vonzatok együtt-előfordulási gyakoriságai.

Mindegyik vizsgált igénél az látszik, hogy a két argumentumtípus együtt-előfordulási gyakorisága kisebb a két argumentum előfordulási gyakoriságának a szorzatánál, azonban míg a *mesél* és a *hív* igék esetében a gyakoriságok szorzata csak kb. 1,5–2-szerese az együtt-előfordulási gyakoriságnak, a *bevon* és a *készül* igék esetében ez 5–20-szoros. A *bevon* és a *készül* igénél ez a várakozásainknak megfelelő, hiszen ott a két bővítmény különböző variánshoz tartozik. Magyarázatra szorul azonban, hogy az egyvariánsos *mesél* esetében miért kevesebb az együtt-előfordulási gyakoriság a vártnál, illetve hogy a kétvariánsos *hív* igénél miért csak ilyen kis mértékű csökkenés figyelhető meg.

Az emberi nyelvhasználat során nem egymástól független információkat közlünk a diskurzus folyamán, hanem egymásra épülő, egymást követő, egymással összefüggő információkat. Az összefüggést a diskurzus témájaként lehet azonosítani: ez sokszor egy személy vagy objektum, amiről új információkat közlünk, vagy egy esemény, amelynek új aspektusait adjuk meg.

A témaként szereplő személy/objektum általában az új információt kifejező ige alanyaként vagy tárgyaként jelenik meg, jellemzően a mondat topik pozíciójában, és gyakran el is hagyjuk, mivel a diskurzusuniverzumban könnyen elérhető, könnyen felidézhető. Ezzel szemben az új információ részét képező igei bővítmények sokkal többször jelennek meg az ige mellett a megnyilatkozásokban. A *hív* ige mindkét variánsánál van tárgyi vonzat, de az első variánsnál (*hívtam Pétert*) a tárgy általában az új információ része, míg a második variáns esetében (*Pétert öcskösnek hívtam*) a tárgy általában az ismert szereplők egyike. A datívuszi vonzatos *hív* ige mellett ennek megfelelően sokkal kisebb gyakorisággal jelenik meg a tárgyi bővítmény, mint az első variánsnál, mint az a 3. táblázatban is látható. A *hív* ige esetében a tárgynak a különböző variánsokban megfigyelhető nagyon eltérő gyakorisági értéke okozza, hogy összességében csak kicsivel kevesebb a tárgy és a datívusz együtt-előfordulási gyakorisága a két bővítmény gyakoriságának a szorzatánál, vagyis a várt nagyobb különbség elmaradásának pragmatikai-szemantikai okai vannak.

A *mesél* ige esetében viszont a RÓL és a datívuszi bővítmény ugyanannak az eseménynek két különböző aspektusát fejezi ki, és általában mindkettő az új információ részének tekinthető. Azonban az emberi nyelvhasználók nem mindig törekednek az események teljes leírására, hanem csak a legfontosabb aspektusokat közlik, így bár mindkét argumentumtípus vonzata az egyetlen variánsnak, az együttes előfordulásuk alacsonyabb lehet a vártnál.

Összegezve tehát elmondhatjuk, hogy a különböző argumentumtípusok egy ige melletti együtt-előfordulási gyakoriságának vártnál alacsonyabb értékéből következtethetünk arra, hogy azok különböző argumentumszerkezeti variánsokban jelennek meg, de a várthoz közelítő érték nem feltétlenül vezet ahhoz, hogy egy variánszhoz tartozónak tekintsük őket.

## 5 Összegzés

Tanulmányunkban Szécsényi (2019) valószínűségi vektor alapú argumentumszerkezeti modelljét alapul véve vizsgáltuk meg az igék helyhatározói bővítményeinek eloszlási mintázatait.

A Szeged Korpusz összes igéjének valószínűségi vektorához viszonyítva az egyes igék argumentumszerkezeti vektorát, három mintázatot különítettünk el. Azokban az esetekben, amelyekben csak az egyik argumentumtípus megjelenési gyakorisága növekedett, az argumentumtípust az ige valódi vonzatának tekintettük – ebben az esetben az ige és a helyhatározó ragos bővítmény (az esetrag szempontjából) nem kompozicionális nem kompozicionális összetétel. Azokban az esetekben, amelyekben több argumentumtípus gyakorisága is megnőtt, de az argumentumtípusok az irányhármasság szempontjából egy csoportba tartoznak, a megnövekedett gyakoriságú argumentumtípusokat egyetlen tematikus vonzat különböző megjelenési formáinak vettük. Végül azokban az esetekben, amikor egyetlen argumentumtípus gyakorisága sem növekedett a Szeged Korpuszban megfigyeltékhez viszonyítva, azt mondtuk, hogy ezeknek az igéknek nincsenek helyhatározói esetragos vonzataik, az ilyen bővítmény mindig szabad bővítmény.

Előfordul olyan eset is, amikor egy ige mellett egynél több (tematikusan nem összefüggő) argumentumtípus előfordulási gyakorisága is megnő a Szeged Korpuszhoz képest. Ekkor a két argumentumtípus lehet ugyanannak az igének két egymástól független vonzata, vagy egy ige két különböző (jelentésű) változatának az egyedi vonzata: ez utóbbi esetben az ige két argumentumszerkezeti variánsát feltételezzük. Ha a két argumentumtípus együttes előfordulásának a gyakorisága lényegesen kisebb, mint az argumentumtípusok előfordulási gyakoriságainak a szorzata, akkor az igének két argumentumszerkezeti variánsa van.

A tanulmányban egy-egy igét kiválasztva esettanulmányokat mutattunk be, amelyek feldolgozásánál egy rögzített feldolgozási láncot használtunk, azt adatokat pedig a MNSz-ből szereztük. Az eredmények jobb általánosíthatósága és további összefüggések megfogalmazhatósága érdekében szükséges a vizsgált igék számának nagyságrendekkel történő emelése.



## Hivatkozások

- Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. In: *Computational Linguistics* 28/3. pp. 245–288. (2002) doi:10.1162/089120102760275983
- Gyulai, L.: Nem kompozicionális igekötős igék argumentumszerkezetének korpuszalapú vizsgálata. In: Ludányi, Zs., Grácsi, T. E. (szerk.) *Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből* 2019. XIII. Alkalmazott Nyelvészeti Doktoranduszkonferencia. pp. 44–58. MTA Nyelvtudományi Intézet, Budapest (2019) doi:10.18135/Alknyelvdok.2019.13.4
- Kálmán, L.: Miért nem vonzanak a régeensek? In: Kálmán, L. (szerk.) *KB 120. A titkos kötet. Nyelvészeti tanulmányok Bánréti Zoltán és Komlósi András tiszteletére*. pp. 229–246. MTA Nyelvtudományi Intézet, Tinta Könyvkiadó, Budapest (2006)
- Kálmán, L.: Bővítmenykeretek mint konstrukciók. In: Kas, B. (szerk.) „Szavad ne feledd” *Tanulmányok Bánréti Zoltán tiszteletére*. pp. 61–72. MTA Nyelvtudományi Intézet, Budapest (2016)
- Kovács, V., Simkó, K., Szécsényi, T.: Szabályalapú szintaktikai elemző szintaktikai szabályok nélkül. In: Tanács, A., Varga, V., Vincze, V. (szerk.) XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). pp. 251–259. Szegedi Tudományegyetem, Szeged (2016)
- Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik (2014)
- Palmer, M., Gildea, D., Kingsbury, P. The Proposition Bank: An Annotated Corpus of Semantic Roles. In: *Computational Linguistics* 31/1. pp. 71–106. (2005) doi:10.1162/0891201053630264
- Sass, B.: „Mazsola” - eszköz a magyar igék bővítmenyszerkezetének vizsgálatára. In: Váradi Tamás (szerk.) *Válogatás az I. Alkalmazott Nyelvészeti Doktorandusz Konferencia előadásaiból*. pp. 117–129. MTA Nyelvtudományi Intézet, Budapest (2009)
- Sass, B.: Az igei szerkezetek algebrai struktúrája, avagy a duplakocka modell. In: *Argumentum* 14. pp. 12–44. (2018)
- Sass, B.: A duplakocka modell és az igei szerkezeteket kinyerő „ugrik és marad” módszer nyelvfüggetlensége, valamint néhány megjegyzés az UD annotáció univerzalitásáról. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 399–407. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2020)
- Szécsényi, T.: Argumentumszerkezet-variánsok korpusz alapú meghatározása. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 315–329. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2019)
- Vincze, V., Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: *Proceedings of the Seventh Conference on International Language Resources and Evaluation*. pp. 1855–1862. European Language Resources Association, Valletta, Málta (2010)
- Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: *Proceedings of RANLP 2013*. pp. 763–771 (2013)



# Lightweight diacritics restoration for V4 languages

Bálint Csanády, András Lukács

Eötvös Loránd University, Institute of Mathematics  
Department of Computer Science, AI Research Group

csbalint@protonmail.ch, lukacs@cs.elte.hu

**Abstract.** Diacritics restoration became a ubiquitous task in the Latin-alphabet-based English-dominated Internet language environment. In this article, we describe a small footprint 1D convolution-based approach, which works on character-level. The model even runs locally in a web browser, and surpasses the performance of similarly sized models. We evaluate our model on the languages of the Visegrád Group, with emphasis on Hungarian.

**Keywords:** diacritics restoration, 1D convolutional neural network, A-TCN, small footprint, V4, Hungarian

## 1 Introduction

Many languages have alphabets in which some characters are derived from other characters using *diacritical marks*. For example, most European languages have alphabets which are derived from the Latin alphabet in this way. The goal of *diacritics restoration* is to restore diacritical marks, given an input text which does not contain (or only partially contains) the proper diacritical marks. Diacritics restoration is a practical task on the Internet, where it is still apparent that computers were built initially with the base Latin alphabet in mind.

Diacritics restoration is a useful preprocessing tool for many NLP tasks, e.g. question answering (Abdelnasser et al., 2014). On the other hand, diacritic restoration is an important tool for language revitalization (Galla, 2009), thus contributing to linguistic diversity, the literacy of endangered languages, and the maintenance of their digital presence (Kornai, 2013). This can be effectively supported by language-independent diacritic restoration tools. Nevertheless, we consider only living languages where large corpora based on the Latin alphabet are available (omitting such exciting cases as Celtic languages or poetry marking). Diacritical marks appear in certain Slavic languages (Czech, Slovak, Polish), some Finno-Ugric languages (Finnish, Hungarian, Latvian), Romanian, Turkish, and, most intensively, in Vietnamese.

Approaches to diacritics restoration have evolved from rule-based and statistical solutions to the application of machine learning models (Yarowsky, 1999). The latter approach can be broken down into solutions using fixed or learned

representations. All solutions with learned representations seem to be based on neural networks connected to the models used in NLP, lately recurrent neural networks models (Hucko and Lacko, 2018) being replaced by transformers (Laki and Yang, 2020; Náplava et al., 2021). In such cases, models used for machine translation are often used to correct diacritical marks (Novák and Siklósi, 2015). Another approach is to consider diacritics restoration as a sequence labeling problem where convolutional neural networks and recurrent neural networks such as BiLSTM-s (Náplava et al., 2018) can be applied. We apply a fast language-independent method with small footprint for automatic diacritic restoration using a neural architecture based on 1D convolutions, the so called Acausal Temporal Convolutional Networks (A-TCN). Models based on A-TCN have a comparable performance to BiLSTM-s (Alqahtani et al., 2019).

We focus on the languages of the Visegrád Group (V4), with emphasis on Hungarian. In Hungarian the characters which can receive diacritical marks are exactly the vowels (e.g.  $u \mapsto \{u, \acute{u}, \ddot{u}, \tilde{u}\}$ ). For Hungarian, the current state of the art is reported by Laki and Yang (2020) and is achieved by neural machine translation. Our main contribution is a lightweight model, which can even be run locally in web browsers, allowing client-side inference. We compared our model with Hunaccent (Ács and Halmi, 2016); both models have a similar size of around 10MB. Our approach outperformed Hunaccent by a large margin.

## 2 Methods

We approached the diacritics restoration problem as a character-sequence labeling task. We chose the output labels as the set of characters in each alphabet. An alternative way to model the restoration task could have been to produce the possible diacritical marks (including the empty mark) on the output side. Our choice is motivated by the expectation that the model’s scope could be expanded, and it might be able to correct other local errors in the text, not only missing diacritical marks.

The neural network architecture we considered for sequence labeling are Temporal Convolutional Networks (TCNs). TCNs are a generic family of models, with notable examples including WaveNet (Oord et al., 2016). TCNs are 1D fully convolutional networks, where the convolutions are causal, and at time  $t$  output is produced in each layer by the convolution of input elements from time  $t - 1$  and earlier (Bai et al., 2018). To increase the effective size of the convolutional windows, dilated convolutions can be used (Yu and Koltun, 2015). The network is built with dilation factors which increase exponentially by the depth of the network (Fig. 1). This ensures that the window on the input sequence, which the network can utilize for the inference of a given label, also increases exponentially.

TCNs also contain residual connections (He et al., 2016). A residual block involves a series of transformations, the result of which are then added to the input. The transformation consists of a dilated convolution followed by a normal-

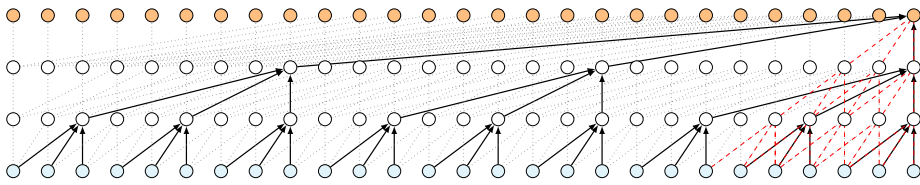


Fig. 1: TCN architecture (kernel size: 3, dilation factors: 1,3,9). Red dashed: without dilation.

ization layer, activation function, and dropout. This is repeated  $b$  times (typically  $b = 2$ ).

TCNs work well for applications where information flow from the future is not permitted. For diacritics restoration it is essential to incorporate future context as well as past context. To achieve this, we have to slightly modify the base TCN architecture as seen in Fig. 2. The modified TCN architecture is called acausal TCN, or A-TCN for short (Alqahtani et al., 2019).

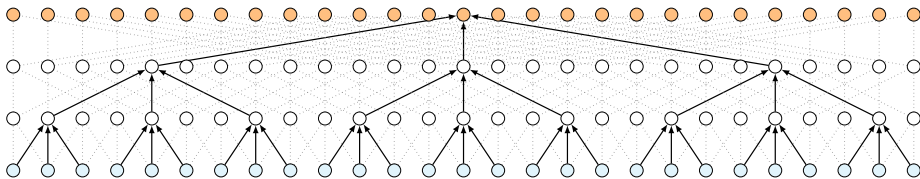


Fig. 2: A-TCN architecture (kernel size: 3, dilation factors: 1,3,9).

### 3 ONNX.js compatibility

Our model can be converted to ONNX (Bai et al., 2017), a cross-platform neural network format. ONNX.js (Wang et al., 2018) is a JavaScript library, which can run models in ONNX format, which in turn makes it possible to run our model in the browser. Inference happens on the clients device, making use of the clients graphical processor with the help of WebGL.

Converting a model to work with ONNX.js requires some care. For example LSTMs are not supported yet, and even 1D convolutions had to be simulated with 2D convolutions. Although they are mathematically equivalent, we found that training the model in PyTorch is much more effective if we reduce a spatial dimension to 1 in the 2D convolution (instead reducing the feature size to 1).

Another difficulty is that the model allows arbitrary input lengths, but in ONNX.js the first inference fixes the input sequence length. The solution is to dynamically reload the model. If the input is longer than the current limit, the model is reloaded with double length.

Our demonstration web page with diacritics restoration for the V4 languages is available at <https://web.cs.elte.hu/~csbalint/diacritics/demo.html>.

## 4 Datasets

### 4.1 LINDAT

We can generate the data for diacritics restoration in a self-supervised fashion. Grammatically correct sentences from the target language provide the annotated data, which means that the removal of the diacritical marks provide the input.

For training on the V4 languages, we used the datasets provided by Náplava et al. (2018). We will refer to these as LINDAT<sup>1</sup>. We cleaned up the datasets by removing the sentences containing exotic characters (if the length of the character was more than 1 after applying the `unidecode` Python function, we considered the character exotic). We also cut off all the sentences to a maximum length of 500.

Language	Train			Dev		
	Sequences	Avg.seq.len.	Characters	Sequences	Avg.seq.len.	Characters
Cze	946 k	107.6	101.8 M	14.5 k	114.4	1.66 M
Hun	1 287 k	108.3	139.3 M	14.7 k	120.7	1.77 M
Pol	1 063 k	116.2	123.6 M	14.8 k	121.3	1.80 M
Svk	609 k	106.7	65.1 M	14.9 k	114.7	1.71 M

Table 1: Statistics of the datasets by Náplava et al. (2018) we used for training.

### 4.2 Hungarian Webcorpora

We also considered two additional corpora for Hungarian. We trained a model on the dataset built from Hungarian Webcorpus 2.0 (HunWeb2)<sup>2</sup> by Nemeskey (2020), and evaluated the models on the dataset built from Hungarian Webcorpus (HunWeb1)<sup>3</sup> by Halácsy et al. (2004). Each corpus contains a large collection of Hungarian text documents. To prepare the data, we extracted sentences from each document until we reached a length limit of 500. After extracting the sequences, we random sampled them, and created the train-dev cuts (Table 2). In the case of HunWeb2, we used the "2017-2018" part of the Common Crawl subcorpus. Not splitting up the documents ensures that the train and dev data do not correlate too much, as we have found that there can be a lot of repeated sentences within one document.

<sup>1</sup> <https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2607>

<sup>2</sup> <https://hlt.bme.hu/en/resources/webcorpus2>

<sup>3</sup> <http://mokk.bme.hu/resources/webcorpus/>

Corpus	Train			Dev		
	Sequences	Avg. seq. len.	Characters	Sequences	Avg. seq. len.	Characters
HunWeb1	96.1 k	405.5	39.0 M	12.0 k	406.8	4.89 M
HunWeb2	73.7 k	498.7	36.8 M	9.2 k	498.4	4.59 M

Table 2: Statistics of the Hungarian Webcorpus based datasets used for training.

We analyzed the datasets in terms of word ambiguity. We first cleaned up the train data by removing all sequences not containing enough diacritical marks. Let us call the base of a word the word we get, after removing the diacritical marks from it. We categorized a base unambiguous, if the data contained only one diacritized version of it. Similarly, a word was categorized ambiguous, if multiple diacritized forms existed in the data. The ambiguity of a word may be due to grammar, or to an error in the corpus, even after the cleanup step performed to decrease the number of such false positives. Unambiguous words can be diacritized with a dictionary-based approach.

In Table 3 we see the statistics related to ambiguous and unambiguous words in the datasets. We see that there are similar amounts of ambiguous and unambiguous words in the data, but the ambiguous words come from a much smaller set of bases. We considered the metric of ambiguous word accuracy versus word accuracy, and we saw that at the beginning of the training ambiguous word accuracy was higher, but as the model improved the two metrics switched places. This can be explained by Table 3.

Corpus	Sequences		Unambiguous		Ambiguous		Ratio	
	Words	Bases	Words	Bases	Words	Bases	Words	Bases
HunWeb1	73.3 k	13.0 M	7.50 M	635 k	5.51 M	16.6 k	1.36	38.2
HunWeb2	69.8 k	14.0 M	7.87 M	746 k	6.08 M	20.2 k	1.29	36.9

Table 3: Word statistics of the filtered Webcorpus based datasets for Hungarian.

## 5 Experimental Setup

In terms of model architecture we used the following hyperparameters. The character embedding dimension was set to 50. After the embedding, the vectors are upsampled to dimension 250, which is the channel size. The network contains 4 residual block layers with dilation factors of 1,2,4, and 8, respectively. Each block contains 2 convolutional layers, each followed by batch normalization, ReLU, and spatial dropout layers with a rate of 0.2, respectively. The convolutions have a kernel size of 5. In the convolutions, zero padding is used to ensure that the output is the same length as the input.

We augmented the training data before each epoch in the training. If a character had a diacritical mark, we removed it with a probability of 80%. In real world use, the absence of diacritical marks might only be partial.

The model implemented in PyTorch was trained on 4 Nvidia RTX 2080 Ti graphics cards. Training took approximately one day per model. Our model is available at [https://github.com/aielte-research/Diacritics\\_restoration](https://github.com/aielte-research/Diacritics_restoration).

## 6 Results

We calibrated our model lightweight enough to be converted to HTML. For Hungarian we took Hunaccent (Ács and Halmi, 2016) as a direct comparison. Hunaccent is decision tree based, and it shares our goal to implement a small footprint restorator. Moreover, it also can be run locally in a browser. To ensure a fair comparison, we set up our model to have a size similar to the 12.1 MB of the trained model of Hunaccent. The raw ONNX file of our trained model is 9.65 MB and our demo HTML file is 12.88 MB. The HTML file contains the ONNX file as a Base64 encoded string.

Model	Train data	Eval data	Character	Vowel	Alpha-word	Sequence
Hunaccent	HunWeb1	HunWeb1	0.9874	0.9619	0.9129	0.0868
		HunWeb2	0.9838	0.9511	0.8942	0.0055
		LINDAT	0.9834	0.9509	0.8934	0.2732
A-TCN	HunWeb2	HunWeb1	<b>0.9968</b>	<b>0.9903</b>	<b>0.9778</b>	<b>0.4223</b>
		HunWeb2	<b>0.9965</b>	<b>0.9893</b>	<b>0.9764</b>	<b>0.3246</b>
		LINDAT	0.9952	0.9876	0.9715	0.6596
A-TCN	LINDAT	HunWeb1	0.9945	0.9834	0.9617	0.2724
		HunWeb2	0.9940	0.9819	0.9596	0.1352
		LINDAT	<b>0.9975</b>	<b>0.9925</b>	<b>0.9824</b>	<b>0.7890</b>

Table 4: Accuracy comparison for Hungarian diacritic restoration between the baseline (Hunaccent) and our model (A-TCN). We used the pretrained Hunaccent model provided by the authors. The numbers indicate the results on non-augmented, fully dediacritized input.

Compared to the baseline, our model achieved significantly better results in all of the metrics we considered (Table 4). *Character* accuracy measures the ratio of the correct characters in the output. *Important character* accuracy is measured on characters for which diacritical marks are applicable. In the case of the Hungarian language, these characters are the vowels. *Alpha-word* accuracy is measured by the ratio of the correct words in the output, where only the words are considered which contain at least one alphabetical character. *Sequence* accuracy is measured by the ratio of flawless sequences, which is inversely proportional to the average length of the sequences.



In Table 5 we can see the effect of the augmentation. Hunaccent performs better on data where all of the diacritics are missing, while our model performs slightly better, but almost the same when we leave about 20% of the diacritical marks.

Eval. type	Hunaccent		A-TCN	
	Vowel	Alpha-word	Vowel	Alpha-word
Augmented	0.9400	0.8705	<b>0.9908</b>	<b>0.9795</b>
Non-augmented	<b>0.9511</b>	<b>0.8942</b>	0.9893	0.9764

Table 5: Performance comparison of Hunaccent and A-TCN (augmented training) on the augmented and the non-augmented task (HunWeb2).

For Hungarian we compared the datasets in terms of performance of the trained models (Table 4). Our tests indicate that our HunWeb2-based dataset yields better results. This is partly due to the fact that as shown in Table 6, the model seems to overfit when trained on the dataset provided by Náplava et al. (2018). The train and dev data are likely not independent enough.

Dataset	Train		Dev	
	Vowel	Alpha-word	Vowel	Alpha-word
HunWeb2	0.9924	0.9828	0.9893	0.9764
LINDAT	0.9922	0.9816	0.9925	0.9824

Table 6: Train and dev accuracies of the same model trained on HunWeb2 and LINDAT. The model seems to overfit on LINDAT.

Language	Character	Important Character	Alpha-word	Sentence
Cze	0.9966	0.9944	0.9783	0.7344
Hun	0.9975	0.9925	0.9824	0.7890
Pol	0.9987	0.9970	0.9903	0.8810
Svk	0.9966	0.9947	0.9784	0.7420

Table 7: Accuracies on V4 languages trained on the dataset provided by Náplava et al. (2018).

The performance of our model on the V4 languages can be seen in Table 7. The results indicate that our model is language-agnostic and works well for

its size for multiple different languages. The alpha-word accuracies are slightly below the ones reported by Náplava et al. (2018).

## 7 Error Analysis

The confusion matrix of the A-TCN model (trained and evaluated on HunWeb2) can be seen in Table 9. Even though our model can output every character in the vocabulary at each position, the only confused characters were vowels with the same base. We included precision (PPV) and recall (TPR) in the table. The overall weighted F1 score for vowels is 0.990.

		Actual Vowel										
		o	ó	ö	ő	PPV	u	ú	ü	ű	PPV	
Predicted Vowel	o	151k	686	523	251	0.990	u	42.3k	180	188	55	0.990
	ó	849	39.1k	58	201	0.973	ú	253	11.6k	18	29	0.975
	ö	399	32	38.2k	118	0.986	ü	170	12	22.1k	19	0.991
	ő	397	145	123	35.1k	0.981	ű	93	26	65	7618	0.976
	TPR	0.982	0.978	0.982	0.984		TPR	0.988	0.982	0.988	0.987	
		a	á	PPV	e	é	PPV	i	í	PPV		
Predicted Vowel	a	337k	2452	0.993	e	376k	3386	0.991	i	164k	286	0.998
	á	2146	131k	0.984	é	2264	125k	0.982	í	478	23.7k	0.980
	TPR	0.994	0.982		TPR	0.994	0.974		TPR	0.997	0.988	

Table 8: Vowel confusion matrix

We performed a small-scale manual evaluation of the A-TCN model. After inference on the evaluation dataset, we selected 100 random errors to be manually classified in the following categories.

1. The error is false positive due to a corpus error, the model output is the correct form.
2. The input is ambiguous at word level, but the model output does not fit grammatically in the sentence.
3. The output is not wrong grammatically, but does not agree with the wider context of the text.
4. Though the model output and the ground truth are different, they both are adequate.
5. The error occurred in a named entity.
6. None of the above, true error.

According to the manual evaluation (Table 9) around 50% of the errors belong to categories 2 and 6. We can reasonably expect to reduce these errors by increasing the size of the model, both to increase the perceived vocabulary of the

Error class	Ratio
1. Corrected Input	0.11
2. Word Ambiguous Input	0.27
3. Grammar Ambiguous Input	0.11
4. Context Ambiguous Input	0.12
5. Named Entity	0.17
6. Incorrect Output	0.22

Table 9: Error classes of the Hungarian A-TCN model.

model, and also to enable a larger context window to draw information from, as some of the grammatical context is likely too far away for the model with the current hyperparameters. Named entity errors are a bit harder to reduce, since they are often less frequent or more ambiguous in the corpus. Errors due to ambiguous input in terms of grammar could be harder to reduce as they sometimes require more insight.

## 8 Conclusion

We presented a model of small size based on 1D convolutional neural network for diacritic restoration. Furthermore, the model is ONNX.js compatible, so it can even be used in a web browser. The model was evaluated for V4 languages and it performed similarly well compared to other larger models and outperformed models of similar size. In the case of the Hungarian language, we considered three data sets and studied the generalizing power of the model between data sets.

Further research is needed to expand the applicability of the model to correcting general errors in texts, including spelling. We plan to try a larger, but still browser-compatible model, and at least in the case of Hungarian, on a larger training data set.

## Acknowledgments

The research was partially supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program, the Hungarian National Excellence Grant 2018-1.2.1-NKP-00008 and the grant EFOP-3.6.3-VEKOP-16-2017-00002.

The second author was supported by project "Application Domain Specific Highly Reliable IT Solutions" implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

We managed our experiments using <https://neptune.ai>. We would like to thank the Neptune Labs team for providing us access to the team version and the technical support.

We would like to thank Dániel Varga for drawing our attention to the problem of lightweight diacritics reconstruction, and Judit Ács for her help with NLP issues.

## Bibliography

- Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N.M., Torki, M.: Al-bayan: an arabic question answering system for the holy quran. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP). pp. 57–64 (2014)
- Ács, J., Halmi, J.: Hunaccent: Small footprint diacritic restoration for social media. In: Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop Programme. p. 1 (2016)
- Alqahtani, S., Mishra, A., Diab, M.: Efficient convolutional neural networks for diacritic restoration. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 1442–1448 (2019)
- Bai, J., Lu, F., Zhang, K., et al.: ONNX: Open Neural Network Exchange. <https://github.com/onnx/onnx> (2017)
- Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
- Galla, C.K.: Indigenous language revitalization and technology: From traditional to contemporary domains. Indigenous language revitalization: Encouragement, guidance & lessons learned pp. 167–182 (2009)
- Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for hungarian. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04) (2004)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
- Hucko, A., Lacko, P.: Diacritics restoration using deep neural networks. In: 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). pp. 195–200. IEEE (2018)
- Kornai, A.: Digital language death. PloS one 8(10), e77056 (2013)
- Laki, L.J., Yang, Z.G.: Automatic diacritic restoration with transformer model based neural machine translation for east-central european languages. In: ICAI. pp. 190–202 (2020)
- Náplava, J., Straka, M., Straková, J.: Diacritics restoration using BERT with analysis on czech language. The Prague Bulletin of Mathematical Linguistics No. 116, pp. 27–42 (2021)

- Náplava, J., Straka, M., Straňák, P., Hajic, J.: Diacritics restoration using neural networks. In: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018) (2018)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)
- Novák, A., Siklósi, B.: Automatic diacritics restoration for Hungarian. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2286–2291 (2015)
- Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
- Wang, Y., Seshadri, H., et al.: ONNX.js. <https://github.com/microsoft/onnxjs> (2018)
- Yarowsky, D.: A comparison of corpus-based techniques for restoring accents in Spanish and French text. In: Natural language processing using very large corpora, pp. 99–120. Springer (1999)
- Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)



## Dinamikus szemantikai modell megszorított predikátumokkal

Szécsényi Tibor<sup>1</sup>, Kovács Viktória<sup>2</sup>

<sup>1</sup> SZTE Általános Nyelvészeti Tanszék  
szecsényi@hung.u-szeged.hu

<sup>2</sup> SZTE Nyelvtudományi Doktori Iskola  
viktoria.kovacs12@gmail.com

**Kivonat:** A tanulmány a diskurzusok dinamikus szemantikai reprezentációját kezelni képes programot mutat be. A diskurzus megnyilatkozásait megfogalmazó predikátumlogikai formulák megadásánál a program meghatározza a diskurzus által leírt lehetséges világok halmazát. A logikai formulákban használt predikátumok megadásánál a predikátumra vonatkozó megszorítások alkalmazásával a lehetséges világok száma nem nő kezelhetetlen nagyságúra. A predikátumokra vonatkozó megszorítások a természetes nyelvi kifejezések, predikátumok ontológiáját követik, lehetőséget adva az osztályozások (ekvivalenciareláció), a rendezések és a fogalmak alá-fölérendeltségével kapcsolatos domain-megszorítások könnyű kezelhetőségének.

### 1 Bevezetés

A természetes nyelvi kifejezések jelentésének megadására többféle rendszer használható. A nyelvtudományban a jelenleg két legelterjedtebb szemantikaelmélet a kognitív nyelvtan konstrukciós, jelentéssémákat használ a jelentések reprezentációjára, a formális vagy modellelméleti szemantika pedig különböző logikai kifejezéseket rendel a természetes nyelvi konstituensekhez, majd a logikai formulákat absztrakt modellekben interpretálja – tanulmányunkban a modellelméleti szemantikai reprezentáció számítógépes alkalmazásának lehetőségeit vizsgáljuk.

A modellelméleti szemantikában az interpretáció történhet egy meghatározott modellben: ekkor a nyelvi kifejezések extenzióját, vagyis a modellbeli jelölését határozhatjuk meg. A bonyolultabb, intenzionális kifejezések jelentését viszont a lehetséges világok, modellek mindegyikében kell értelmeznünk. Lehetséges világnak tekinthetjük az interpretálandó kifejezésekben szereplő atomi kifejezések által meghatározott összes lehetséges modellt, vagy ezeknek egy olyan részhalmazát, amelyet valamilyen modalitás alapján választunk ki. Ha például egy kijelentésünk a  $p$ ,  $q$  és  $r$  atomi kijelentésekből épül fel, akkor ez a három atomi kijelentés nyolc különböző lehetséges világot határoz meg. Ha azonban feltételezzük, hogy csak azok a világok az elfogadhatóak, amelyek a hétköznapi megfigyeléseinknek megfelelnek, és a  $p$  kijelentés azt állítja, hogy *derült az ég*, a  $q$  pedig azt, hogy *villámlik*, akkor azok a világok, amelyekben a  $p$  és  $q$  állítás egyaránt igaz, nem lehetséges: ez a modalitás hat lehetséges világot eredményez.

A dinamikus szemantika a (monológ vagy társas) diskurzust úgy értelmezi, hogy a diskurzus során megfogalmazott kijelentések a kezdeti teljes, minden megszorítástól mentes lehetségesvilág-halmazt folyamatosan szűkítik: az új kijelentések új információkat közölnek a világról a hallgató számára, aki így a diskurzus során egyre pontosabb ismeretekkel rendelkezik arról, hogy melyek a lehetséges világok – ideális esetben egészen addig, amikor már annyira leszűkül a lehetséges világok halmaza, hogy csak egy eleme marad, a beszélő által leírni kívánt világ modellje. Így a természetes nyelvi kifejezések, kijelentések jelentése a dinamikus szemantikában valójában a diskurzusban részt vevők által lehetségesnek tartott világokra gyakorolt hatásként értelmezhető. Emiatt a jelentés dinamikus értelmezésében modellnek a megnyilatkozások előtti, illetve azt követő világhalmazt tekinthetjük, így a továbbiakban a modell terminust ilyen értelemben használjuk.

Tanulmányunkban egy olyan számítógépes eszközt kívánunk bemutatni, amely ezeket a világmódel-megváltoztatási folyamatot képes nyomon követni elsőrendű logikai kifejezések használatával. A `dynsem` program<sup>1</sup> prolog nyelven íródott, kipróbálva az SWI-Prolog 8.0.1 verzióján lett. A program három fájlból áll. A `modellChecker3.pl` és a `comsemPredicates.pl` fájlok Patrick Blackburn és Johan Bos (2005) könyvéhez tartozó programkönyvtár két darabjának kibővített változata, a harmadik fájl pedig maga a főprogram. További két fájl a létrehozott modellek exportálását és importálását teszi lehetővé. A program a Szécsényi (2013)-ban bemutatott rendszer továbbfejlesztett változata.

A tanulmányban először megmutatjuk, hogy a dinamikus szemantikai modell hogyan kapcsolható a szemantikus következtetéshez. Ezután felvázolunk egy olyan dinamikus interpretáló rendszert, ami egy teljes, megszorításokat nem tartalmazó világhalmazból indul ki, rámutatva egyúttal ennek a hiányosságaira is: nem túl bonyolult atomi kifejezéseket használva is kezelhetetlen mennyiségű lehetséges világgal kell számolni. Ezt követően bemutatjuk, hogyan lehet optimalizálni a modellépítést, amely során mindig csak a feltétlenül szükséges predikátumok által meghatározott lehetséges világokat hozzuk létre, amit aztán a diskurzusban megjelenő kijelentések révén szűkítünk le, majd hogy újabb kijelentéseket is megfogalmazhassunk, újabb predikátumokat vezetünk be, ezáltal ismét növelve a lehetséges világok számát. A következő szakaszban, felismerve, hogy az új predikátumok bevezetése után sokszor olyan típusú megszorításokat kell elvégezni, amelyek a természetes nyelvi diskurzusok során nem expliciten jelennek meg, már a predikátumok bevezetésénél közvetlenül alkalmazzuk a tipikus megszorításokat: a természetes nyelvben használt kifejezések általában nem szabad predikátumként értelmeződnek, hanem például a *fiú* és a *lány* predikátumok osztályozásként, a *fiatalabb* pedig lineáris rendezésként. A dinamikus szemantikai modell bemutatásának a végén egy további megszorítástípust mutatunk be: az újonnan bevezetendő predikátumok nem mindig a teljes modellbeli individuumhalmazon értelmezendők, hanem sokszor annak egy meghatározott részhalmazán: a *fiú* predikátum soha nem tartalmaz például élettelen individuumokat. Az ilyen megszorított predikátumbevezetéssel a dinamikus szemantikai értelmező által használt lehetségesvilág-halmaz folyamatosan kezelhető méretű marad.

<sup>1</sup> A program és annak részletesebb leírása megtalálható a következő github projektoldalon: <https://github.com/szecsényi/dynsem.pl>.



A *dynsem* program csak koherens, javításmentes diskurzusokat képes kezelni, a diskurzusok kijelentéseinek a visszavonásáról vagy a diskurzusban nem szereplő individuumokra történő utalásról Dyekiss (2011) és Dyekiss (2013) ad elméleti útmutatást.

## 2 Szemantikai és formális következtetés

A bemutatott dinamikus szemantikai modell hasonlóságot mutat a logikai következtetések, következtetőrendszerek működésével. A diskurzus során korábban elhangzott megnyilatkozások tekinthetők premisszáknak, és az új megnyilatkozás által kifejezett állítást vizsgáljuk ezzel a premisszahalmazzal kapcsolatban. A különbség azonban nyilvánvaló. A következtetés során azt várjuk, hogy az új állítás (azaz a konklúzió) igaz legyen a premisszák által meghatározott lehetségesvilághalmaz minden elemében, azaz vonzata legyen annak (Russell és Norvig, 2010: 240). A diskurzusok során viszont az a cél, hogy az új állítás új információt közöljön a világról, tehát ne legyen az összes lehetséges világban igaz. A következtetés során tehát *valid*, a diskurzus során pedig *informatív* állítás megfogalmazása a cél (Blackburn és Bos, 2005: 28).

A következtetésnek ez az értelmezése feltételezi, hogy a premisszákkal, illetve a diskurzus korábbi megnyilatkozásaival kompatibilis lehetséges világok a rendelkezésünkre állnak, és ezekben a lehetséges világokban interpretáljuk az új állítást. A következtetésnek ezt a módját szemantikusnak nevezzük, mivel nem közvetlenül a premisszákból kaphatjuk meg a konklúziót, hanem a premisszák és a konklúzió interpretációin keresztül, azok összevetéseként. Kizárólag a premisszák alakja alapján is végezhetünk következtetést, ezt nevezzük formális következtetésnek. A formális következtetés az interpretációs modellek összetettsége esetén hatékonyabbak, mivel nem kell létrehozunk a nagy számú lehetséges világot. A formális következtetési eljárások képesek az összes vonzat levezetésére, lásd (Russell és Norvig, 2010: 255).

A természetes nyelvi diskurzusok megnyilatkozásainak leírása során is használhatunk formális és szemantikai megközelítést is, azonban itt nem egy bináris döntés meghozatala a cél (vonzat vagy nem vonzat), hanem a megnyilatkozás által kifejezett állítás modellváltoztató képességének a megadása, ezért a szemantikai összefüggés megadása tekinthető közvetlenebb módszernek.

## 3 Megszorítatlan modellből kiinduló szemantikai reprezentáció

Tanulmányunkban a diskurzust  $\alpha_1\text{-}\alpha_2\text{-}\alpha_3\text{-}\dots\text{-}\alpha_n$  kijelentések lánacának tekintjük, ahol az  $\alpha_i$  olyan elsőrendű logikai formula, amelyben nevek és individuumváltozók (termek), a változókat kötő egzisztenciális és univerzális kvantorok, egy- és kétargumentumú predikátumok, azonosság reláció, termeket definiáló  $\iota$  operátor, valamint a szokásos nulladrendű logikai műveletek találhatóak (negáció, konjunkció, diszjunkció és kondicionális). Feltételezzük, hogy a diskurzus modelljében előre rögzített az individuumok (véges) halmaza:  $U$ . Ezt diskurzusuniverzumnak nevezzük.

A diskurzust leíró dinamikus szemantikai eszközhöz első lépésként meg kell határozni a kezdeti, az összes lehetséges világot tartalmazó  $M_0$  modellt. A nevek interpretációja minden világban ugyanaz, a predikátumok interpretációja azonban változó. Egy világ definiálásához meg kell adni az összes egy- és kétargumentumú predikátum világbeli interpretációját: egyargumentumú predikátumoknál az  $U$  egy részhalmazát, kétargumentumú predikátumoknál pedig  $U \times U$  egy részhalmazát. Az egyargumentumú predikátumok interpretációja így  $U$  hatványhalmazának ( $\mathcal{P}(U)$ ) egy eleme, a kétargumentumúaké pedig  $\mathcal{P}(U \times U)$  egy eleme. A predikátumok interpretációi egymástól függetlenek, ezért  $M_0$  a kijelentésekben használt predikátumok lehetséges interpretációinak a Descartes-szorzataként értelmezhető. Ha a kijelentések  $n$  darab egyargumentumú és  $m$  darab kétargumentumú predikátumot tartalmaznak, akkor az  $M_0$  modell által tartalmazott lehetséges világok halmaza a következőképpen adható meg (a indexben megadott számok csak a predikátumok számlálását segítik):

$$M_0 = \mathcal{P}(U)_1 \times \mathcal{P}(U)_2 \times \dots \times \mathcal{P}(U)_n \times \mathcal{P}(U \times U)_1 \times \mathcal{P}(U \times U)_2 \times \dots \times \mathcal{P}(U \times U)_m \quad (1)$$

Az egyes  $\alpha$  kijelentéseket mindegyik lehetséges világban interpretáljuk, így az  $\alpha$  tekinthető az  $M_0$  modellen értelmezett karakterisztikus függvénynek is, az  $M_0$ -t alkotó lehetséges világok mindegyikéhez igaz vagy hamis értéket rendelve. Az  $\alpha$  kijelentés által  $M_0$ -ból kiválasztott  $M_1$  részhalmaz lesz a dinamikus interpretáció modellje, vagyis minden  $\alpha$  kijelentés valójában  $M_0$  részhalmazaihoz  $M_0$  részhalmazait rendeli monoton csökkenő módon:

$$\alpha \in \mathcal{P}(M_0)^{\mathcal{P}(M_0)} \quad (2)$$

$\alpha(M) \subseteq M$  tetszőleges  $M \in \mathcal{P}(M_0)$  esetén

Az  $\alpha_1 - \alpha_2 - \alpha_3 - \dots - \alpha_n$  diskurzus tehát egy  $M_0 - M_1 - M_2 - \dots - M_n$  modellsorozatként értelmezhető, ahol  $M_i = \alpha_i(M_{i-1})$  és  $M_i \subseteq M_{i-1}$ .

A `dynsem` programban a modell individuumainak  $U$  halmazát és a modellben használható predikátumokat külön adhatjuk meg. Először a diskurzusban szereplő individuumokat rögzítjük a `createModel` paranccsal, pl.: `createModel(beatles, [john, paul, ringo], [])`, ahol a `beatles` a diskurzusnak, illetve a diskurzus aktuális modelljének az azonosítója, a `[john, paul, ringo]` lista pedig a diskurzusban szereplő individuumok halmaza – ezek egyúttal névként is használhatók a diskurzusban. Ekkor egy olyan modell keletkezik, amely egyetlen lehetséges világból áll, mivel még semmilyen predikátumot nem vezettünk be a diskurzusba: `index(0): beatles=[john,paul,ringo]` (az `index` az egyes világokat azonosítja, a `beatles` pedig a modell  $U$  univerzumával egyezik meg minden világban, tkp. egy konstans egyargumentumú predikátum). Új predikátumot az `addNewPredicate` paranccsal vezethetünk be, pl.: `addNewPredicate(boy, 1)`, ahol a `boy` az új predikátum neve, a másik argumentum pedig a bevezetett predikátum argumentumszáma: jelen esetben a `boy` predikátum egyargumentumú lesz. Az új predikátum alapértelmezésben a legutóbb használt diskurzusban jelenik meg. Az új predikátum bevezetésével az aktuális modellben levő világok száma meg többszöröződik a korábban leírt módon, itt most a következő világok alkotják a *beatles* diskurzus modelljét:

```

index(7): beatles=[john,paul,ringo], boy=[ringo]
index(6): beatles=[john,paul,ringo], boy=[paul,ringo]
index(5): beatles=[john,paul,ringo], boy=[paul]
index(4): beatles=[john,paul,ringo], boy=[john,ringo]
index(3): beatles=[john,paul,ringo], boy=[john,paul,ringo]
index(2): beatles=[john,paul,ringo], boy=[john,paul]
index(1): beatles=[john,paul,ringo], boy=[john]
index(0): beatles=[john,paul,ringo], boy=[]

```

Ha a diskurzusban más predikátumokat is használunk, azokat is hasonlóan adjuk a modellhez. Az így felépített  $M_0$  modellre kell ezután a diskurzus  $\alpha$  kijelentéseit alkalmazni. A diskurzust a megnyilatkozások által kifejezett kijelentések konjunkciójaként értelmezve a `dynsem` programban ezt az egyargumentumú `and` paranccsal tehetjük meg, argumentumként pedig a soron következő  $\alpha$  megnyilatkozás kijelentéslogikai megfelelőjét használjuk, pl.: `and(boy(john))`. A *beatles* diskurzus aktuális modellje ennek hatására a következő világokból áll:

```

index(4): beatles=[john,paul,ringo], boy=[john,ringo]
index(3): beatles=[john,paul,ringo], boy=[john,paul,ringo]
index(2): beatles=[john,paul,ringo], boy=[john,paul]
index(1): beatles=[john,paul,ringo], boy=[john]

```

A megnyilatkozásnak megfelelő kijelentéslogikai formula a modellbe bevezetett neveken és predikátumokon kívül használható negáció, konjunkció, diszjunkció, kondicionális műveleteket, valamint univerzális és egzisztenciális operátorokat (megfelelő individuumváltozók használatával), ióta operátorokat és azonossági relációkat tartalmazhat.

A `dynsem` az eddig bemutatott lehetőségekkel funkcionalitásában megegyezik a (Szécsényi, 2015) tanulmányban ismertetett rendszerrel.

## 4 Dinamikus modellekkel végezhető műveletek

A program egyszerre több modellt/diskurzust is tud kezelni, amelyekre a modell létrehozásakor használt nevekkal lehet hivatkozni. Alapértelmezett esetben a legutóbb létrehozott modellbe lehet az `addNewPredicate` utasítással új predikátumot bevezetni, illetve az `and` utasítással a diskurzus egy megnyilatkozása által megfogalmazott új információt bevinni. Ha ettől eltérő modellel szeretnénk dolgozni, a `changeActualModel` egyargumentumú paranccsal válthatunk az aktuális modellek között.

Lehetőség van a modellek elmentésére és beolvasására, ekkor a modell jelenlegi állapotát, vagyis a modell nevét, az individuumok listáját, a deklarált predikátumokat, és a modellben elérhető lehetséges világokat egyaránt lemezre írhatjuk. Az egyes lehetséges világok a használható predikátumok interpretációját tartalmazza.

Új modellt nem csak a `createModel` utasítással lehet létrehozni, hanem két már meglévő modell kombinációjával is a `combineModels(a,b,new)` utasítással, ahol *a* és *b* a két már meglévő modell azonosítója, *new* pedig az új modell azonosítója. Két modell akkor kombinálható, ha a bennük bevezetett predikátumok különbözőek: nincs olyan predikátum, amely ugyanolyan argumentumszámmal szerepel mindkét

modellben. A kombinált modell diskurzusuniverzuma, azaz a modellben szereplő individuumok halmaza a kombinálandó modellek diskurzusuniverzumainak az uniója lesz, a kombinált modellben pedig minden olyan predikátum elérhető lesz, ami a kombinálandó modellek valamelyikében elérhető volt. A kombinált modell lehetséges világai a két modell világainak összes lehetséges egyedi kombinációjaként jönnek létre: ha az  $a$  modellben  $n$  lehetséges világ volt, a  $b$  modellben pedig  $m$ , az új modellben  $n \cdot m$  lehetséges világ lesz.

A modellek kombinálásának bemutatására példaként vegyünk két modellt,  $a$ -t és  $b$ -t. Az  $a$  modellben legyen két individuum:  $i$  és  $j$ , két predikátum:  $p_1$  és  $p_2$ , és a modellben két lehetséges világ legyen:

```
index(0): a=[i,j], p1=[i], p2=[j]
index(1): a=[i,j], p1=[], p2=[i,j]
```

A  $b$  modellbe a  $j$  és  $k$  individuumok és a  $p_3$  egyargumentumú predikátum legyen bevezetve három lehetséges világgal:

```
index(0): b=[j,k], p3=[]
index(1): b=[j,k], p3=[j]
index(2): b=[j,k], p3=[k]
```

A kettő kombinációjaként létrehozott  $new$  modellben hat lehetséges világ található:

```
index(0): new=[i,j,k], a=[i,j], b=[j,k], p1=[i], p2=[j], p3=[]
index(1): new=[i,j,k], a=[i,j], b=[j,k], p1=[i], p2=[j], p3=[j]
index(2): new=[i,j,k], a=[i,j], b=[j,k], p1=[i], p2=[j], p3=[k]
index(3): new=[i,j,k], a=[i,j], b=[j,k], p1=[], p2=[i,j], p3=[]
index(4): new=[i,j,k], a=[i,j], b=[j,k], p1=[], p2=[i,j], p3=[j]
index(5): new=[i,j,k], a=[i,j], b=[j,k], p1=[], p2=[i,j], p3=[k]
```

## 5 A modellépítés optimalizálása

A (2) képletben megadott összefüggésnek megfelelően egy  $k$  individuumot  $n$  egyargumentumú predikátumra és  $m$  kétargumentumú predikátumra épülő kezdeti modell  $2^{k \cdot n + k \cdot m}$  különböző világot tartalmaz. Ez már viszonylag kis  $k$ ,  $n$  és  $m$  értékek esetén is nagy mennyiség lesz: háromelemű univerzumnál 4 egyargumentumú és 2 kétargumentumú predikátummal  $2^{3 \cdot 4 + 3 \cdot 2} = 2^{30}$  lehetséges világ. Bármilyen kompakt is az egyes világok reprezentációja, az egyes kijelentések modellváltoztató képességének meghatározásánál az összes lehetséges világban meg kell határozni a kijelentés igazságértékét, ami kezelhetetlen feldolgozási időt eredményezhet.

Hogy a dinamikus diskurzusreprezentációs szemantikai modell mégis használható legyen, nem az összes, diskurzus során használt predikátumot és azok összes lehetséges interpretációit tartalmazó  $M_0$  modellből kell kiindulni, hanem egy olyan modellből, ami ténylegesen nem tartalmaz semmilyen információt a világról, csak azt, hogy mely individuumok találhatók meg benne: a `createModel` utasítással létrehozott üres, egyvilágos modellből. Az aktuális modellben a lehetséges világok számát egy predikátum bevezetésével pedig csak akkor kell megnövelni, amikor a diskurzusban az adott predikátum használatba kerül: a `boy` egyargumentumú predikátumot csak akkor deklaráljuk a modellben az `addNewPredicate` utasítással,

amikor a diskurzus egy kijelentése miatt szükséges annak használata. Ekkor további predikátumok bevezetése előtt, ami megnövelné a lehetséges világok számát, a megnyilatkozás által megfogalmazott új információ lecsökkenti azt. Például a három individuumot, John, Pault és Ringót tartalmazó diskurzusuniverzumban a *John fiú*, *Paul gitározik* és *Ringó ismeri John* diskurzusban ilyen optimalizálás nélkül és optimalizálással a következőképpen alakul a lehetséges világok száma:

optimalizálás nélkül	világok száma
<code>createModel(beatles, [john, paul, ringo], [])</code>	1
<code>addNewPredicate(boy, 1)</code>	8
<code>addNewPredicate(play_guitar, 1)</code>	64
<code>addNewPredicate(know, 2)</code>	32768
<code>and(boy(john))</code>	16384
<code>and(play_guitar(paul))</code>	8192
<code>and(know(ringo, john))</code>	4096
optimalizálással	világok száma
<code>createModel(beatles, [john, paul, ringo], [])</code>	1
<code>addNewPredicate(boy, 1)</code>	8
<code>and(boy(john))</code>	4
<code>addNewPredicate(play_guitar, 1)</code>	32
<code>and(play_guitar(paul))</code>	16
<code>addNewPredicate(know, 2)</code>	8192
<code>and(know(ringo, john))</code>	4096

Látható, hogy az aktuális modellben levő világok maximális száma az ilyen optimalizálással nagyságrendekkel kisebb, mint az optimalizálás nélküli modellekben.

## 6 Explicit információ és háttértudás: predikátumokra vonatkozó megszorítások

A természetes nyelvi diskurzusokban nem csak a megnyilatkozások által expliciten megfogalmazott állítások alakítják a modellt, hanem a diskurzus résztvevőinek közös háttértudása is. Ez a háttértudás a diskurzusban használt predikátumok által kifejezett fogalmak ontológiájaként értelmezhető.

A *younger* kétargumentumú predikátummal kifejezett *fiatalabb* reláció esetében például ilyen ontológiai információ az, hogy a reláció a diskurzusuniverzum individuumainak lineáris rendezését valósítja meg. A *same\_class* kétargumentumú predikátum által kifejezett *osztálytársa* reláció az individuumok halmazán értelmezett ekvivalenciareláció vagy osztályozás, vagyis a diskurzusuniverzum elemeinek diszkrét osztályozását végzi. A *male* és *female* egyargumentumú predikátumok is ilyen diszkrét osztályokként viselkednek, mintha a *same\_sex* kétargumentumú predikátum által létrehozott osztályokat fejeznék ki. A *male* és a *bachelor* egyargumentumú predikátumok által kifejezett individuumhalmazok között viszont más jellegű viszony van: csak az lehet *aggregény*, ami *hímnemű*.

Az ilyen jellegű háttértudásunkat, vagyis a fogalmak tulajdonságait és egymáshoz való viszonyukat a fogalmakat kifejező predikátumok bevezetése után fogalmazhatjuk

meg. Az `addNewPredicate` használatával a kiinduló  $M_i$  modell lehetséges világainak a számát megnöveljük, az  $M_{i+1}$  modellben már a korábbi modell lehetséges világai helyett már azok azon változatai szerepelnek, amelyek az új predikátum összes lehetséges interpretációját tartalmazzák. Ezután az `and` diskurzusépítő művelettel egyenként módosítjuk az aktuális modellt a bevezetett fogalom ontológiai megszorításaival, melynek során a lehetséges világok száma csökken.

A `dynsem` programban az ilyen speciális, ontológiai háttérinformációkat már a predikátum bevezetésénél alkalmazhatjuk a predikátum ontológiai típusának jelzésével, így az  $M_{i+1}$  modellben már eleve csak a csökkentett (=kisebb mértékben növelt) lehetséges világok jelennek meg.

## 6.1 Új lineáris rendezés bevezetése

A `younger` kétargumentumú predikátummal kifejezett *fiatalabb* reláció a diskurzusuniverzum individuumainak lineárist rendezését valósítja meg, vagyis tetszőleges  $x$  és  $y$  esetén az  $x$  *fiatalabb*  $y$ -nál,  $y$  *fiatalabb*  $x$ -nél és  $x$  *azonos*  $y$ -nal állítások közül pontosan az egyik teljesül. Ezekkel a háttérállításokkal ugyanúgy módosítani kell a modellt, mint a megnyilatkozásokkal kifejezett explicit állításokkal. Mivel azonban ez a háttértudás nem a diskurzus egyes megnyilatkozásaihoz kapcsolódik, hanem a használt fogalmakhoz, predikátumokhoz, ezért a predikátumok bevezetésével együtt érdemes hozzáadni a modellhez ezeket az információkat.

```
addNewPredicate(younger, 2).
and(all X: all Y: (younger(X,Y) -> ~younger(Y,X))).
and(all X: all Y: all Z: ((younger(X,Y) & younger(Y,Z)) ->
    younger(X,Z))).
and(all X: all Y: (younger(X,Y) | younger(Y,X) | X=Y)).
```

A példa első sora bevezeti az új kétargumentumú predikátumot, ezzel a predikátum bevezetése előtti  $M_i$  modell világainak a számát  $2^{n^n}$ -szeresére növeli  $n$  individuum esetén, a második sor a reláció aszimmetrikusságát és irreflexivitását deklarálja, a harmadik sor a tranzitivitást fejezi ki, az utolsó pedig lineárisá teszi a rendezést. Az utolsó három sor csökkenti a modellbeli világok számát, az  $M_i$  modell világ-számnak végül  $n!$ -szorosa lesz az  $M_{i+4}$  modellben az elérhető világok száma.

A `dynsem` programban egy lépésben is bevezethetünk új lineáris rendezést az `addNewOrdering` utasítással: `addNewOrdering(younger)`.

Az utasítás használatával közvetlenül vezetjük be a rendezést, anélkül, hogy közben  $2^{n^n}$ -szeresére nőne a lehetséges világok száma. Az utasítás használata során ugyanis nem a megszorítatlan relációból kiindulva hozzuk létre az új lehetséges világokat, hanem a diskurzusuniverzum individuumainak a permutációiból: először létrehozuk az individuumok lehetséges sorbarendezéseit ( $n!$ ), majd a korábbi modell összes lehetséges világát kibővítjük ezekkel a rendezésekkel, így minden korábbi világnak  $n!$  változata lesz az új predikátum bevezetése utáni modellben.

## 6.2 Új osztályozó bevezetése

A természetes nyelvi fogalmak között nem csak a lineáris rendezésként értelmezhetőek fordulnak elő gyakran, egy másik gyakori fogalomtípus az individuumok osztályozását szokta eredményezni. Egy iskola tanulóiról szóló diskurzusban például az *osztálytársa* fogalom, az állatokról szólóban a *kutya*, *macska*, *egér* ... fogalmak, vagy az emberek esetében a *férfi* és a *nő* fogalmak a diskurzusuniverzum individuumainak diszkrét, a diskurzusuniverzumot mégis teljesen lefedő részhalmazait, osztályait határozzák meg. Ezek az osztályozók a megnyilatkozások logikai reprezentációiban megjelenhetnek egy kétargumentumú predikátum által kifejezett ekvivalenciarelációként (pl.: *classmate*, *same\_species*, *same\_sex*), vagy egy ilyen ekvivalenciareláció által meghatározott ekvivalenciaosztályokat definiáló egyargumentumú predikátumok sorozataként (pl.: *dog*, *cat*, *mouse* ... vagy *male*, *female*).

A *dynsem* program eddig ismertetett lehetőségeit használva – lineáris rendezésnél leírtakhoz hasonlóan – az osztályozó predikátumokat is először megszorítatlan formában deklaráljuk, így a kétargumentumú predikátumok esetében  $2^{n \cdot n}$ -szeresére növelve a predikátum bevezetése előtti  $M_i$  modellben levő lehetséges világok számát, vagy az  $m$  darab diszjunkt osztályt meghatározó egyargumentumú predikátumoknál  $2^{n \cdot m}$ -szeresére. Ezt követi a bevezetett predikátumok megszorítása, hogy azok osztályozóként viselkedjenek: kétargumentumú predikátumnál az ekvivalenciarelációra jellemző megszorításokkal, egyargumentumú predikátumnál pedig az ekvivalenciaosztályok diszjunktóságának biztosításával. A megszorítások bevezetése után az eredetinek az  $n^m$ -szeresére csökken lehetséges világok száma, ahol  $m$  az ekvivalenciaosztályok száma.

A *dynsem* programban azonban közvetlenül egy lépésben, rögtön a szűkített lehetségesvilág-halmaz bevezetésével is be lehet vezetni ilyen osztályozót, egyszerre megadva az ekvivalenciarelációként értelmezett kétargumentumú predikátumot, és az ekvivalenciareláció által létrehozott ekvivalenciaosztályokat is. Az `addNewClassifier(same_sex, [male, female])` utasítással az aktuális modellt kibővíthetjük a *same\_sex* kétargumentumú predikátummal és a *male* és *female* egyargumentumú predikátumokkal, ahol a kétargumentumú predikátum olyan ekvivalenciarelációként fog funkcionálni, amely (ebben az esetben) két ekvivalenciaosztályt határoz meg, amelyekre a *male* és a *female* predikátumokkal hivatkozhatunk a későbbiekben. Az `addNewClassifier` második argumentumaként tetszőleges számú ekvivalenciaosztály nevét megadhatjuk listaként.

Az osztályozók által használt ekvivalenciaosztályok a WordNet koordinált *synsetjeinek* felelnek meg (Fellbaum 1998).

## 6.3 Az új predikátumok domain-megszorításai

A diskurzusba újonnan bevezetett predikátumoknak sokszor nem a teljes diskurzusuniverzum az értelmezési tartománya, hanem annak csak egy része. A predikátumok értelmezési tartománya, domainje néha lexikailag meghatározott, azaz más predikátumra vezethető vissza, például *agglegény* csak az lehet, aki egyúttal *férfi* is, a *kutya* tulajdonságú individuumok az *emlős* tulajdonságúaknak egy részhalmaza,

amelyek a *gerinces*-eknek egy részhalmaza stb. – ez a WordNet hiperonim/hiponim synset-relációjának felel meg. Más esetben viszont a diskurzus által leírt világ(ok) egyedi tulajdonsága az, hogy ott egy predikátum valamely argumentumának mi az értelmezési tartománya: az hogy egy szituációban kutyák kergetnek macskákat, nem a *kerget* kétargumentumú predikátum inherens lexikai tulajdonsága, hanem a szituációé. Azonban ha egy predikátumról van ilyen lexikai vagy egyedi világismeretből származó háttértudásunk, akkor azt is ki kell fejeznünk a dinamikus diskurzusreprezentációs modellben.

A domain-megszorításokat a predikátum bevezetése után fogalmazhatjuk meg. Például az `addNewPredicate(bachelor, 1)` használatával először bevezetjük a *bachelor* egyargumentumú predikátumot, ezzel  $2^n$ -szeresére növelve a lehetséges világok számát, majd az `and(all X: (bachelor(X) -> male(X))` utasítással leszűkíthetjük a predikátum domainjét a már korábban bevezetett *male* predikátumra, ezzel csökkentve a lehetséges világok számát. Vagy az `addNewPredicate(chase, 2)` segítségével bevezetjük a kétargumentumú predikátumot, és a predikátum egyes argumentumainak a domainjét leszűkítjük a *dog* és *cat* predikátumra az `and(all X: all Y: (chase(X, Y)) -> (dog(X) & cat(Y)))` utasítással. Így azonban először mindig a domain-megszorítás nélküli predikátumból indulunk ki, megnövelve a modellben a lehetséges világok számát, és csak utána csökkentjük le ezt a számot a domain-megszorításokkal.

A *dynsem* programban azonban már a predikátumbevezetéskor is megadhatjuk a predikátum argumentumainak a domainjét, ha az különbözik a diskurzusuniverzum individuumainak halmazától. A domain megadása történhet abszolút vagy relatív módon. Abszolút domainmegadásnál a diskurzusuniverzum individuumainak egy explicit részhalmozát adjuk meg domainként (a domainmegszorítás félkövérral kiemelve): `addNewPredicate(hungry, 1, [john, paul])`, amely hatására a *beatles* diskurzusba a *hungry* egyargumentumú predikátum úgy kerül bevezetésre, hogy csak a *john* és a *paul* individuumok lehetnek éhesek (de nem feltétlenül azok), a *ringo* nem. Ekkor a modell lehetséges világainak a száma nem nyolcszorosára nő ( $2^3$ ), hanem csak négyszeresére.

Relatív domain-megadásnál a predikátum bevezetése előtti  $M_i$  modell minden lehetséges világánál külön meghatározzuk a predikátum értelmezési tartományát. Egy szabad individuumváltozót tartalmazó logikai állítás segítségével először kiválasztjuk azokat az individuumokat a diskurzusuniverzumból, amelyek az adott világban igazgá teszik a nyitott állítást: az  $\alpha(x)$  kifejezés esetén a  $\{x \mid \alpha(x)\}$  individuumokat. Relatív domain-megadásnál a domain-megszorítást a `addNewPredicate(bachelor, 1, x: male(x))` alakban adhatjuk meg, ahol a *male(x)* a domaint meghatározó nyitott mondat, *x* pedig a lekötetlen változó.

Többargumentumú predikátumok esetében egyszerre az összes argumentumra is lehet domain-megszorítást tenni, vagy argumentumonként külön-külön is, pl: `addNewPredicate(chase, 2, [x: dog(x), y: cat(y)])`. Rendezések és osztályozók bevezetésénél is lehet domain-megszorítást tenni, de a relációk jellegéből kifolyólag az argumentumokra itt egyetlen megszorítás értelmezhető csak: `addNewClassifier(same_species, [dog, cat, mouse], x: mammal(x))`.



## 7 Összefoglalás

A tanulmány bemutatta a *dynsem* programot, amellyel a diskurzusok jelentését olyan modellként adja meg ami a diskurzus által meghatározott lehetséges világok halmazát tartalmazza. A diskurzus megnyilatkozásainak a jelentése a modell változását eredményezi. A programban a megnyilatkozások mint elsőrendű logikai formulák adhatók meg.

Hogy a lehetséges világok száma ne legyen kezelhetetlenül nagy, a megnyilatkozásokban használható predikátumokra a programban különböző megszorításokat tehetünk, ami a természetes nyelv fogalmainak ontológiáján alapul.

## Hivatkozások

- Blackburn, P., Bos, J.: Representation and Inference for Natural Language. A First Course in Computational Semantics. (CSLI Studies in Computational Linguistics 1) CSLI Publications, Stanford (2005)
- Dyekiss, E. G.: Hivatkozás nem létező diskurzus-referensekre? In: Gécseg, Zs. (szerk.) LingDok10: Nyelvész-doktoranduszok dolgozatai. pp. 9–28. SZTE Nyelvtudományi Doktori Iskola, Szeged (2011)
- Dyekiss, E. G.,: Nem teljesen megbízható információkat hordozó dialógusok értelmezése véges állapotú súlyozott automaták segítségével. In: Gécseg, Zs. (szerk.) LingDok12. Nyelvész-doktoranduszok dolgozatai. pp. 51–71. SZTE Nyelvtudományi Doktori Iskola, Szeged (2013)
- Fellbaum, C. (szerk.), WordNet: An Electronic Lexical Database. (Language, Speech, and Communication). The MIT Press, Cambridge (1998)
- Russell, S. J., Norvig P.: Artificial Intelligence. A Modern Approach. 3<sup>rd</sup> edition. Prentice Hall (2010)
- Szécsényi, T. A lehetőség-halmazok meghatározása az inkvizitív szemantikában. In: Tanács A., Vincze, V. (szerk.) IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 205–212. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2013)
- Szécsényi, T.: Van-e a nyuszinak sapkája? A számítógépes jelentésábrázolás problémái. In: Szeverényi, S., Szécsényi, T. (szerk.) Érdekes nyelvészet. pp. 127–144. JATEPress. Szeged (2015)



# Neme?

## Nemek közötti nyelvhasználati különbségek a Facebook bejegyzésekben

Kelemen Sára Bernadett<sup>1</sup>, Yang Zijian Győző<sup>2</sup>, Kmetty Zoltán<sup>3,4</sup>

<sup>1</sup> Clementine

1115 Budapest, Bartók Béla út 105-113 I/B.  
skelemen@clementine.hu

<sup>2</sup> Nyelvtudományi Kutatóközpont,  
1068 Budapest, Benczúr u. 33.

yang.zijian.gyozo@nytud.hu

<sup>3</sup> Eötvös Loránd Tudományegyetem, Társadalomtudományi Kar, Szociológia Tanszék,  
1117 Budapest, Pázmány Péter sétány 1/C.

kmetty.zoltan@tatk.elte.hu

<sup>4</sup> Társadalomtudományi Kutatóközpont, CSS-Recens kutatócsoport  
1097 Budapest, Tóth Kálmán utca 4.

**Kivonat:** Online szövegelemzések esetében gyakran elhangzó kritika, hogy a szövegek íróiról nincs háttérinformációnk. Ha nyelvhasználatuk alapján képesek lennénk nemet becsülni, értékes plusz tudást szerezhethetnénk. Jelen kutatásunk célja 110 nő és 37 férfi magyar nyelvű Facebook posztjainak felhasználásával annak a kérdésnek a megválaszolása volt, hogy az alkalmazott modellek közül melyik az, amelyik a legsikeresebben külön tudja választani a nemeket nyelvhasználatuk alapján. Ennek keretén belül nem csak több, hagyományosnak tekinthető felügyelt gépi tanulási módszert vizsgáltunk meg, hanem egy transzformer modelt is finomhangoltunk szövegosztályozásra. A modellek teljesítményét a teszt-halmazon összetett mutatók segítségével értékeltük.

**Kulcsszavak:** közösségi média, Facebook, nyelvhasználat, nem, felügyelt gépi tanulás, szövegosztályozás, huBERT

## 1 Bevezetés

A szerzők tetszőleges demográfiai változó mentén való kategorizálása gyakori klasszifikációs probléma napjainkban. A nem, életkor vagy akár nyelvhasználat szerinti osztályozás az üzleti alkalmazásokban segítheti a fogyasztók hatékonyabb megismerését.

Az internet térhódításával, valamint a digitális forradalomnak köszönhetően egyre nagyobb mennyiségű elektronikus szöveg válik hozzáférhetővé. A közösségi oldalak, mint például Facebook vagy Twitter, felhasználói által hagyott „digitális lábnyomok”<sup>1</sup>

---

<sup>1</sup> A 20. század végétől kezdő, számítógépek és a digitalizálás által kiváltott áttörést értjük digitális forradalom alatt. A kifejezés a számítástechnikai és távközlési eszközök, a számítógép és a telefon elterjedésével járó hatások leírásával fejezhető ki. (Karvalics, 2012)

elemzése egyre népszerűbbé válik, hiszen relatív egyszerűen elérhető, hatalmas adatmennyiségről van szó. A közösségi média oldalakról elérhető strukturálatlan adatok elemzése azért is „kifizetődő”, mert társadalmunk nagy része megtalálható ezeken a platformokon.

A felhasználók posztjaiból kiinduló, az ő nyelvhasználati szokásaikat kutató, klaszifikációt alkalmazó cikkek száma is megnövekedett. Ezekben rendszerint Twitter-ről, vagy különböző blogokról származó szövegeket elemeznék (Fosch-Villaronga és mtsai, 2021; Vashisth és mtsai, 2020; Zhang és Zhang 2010), Facebook-on megjelenő szöveges posztokat nagyon ritkán elemeznék ilyen szempontból, elsősorban az adathozzáférés nehézségei miatt. Kutatásunkban azt vizsgáltuk vajon mennyire alkalmasak a Facebookról származó bejegyzések ennek a témakörnek a tanulmányozására. A fellelhető szakirodalmak elsősorban angol nyelvűek, és jelentős részük angol vagy más, magyartól eltérő, idegen nyelvű posztokon keresztül vizsgálja a felhasználók nyelvhasználati szokásait. A netnyelvről, valamint arról, hogy a fiatalok csevegéssel töltött ideje, hogyan befolyásolja élőbeszédüket számos értekezés elérhető magyar nyelven is. Előbbiek azonban inkább csak történeti összefoglalók, sokszor mindenféle saját kutatást nélkülözve. Utóbbiak pedig többnyire valamilyen kvantitatív interjú megkérdezésre és különböző iskolai feladatként írt fogalmazások analizálásra korlátozódnak. Az utóbbi években azonban egyre inkább találkozhatunk beszéltnyelvi leiratozásokra alapozott analízisekkel (Vincze és mtsai, 2021).

## 2 Kapcsolódó munkák

Azzal, hogy miben különbözik a nemek közötti nyelvhasználat, a kutatók már jóval azelőtt elkezdtek foglalkozni, minthogy a közösségi médiák platformjai megjelentek volna. Később, miután az interneten egyre inkább elszaporodtak a különböző blogok, megjelentek a közösségi média platformjai, folyamatosan nőtt az igény és az érdeklődés az előbb említett felületekről származó hatalmas adatmennyiség felhasználhatóságának vizsgálatára a témában. A leggyakoribbak a különféle blogbejegyzésekből kiinduló kutatások (Argamon és mtsai, 2003; Schler és mtsai, 2006; Zhang és Zhang 2010), de elterjedtek a Twitter-ről származó tweet elemzések (Markov és mtsai, 2017; Aragón és López, 2018; Vashisth és mtsai, 2020) is. Facebook posztokra alapozott vizsgálatokkal, sokkal ritkábban<sup>2</sup>, de szintén találkozhatunk (Rangel és Rosso, 2013; Sap és mtsai, 2014).

Newman és munkatársai 2008-as tanulmányukban összegyűjtötték, hogy milyen, gyakran ellentmondásos eredmények születtek a korábbi kutatások során. Kihangsúlyozták, hogy a szavakra alapozott szöveganalitikák természetüknél fogva nem képesek megragadni azt a kontextust, amelyben a szavakat használják. A nemek közötti különbségek értelmezése árnyalt ügy, a társadalmi célok, situációs igények és szocializáció komplex kombinációja. A tanulmányban saját kutatásukat is ismertetik. Elemzéseikben kicsi, de szisztematikus különbségeket találtak női és férfi nyelvhasználat között. Ezt négy aspektusból vizsgálták: mikor, hol, miért és hogyan választható szét a

<sup>2</sup> A Facebook esetében az adathozzáférés sokkal nehezebb, mint a Twitter vagy egyes blogok esetében, különösen amióta a Facebook elzárta az API hozzáférést a kutatók elől. Elsősorban ezért a Twitter az elsődleges platformja a nyelvhasználati kutatásoknak.

nyelvhasznált mind abban a tekintetben, hogy mit mondanak, mind abban, hogy hogyan fejezik ki magukat a nők és férfiak. Esetükben a hangsúly funkcionális szavakon volt. Eredményeik segítségével arra mutattak rá, hogy a szó-számolási stratégiák életképesek, nagyhatékonyságú alternatívái az emberi kódokon alapuló nyelvi elemzésnek (Newman és mtsai, 2008).

Huffaker és Calvert 2005-ös írásukban tinédzserek webblogjain keresztül vizsgálták, hogy hogyan fejezik ki a serdülők magukat nyelviileg, milyen érzelmi kódokat használnak. A szerzőpáros nem talált nemi különbségeket a hangulatjelek használatának gyakoriságában. Sőt meglepő módon azt tapasztalták, hogy azok közt, akik használnak emotikonokat több a fiú. Nem tapasztaltak több agressziót a fiúk esetében, vagy nagyobb passzivitást a lányoknál. A korábbi eredményekkel ellentétben nem találtak a nemek között különbséget törődés, illetve együttműködés terén. Arra az eredményre jutottak, hogy a fiúk aktívabb, határozottabb és rugalmatlanabb nyelvet használtak. A lányok ellenben nem használtak passzívabb, kooperatívabb, vagy alkalmazkodóbb nyelvet. E mögött meghúzódó lehetséges ok szerintük, hogy a nyelv és az interneten történő szociális interakciók változnak, talán pont amiatt, mivel a résztvevők is változnak.

Huffaker és szerzőtársa egy tartalomelemző szoftvercsomag segítségével értékelték ki a dokumentumokat a szavak száma, tartalomtípus és nyelvi hangnem szempontjából. A használt nyelv értékelését is a tartalomelemző szoftverre bízta, amely figyelembe vette a nyelvi kontextust, valamint a szavak gyakoriságát. A blogok összehasonlítására Khi-négyzet próbákat és független T-próbákat futtattak (Huffaker és Calvert, 2015).

Rangel és Rosso (2013) spanyol anyanyelvű emberek nyelvhasználatát vizsgálta, hogy milyen grammatikai kategóriákat használnak Facebook posztjaikban. Azokra a kognitív tulajdonságokra összpontosítottak, amik nem és kor szerint különbözővé tesznek minket. Nemekre lebontva eredményeik alapján a férfiak több előszót használnak, talán amiatt, hogy megpróbálják a környezetükben lévő dolgokat hierarchikusan besorolni. A nők ezzel szemben több determinánst, névmást és közbeiktatást alkalmaznak, valószínűleg amiatt, mert több érdeklődéssel vannak a társas kapcsolatok iránt.

Adataik nem szerint igen, de kor szerint nem voltak kiegyensúlyozottak. Szupport Vektor Gép osztályozási eljárást alkalmaztak. Végeredményben pedig arra a megállapításra jutottak, hogy az általuk alkalmazott stilisztikai dimenziók jobban teljesítenek a kor, mint a nem azonosítására, esetleg annak köszönhetően, hogy az írástílust inkább a szerző korától és nem a nemétől függ (Rangel és Rosso, 2013).

Sap és munkatársai (2014) abból indultak ki, hogy a férfiak és nők jelentősen különböznek érdeklődési és munkabéli preferenciáikban. Az életkor előrehaladtával a személyiség fokozatosan alakul. Emellett a közösségi média nyelve a kor és nem függvényében változik. Facebook posztokat elemeztek, súlyozott szótárt alkalmaztak, amelyet a lineáris többváltozós regressziós és osztályozási modellek együtthatóinak felhasználásával hoztak létre. A nyelv kollinearitás miatt többváltozós lexikon fejlesztési megközelítést alkalmaztak, mely figyelembe veszi a kovarianciát. A nem előrejelzéséhez Szupport Vektor Gép osztályozó eljárást használtak. A tanító adathalmaz felhasználásával több modellt is teszteltek a legjobb modell megtalálásának érdekében. Mind a kor, mind a nem esetében az általuk elért pontosság lényegesen magasabb, mint az alapérték. Azonban azt tapasztalták, hogy minél kevesebb bejegyzés áll rendelkezésre egy-egy felhasználótól, annál kevésbé pontos a nem és az életkor előrejelzése (Sap és mtsai, 2014).

### 3 Alkalmazott módszerek bemutatása

Többféle osztályozó teljesítményét vizsgáltuk kutatásunk során: Szupport Vektor Gép (Support Vector Machine, továbbiakban SZVG), Bináris Logisztikus Regresszió (továbbiakban LR), Véletlen Erdő (Random Forest, továbbiakban VE), Naiv Bayes (továbbiakban NB) és transzformer modellek. A különféle gépi tanulási modellek tesztelése segít, hogy kiderítsük melyik illeszkedik jobban az adatokra, melyikkel tudjuk a legmegfelelőbbben megfogni a klasszifikált változó értékeit reprezentáló pontok és osztálycímkékük közötti kapcsolatot. Emellett annak érdekében, hogy minél pontosabb klasszifikációt kapjunk érdemes a modellek és hiperparamétereik kombinációjával kísérletezni.

A hagyományosnak tekinthető modellek mellett kíváncsiak voltunk arra is, hogy egy újabb, manapság nagy népszerűségnek örvendő mélyneurális hálózaton alapuló transzformer modell milyen eredményeket produkál ugyanazon az adatbázison. Ebből kifolyólag erre a feladatra egy magyar nyelvű transzformer modellt választottuk, a huBERT modellt.

**SZVG.** A Szupport Vektor Gép szövegbányászati feladatok során (is) nagy népszerűségnek örvendő modell stabil teljesítmény nyújtása miatt. A nemek osztályozásakor képes valamilyen fajta választ adni a címkéket meghatározó kategória sajátosságaira. Azokban az esetekben hatékony, mikor a dimenziók száma nagyobb, mint a minták száma. Nagy adathalmazok esetén a tanításhoz szükséges idő nagysága miatt nem teljesít túl jól, valamint a zajjal rendelkező adatkészletek is problémát jelentenek számára a target osztályok átfedése miatt (Rakovics, 2016).

**LR.** A Logisztikus Regresszió olyan klasszifikációs eljárás, mely során előre definiált, egymást kölcsönösen kizáró csoportok egyikébe soroljuk a megfigyeléseket a magyarázó változó(k)ból kinyert információ alapján. Bináris Logisztikus Regresszió esetében a magyarázó változóval/változókkal annak a bekövetkezési esélyét szeretnénk prediktálni, hogy a kimenet 1 lesz (Kovács, 2014).

**VE.** A Véletlen Erdő előnyei közé sorolható, hogy pontos klasszifikációra képes, nagy adatok esetén is gyorsan lefut, valamint, hogy becsléseket ad arra, melyik változók bírnak nagy jelentőséggel. Nagy mértékben a döntési fák elvén alapul, a zsákolás (bagging) egy speciális alelete, melynél az egyedi döntési fák kvázi korrelálatlanok. Több gyenge osztályozó átlagos teljesítménye alapján klasszifikál, több kisebb méretű döntési fát is épít. Addig folytatja a fák építését, míg az előre rögzített mélységet el nem éri. Az egyes erdők hatékonysága függ a generált fák számától és minőségétől, valamint a fák közötti korrelációtól (ha nő, az eredmény romlik) (James és mtsai, 2013).

**NB.** A Naiv Bayes féle osztályozók előnye, hogy robusztusok izolált zajos pontokra, illetve az irreleváns attribútumokra (feature). Az osztályozás lényege a Bayes tételre alapszik. Az alapvető feltevés, hogy minden attribútum függetlenül és egyenlően (minden tulajdonság azonos súlyt vagy fontosságot kap) járul hozzá az eredményhez. A szükséges paraméterek becsléséhez kisméretű tanító halmaz is elegendő, valamint a kifinomultabb osztályozókhöz képest sokkal gyorsabban képesek eredményt adni, különösen hasznos nagyon nagy adathalmazok esetén (Tan és mtsai, 2006).

**GNB.** Gauss Naiv Bayes esetén feltételezzük, hogy az egyes attribútumokhoz tartozó folyamatos értékek egy Gauss-eloszlás szerint oszlanak el, abból kifolyólag, hogy a prediktorok folytonos értéket vesznek fel, és nem diszkrét.

**BNB.** Bernoulli Naiv Bayes során az osztályváltozó megjöléséhez használt paraméterek bináris változók. Kifejezetten bünteti az olyan attribútum elő nem fordulását, amely egy osztály indikátora.

**KNB.** A Kiegészítő Naiv Bayes (Complement Naiv Bayes) különösen alkalmas kiegyensúlyozatlan adathalmazok esetében. Az egyes osztályok kiegészítéséből származó statisztikákat használja a modell súlyok kiszámításához. Egy dokumentumot ahhoz az osztályhoz rendeli, amelynek a legszegényebb a bővítmény egyezése (Zhang, 2004).

**huBERT.** Egy magyar „BERT base” (Devlin és mtsai, 2019) modell, mely a Webcorpus 2.0 korpuszon (Nemeskey, 2020) lett betanítva, ami közel 9 milliárd tokennel rendelkezik. A BERT egy csak enkóderrel rendelkező transzformer architektúrájú neurális nyelvi modell. Jelenleg a huBERT a legjobb teljesítményű BERT modell magyar nyelvre. Több feladatban „state of the art” eredményt ér el, mint például a maszkolt nyelvi modellezésben, névelem-felismerésben vagy főnévi csoport felismerésben.

## 4 Kutatás

Az elvégzett elemzés célja volt választ kapni azokra a kérdésekre, hogy el lehet-e különíteni a nemeket nyelvhasználatuk alapján, és ha igen, akkor hogyan, milyen mértékben lehet megkülönböztetni őket? Többféle klasszifikációs modell futtatásának segítségével szerettük volna kideríteni, hogy melyik algoritmussal lehet hatékonyabban, helyesen besorolni a saját nemüknek megfelelő osztályba a férfiakat és a nőket.

### 4.1 Felhasznált adatok

A kutatáshoz nagy időintervallumra kiterjedő Facebook adatokat használtunk fel. A Magyarországon zajló 2019-es kutatás<sup>3</sup> során a résztvevőket egy önkitöltős kérdőív kitöltése után megkérték, hogy töltsék le Facebook adataikat. A minta nem valószínűségi, kvótás (nem és kor), kényelmi minta volt. Ez korlátozza a kutatás megállapításainak általánosíthatóságát, ugyanis a minta fiatalabb volt, mint a magyarországi Facebook-felhasználók átlagos életkora (35-37 év), valamint nagyobb volt a női résztvevők aránya (75%).

Egy fontos kritériuma volt a mintába kerülésnek, hogy adott résztvevőnek rendszeres Facebook felhasználónak kellett lennie. A rendszerességet ebben az esetben úgy definiálták, hogy legalább heti gyakorisággal használja a platformot. Az adatok a résztvevők Facebook használatának teljes időtartamát lefedik, azaz a regisztrációjuktól kezdve a letöltés pillanatáig a felületen végrehajtott összes tevékenységüket, beleértve azokat is, amiket később eltávolítottak. Néhány esetben ez több mint tíz évnyi adatot jelent. A letöltött adatprofilok nem tartalmaznak privát üzeneteket, sem keresési előzményeket vagy audiovizuális tartalmakat (fotók és videók). Valamint kizárták az olyan

---

<sup>3</sup> A kutatás az NKFI-től nyert támogatást a Fialat Kutató Témapályázaton. A kutatás azonosítója: FK: 128981

tevékenységeket - mint például a Marketplace - melyeket a felhasználók ritkán, vagy szinte soha nem használtak (Breuer és mtsai, 2021).

Az elemzett posztok 20 típusba sorolhatóak. Gyakoriságukat tekintve az első három legnagyobb mértékben előforduló cselekedet típusok a posztok/bejegyzések valakinek az idővonalára, az állapotfrissítések és a csoportba írt bejegyzések voltak. De ezek mellett még sűrűn előforduló posztok voltak a fotók, illetve videók feltöltése, (csak a feltöltött tartalomhoz írt szöveg, egyes esetekben a videó/kép url-je) különböző linkek megosztása, saját idővonalon bejegyzések közzététele. Ritkábban, de az adatbázisban találhatunk többek között emlékmegosztásokat, vagy akár „geotag”-eket (hol járt a felhasználó) is.

A mintát 110 nő és 37 férfi teszi ki. Az átlagos életkor 30 év, a legfiatalabb egyén 18 éves, a legidősebb pedig 71. Az adatbázisban 149.471 poszt szerepel, egy sor egy bejegyzésnek felel meg. Minden poszt szövege mellett szerepel az egyoldali kulccsal anonimizált szerző. Mindegyik résztvevőt egy egyéni azonosítóval láttak el. Ebből a poszt íróját nem lehet visszakövetni, de segítségével tudjuk, hogy melyik posztok származnak ugyanattól a felhasználótól. Emellett fel van tüntetve a felhasználó neme, és születési éve. Valamint szerepel a bejegyzések típuskódja is, hogy hova lett bejegyezve a poszt (saját idővonalra, egy ismerőse idővonalára, csoportba, eseményhez). Ezek mellett a posztok időbélyeggel (timestamp) is el lettek látva, ami alapján másodpercre pontosan meg lehet mondani, hogy mikor született a bejegyzés. A posztok átlagosan 118 szóból állnak. A leghosszabb poszt 13.610 mondatalkotó elemből áll. A nők jelentősebb része 50 és 90 közötti szóból álló bejegyzéseket tett közzé, velük szemben a férfiak nagyobb része 45 és 125 közötti szót használt. A leghosszabb bejegyzések valamilyen csoportba történő posztoláskor születtek.

	Eredeti		Tisztított	
	Tanító	Teszt	Tanító	Teszt
Poszt	38.115	19.334	30.376	16.144
Token	610.574	332.651	523.058	311.243
Type	146.348	91.111	83.305	51.026
Poszt átlagos hossza	16,01 medián: 6	17,21 medián: 8	17,22 medián: 7	19,27 medián: 10
Osztálycímkék	Nő: 26.6204 Férfi: 11.511	Nő: 14.239 Férfi: 5.095	Nő: 21.000 Férfi: 9,376	Nő: 11.365 Férfi: 4.788

**1. táblázat.** A korpuszra jellemző nyelvtechnológiai értékek.

A posztokban a négy leggyakoribb szófaj a főnevek, melléknevek, igék és határozószók voltak. Nemek szerint bontva nincs jelentős különbség, közel azonos arányban használják mind a négy (és az összes többi) szófajt, bár a nők minden esetben egy-két százalékponttal ugyan, de magasabb használati arányt mutattak, mint a férfiak.

Az elemzett adatbázisban a személyes névmások használata igen gyakori volt. Az egyes szám második és harmadik személyt kifejező „te” és „ő” szavak a 15 leggyakoribb kifejezés közt szerepeltek mind a két nem esetében. Az E/3-t kifejező személyes névmás használata férfiak esetében magasabb volt, és az „ők” személyes névmást is –



bár nagyon elhanyagolható mértékben – de ők alkalmazták többször. Az egyes szám első személyt kifejező „én” szó gyakrabban szerepelt a nők által írt bejegyzésekben, ahogy a „te” és „mi” kifejezések is.

## 4.2 Adatelőkészítési eljárások

Első lépésként megtisztítottuk a korpuszt minden olyan tartalmi elemtől, melyeknek a szöveg mondanivalójára nézve nincs hozzáadott értéke, nem erősítik a műveletek eredményességét.

Az egyes típus- poszt/bejegyzés valakinek az idővonalára - elhanyagolható részt tekintve csupán születésnapjára és névnapi köszöntések különböző formáit, valamint egyéb ünnepekkel kapcsolatos jókívánságokat tartalmazott. Mivel úgy véltük a számtalan féle, gyakran ismétlődő gratulációk magához a klasszifikáció folyamatához nem tesznek hozzá semmilyen plusz értéket, ezért következő lépésként ezt a típust eltávolítottuk az adatbázisból 61.142 sorosra csökkentve ezáltal a korpuszt.

Ezt követően töröltünk minden linket, és e-mail címet a posztokból. Az adatbázis ekkor 57.479 sort számlált. Abból kifolyólag, hogy születésnapjára köszöntések (például saját üzenőfalán tette közzé adott egyén a születésnapjára felkötést, és az érintett embert csak megjelölte a posztban, vagy saját üzenőfalán megköszönte azoknak, aki gondoltak rá születésnapján), valamint egyéb jókívánságok, mint “boldog karácsonyt”, “kellemes ünnepeket”, “sikeres új évet”, “boldog húsvétot” nem csak azok között a posztok között voltak megtalálhatóak, amiket valaki másnak címeztek, hanem a mintában szereplők saját üzenőfalán is rengeteg közzé lett téve, ezért ezeket a többi bennmaradt típusból is kiszűrtük, ami 3.364 sornyi csökkenést eredményezett az adathalmazon.

A posztokban szereplő személyek anonimizálásakor a neveket egy „@”-jelet követő szám és betűkombinációval helyettesítették. Mivel plusz információval ezek az egységek sem szolgáltak, ezért szintén eltávolítottuk őket.

Mind az emotikonok, mind az emoji használata információt tartalmaz(hat) a szerzőről, ezért nem töröltük őket. Azonban a modellek futtatásakor a speciális karakterek problémát okozhatnak, ezért mind az emotikonokat, mind az emojiakat átkódoltuk, egy-egy egyedi névvel láttuk el őket.

Orosz György spaCy fejlesztését<sup>4</sup> felhasználva, ami kifejezetten a magyar nyelvű szövegeken való alkalmazásra lett készítve, 46.514 sornyi szöveget tokenizáltunk, majd lemmatizáltunk. Ezt követően „Part of Speech” (Pos) tag-gel is elláttuk őket, azaz szó-faj szerint azonosítottuk azokat.

A tiltólistás szavak eltávolítására a Python Natural Language Toolkit (NLTK)<sup>5</sup> nevű csomagját választottuk, azon okból kifolyólag, hogy már magyar nyelvű stopszólistával is rendelkezik. A módszer nem kezeli a szavak sorrendjét, csupán a gyakoriságát. Nem ismeri fel a hétköznapi nyelvhasználatot sem, így a helyesírási hibákat, elgépeléseket nem tudja kezelni, ezek problémát jelenthetnek. A sokat használt, illetve nyelvtani funkciót betöltő szavak törlése javíthatja az elemzésünk végeredményét, hiszen a leggyakrabban használt szavak vizsgálatakor így nem az „a”, „az” vagy épp az „egy” szavakat fogjuk visszakapni. Feltételezhetőleg információvesztéssel járt volna minden

<sup>4</sup> <https://github.com/oroszgy/spacy-hungarian-models>

<sup>5</sup> <https://www.nltk.org/>

stopszó eltávolítása, amit az NLTK stopszó listája tartalmaz, hiszen lehet, hogy a személyes névmások használata is olyan sajátosságot jelent, amely alapján el lehet különíteni a férfiakat és nőket nyelvhasználatuk alapján. Ezért a fentebb említett listát módosítva alkalmaztuk, és csak a kötőszókat, névelőket, határozószókat, valamint néhány olyan általunk megadott szót távolítottunk el, melyekről a leggyakrabban használt unigramok vizsgálatát követően kiderült, hogy a posztokban gyakran alkalmazták, de érdemi információt nem hordoznak, nem adnak többletinformációt a kutatáshoz (pl.: debrecen, delon).

### 4.3 Vektorizálás

Vektorizálás során a tanító halmazból kinyert  $n$ -grammok határozzák meg a magyarázó változókat, a későbbiekben teszteléshez és validáláshoz is ezeket kell kinyerni a korpuszból. Emiatt még vektorizálás előtt felosztottuk az adathalmazt tanító és tesztalmazra.

A 46.514 bejegyzésből álló adatbázist véletlenszerűen  $\frac{2}{3}$  és  $\frac{1}{3}$  arányban osztottuk fel. A modellek futtatásakor azonban ez túl soknak bizonyult egyes algoritmusok esetében (kifejezetten SZVG, de a Logisztikus Regresszió is lassan futott le), ezért, hogy ne ennyi emberről (egy sor egy egyének feleltethető meg, ha nem vesszük figyelembe, hogy egy id-hoz hány darab poszt tartozik) kelljen eldöntenie a modelleknek, hogy női vagy férfi szerzőről van szó, a bejegyzéseket összevontuk. Későbbiekben az adatbázis egy sorában az összes, adott azonosítóval rendelkező személy által írt poszt egyben, egy cellában szerepelt. A bejegyzések összevonását követően az adatbázis már csak 147 sort számlált. Így végül a tanító halmazba 98, míg a tesztalmazba 49 egyén került.

A futtatott modellek a TF-IDF vektorizálást követően produkálták a legjobb értéket. A modellek a legjobb előre jelző erővel abban az esetben bírtak, mikor unigramokat és bigramokat vizsgáltunk, az 5 és afeletti előfordulási gyakoriságú és a dokumentumok 70%-nál kevesebben előforduló szavakra korlátoztuk a szókincs összeállításakor felhasznált kifejezéseket, valamint a kifejezésgyakoriság szerint rendezett legfelső 500 attribútumot vettük figyelembe.

### 4.4 A hagyományos modellek hiperparamétereinek hangolása

Miután kiválasztottuk, hogy mely paramétereket fogjuk hangolni, definiáltunk a lehetséges értékekből egy rácsot és „Randomized Search”-öt hajtottunk végre ötszörös keresztvalidálással, mivel így az egyes hiperparaméterek szélesebb értéktartományát tudtuk lefedni nagy végrehajtási idő nélkül. Miután megkaptuk a legjobb hiperparaméterekkel rendelkező modellt - úgy, hogy leszűkítettük az egyes tartományokat - „Grid Search”-öt végeztünk szintén ötszörös keresztvalidálással, kifejezetten meghatározva a kipróbálandó beállítások minden kombinációját, hogy a hiperparaméter térben megtaláljuk a legjobban teljesítő kombinációt.

Ezt követően, mikor megtaláltuk a legjobb kombinációját a hiperparamétereknek, elvégeztük a hiperparaméter hangolást a tanító adatokkal, és ráillesztettük a modellt a tanítóadatokra, értékeltük a teljesítményét a tesztalmazon. Ha az osztályeloszlás kiegyensúlyozatlan, az abszolút pontosság (accuracy) rossz választásnak számít, mivel

minden osztályt egyforma fontosságúként kezel, magas pontszámot ad azoknak a modelleknek, amelyek csak a leggyakoribb osztályt jósolják. Emiatt összetett mutatókat választottunk az abszolút pontosság mellett a modellek teljesítményének értékelésére, mint fedés (recall), relatív pontosság (precision), f-mérték (f1-score).

#### 4.5 huBERT finomhangolása

Kutatásunk során finomhangoltuk a huBERT nyelvi modellt szövegosztályozási feladatra. A tanításhoz a Huggingface által közzétett „transformers text classification” könyvtárat<sup>6</sup> használtuk. A finomhangolást az alábbi módosított hiperparaméterekkel végeztük: maximum bemeneti hossz: 128; batch méret: 8 / GPU (4 darab GeForce GTX 1080 Ti); tanulási ráta: 2e-5; epoch: 10.

Kétféle modellt tanítottunk. Eredetiként arra a modellre utaltunk, mely esetén az eredeti adatbázissal dolgoztunk, amelyből eltávolítottuk az egyes típusú posztokat. A bent hagyott posztokból nem lettek kiszűrve a felkőszöntések. A bejegyzésekből nem töröltük a linkeket, e-mail címeket, maszkolásokat, sem a speciális karaktereket. Ez a szöveg nem lett tokenizálva. Tisztítottként neveztük el azt a modellt, amely a teljesen megtisztított 46.514 sorból álló adatbázisra lett futtatva.

Mindkét modellt a posztok szintjén tanítottuk és 60 lépésenként készítettünk mentési pontokat. Minden egyes mentéskor végeztünk egy posztszintű kiértékelést, majd a végén kiválasztottuk a legjobb eredményt (huBERT tisztított: 76,01%; huBERT eredeti: 76,45%) elérő mentési pontot. Ezután a kiválasztott mentési pontok segítségével elvégeztük a posztszintű prediktálást, majd összevontuk őket emberszintre. Az emberszintű összevonáskor megszámloltuk az adott emberhez tartozó prediktált osztálycímkeket és amelyikből több volt, az lett az adott emberhez hozzárendelt végső osztálycímke. Azonban azt vettük észre, hogy ezen modellek alkalmazásakor, az emberszintű összevonás után, eredményként azt kaptuk, hogy mindegyik ember nő lett. Ezért az összevonás szempontjából, ezek a modellek nem használhatóak számunkra. Hogy megtaláljuk az összevonás tekintetében a legjobban teljesítő modellt, minden egyes elmentett modellre elvégeztük az emberszintű kiértékelést is. Végül a tisztított modellünk esetében az 1900-as mentési pont, míg az eredeti modellünk esetében a 2160-as mentési pont került kiválasztásra. Ezen modellek posztszintű pontossága: huBERT tisztított: 70,76%; huBERT eredeti: 71,36%.

## 5 Eredmények

A Véletlen Erdő módszer nem hozott értékelhető eredményt, ezért ennek további közlésétől eltekintünk. Az algoritmus a hiperparaméterek semelyik kombinációjában sem volt képes elkülöníteni a férfiakat és nőket. Csak nőket prediktált a tanító halmaz méretének növelését követően, ahogy keresztvalidáció végrehajtása után is. A modell előtt futtatott SZVG és VE kapcsán azt tapasztaltuk, hogy a két nem elkülönítése nem lehetetlen feladat. A VE „kudarca” mögött meghúzódó okot valószínűsíthetően a kis férfi mintaelemszámban kell keresni.

<sup>6</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification>

	Abszolút pontosság	Relatív Pontosság		Fedés		F-mérték	
		Férfi	Nő	Férfi	Nő	Férfi	Nő
SZVG	90%	70%	95%	78%	93%	74%	94%
LR	80%	47%	97%	89%	78%	62%	86%
KNB	79%	47%	94%	78%	80%	58%	86%
BNB	82%	50%	97%	89%	80%	64%	88%
GNB	87%	64%	95%	78%	90%	70%	92%
huBERT (tisztított)	94%	88%	95%	78%	98%	82%	96%
huBERT (eredeti)	92%	86%	93%	67%	98%	75%	95%

**2. táblázat.** Az egyes modellek által elért eredmények.

	Abszolút pontosság		Relatív Pontosság		Fedés		F-mérték	
	Tanító	Teszt	Tanító	Teszt	Tanító	Teszt	Tanító	Teszt
SZVG	100%	90%	100%	70%	100%	78%	100%	74%
LR	93%	80%	84%	47%	93%	89%	89%	62%
KNB	89%	79%	74%	47%	93%	78%	83%	58%
BNB	87%	82%	70%	50%	93%	89%	80%	64%
GNB	94%	87%	84%	64%	97%	78%	90%	70%

**3. táblázat.** A hagyományos modellek által elért mutató értékek a tanító és a teszt-halmazon.

A futtatott modellek által elért abszolút pontosság, relatív pontosság, fedés és f-mérték értékeket a 2. táblázatban foglaltuk össze.

A hagyományosnak tekinthető, jellemzőkkel dolgozó modellek tekintetében, összetett mutatók szerint sorba rendezve a futtatott osztályozókat az SZVG modell áll az első helyen. Figyelembe véve azonban az SZVG által a tanító halmazon elért nagyon magas eredményeket, és az ezzel szembeni teszt-halmazon elért alacsonyabb összetett mutató értékeket (3. táblázat első sora) a Bernoulli Naiv Bayes modellt választottuk, mint az az osztályozási módszer, mely az adatokat a legjobban/legpontosabban el tudta különíteni. Az SZVG esetében a tanító halmazon elért 100%-os érték tökéletes modellt jelent, azonban ez az optimális érték túllílesztésre utal(hat). Egy tanító adatokra túlon túl jól illeszkedő modell pedig rosszabb általánosítási hibával rendelkezhet, mint egy nagyobb tanítási hibával rendelkező. A Bernoulli Naiv Bayes modell abban a tekintetben is jó helyen végzett, hogy összesen hány egyént sorolt rossz osztályba. Noha az SZVG összességében nézve csak 5 embert „rontott el”, de a fentebb is említett valószínűsíthető túllílesztés miatt, új, eddig nem látott adatokon nem biztos, hogy ilyen jól teljesítene.

A BNB modell mikor egy személyt férfiként jósol meg az esetek 50 százalékában helyesen jár el. A nőként való jóslás esetében 3 százalékban hibázik. A férfi osztály elemeinek 89 százalékát, míg a nők esetében 80 százalékát jósolja a megfelelő egységbe. Az algoritmus a tesztalmazba került 9 férfi esetében 8-at jósol helyesen férfinak, és 1-et tévesen nőnek. Míg a 40 nő esetében 32-t prediktált nőnek, 8-at pedig férfinak (2. táblázat BNB sora).

Nem csak a BNB modell, hanem az összes többi futtatott módszer esetében általánosságban elmondható, hogy a férfiak pontosabb prediktálása a nők helyes jóslási eredményének csökkenésével - és fordítva - járt együtt.

A kontextuális beágyazáson alapuló huBERT esetén a 2. táblázat alján, dupla elválasztóvonal alatt láthatóak a modelljeink által elért eredmények. A huBERT összességében hasonló (kissé jobb) eredményeket produkált a tisztított adatainkon ember szinten összevonva, mint a felügyelt tanulási algoritmusok. Bár esetében - ahogy SZVG kapcsán is - azt tapasztaltuk, hogy nagyon könnyű a modellt túltanítani.

A huBERT<sub>2</sub> modell teljesen tisztított, emberszinten összevont adatok esetén mikor egy személyt férfiként jósol meg az esetek 12 százalékában jár el helytelenül. A nőként való jóslás esetében 5 százalékban hibázik. A férfi osztály elemeinek 78 százalékát, míg a nők esetében 98 százalékát jósolja a megfelelő egységbe.

## 6 Összegzés

A futtatott modellek által elért eredmények alapján megállapíthatjuk, hogy a nemek közötti nyelvhasználat különbség a Facebookon is megjelenik, a férfiak és nők elkülöníthetőek egymástól nyelvhasználatuk alapján. A korábbi strukturálatlan szövegeken végzett azonos témájú kutatásokkal való teljeskörű összehasonlítás nem igazán lehetséges. Legfőképp amiatt, hogy korábban angol nyelvű (Facebook) szövegeket felhasználva végeztek felméréseket, tudomásunk szerint magyar Facebook posztokon alapuló nyelvhasználati kutatást még nem hajtottak végre.

A kapott eredmények önmagukban is hasznosíthatóak, ám a nem valószínűségi kényelmi minta miatt általánosíthatóságuk korlátozott. Önálló felhasználásuk mellett ideális kiindulópontot jelenthetnek egy olyan gyakorlatban hasznosítható kutatáshoz, melyben esetlegesen a szerzőket más demográfiai változók mentén is kategorizálják (pl.: iskolai végzettség, életkor, társadalmi státusz) és amely aztán a személyre szabott marketing, vagy akár a digitális bűnüldözés és kiberbiztonság területén alkalmazható.

Jövőbeni további munkálatokként érdemes megnézni a legjobbnak választott modell univerzalitását, hogy lehetséges-e a kapott eredményeket a konkrét mintán kívül adott műfajon belül más mintára is általánosítani. Esetlegesen – ha a későbbiekben ilyen adatok is elérhetővé válnak – megvizsgálni, hogy a társadalmi státusz vagy foglalkozás bevonásával lehetséges lenne-e választékosabb modell kiépítésére, mely segítené a nem és a nyelvhasználat kapcsolatának mélyebb megértését. Ezen kívül a keletkezett modelleket érdekes lehetne olyan szempontból is megvizsgálni, hogy attribútum vektorokkal kiegészítve – mint például használt emotikonok halmazódása, a leggyakoribb szófajok, vagy a személyes névmások aránya – az illeszkedésük javítható lenne-e.

## Hivatkozások

- Aragón, M. E., López-Monroy, A. P.: A Straightforward Multimodal Approach for Author Profiling. Notebook for PAN at CLEF 2018. In: CLEF 2018 Evaluation Labs and Workshop -- Working Notes Papers. CEUR-WS.org, Padova (2018)
- Argamon, S., Koppel, M., Fine, J., Shimoni, A. R.: Gender, genre, and writing style in formal written texts. In: Text 23(3). pp. 321–346. (2003)
- Breuer, J., Kmetty, Z., Haim, M., Stier, S.: User-focused approaches for collecting Facebook data in the “post-API” age. Kézirat (2021)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deepbidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
- Fosch-Villaronga, E., Poulsen, A., Sora, R. A., Custers, B. H. M.: A little bird told me your gender: Gender inferences in social media. In: Information Processing & Management 58(3). pp. 1-13. (2021)
- Huffaker, D., Calvert S.: Gender, Identity, and Language Use in Teenage Blogs. In: Journal of Computer-Mediated Communication, 10(2.) (2005)
- James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning. pp. 303-332. Springer, New York (2013)
- Karvalics, L. Z.: Információs kultúra, információs műveltség - egy fogalomcsalád értelme, terjedelme, tipológiája és története. In: Információs Társadalom – 12(1.) pp. 7-43. (2012)
- Kovács, E.: Többváltozós adatelemzés. pp. 126-146. Typotex Könyvkiadó, Budapest (2014)
- Markov, I., Gómez-Adorno, H., Sidorov, G.: Language- and Subtask-Dependent Feature Selection and Classifier Parameter Tuning for Author Profiling. In: CLEF 2017 Evaluation Labs and Workshop -- Working Notes Papers, 11-14 September, Dublin, Ireland. CEUR-WS.org, Dublin (2017)
- Nemeskey, D. M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021)
- Nemeskey, D. M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értékezés, Eötvös Loránd Tudományegyetem, Budapest (2020)
- Newman, M. L., Groom, C. J., Handelman, L. D., Pennebroke, J. W.: Gender Differences in Language Use: An Analysis of 14,000 Text Samples. In: Discourse Process. pp. 211-236. (2008)
- Rakovics, M.: Adattudomány jegyzet. pp. 73-76. Eötvös Loránd Tudományegyetem, Budapest (2018)
- Rangel, F., Rosso, P.: Use of Language and Author Profiling: Identification of Gender and Age. In: Workshop on Natural Language Processing and Cognitive Science (NLPCS-2013). Marseille, Franciaország (2013)
- Sap M., Park G., Eichstaedt J. C., Kern M. L., Stillwell D., Kosinski M., Ungar L. H., Schwartz H. A.: Developing Age and Gender Predictive Lexica over Social Media. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1146-1151. Doha, Katar (2014)
- Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of Age and Gender on Blogging. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. pp. 199-205. Stanford, California, USA (2006)
- Tan, Pang-Ning, Steinbach, M., Kumar, V.: Bevezetés az adatbányászatba. Panem Kft, Budapest (2011)
- Vashisth, P., Meehan K.: Gender Classification using Twitter Text Data. In: 31st Irish Signals and System Conference (ISSC). pp. 1-6. Letterkenny, Írország (2020)

XVIII. Magyar Számítógépes Nyelvészeti Konferencia      Szeged, 2022. január 27–28.

- Vincze, V., Üveges, I., Szabó, M. K., Takács, K.: A magyar beszélt és írott nyelv különböző korpuszainak morfológiai és szófaji vizsgálata. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2021)
- Zhang, C., Zhang, P.: Predicting gender from blog posts. Technical Report. University of Massachusetts Amherst, USA (2010)
- Zhang H.: The Optimality of Naive Bayes. In: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004). pp. 562-567. Miami Beach, Florida, USA (2004)





# Vállalati rendszerbe integrálható természetesnyelv-feldolgozó alkalmazás készítése digital-twin-distiller platformmal

Orosz Tamás<sup>1</sup>, Csányi Gergely Márk<sup>1,4</sup>, Gadó Krisztián<sup>1</sup>, Üveges István<sup>1</sup>,  
Vági Renátó<sup>1,2</sup>, Vadász János Pál<sup>1,3</sup>, Nagy Dániel<sup>1</sup>

<sup>1</sup>MONTANA Tudásmenedzsment Kft.

<sup>2</sup>Eötvös Lóránd Tudományegyetem

<sup>3</sup>Nemzeti Közszerológáti Egyetem

<sup>4</sup>Budapesti Múszaki és Gazdaságtudományi Egyetem

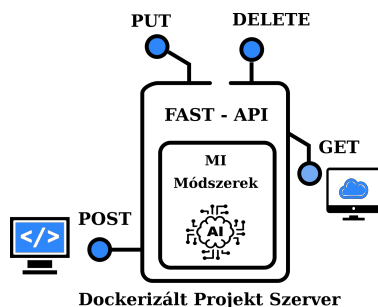
{orosz.tamas,csanyi.gergely,gado.krisztian,uvegesi,vagi.renato,vadasz.pal,nagy.daniel}-  
@montana.hu

**Kivonat** Nagy méretű, mesterséges intelligenciát vagy egyéb numerikus módszert használó projektek esetén a modellek megalkotásán és finomhangolásán kívül számos kihívást jelentő feladat van, amelyeknek a megoldása egyre nehezebbé válik a projekt méretének és a projekten dolgozó mérnökök, kutatók számának a növekedésével. A cikkben egy olyan számítási platformot mutatunk be, amelynek az segítségével a benne készített numerikus modellek egy parancs kiadásával automatikusan átalakíthatók egy vállalati rendszerbe integrálható, szabványos REST-API-t tartalmazó alkalmazássá. A cikk bemutatja az eszköz fontosabb részeit, és néhány rövid nyelvtechnológiai példán keresztül szemlélteti a kapcsolódó eszközöket.

## 1. Bevezetés

Ipari környezetben alkalmazott mesterséges intelligencia megoldásokat alkalmazó projektek fejlesztése során fontos kérdés, hogy az ügyfél igényeinek megfelelő, a célrendszerbe könnyen integrálható alkalmazást hozzunk létre. Az időnyomás és az infrastruktúra kialakítása fontos kérdés, amelyeknek az architektúrális megalapozása és elkészítése komolyabb tervezést és erőforrásokat igényelhet, mint magának a szövegbányászati modellnek a betanítása. A természetesnyelv-feldolgozást támogató eszközök közül a `spaCy` az, amelyik ipari alkalmazások létrehozását is támogató, template-szerű megoldásokat is tartalmaz `spaCy` (Schmitt és mtsai, 2019). Azonban, ahogy az a keretrendszer honlapján is olvasható (Honnibal és Montani, 2017), egy keretrendszer használata esetén előfordulhat, hogy saját platformszerű környezetet kell kialakítani, a különböző igényeket kielégítő, különböző eszközök kombinálásával. Ez a feladat megoldása során nagyobb szabadsági fokot biztosít a programozóknak, azonban lassabb, és kevésbé támogatja a fejlesztést, mint egy számítási platformszerű megoldás, ahol ezek

az igények egyetlen pontból, szorosan integrált módon elégíthetők ki (smartface, 2021). A `digital-twin-distiller` egy szövegbányászatot és numerikus modellezési feladatok fejlesztési folyamatát támogató számítási platform (Orosz és mtsai, 2021b,a). A Netflix által fejlesztett Metaflow-hoz hasonlóan (Ahmad (2021)), csoportmunkát és felhőben végzett számításokat automatikusan támogató, nyílt forráskódú Python-csomag `digital-twin distiller`<sup>1</sup>. Mindkét projekt azt használja ki, hogy a mesterséges intelligencia alapú, szövegbányászati, képfelismerési vagy egyéb komplex mérnöki problémák megoldása során sok az ismétlődő (rész)feladat (Orosz és mtsai, 2021a). A Metaflow ezeket a megoldási lépéseket egy gráfként reprezentálja, amelynek az egyes lépései újrahasznosíthatóak. A `digital-twin-distiller`-ben a feladatok egy fastruktúrába képezhetők le, majd a tanítás után automatikusan deployolhatóak is a beépített REST-API-n keresztül (1. ábra). Egyes felmérések szerint, ennek a feladatnak a megoldása a projekt teljes idejének körülbelül a 10%-át veszi igénybe (Hecht, 2019).



1. ábra. API megvalósítása `digital-twin distiller` framework segítségével.

A `digital-twin distiller`-ben készített projektek automatikusan szállíthatók egy dockerizált (Merkel és mtsai, 2014) környezetbe csomagolva, ahol a végfelhasználó egy szerverként tudja futtatni a megvalósított digitális ikerként funkcionáló alkalmazást.

A digitális iker fogalmát eredetileg a gyártási folyamatok fejlesztése során alkották meg, ez azonban az utóbbi években átalakult, kitért, immár többféle definíciója létezik. Az élő és élettelen entitások olyan digitális replikációját jelentik, ahol a termék teljes életciklusa alatt keletkezett adatokat, numerikus szimulációkat, vagy akár a mesterséges intelligencián alapuló nyelvi modelleket tartalmazza (Vogel-Heuser és mtsai, 2021; Rassolkin és mtsai, 2021).

A dockerizálásnak köszönhetően, az így előállított digitális ikrek később, akár évek múlva is futtathatóak lesznek, a rendszer mélyebb ismerete nélkül. A vállalati kompetencia fluktuációja, a vendor lock-in és a szoftverkörnyezet változása

<sup>1</sup> <https://github.com/montana-knowledge-management/digital-twin-distiller> (2021.11.09.)

miatt van előnye a fent vázolt alkalmazásnak az ipari környezetben történő alkalmazás esetén.

A vállalati kompetencia nem egy állandó érték, hanem időről időre változik, ahogyan a munkavállalók feladatkört váltanak, vagy elhagyják a munkáltatójukat. Minél gyorsabban elsajátíthatók a vállalati rendszer egyes eszközei, annál könnyebben biztosítható a folytonos üzletmenet. Hiszen, egy termelési láncba állított kód esetén nem engedhető meg, hogy azt kizárólag abban az esetben lehessen megbízhatóan működtetni, ha a teljes kifejlesztéshez szükséges kompetencia rendelkezésre áll.

A vendor lock-in az előzőhöz hasonló problémakör, csak ez esetben nem valaminek a hiánya, hanem az egyes szoftverek, szoftver komponensek adott szolgáltatóhoz vagy gyártóhoz való kötöttsége miatt állhat elő az a helyzet, hogy a vállalat nem ura a saját kódjának, ezért kiszolgáltatottá válik az üzletmenet terén.

A népszerű természetesnyelv-feldolgozásra alkalmas keretrendszerek, mint a Gensim, a spaCy vagy a Keras elérhetőek a distillerben történő fejlesztés során is, azonban ezen könyvtárak mellett egyéb pluginként használható, természetesnyelv-feldolgozást támogató könyvtárak is használhatók. A következő fejezetekben bemutatjuk a `digital-twin distiller` keretrendszerben jelenleg elérhető főbb modulokat, plugineket.

## 2. I/O funkciók támogatása

Egy projekt megvalósítása során az egyik legalapvetőbb, és egyben leggyakrabban ismétlődő feladat a projekt során használt adatok fájlból beolvasása és fájlba írása. A `digital-twin distiller` keretrendszer mind beolvasás, mind fájlba írás esetében többféle formátumot is támogat, melyeket az 1. táblázat mutat be.

Támogatott formátum	Modulok		
	Reader	Writer	Data Snapshot
json	O	O	O
txt	O	O	X
pdf	O	X	X
zip	X	X	O

1. táblázat. A `digital-twin distiller` által támogatott formátumok. Jelölés: O: jelenleg támogatott, X: jelenleg még nem támogatott

A framework támaszkodik az Apache Tika<sup>2</sup> szoftverre, amely segítségével lényegében bármely szöveges formátum könnyedén beolvasható. Alapértelmezettként a pdf, txt és json fájlok beolvasására alkalmas osztályok érthetők el. A projektek életciklusa során gyakori feladat a különféle előfeldolgozási lépések

<sup>2</sup> <https://tika.apache.org/> (Elérés: 2021.11.12.)

(pl. lemmatizálás, tokenizálás, szótövezés, POS-taggelés, egyéb adatok kinyerése stb.) elvégzése, amelyek igen időigényesek is lehetnek. A hatékonyabb fejlesztés célját szem előtt tartva a **digital-twin distiller**ben implementáltunk egy `DataSnapshot()` névre hallgató osztályt, amely egy zip fájlba képes json kiterjesztésű adatok listáját kimenteni, illetve onnan betölteni. Ezzel egyszerűen megvalósítható egy hosszadalmas előfeldolgozás utáni adat gyors betöltése és további használata, amellyel jelentős idő takarítható meg.

### 3. Modellek mentése

Bizonyos gépi tanulási módszereket implementáló Python könyvtárak, mint a **scikit-learn** (Pedregosa és mtsai, 2011) nem biztosít modellmentési lehetőséget a keretrendszeren belül, így legtöbbször ezeket a modelleket **pickle** (Van Rossum, 2020), vagy **joblib** (Team, 2020) formátumban szokták a felhasználók elmenteni. A Python **pickle** modulja azonban kockázatot jelent a gépi tanulási modellek üzemeltetésével kapcsolatban, hiszen az adott modellt az összes osztályváltozójával együtt menti el. Ha az osztály egyik adattagja egy refaktorálás során megváltozik, akkor az pl. egy frissített Python könyvtárral már nem használható újra, így a modell tanításával töltött idő elvész.

A **digital-twin distiller**, az AGPL v3 licenstes **sklearn2json** modul<sup>3</sup> használja a **scikit-learn** modellek mentésére. Ez egy különálló, önmagában is használható modul, amely egy json formátumú fájlba menti az adatokat, így pl. egy frissítés során az osztályváltozók nevének változása esetén ez az egy helyen karbantartott csomag képes ezeket a változásokat lekezelní, így újra felhasználhatóvá tenni őket. Az **sklearn2json** emellett számos osztály mentésére kínál lehetőséget, legyen az klasszifikáló-, regressziós problémára alkalmazható-, klaszterező- vagy vektorizáló osztály.

A probléma kezelésére léteznek más csomagok is, például az **sklearn2pmm1**<sup>4</sup>, amelyet egy kisebb észt startup fejleszt, ez a megoldás azonban egy Java kódból generált Python API-val használható. Egyre nagyobb nemzetközi figyelmet és támogatottságot tudhat maga mögött az ONNX projekt<sup>5</sup>, amely a különböző keretrendszerekben tanított modellek portabilitásáért felel, így nagyvállalatok által fejlesztett ipari standarddá is válhat. Az onnx, mint közös fájlformátum lehetővé teszi a gépi tanulási fejlesztők számára, hogy a modelleket különböző keretrendszerekkel, és eszközökkel használhassák.

### 4. Augmentálás

Habár napjainkban rengeteg digitális adat keletkezik, mégis a gépi tanulást is alkalmazó projektek esetében nem ritka, hogy az adott (rész)feladat megoldásához nincsen elegendő mennyiségű tanítóadat. Ilyen esetekben gyakran alkal-

<sup>3</sup> <https://github.com/montana-knowledge-management/sklearn2json>

<sup>4</sup> <https://github.com/jpmm1/sklearn2pmm1> (2021.11.12.)

<sup>5</sup> <https://github.com/onnx/onnx>

mazott megoldás valamilyen adathalmaz, más szóval augmentáló eljárás alkalmazása, amely során a meglévő adatok mellé különböző eljárásokkal szintetikusán előállított tanítóadatot lehet képezni. Az így kapott kibővített adathalmaz már jellemzően alkalmas pontosabb modellek tanítására. Fontos megjegyezni, hogy a szintetikusán előállított adatok minősége jellemzően elmarad a valós adatokétól, sok esetben mégis jelentős pozitív hatással vannak a modellek teljesítményére.

Szöveges adatok esetében az augmentálási módszereket két nagyobb csoportra oszthatjuk aszerint, hogy magát a szöveget módosítjuk, augmentáljuk (pl. szavak törlése, szinonimák hozzáadása, cseréje stb.), vagy pedig annak vektorreprezentációját (pl. SMOTE (Chawla és mtsai, 2002)). Ezek közül a **digital-twin distiller** jelenleg csak az előbbi típust támogatja, erre azonban több különböző megoldást is kínál.

Fontos megjegyezni, hogy az augmentálás során előfordulhat olyan eset, hogy bizonyos kifejezéseket nem szeretnénk, hogy módosítson az augmentáló algoritmus. Ezért elláttuk az összes jelenleg elérhető augmentálási módszerünket egy **védett szólista** megadásának lehetőségével.

A következő alfejezetekben sorra vesszük a **digital-twin distiller** által jelenleg támogatott augmentálási módszereket.

#### 4.1. Easy Data Augmentation (EDA) (Wei és Zou, 2019)

Az ún. Easy Data Augmentation (EDA) négy egyszerű eljárást takar. Ezek a szinonima csere (Synonym replacement, SR), véletlenszerű törlés (Random deletion, RD), véletlenszerű szó szinonimájának beszúrása a szövegbe (Random insertion, RI), és a szövegben előforduló szavak véletlenszerű cserélgetése (Random swap, RS). A szinonimákat az eredeti megoldásban angol nyelvű WordNet segítségével keressük meg (Wei és Zou, 2019), a mi eszközünk pedig alapértelmezettként a magyar nyelvet támogatja a magyar WordNetre építve (Miháltz és mtsai, 2008).

Valamennyi megoldás esetén beállítható a művelet elvégzésének valószínűsége tokenenként, vagy a művelet elvégzésének száma.

Wei és Zou (2019) rövid szövegeken bizonyította az eljárások hatékonyságát, de természetesen ez nagyban függ a konkrét alkalmazási területtől is.

#### 4.2. Szóbeágyazás alapú augmentálás

Egy tipikus augmentálási megoldás a szemantikai hasonlóság alapján történő augmentálás (Csányi és Orosz, 2021). Ennek során véletlenszerűen kiválasztott szavakat egy betanított nyelvi modell segítségével lecserélünk a hozzájuk leghasonlóbb szavakra, ezzel az eredeti mondathoz lényegileg hasonló, mégis új adatot hozva létre.

Ilyen típusú augmentálás megvalósítására alkalmasak pl. a különböző szóbeágyazási modellek, amilyen a Word2Vec (Mikolov és mtsai, 2013), GloVe (Pennington és mtsai, 2014) vagy a FastText (Bojanowski és mtsai, 2017).

A keretrendszerben lehetővé tettük az ilyen eljárással történő augmentálást, melyet kutatási célra már alkalmaztak is (Úveges és mtsai, 2022). Az augmentálás során itt is beállítható, hogy milyen eséllyel legyen kiválasztva augmentálásra egy adott szó, valamint, hogy egy-egy szó cseréje során hány darab, hozzá leginkább hasonló más szóból történjen a kiválasztás. Utóbbi esetén szintén megadható, hogy ezekből a mintavételezés súlyozottan vagy teljesen egyenletes eloszlást követve történjen-e, esetleg az algoritmus minden esetben a leghasonlóbb szót válassza ki helyettesítésre.

Mivel ezek a szóbeágyazási modellek nem alkalmasak a kontextus figyelembe vételére, így egy adott szóhoz a leghasonlóbb szavakból építhető egy szótár (Python dictionary), amely az adott korpusz összes különböző szavához tartalmazza a legközelebbi vektorrepresentációval bíró szavakat. Ezen szótár fájlba mentésével jelentős időt lehet megtakarítani, hiszen nem szükséges minden alkalommal a szóbeágyazási modelltől lekérdezni a leghasonlóbb szavakat, illetve ha egy szó többször is előfordul a szövegben, úgy a lekérdezést szintén elég egy alkalommal elvégezni.

Jelenleg a keretrendszer által a `fasttext` és a `gensim` library-k által támogatott modellek használhatóak, az előbbiből 157 különböző nyelven érhetőek el modellek, így a módszer könnyen alkalmazható több különböző nyelvre is<sup>6</sup>.

### 4.3. Kontextusfüggő beágyazás alapú augmentálás

Egy másik, szofisztikáltabb megoldás a kontextusfüggő, maszkolt nyelvi modellek használata augmentálásra, mint például a BERT (Devlin és mtsai, 2018). Mivel a nyelvi modellek tanítása során jellemzően az egyik célfeladat a kimaszkolt szövegrészek predikciója, így az eredeti szövegből maszkolással új, az eredetihez nagyon hasonló jelentésű szövegek hozhatók létre, amelyek nyelvtanilag lényegesen nagyobb százalékban eredményeznek helyes mondatot, mint az előzőekben bemutatott mondatok. Az előzőekhez hasonlóan az augmentálás aránya itt is megadható. Az augmentáláshoz használt modellek a `huggingface` oldaláról<sup>7</sup> érhetőek el, igen széles választékban.

## 5. Egyéb modulok

### 5.1. A `hungarian-stemmer` szótövező

Az utóbbi években nagy népszerűségnek örvendenek a hunspell-alapú szótövező eszközök (Halácsy és mtsai, 2004) mind a kutatók, mind az ipari felhasználók körében. Gyakori probléma volt azonban a magyar nyelvű verzióban, hogy a szótövező "túltövez", gyakran olyan töveket is visszaadva, amelyek jelentésben igen messze vannak a kiindulási szótól. Emellett tapasztalható volt, hogy az igekötős igék esetén a korábbi verzió igekötő nélküli szótövet adott vissza, illetve nem történt szótövezés néhány olyan esetben, amikor szükséges lett volna (lásd 2.

<sup>6</sup> <https://fasttext.cc/docs/en/crawl-vectors.html>

<sup>7</sup> <https://huggingface.co/models>

táblázat). Erre válaszul készült el a **hungarian-stemmer**, amely ezt a problémás működést kiküszöbölve orvosolja a fent említett problémákat. A különbségekről a 2. táblázat ismerteti néhány példát.

2. táblázat. A hunspell és hungarian-stemmer közti különbség

Szó	hunspell	hungarian-stemmer
mással	más, ma, mi	más
meg történt	történik	meg történt, megtörténik
köztestületi	köztestületi	köztestület

## 5.2. Szöveg vektorizálás

Egy ismétlődő, gyakori feladat lehet annak vizsgálata, hogy az adott szövegekre egyszerűbb vektorrepresentációs formák használatával betanított modellek hogyan teljesítenek. A keretrendszer ezért tartalmaz egy dokumentum vektorizáló modult, amely jelenleg háromféle módon képes dokumentumokhoz vektorokat rendelni:

- dokumentumban szereplő szavak szóbeágyazásainak átlaga
- dokumentumban szereplő szavak szóbeágyazásainak a szavak idf értékeivel súlyozott átlaga
- Doc2Vec vektor készítése (Le és Mikolov, 2014).

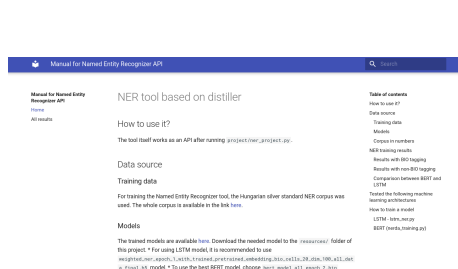
## 5.3. Pozitív-címkézetlen tanulás

A gyakorlatban sok esetben nem csak a gépi tanulási modell tanításához szükséges adatok elégtelen száma jelent problémát, hanem az adatok elégtelen címkézése is. Ilyen esetben például egy bináris klasszifikálási feladatnál csak az ismert, hogy a pozitív minták valóban pozitívak, azonban a többi adatról nem rendelkezünk egyéb információval; azok egyaránt lehetnek negatív és pozitív címkéjű adatok is. Ez a problémakör a pozitív-címkézetlen tanulás (Positive Unlabeled learning, PU learning) (Li és Liu, 1970; Bekker és Davis, 2020). A többféle megközelítés közül a **digital-twin distiller** keretrendszerben jelenleg az Elkan és Noto (2008) által közölt megoldás érhető el.

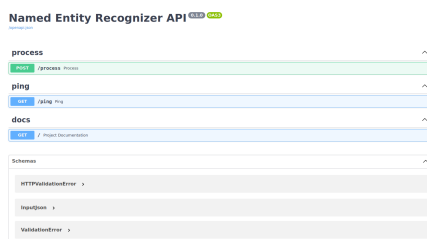
A plugin minden olyan gépi tanulási algoritmust támogat, amely képes valószínűség predikciójára, tetszőleges vektorizálási formán, tehát egyaránt alkalmas pl. tf-idf vagy szóbeágyazás alapú vektorokon is működni. A plugin rendelkezik egy olyan funkcióval is, hogy a pozitív entitások számát becsülje az Elkan és Noto (2008) által közöltek szerint, valamint használható a kézi annotálás hatékonyabbá tételére is.

## 6. Dokumentáció

Egy projekt megvalósításának az egyik legfontosabb sarokköve (különösen a későbbi üzemeltetés szempontjából) a megfelelő dokumentálás. Ennek a megvalósításában is segít a **digital-twin distiller** keretrendszer, amely a népszerű **mkdocs**<sup>8</sup> dokumentáció készítő szoftverre épít. A keretrendszerbe integráltuk továbbá az Open API<sup>9</sup> (korábban Swagger) által biztosított API dokumentációt bemutató eszközt is, amellyel nem csak az elkészített API végpontjainak dokumentációját lehet elolvasni, hanem ki is lehet próbálni ezeket. Ennek segítségével a kódolásban járatlanok is könnyedén tesztelhetik az elkészített API működését. Erre mutat egy példát a 2. ábra.



(a) Mkdocs-alapú dokumentáció kezelése



(b) Egy elkészült névelem-felismerő rendszer Open API dokumentációja

2. ábra. Dokumentáció a **digital-twin distiller** keretrendszeren keresztül

## 7. Alkalmazási példák

A keretrendszer működését, alkalmazását néhány rövid, gyakorlati példán is szemléltetjük.

### 7.1. Augmentálás

Az 4. fejezetben bemutatott augmentálási módszerek eredményeit egy példamondaton a 3. táblázat mutatja be. Mindegyik módszer esetében  $\alpha = 0,5$ -ös beállítást alkalmaztunk, amely esetén az adott algoritmus a mintapélda tokenjein átlagosan 50% eséllyel alkalmazta az adott augmentálási módszert. A módszerek időigényét egy eredeti mondatot 100 példányra augmentálva tüntettük fel, és a modellek betöltésének idejét az értékekbe nem számítottuk bele.

Látható, hogy maguk a módszerek különböző előnyökkel és hátrányokkal rendelkeznek. Az EDA algoritmusok jelentik a minőségben leggyengébb megoldást, azonban futási idejük viszonylag alacsony.

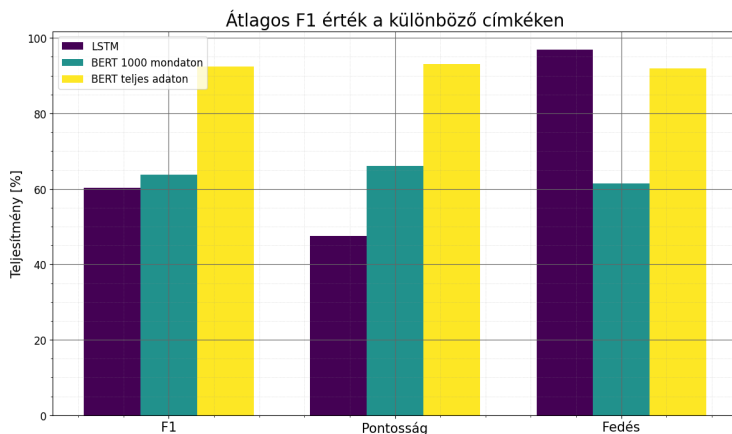
<sup>8</sup> <https://www.mkdocs.org/>

<sup>9</sup> <https://www.openapis.org/>



Módszer	Futási idő [s]	Augmentálás eredménye
Eredeti mondat	-	A horvát élvonalban játszó Kleinheisler László biztosan nem fog felépülni a magyar válogatott következő két vb selejtezőjére.
EDA/RD	0,61	A horvát Kleinheisler László
EDA/RS	0,52	A felépülni magyar játsszó Kleinheisler selejtezőjére. következő biztosan fog vb a élvonalban nem László két horvát válogatott
EDA/RI	0,50	A horvát élvonalban makulátlanul játszó Kleinheisler László biztosan nem eszert eredő fog felépülni a magyar válogatott következő két vb selejtezőjére.
EDA/SI	0,46	A horvát élvonalban játszó Kleinheisler László ügyesen nem eszert felépülni a magyar válogatott rákövetkező két vb selejtezőjére.
FastText	24,92/0,12*	A horvát élvonalban játszik Kleinheisler Tamás biztosan nem fogok felépülhet a külföldi válogatott fenti Két vb selejtezőjére.
BERT	44,67	A magyar színekben játszó Tóth már biztosan nehezen fog felépülni a magyar válogatott következő két vb selejtezőjére.

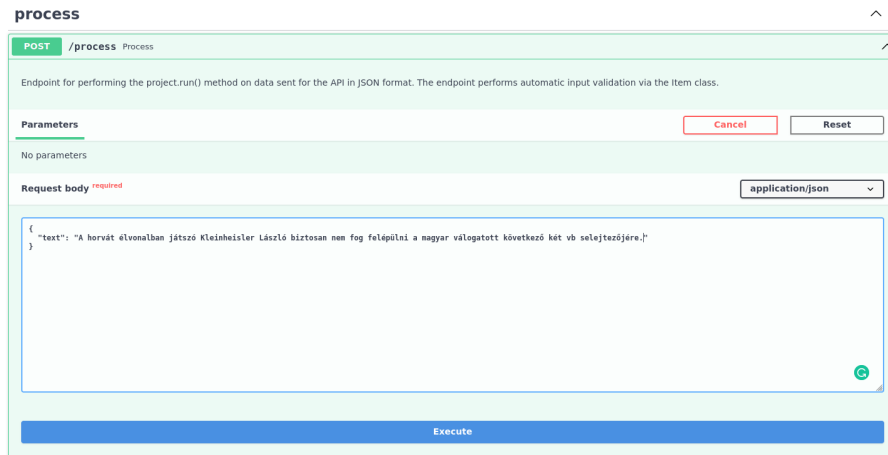
3. táblázat. Különböző augmentálási módszerek működése,  $\alpha = 0,5$  beállítással.  
\*: előre elkészített szótár használatával



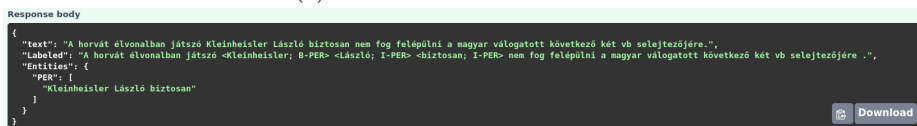
3. ábra. Névellem-felismerés eredménye

A minőségben jelentősebb előrelépést jelentő FastText-et használó augmentálás lényegesen lassabb amíg a szöveg szavaihoz leghasonlóbb szavakat tartal-

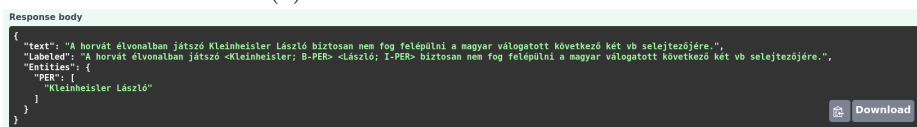
mazó szótár elkészül, azonban ha ezzel már rendelkezünk, akkor jelentősen javul az augmentálás sebessége is. A példa alapján is látszik, hogy a kontextusfüggő BERT módszer adja a legjobb minőségű augmentált adatot, nem ritkán nyelvtanilag tökéletes mondatokat alkotva, azonban ehhez is szükséges a legtöbb idő. Megfigyelhető tehát a gyakran megjelenő performancia-idő tradeoff (Baeza-Yates és Liaghat, 2017), másképpen a „jó munkához idő kell” népi bölcsesség.



(a) Névelem-felismerő tesztelése



(b) LSTM modell által talált entitás



(c) BERT modell által talált entitás

#### 4. ábra. Különböféle NER modellek egyszerű cserélhetősége

## 7.2. Névelem-felismerő rendszer

Ebben a fejezetben egy elkészült névelem-felismerő rendszer működését mutatjuk be. A keretrendszer képes bármilyen fájlba menthető illetve onnan beolvasható gépi tanulási modellt kezelni egy projekten belül, tehát egyaránt alkalmas pl. `sklearn`, `keras`, `huggingface` stb. alapú modellek hatékony használatára API-ként, ezáltal lehetőséget biztosít, hogy minimális módosítások elvégzésével a különböző módokon elkészült modellek egyaránt használhatóak legyenek.

Példaként egy egyszerű névelem-felismerőt tanítottunk fel a hunNERwiki korpuszon (Simon és Nemeskey, 2012), amely IOB2 taggelési formátumot használ.

Kétféle különböző modellt tanítottunk fel; az egyik egy **keras** segítségével feltanított LSTM (Hochreiter és Schmidhuber, 1997) alapú megoldás, a másik pedig a kisméretű többnyelvű BERT nagybetűket is kezelő változatának (bert-base-multilingual-cased) finomhangolása volt. Tanításhoz az adatok 80%-át használtuk, 10% validálási és 10% tesztalmoz felosztásban.

A tesztalmozon elért eredményeket a 3. ábra mutatja be, ahol az egyes entitások F1 értékeinek nem súlyozott átlagait tüntettük fel.

Látható, hogy az LSTM modell nagyobb fedést volt képes elérni, mint a többnyelvű BERT bármelyik verziója, azonban az F1-mérték átlagát tekintve jelentősen jobbnak bizonyult az utóbbi modell, amely már ezer(!) nem csak O entitást tartalmazó mondat esetén jobban teljesített, mint a több epochon keresztül tanított LSTM modell.

Az elkészült API tesztelését a különböző modellek segítségével a 4. ábra mutatja be. Látható, hogy mindkét modell megtalálta Kleinheisler László nevét, azonban az LSTM modell hozzátartozónak ismerte föl a „biztosan” szót is.

### 7.3. Anonimizálás

Az előző fejezetben feltanított névelem-felismerőnket ezt követően a **digital-twin distiller** keretrendszer segítségével egy kezdetleges anonimizáló eszközzé is formálhatjuk. Elkerülve a monogramok használatát, a neveket „X” karakterre cseréltük kizárólag az első karaktert meghagyva és a rövidítést ponttal jelölve. Ennek eredménye a 5. ábrán látható.

```
{
  "text": "A horvát élvonalban játszó Kleinheisler László biztosan nem fog felépülni a magyar válogatott következő két vb selejtezőjére.",
  "labelled": "A horvát élvonalban játszó <I-PER> Kleinheisler, <B-PER> László, <I-PER> biztosan nem fog felépülni a magyar válogatott következő két vb selejtezőjére.",
  "entitles": {
    "PER": [
      "Kleinheisler László"
    ]
  },
  "Anonymized": "A horvát élvonalban játszó X. X. biztosan nem fog felépülni a magyar válogatott következő két vb selejtezőjére."
}
```

5. ábra. Anonimizálás eredménye

A példamondat alapján jól látszik, hogy ez a művelet miért nem tekinthető a GDPR-nak megfelelő anonimizálásnak (Csányi és mtsai, 2021). Ehhez egy statisztikai analízisre is szükség lenne, amelynek a segítségével bizonyítható, hogy a szövegben található személy személyazonossága nem ismerhető fel a mondatban található mikroadatokból.

Ismertek ugyanis a következő információk:

- sportolóról van szó,
- aki a horvát élvonalban játszik,
- a magyar válogatott tagja,
- és vb selejtezőn nem tud részt venni, tehát olyan sportágban, ahol ilyen egyáltalán létezik.

Ezek alapján feltételezve az egyik legnépszerűbb sportágat, a labdarúgást, valamint ha ismert, hogy mikor keletkezett a szöveg (2021), könnyen megtalálhatjuk, hogy kiről is van szó, hiszen jelenleg egyetlen labdarúgó játékos van, akire illik ez a leírás (Lovrencsics Gergő ugyanis visszamondta a válogatottságot). Ha ezen pszeudo-azonosítók (Dalenius, 1986) – amik önmagukban nem, de együttesen már alkalmasak lehetnek az egyén azonosítására – nincsenek megfelelően kezelve, akkor a fent bemutatott anonimizálási megoldás lényegében haszontalan. Fontos ezért információ tartalom szerint is figyelembe venni a pszeudoinformációkat anonimizálásakor, ami az egyes információkhoz tartozó ekvivalenciaosztályok számosságának a becslésével lehetséges (Csányi és mtsai, 2021). Ha az ekvivalenciaosztály elegendően nagy, azaz a társadalom egy elegendően nagy részére illik az adott mikroadat, azzal biztosítható, hogy nehezen lehessen felismerni az adott szövegben szereplő egyén személyazonosságát. Ha a példamondatból kivesszük a két leginkább szűkítő információt, tehát, hogy horvát élvonalban játszik, és hogy magyar válogatott, illetve ezekből csak a nemzetiségneveket, akkor az már meggátolhatja a konkrét személy beazonosítását.

## 8. Összegzés

A cikkben egy Python alapú eszközt, a `digital-twin-distillert` mutattuk be, amelynek a segítségével készített szövegfeldolgozási modellek egy parancs segítségével konténerizálhatóak egy digitális ikerként. Az ilyen módon előállt, kapszulázott mesterséges intelligencia modell könnyen és szabványos módon integrálható bármilyen más rendszerbe a projekthez csomagolt REST-API segítségével, emellett akár önállóan is használható a beépített webes alkalmazás segítségével. A számítási platform a modellalkotáshoz és a kutatási projektek támogatásához számos plugint tartalmaz, melyek tetszőlegesen bővíthetők. A cikkben három gyakorlati példán - egy augmentált példamondaton, több módszerrel feltanított névelem-felismerőn mutattuk be hogy hogyan lehet kutatási és ipari projektek támogatására, közzétételére használni a `digital-twin-distillert`. A számítási platformhoz számos plugin tartozik, amelyek segítségével a modellek tanítása és a szövegek preprocessálása egyszerűsíthető, ezáltal jelentős idő megtakarítást elérve. Jó példa erre, hogy a hosszú ideig preprocessált adatainkat kimenthetjük és beolvashatjuk egy snapshotba (`DataSnapshot`), szintetikusan létrehozott mintákkal javíthatjuk a gépi tanulási modelljeink teljesítményét (augmentáció), nem teljesen címkézett adatokon taníthatunk hatékonyabb gépi tanulási modelleket (PU-learning), dokumentumokat vektorizálhatunk egyszerűen, vagy szótövezhetünk. A `digital-twin distiller` keretrendszer a fentiekben bemutatott pluginokkal és modulokkal együtt szabadon elérhető<sup>10</sup>.

<sup>10</sup> <https://bitbucket.org/montanatudasmenedzsmentkft/distiller/src/master/>  
(2021.11.09.)

## Köszönetnyilvánítás

A tanulmány a 2020-1.1.2-PIACI-KFI-2020-00049 számú projekt a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatásával valósult meg, a 2020-1.1.2-PIACI KFI finanszírozási rendszer keretében.

## Hivatkozások

- Ahmad, H.: How netflix metaflow helped us build real-world machine learning services (2021), (2021.11.12.)
- Baeza-Yates, R., Liaghat, Z.: Quality-efficiency trade-offs in machine learning for text processing. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 897–904 (2017)
- Bekker, J., Davis, J.: Learning from positive and unlabeled data: A survey. *Machine Learning* 109(4), 719–760 (2020)
- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146 (2017)
- Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16, 321–357 (2002)
- Csányi, G., Orosz, T.: Comparison of data augmentation methods for legal document classification. *Acta Technica Jaurinensis* (2021)
- Csányi, G.M., Nagy, D., Vági, R., Vadász, J.P., Orosz, T.: Challenges and open problems of legal document anonymization. *Symmetry* 13(8), 1490 (2021)
- Dalenius, T.: Finding a needle in a haystack or identifying anonymous census records. *Journal of official statistics* 2(3), 329 (1986)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- Elkan, C., Noto, K.: Learning classifiers from only positive and unlabeled data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 213–220 (2008)
- Halácsy, P., Kornai, A., László, N., András, R., Szakadát, I., Viktor, T.: Creating open language resources for hungarian (2004)
- Hecht, L.E.: Add it up: How long does a machine learning deployment take? (2019), <https://thenewstack.io/add-it-up-how-long-does-a-machine-learning-deployment-take/>, (2021.11.12.)
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* 9(8), 1735–1780 (1997)
- Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
- Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196. PMLR (2014)

- li, X., Liu, B.: Learning from positive and unlabeled examples with different data distributions. pp. 218–229 (01 1970)
- Merkel, D., és mtsai: Docker: lightweight linux containers for consistent development and deployment. *Linux journal* 2014(239), 2 (2014)
- Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., Váradi, T.: Methods and results of the hungarian wordnet project. In: *Proceedings of The Fourth Global WordNet Conference*. pp. 311–321 (2008)
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
- Orosz, T., CSányi, G., Nagy, D.: Mesterséges intelligenciát alkalmazó szövegbányászati eszközök készítése a distiller keretrendszer segítségével–jogi szövegek automatikus feldolgozása: Development of artificial intelligence-based text mining tools with the distiller-framework–in case of legal documents. *Energetika-Elektrotechnika–Számítástechnika és Oktatás Multi-konferencia* pp. 62–69 (2021a)
- Orosz, T., Gadó, K., Katona, M., Rassólkin, A.: Automatic tolerance analysis of permanent magnet machines with encapsulated fem models using digital-twin-distiller. *Processes* 9(11) (2021b), <https://www.mdpi.com/2227-9717/9/11/2077>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., és mtsai: Scikit-learn: Machine learning in python. *the Journal of machine Learning research* 12, 2825–2830 (2011)
- Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543 (2014)
- Rassolkin, A., Orosz, T., Demidova, G.L., Kuts, V., Rjabtšikov, V., Vaimann, T., Kallaste, A.: Implementation of digital twins for electrical energy conversion systems in selected case studies. In: *Proc. Est. Acad. Sci. vol. 70* (2021)
- Schmitt, X., Kubler, S., Robert, J., Papadakis, M., LeTraon, Y.: A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. pp. 338–343. IEEE (2019)
- Simon, E., Nemeskey, D.M.: Automatically generated ne tagged corpora for english and hungarian. In: *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*. pp. 38–46. Association for Computational Linguistics, Jeju, Korea (July 2012), <http://www.aclweb.org/anthology/W12-4405>
- smartface: Add it up: How long does a machine learning deployment take? (2021), [What is the Difference Between a Platform and a Framework?](https://www.smartface.com/blog/2021/11/12/what-is-the-difference-between-a-platform-and-a-framework/), (2021.11.12.)
- Team, J.D.: Joblib: running python functions as pipeline jobs (2020), <https://joblib.readthedocs.io/>, (2021.11.12.)
- Van Rossum, G.: The python library reference, release 3.8. 2. Python Software Foundation 16 (2020)
- Vogel-Heuser, B., Ocker, F., Weiß, I., Mieth, R., Mann, F.: Potential for combining semantics and data analysis in the context of digital twins. *Philosophical Transactions of the Royal Society A* 379(2207), 20200368 (2021)

XVIII. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2022. január 27–28.

Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)

Üveges, I., Orosz, T., Csányi, G., Orsolya, R.: Szövegaugmentálási módszerek összehasonlítása politikai szövegek szentimentanalízise során. XVIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2022) (2022)





# A kis HIL-ELECTRA, HIL-ELECTRIC és HIL-RoBERTa Magyar kísérleti nyelvi modellek tanítása kevés erőforrással

Yang Zijian Győző<sup>1,3</sup>, Feldmann Ádám<sup>2</sup>, Váradi Tamás<sup>1</sup>

<sup>1</sup>Nyelvtudományi Kutatóközpont

{yang.zijian.gyozo, varadi.tamas}@nytud.hu

<sup>2</sup>Pécsi Tudományegyetem, Általános Orvostudományi Kar, Magatartástudományi  
Intézet, Alkalmazott Adattudomány és Mesterséges Intelligencia Csoport  
feldmann.adam@pte.hu

<sup>3</sup>MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport  
1083 Budapest, Práter u. 50/a.  
yang.zijian.gyozo@itk.ppke.hu

**Kivonat** Napjainkban a természetes nyelvfeldolgozás (NLP) területén rohamos tempóban uralkodóvá válnak a neurális hálós technológiák, amelyekhez az egyre nagyobb méretű nyelvi modellek kifejlesztése adja az alapot. Az általános célú nyelvi modellek, mint például a BERT, előtanításához rendkívüli számítási kapacitást és hasonlóan extrém nagy mennyiségű adathalmazra van szükség, amelyek nem mindenütt állnak rendelkezésre. Ezért az egyre nagyobb méretű nyelvmodellek létrehozásával egyidőben kialakult egy ellentétes fejlesztési irány, amely a kisebb modellek kifejlesztését célozta. A kutatásunk fókuszában az kérdés áll, hogy mire képesek a limitált erőforrások mellett kifejleszhető és alkalmazható kisebb nyelvmodellek. Mivel a BERT modell kisebb magyar nyelvű változatairól már születtek tanulmányok, így jelen kutatásunk keretében két másik modell, az ELECTRA és az ELECTRIC, valamint a RoBERTa betanítására fókuszáltunk. A Google Inc. által fejlesztett ELECTRA és ELECTRIC modell sokkal kevesebb erőforrással és futási idővel tanítható be, mint a hagyományos BERT-alapú modellek. A RoBERTa modell nemzetközi viszonylatban kiemelkedő eredményeket ért el számos nyelvelemzési feladatban, mindez indokoltá teszi a magyar nyelvre történő alkalmazását. Eredményeink alapján elmondható, hogy bár természetesen kísérleti modelljeink nem képesek elérni a jelenlegi csúcsteljesítményű magyar nyelvű hagyományos BERT modelleket, azonban modelljeink közel olyan jó, értékelhető eredményeket érnek el kisebb erőforrásigény mellett, ami környezettudatosabb megoldások kifejlesztéséhez vezethet. Továbbá a kis modellek alkalmazása felhasználóbarát, mivel hasonlóan jó eredmények elérése lehetséges lényegesen kisebb modellek alkalmazásával, ami adott alkalmazásban kulcsfontosságú szempont lehet.

**Kulcsszavak:** ELECTRA, ELECTRIC, RoBERTa, nyelvi modellek, kevés erőforrás, névelemfelismerés, főnévi csoport felismerés, szentiment analízis, mondatszintű osztályozás, extraktív összefoglalás

## 1. Bevezetés

Az utóbbi évtizedben a neurális háló alapú technológiák robbanásszerű fejlődést hoztak a természetes nyelvfeldolgozás terén. Az első áttörés a szóbeágyazás módszere volt (Mikolov és mtsai, 2013b,a), amely a szavakat többdimenziós, folytonos vektorokként ábrázolja. A szavak vektorrepresentációja egy szemantikus teret eredményez, amelyben a hasonló jelentésű szavak közel helyezkednek el egymáshoz. A szemantikai tartalom mellett a szóvektorok szintaktikai jellemzők reprezentálására is képesek. Az egyik legfőbb hátránya ennek a módszernek, hogy a vektorok szóalakokat képviselnek, és egybemosnak olyan szóalakokat, amelyek különböző vagy egymástól független jelentéssel bírnak. E probléma kiküszöbölésére olyan alternatív modelleket fejlesztettek ki, melyek kontextuális beágyazáson alapulnak, mint például az ELMo (Peters és mtsai, 2018), BERT (Devlin és mtsai, 2019) vagy a BERT származékok (mint például a RoBERTa (Liu és mtsai, 2019)), melyekben a szóvektorok képesek tükrözni kontextusfüggő jelentésváltoztatokat. A BERT (Devlin és mtsai, 2019) (Bidirectional Encoder Representations from Transformer) úgy definiálható, mint egy többszintű, kétirányú transzformer enkóder (Vaswani és mtsai, 2017). A BERT modell két nyelvmodellezési feladaton van előtanítva: szómaszkolás és következő mondat predikciója. A szómaszkolás során a tanításhoz használt korpuszban a szavak 15 százaléka véletlenül maszkolásra kerül, a rendszernek pedig ki kell találnia ezeket az elrejtett szavakat. A következő mondat predikciója során pedig a feladat annak kitalálása, hogy két mondat a szövegben egymás után következik-e vagy csak véletlenül állnak egymás után. A szótár méretének csökkentése érdekében egy tokenizáló algoritmus (Schuster és Nakajima, 2012) a szavakat szóelemekre (word pieces), azaz statisztikai alapon megállapított betűszekvenciákra bontja. A BERT kezdeti betanítását követően az előtanított modell finomhangolásra kerül, hogy optimális legyen az adott célfeladat elvégzésére.

Az első natív magyar nyelvű BERT modell Nemeskey (2021) által került publikálásra huBERT néven. Három huBERT modell készült el:

- huBERT: BERT base modell, mely a Hungarian Web Corpus 2.0 korpuszon (Nemeskey, 2020b) lett betanítva
- huBERT Wikipedia cased modell: BERT base modell, mely a magyar nyelvű kisbetűsítés nélküli Wikipédián lett betanítva
- huBERT Wikipedia lowercased: BERT base modell, mely a kisbetűsített magyar nyelvű Wikipédián került betanításra

A Hungarian Web Corpus 2.0 korpuszon betanított huBERT base modell kiemelkedő eredményeket ért el az olyan feladatokban, mint a névelemfelismerés és a főnévi csoportfelismerés (Nemeskey, 2020a).

Az egyik fő korlátja a neurális háló alapú modellek fejlesztésének, hogy rendkívül erőforrásigényes folyamatnak számít mind adat mind pedig számítási kapacitás szempontjából. Az imént említett limitációra megoldást kínál a Google Inc. által fejlesztett ELECTRA modell (Clark és mtsai, 2020a), amely képes a hagyományos BERT modellhez hasonló eredmények elérésére kevesebb erőforrás felhasználásával (1 GPU) és kevesebb idő alatt. Az ELECTRA további előnye

az, hogy a betanított modellek mérete lényegesen kisebb, ami kulcsfontosságú az egyre mobilisabbá váló világunkban.

Kutatásunk során az ELECTRA, ELECTRIC és RoBERTa modellekkel kísérleteztünk kevesebb tanítóanyaggal és kisebb erőforrással. Modelljeink megtalálhatóak a Github<sup>1</sup> és Hugging Face<sup>2</sup> oldalainkon.

## 2. Felhasznált modellek és korpuszok

### 2.1. ELECTRA és ELECTRIC

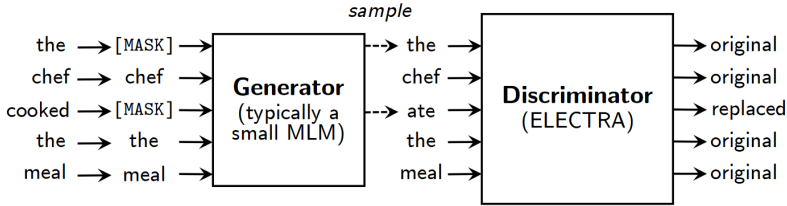
Az ELECTRA (Clark és mtsai, 2020a) az úgynevezett GAN (Generative Adversarial Network) (Goodfellow és mtsai, 2014) módszeren alapul (lásd 1 ábra). A módszer alapját az képezi, hogy két hálózat, egy generátor és egy diszkriminátor van betanítva. Az előtanítás során a generátor véletlenül képzett vektorrepresentációkból álló kimenetet állít elő. Ezt követően az eredeti kimenet bemutatásra kerül a generátornak, amely ez alapján javítja a véletlen vektor generálási folyamatot. A folyamat eredményeként a generátor egyre okosabb lesz a betanítási folyamat végére és képes lesz az eredetihez hasonló kimenet létrehozatalára. Mindeközben a diszkriminátor annak a megkülönböztetésére van betanítva, hogy egy adott adathalmaz eredeti-e vagy hamis. Ennek érdekében a betanító korpuszban szereplő eredeti szövegek és a generátor által képzett szövegek lesznek bemutatva a diszkriminátornak. A BERT szómaszkoló megközelítésével ellentétben itt a rendszer feladata nem az, hogy kitalálja a maszkolt szavakat, hanem hogy azt megtanulja, hogy egy adott szó eredeti-e vagy sem. Fontos kiemelni, hogy a rendszernek az összes szóra el kell végeznie ezt a döntéshozatali folyamatot és nemcsak a teljes szöveg véletlenül kiválasztott 15 százalékára. A két hálózat munkája összekapcsolt és kölcsönösen segítő a betanítási folyamat során. Az ELECTRA oly módon alakítja át a GAN módszert, hogy az képes legyen nyelvmodellek betanítására. Az ELECTRA abban különbözik a BERT modelltől, hogy a hálózat nem maszkolt szavakat prediktál, hanem a generátor állít elő maszkolt szavakhoz tartozó szavakat, majd a diszkriminátor van betanítva a szavak eredetiségének eldöntésére. Tehát a generátor feladata annak megtanulása, hogy mely szavak illeszkednek leginkább a maszkolt szavakhoz, míg a diszkriminátor azt tanulja meg, hogy a szavak egy adott bemeneti szövegben eredetiek-e vagy sem. A betanítást követően a generátor elvetésre kerül, csak a diszkriminátor lesz felhasználva a soron következő finomhangolás során. Google Inc. három különböző ELECTRA modellt fejlesztett ki:

- ELECTRA small: 12 réteg, rejtett réteg mérete: 256; 14 millió paraméter
- ELECTRA base: 12 réteg, rejtett réteg mérete: 768; 110 millió paraméter
- ELECTRA large: 24 réteg, rejtett réteg mérete: 1024; 335 millió paraméter

Az ELECTRA small modell igényli a legkevesebb erőforrást, ezért kutatásaink során csak ezzel a modellel végeztük kísérleteinket.

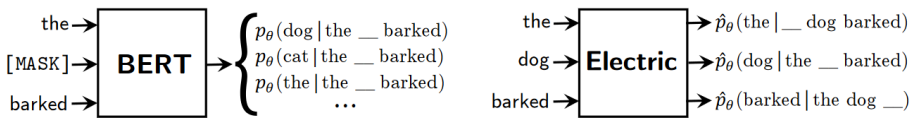
<sup>1</sup> <https://hilanco.github.io>

<sup>2</sup> <https://huggingface.co/hilanco>



1. ábra: ELECTRA modell (Clark és mtsai, 2020a)

Az ELECTRIC az ELECTRA egy változata (Clark és mtsai, 2020b). Az alapelv teljesen megegyezik az ELECTRA módszerével. Az ELECTRA osztályozója (diszkriminátor) egy szigmoid réteg a transzformer réteg fölött, ahol azt az információt, hogy egy adott token igazi-e vagy sem két tényezőből számolják ki: az adatok valószínűségi eloszlásából vagyis abból, hogy mi a valószínűsége annak, hogy egy adott token az adott környezetben fordul elő, valamint a hozzá tartozó zajeloszlásból, amit szintén megtanul a neurális háló a tanítás folyamán. Az ELECTRA és az ELECTRIC közötti különbség az, hogy az ELECTRIC nem használja a zajeloszlást. Ez a gyakorlatban azt jelenti, hogy az ELECTRA maszkoló algoritmust használ, az ELECTRIC esetében ez nem lehetséges, helyette egy úgynevezett „Two-Tower Cloze Model”-t (Baevski és mtsai, 2019) használ, ami az adott token mindkét oldalán lévő kontextust figyelembe veszi, két transzformer modellt használva. A másik nagy különbség az, hogy az ELECTRA esetében (mint általában a szómaszkolás feladatoknál) a kimeneti eloszlás csak a maszkolt token pozíciókra számolódik ki, az ELECTRIC esetében pedig, mivel nincsen maszkolás, az egész bemeneti szövegre (lásd 2. ábra).



2. ábra: ELECTRIC modell (Clark és mtsai, 2020b)

## 2.2. RoBERTa

A RoBERTa modell a BERT esetében alkalmazott előtanítási procedúrát követi, a teljesítmény növelése érdekében a következő módosításokkal:

- Az előtanító korpusz nagyobb mérete: a BERT előtanító korpusza magában foglalja az angol nyelvű Wikipédiát illetve a 3,4 millió szóból álló, 16 GB

nyers szöveget tartalmazó Google Books korpuszt. A RoBERTa tanításához tízszer nagyobb méretű korpuszt használnak (160 GB adat, amely 5 különböző korpuszból származik és 30 millió szóból áll).

- Több modell betanítási lépés: kísérletes munka el lett végezve 100 ezer, 300 ezer illetve 500 ezer lépéssel. Az eredmények alapján kirajzolódik, hogy több lépéssel jobb eredményt lehet elérni.
- Nagyobb batch méret: kísérletes munka során különböző batch méreteket alkalmaztak, ezek 256, 2000 és 8000. Kimutatták, hogy nagyobb batch méret jobb teljesítményhez vezet. A legjobb eredményt 8000-es batch mérettel érték el. A betanítási lépések száma és a batch méret összefügg egymással, nagyobb batch méret mellett kevesebb betanítási lépéssel is el lehet érni ugyanazt a kimenetelt. Következésképpen, az előtanítás időtartama lerövidül, ugyanakkor a memóriahasználat mértéke azonban megnövekszik. Ezen felül, amennyiben egyszerre növeljük a betanítási lépések számát és a batch méretet, akkor a rendszer teljesítménye exponenciálisan növelhető, ezért a RoBERTa modell szerzői mindkét paraméter együttes növelését alkalmazták.
- Következő mondat predikció (Next Sentence Prediction, rövidítve NSP) feladat kizárása a folyamatból: A szerzők tapasztalatai alapján a következő mondat predikciós feladat nem járul hozzá szignifikánsan a rendszer tanításához, amely ezen feladat kizárását eredményezte.
- Hosszabb bemeneti szövegek: A RoBERTa kihasználja a maximális 512-es szekvenciahosszt. A mondatok szekvenciákba egyesével történő betöltése és a fennmaradó részek üres PAD karakterekkel történő telítése helyett a szekvenciák teljesen fel vannak töltve egészen addig, amíg el nem érik a 512 karakter hosszúságot. A dokumentum vége általános esetben új szekvencia nyitását jelentené, itt azonban egy dokumentum szeparátor beillesztése után a szöveg többi része kerül betöltésre.
- Dinamikus maszkolás: Egy fontos jellegzetesség a dinamikus maszkolás használata a statikus helyett. A hagyományos BERT modellben statikus maszkolást alkalmaznak, miszerint a szöveg előfeldolgozása során annak 15 százaléka kerül maszkolásra, és ez a 15 százalék azonos marad az előtanítás során. Ezzel ellentétben a RoBERTa dinamikus maszkolást használ, ami azt jelenti, hogy a szavak maszkolási mintázata újra van képezve mielőtt a szekvencia bemutatásra kerül a rendszernek.
- BPE kódolás: A RoBERTa a szavak belső reprezentációját a Byte Per Encoding (rövidítve BPE) módszerre (Gage, 1994) kódolja, amely a szó- és karakteralapú reprezentáció hidbridje. A szóelemek karakter ngramok iteratív egyesítésének eredményeként jönnek létre, és ezek nem UNICODE karaktereken alapulnak, hanem bájtokon. Ez egy lényegesen gazdaságosabb megoldásnak bizonyult, továbbá a BPE kódolás kevesebb ismeretlen szót eredményez a BERT által használt WordPiece rendszerrel szemben.

Három különböző RoBERTa modell méret létezik:

- RoBERTa small: 6 réteg; rejtett réteg mérete: 768; 65 millió paraméter
- RoBERTa base: 12 réteg; rejtett réteg mérete: 768; 125 millió paraméter

- RoBERTa large: 24 réteg; rejtett réteg mérete: 1024; 355 millió paraméter

Kutatásaink során a RoBERTa small modellel kísérleteztünk.

### 2.3. Korpuszok az előtanításhoz

Az ELECTRA, ELECTRIC és RoBERTa modellek betanításához a következő korpuszokat használtuk:

- Magyar nyelvű Wikipédia (wiki) (Nemeskey, 2020b): 13.098.808 szegmens (mondatok); 163.772.783 token. ELECTRA és RoBERTa tanításához használtuk.
- NYTK korpusz (nytk) (Feldmann és mtsai, 2021): 283.099.534 szegmens (mondatok); 3.993.873.992 token. ELECTRA és ELECTRIC tanításához használtuk.

Szótár méretei a különböző modellekhez:

- A szótár mérete, amelyet az ELECTRA wiki modell tanításához használtunk: 64.000.
- A szótár mérete, amelyet az ELECTRA nytk 31 modell tanításához használtunk: 31.101.
- A szótár mérete, amelyet az ELECTRA nytk 64 modell tanításához használtunk: 64.000.
- A szótár mérete, amelyet az ELECTRIC nytk modell tanításához használtunk: 31.101.
- A szótár mérete, amelyet a RoBERTa modell tanításához használtunk: 30.000.

### 2.4. Korpuszok a finomhangoláshoz

Az előtanított modellek finomhangolását mondatszintű szentiment analízis feladatra a Precognox Kft.<sup>3</sup> által készített Hungarian Twitter Sentiment Corpus<sup>4</sup> használatával végeztük el. Nemzetközi sztenderdeknek (Wang és mtsai, 2018) megfelelően ezt három részkorpuszra bontottuk:

- 2-osztályzatú (2-class): bináris osztályozású részkorpusz. Az értékeket a következőképpen konvertáltuk: 1-es és 2-es értékek 0-vá, mint negatív értékek, 4-es és 5-ös értékek 1-essé, mint pozitív értékek. A 3-as értéket figyelmen kívül hagytuk, hogy elkerüljük félreérthető eseteket.
  - Tanító korpusz: 2468 szegmens. Tesztelő korpusz: 269 szegmens.
- 3-osztályzatú (3-class): az 1-es és 2-es érték negatív, a 3-as semleges, míg a 4-es és 5-ös értékek pozitív osztályba sorolódnak.
  - Tanító korpusz: 3600 szegmens. Tesztelő korpusz: 400 szegmens.
- 5-osztályzatú (5-class): az eredeti 5 osztályra felbontott skálájú korpusz. 1: nagyon negatív, 2: negatív, 3: semleges, 4: pozitív, 5: nagyon pozitív.

<sup>3</sup> <https://www.precognox.hu>

<sup>4</sup> <http://opendata.hu/dataset/hungarian-twitter-sentiment-corpus>

- Tanító korpusz: 3600 szegmens. Tesztelő korpusz: 400 szegmens.

A névelemfelismerés (Name Entity Recognition, rövidítve NER) feladat finomhangolásához a NYTK-NerKor (NerKor) (Simon és Vadász, 2021) és a Szeged NER (SzNer) korpuszt (Szarvas és mtsai, 2006) használtuk, a főnévi névcsoport felismeréshez (Noun Phrase Recognition, rövidítve NP) pedig a Szeged Treebank korpuszt (Csendes és mtsai, 2005). Az összehasonlíthatóság érdekében ugyanazon korpuszokat használtuk a NER és NP finomhangoláshoz, mint az emBERT kutatás (Nemeskey, 2020a) esetében. A korpuszok mérete:

- NerKor (mondat): tanítóanyag: 1.633.952; validációs anyag: 211.392; tesztanyag: 210.884
- SzNER (mondat): tanítóanyag: 8.484; validációs anyag: 514; tesztanyag: 932
- NP (mondat): tanítóanyag: 65.679; validációs anyag: 8.209; tesztanyag: 8.209

Az extraktív összefoglaló (SUM) feladathoz a HVG online (2012–2020) cikkeit és hozzátartozó leadeket használtuk. A korpusz méretei:

- SZUM (dokumentum): tanítóanyag: 474.660; validációs anyag: 3.000; tesztanyag: 3.000

### 3. Kísérletek

#### 3.1. Előtanítási kísérletek

Minden általunk előtanított modell a HILANCO Consortiumra<sup>5</sup> utaló HIL előtaggal van ellátva, ahol a kutatásban leírt modellek is elérhetőek.

Jelen kutatás keretében különböző architektúrájú ELECTRA, ELECTRIC és RoBERTa modelleket tanítottunk be. Az ELECTRA és ELECTRIC modellek betanításához a Google által rendelkezésre bocsátott kódot<sup>6</sup> használtuk. A következő öt modell lett betanítva, mindegyik modellt (kivéve az HIL-ELECTRIC nytk 10%, amit 100 ezer tanítási lépéssel) 1 millió tanítási lépéssel:

- HIL-ELECTRA wiki: a magyar Wikipédián betanított ELECTRA small modell, a betanítás körülbelül 5 napig tartott.
- HIL-ELECTRA nytk 31: a NYTK korpuszon betanított ELECTRA small modell (szótár méret: 31.101), a betanítás időtartama körülbelül 6 nap.
- HIL-ELECTRA nytk 64: a NYTK korpuszon betanított ELECTRA small modell (szótár méret: 64.000), a betanítás időtartama körülbelül 7 nap.
- HIL-ELECTRIC nytk: NYTK korpuszon betanított ELECTRIC small modell (szótár méret: 31.101), a betanítás időtartama körülbelül 5 nap.
- HIL-ELECTRIC nytk 10%: NYTK korpuszon betanított ELECTRIC small modell (szótár méret: 31.101), a betanítás időtartama körülbelül 0,5 nap (10% időtartam, lásd alább).

<sup>5</sup> <https://hilanco.github.io/>

<sup>6</sup> <https://github.com/google-research/electra>

Mindegyik ELECTRA és ELECTRIC modellek betanításához egy darab GeForce RTX 2080 Ti típusú videokártyát használtunk. A betanításhoz az alapbeállításokat alkalmaztuk (tanulási ráta (learning rate) =  $5e-4$ ; súlycsökkentési ráta (weight decay rate) = 0.01; szóbeágyazási méret (embedding size): 128; 1 millió tanítási lépés); egyedül a batch méret lett megváltoztatva, az ELECTRA nytk 31 modell esetében 90, az ELECTRA nytk 64 modell esetében 80, az ELECTRIC modell esetében 70, magasabb érték alkalmazása nem volt lehetséges a CUDA memória méretkorlátai végett. Mindegyik kísérletnél ki kellett egészíteni a kódot és beállítani a „per process gpu memory fraction” értéket 0,95-re, máskülönben csak a 10%-át használta a GPU-nak.

Az ELECTRIC modelleknél egy külön vizsgálatot végeztünk, ahol megvizsgáltuk, hogy indokolt-e az 1 millió tanítási lépés. Hogy nagyobb legyen a különbség, vettük a 100 ezer tanítási lépésnél (10%) elmentett mentési pontot, majd azzal végeztünk különböző finomhangolásokat.

A RoBERTa betanítása egy 4 GeForce GTX 1080Ti GPU kártyát tartalmazó rendszerrel folyt, mind a 4 GPU-t egyidőben használva. Minden egyes GPU 11 GB memóriával rendelkezik, ez összesen 44 GB felhasználható kapacitást jelentett, amely előtanításra volt fordítható. Az előtanítás teljes időtartama 214 óra volt a kártyánkénti 8-as batch méret mellett, összesítve tehát 32-es batch méretet használtunk. Az előtanítás 1.250.000 lépésből állt, a veszteségi görbe 8,7-ről 2,5-re esett vissza, majd ezen érték körül stabilizálódott. Az előtanításhoz a következő hiperparamétereket alkalmaztuk: tanulási ráta (learning rate) =  $1e-4$ , epoch = 5, batch méret = 8. A RoBERTa modell tanításához a Huggingface utasításait követtük. Egy RoBERTa modell került betanításra:

- HIL-RoBERTa wiki: a magyar Wikipédián betanított RoBERTa small modell, a betanítás időtartama körülbelül 9 nap volt.

### 3.2. Finomhangolások kísérletek

A nyelvi modellek tesztelésére 3 különböző kísérletet végeztünk el:

- Mondatszintű szentiment analízis 2, 3, illetve 5 osztállyal.
- Névelemfelismerés (NER).
- Maximális főnévi csoport felismerés (NP).
- Extraktív összefoglaló generálás (SZUM).

A mondatszintű szentiment analízis esetében, a RoBERTa finomhangolásához a Huggingface által rendelkezésre bocsátott „transformers text classification” könyvtárának<sup>7</sup> segítségével történt. A következő módosított paramétereket használtuk: tanulási ráta (learning rate):  $2e-4$ ; batch méret: 32; maximális szekvenciahossz: 128.

Az ELECTRA és az ELECTRIC modellek a Google által kifejlesztett és felhasználható kód<sup>8</sup> segítségével lettek finomhangolva. A következő módosított

<sup>7</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification>

<sup>8</sup> <https://github.com/google-research/electra>



paramétereket használtuk: tanulási ráta (learning rate):  $2e-4$ ; batch méret: 32, maximális szekvenciahossz: 128. Az összehasonlíthatóság érdekében a predikciós teszt eredményeket 1 és 10 közötti epoch értékeknél rögzítettük. A mondatszintű szentiment analízis értékeléséhez a pontosságot (accuracy) használtuk.

A NER és NP feladatokra történő finomhangolásnál az ELECTRA és az ELECTRIC modellekhez a Google által megadott kódot használtuk (a szentiment analízis során alkalmazottakhoz hasonlóan). A következő módosított paramétereket használtuk: tanulási ráta (learning rate):  $1e-3$ ; batch méret: 4; maximális szekvenciahossz: 128. A kiértékeléshez az F-mértéket használtuk.

A RoBERTa finomhangolása a Huggingface által rendelkezésre bocsátott „transformers token classification” könyvtárát<sup>9</sup> segítségével lett elvégezve. A következő módosított paramétereket alkalmaztuk: tanulási ráta (learning rate):  $1e-4$ ; batch méret: 4; maximális szekvenciahossz: 128. A huBERT kísérletekkel (Nemeskey, 2020a) való jobb összehasonlíthatóság érdekében 4-es epoch értéket használtunk. A NER és NP feladatok kiértékeléséhez az IOB-alapú seqeval módszert (Nakayama, 2018) alkalmaztuk.

Az extraktív összefoglalás finomhangolásához a BertSum (Liu és Lapata, 2019) eszközt használtuk az alapbeállításokkal, integrálva a huBERT, ELECTRA és ELECTRIC modelleket.

## 4. Eredmények

Az 1-es és a 2-es táblázatban összefoglalt eredményeink alapján elmondható, hogy az általunk alkalmazott és betanított modellek nem tudják felülmúlni a jelenlegi state-of-the-art huBERT modellt. Ez az eredmény a várakozásainknak megfelelő, mivel az általunk használt modellek gyengébbek számos jellemzőjük tekintetében (kisebb architektúra, kevesebb betanításra használható adat, kisebb batch méret stb.). Annak ellenére, hogy az általunk használt modellek kevesebb paraméterrel bírnak, meglehetősen magas eredményeket érnek el. A szentiment analízis kísérletekben például a modelljeink csak 5-7 százalékkal maradnak alul a huBERT-hez képest. Továbbá megfigyelhető, hogy a HIL-ELECTRA nytk 64 gyengébben teljesít a HIL-ELECTRA nytk 31 modellhez képest. Ennek egyik oka a batch méretbeli különbség lehet. Az eredményekből az figyelhető még meg, hogy az ELECTRIC modellek jobban teljesítenek mondatszintű osztályozás feladatain az ELECTRA modelleknél, sőt a 10%-os modell, ami fél napot tanult csak, a huBERT-el versenyképes eredményt ért el.

A 2-es táblázat mutatja be a NER és az NP kísérletek eredményeit. Az emBERT (Nemeskey, 2020a) kísérletek során a legjobb eredmények 30 epoch értéknél mutatkoztak, de az összehasonlíthatóság érdekében megismételtük a huBERT kísérleteket 15 epoch értékkel az emBERT cikkben ismertetett kód alapján. Majd mindegyik esetben vettük a 15 epochból a legjobb eredményt. A 2-es táblázat alapján látható, hogy a SzNer és az NP kísérletekben az ELECTRA

<sup>9</sup> <https://github.com/huggingface/transformers/tree/master/examples/pytorch/token-classification>

	2-class	3-class	5-class
huBERT	85,92	72,18	68,50
HIL-ELECTRIC nytk 10%	81,85	66,42	63,25
<b>HIL-ELECTRIC nytk</b>	<b>82,22</b>	<b>68,92</b>	<b>65,25</b>
HIL-ELECTRA wiki	76,95	62,66	58,40
HIL-ELECTRA nytk 31	79,55	67,67	60,90
HIL-ELECTRA nytk 64	77,41	62,91	60,25
HIL-RoBERTa wiki	80,00	64,66	61,00

1. táblázat. Mondatszintű szentiment analízis osztályozás eredménye

small modellünk csak 2-3 százalékkal marad el a huBERT modellétől. A NerKor kísérletben viszont a RoBERTa közel 4 százalékkal marad csak le. A fő különbség a NerKor esetében tapasztalható meg, ahol a small modellek megmutatják a korlátaikat. A huBERT több paraméterrel több dolgot tanul meg. A tokenszintű osztályozás feladatában az ELECTRIC modellek gyengébben teljesítettek az ELECTRA modelleknél.

	NerKor	SzNER	NP
huBERT	90,18	97,51	96,79
HIL-ELECTRIC nytk 10%	72,84	86,01	90,73
HIL-ELECTRIC nytk	78,82	93,63	94,14
HIL-ELECTRA wiki	77,37	94,19	94,14
HIL-ELECTRA nytk 31	79,34	95,39	94,50
HIL-ELECTRA nytk 64	77,35	93,59	94,09
<b>HIL-RoBERTa wiki</b>	<b>86,04</b>	<b>96,25</b>	<b>94,4</b>

2. táblázat. NER és NP eredmények

Kiemelendő, hogy bár az ELECTRA small modellek nem képesek túlteljesíteni a BERT modelleket, az ELECTRA small modellek közel olyan jó eredményeket tudnak elérni kevesebb paraméterrel, mindössze 1 GPU használata mellett.

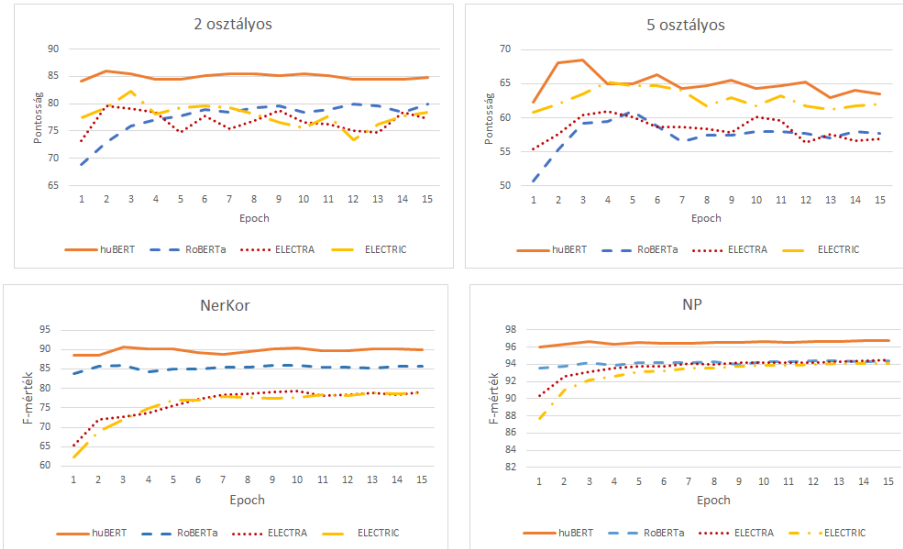
Az 3-es ábra mutatja a mondatszintű szentiment analízis és NerKor kiértékelését az epoch értékek függvényében.

Általánosan elmondható, hogy az ELECTRA és ELECTRIC modellek a mondatszintű szövegosztályozás feladataiban hamarabb elérik az optimális szintet, mint a RoBERTa modell.

A tokenszintű osztályozásnál az látható, hogy mindegyik modell később éri csak el a maximumot, sőt azt látjuk, hogy még nem érték el az optimumot, Nemeskey (2020a) kutatásában 30 epochnál érte el a maximumot.

Érdekes még továbbá, hogy az ELECTRA a tokenszintű osztályozás esetében magasabb eredményt ér el, mint az ELECTRIC, míg a mondatszintű szentiment analízis esetében fordított a helyzet. Ezt az előtanítás különözöségének tulaj-

donítjuk, tekintve, hogy az ELECTRA, a szómaszkolási feladat révén, jobban teljesít a tokenszintű feladatban.



3. ábra: Szentiment analízis, NerKor és NP kísérlet eredményei az epoch függvényében

A 3. táblázatban láthatóak az extraktív összegzés eredményei. A kiértékeléshez a nemzetközileg használt ROUGE (Lin, 2004) mértéket (ROUGE-1/ROUGE-2/ROUGE-L) használtuk. Hogy össze tudjuk hasonlítani a korábbi kutatás eredményeivel (Yang és mtsai, 2021), az eredmények táblázatába betettük a fedés értékeket, azonban a nemzetközi kutatásokban az F-mérték a mérvadó, ezért mindkét eredményt feltüntettük. Várakozásainknak megfelelően a huBERT éri el a legjobb eredményt. Az ELECTRA és ELECTRIC modelljeink is alulteljesítenek, de csak nagyon kicsi mértékkel. Annak ellenére, hogy csak 1 GPU-n tanult, architektúrájában is kisebb és kevesebb anyagon tanult, az ELECTRA modelljeink versenyképes eredményt értek el.

Az ELECTRIC kísérletekben az mutatható ki, hogy bár az 1 millió lépéssel magasabb eredményt tudtunk elérni, mint a fél napos 100 ezer tanítási lépéssel, azonban eredményekben nem ér el szignifikánsabban magasabb értéket az 1 millió lépéses modell. Energiafelhasználás és eredmény függvényében úgy gondoljuk a 10%-os modell is elegendő lehet.

	Fedés	F-mérték
huBERT	49,45/21,07/40,14	27,35/10,78/21,97
HIL-ELECTRIC nytk 10%	49,05/20,54/39,77	26,38/10,13/21,16
<b>HIL-ELECTRIC nytk</b>	<b>49,07/20,56/39,79</b>	<b>26,40/10,14/21,17</b>
HIL-ELECTRA wiki	49,02/20,52/39,74	26,36/10,11/21,13
HIL-ELECTRA nytk 31	49,04/20,53/39,76	26,37/10,12/21,15
HIL-ELECTRA nytk 64	48,99/20,51/39,70	26,38/10,13/21,13

3. táblázat. Extraktív összefoglalás ROUGE eredményei

## 5. Összegzés

Kutatásunk során három magyar nyelvű modellel kísérleteztünk, kevés erőforrással. Előtanítottunk ELECTRA, ELECTRIC és RoBERTa modelleket, majd leteszteltük a teljesítményüket különböző nyelvtchnológiai feladatokon. A teszteléshez az előtanított modelleket finomhangoltuk. Három mondatszintű szentiment analízis feladat, kettő névelemfelismerés, egy főnévi csoport azonosítás és egy extraktív összefoglaló generálás került alkalmazásra a modellek teljesítményének megítélésére. Mindegyik feladatban várakozásainknak megfelelően, az általunk betanított modellek nem tudták felülmúlni a state-of-the-art-nak számító huBERT modelleket, de versenyképesen magas eredményeket produkáltak a kevesebb paraméter, az ELECTRIC esetében kevesebb idő (fél nap) és a kevesebb betanításhoz használt adat ellenére.

Eredményeink figyelemre méltóak, mert lényegesen kisebb hardverigénnyel és kevesebb adattal vagyunk képesek versenyképes és magas színvonalú eredményeket elérni, amely többek között két fontos előnnyel bír. Egyrészt, az ilyen modellek használata csökkenti a használatból fakadó ökológiai lábnyom méretét kevesebb elektromos energia felhasználása révén, másrészt, a kisebb méretű modellek alkalmazása felhasználóbarát, hiszen kevesebb tárhelyre van szükség és gyorsabban lefutathatók. Mindamellet, ezen modellek csökkentett komplexitása növelheti a kísérletező kedvet más területek felhasználói körében, amely pozitívan hat a tudástranszferre különböző tudományterületek között, például a számítógépes lingvisztika és a kognitív infokommunikáció között.

Jelenleg további kísérleteink vannak folyamatban magyar nyelvű GPT2 (Radford és mtsai, 2019), ALBERT (Lan és mtsai, 2019), mondat-BERT (sentence BERT) (Reimers és Gurevych, 2019) és BART (Lewis és mtsai, 2020) modellekkel.

## Hivatkozások

Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L., Auli, M.: Cloze-driven pre-training of self-attention networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5360–5369. Association for Computational Linguistics, Hong Kong, China (Nov 2019)

- Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training text encoders as discriminators rather than generators. In: ICLR (2020a)
- Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Pre-training transformers as energy-based cloze models. In: EMNLP (2020b), <https://www.aclweb.org/anthology/2020.emnlp-main.20.pdf>
- Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (szerk.) Text, Speech and Dialogue. pp. 123–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
- Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Gage, P.: A new algorithm for data compression. *C Users J.* 12(2), 23–38 (Feb 1994)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (szerk.) *Advances in Neural Information Processing Systems*. vol. 27, pp. 2672–2680. Curran Associates, Inc. (2014)
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR* (2019)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
- Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out*. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
- Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: *IJCNLP*. Hong Kong, China (2019)
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. *CoRR* (2019)
- Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *CoRR abs/1301.3781* (2013a)

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* pp. 3111–3119 (2013b)
- Nakayama, H.: *seqeval: A python framework for sequence labeling evaluation* (2018), <https://github.com/chakki-works/seqeval>, software available from <https://github.com/chakki-works/seqeval>
- Nemeskey, D.M.: Egy embERT próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged (2020a)
- Nemeskey, D.M.: *Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University* (2020b)
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
- Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
- Schuster, M., Nakajima, K.: Japanese and korean voice search. In: *ICASSP*. pp. 5149–5152. IEEE (2012)
- Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) *Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science*, vol. 12848, pp. 222–234. Springer (2021)
- Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. In: Todorovski, L., Lavrač, N., Jantke, K.P. (szerk.) *Discovery Science*. pp. 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc. (2017)

- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018)
- Yang, Z.G., Agócs, Á., Kúspér, G., Váradi, T.: Abstractive text summarization for hungarian. *Annales Mathematicae et Informaticae* 53, 299–316 (2021)





## Közérthetőség mint osztályozási probléma (?) - gépi tanulási kísérlet kézzel címkézett korpuszon

Üveges István<sup>1,2</sup>

<sup>1</sup>Szegedi Tudományegyetem, Nyelvtudományi Doktori Iskola

<sup>2</sup>MONTANA Tudásmenedzsment Kft.

uvegesistvan898@gmail.com, uvegesi@montana.hu

**Kivonat** Cikkünkben bemutatjuk a laikusoknak címzett hivatalos szövegek osztályozási kísérletét felügyelt gépi tanuló algoritmusok segítségével. Vizsgálatunkhoz szakértők által, kézzel készített korpuszt használtunk, amely közérthetőre fogalmazott és még átfogalmazás előtt álló mondatokat tartalmazott. Célunk ezzel egy olyan gépi tanult modell készítése, amely alkalmas lehet a szakértők figyelmét felhívni azon mondatokra egy-egy hivatalos szövegben, amelyek további megfontolást érdemelnek a szöveg közérthetőbbre alakítása során, ezzel gyorsítva a szakértői munkát. A kísérletet pilot jelleggel végezzük, az eredmények függvényében korszerűbb módszerek (pl. LSTM, BERT) esetleges kipróbálása előtt, a tapasztalatokat pedig a fentiek szerinti bináris klasszifikációs problémára jellemzően értékeljük.

**Kulcsszavak:** közérthetőség, Plain Language Movement (PLM), Support Vector Machine (SVM), kézzel címkézett korpusz

### 1. Bevezetés

A jogállamiság egyik követelménye a jog átláthatósága és kiszámíthatósága. Ez a fajta kiszámíthatóság különösen hangsúlyosan jelenik meg az olyan helyzetekben, amikor laikus, az adott szakma nyelvét "nem beszélő" célközönség találkozik állami szervek által neki címzett hivatalos / gyakran jogi tartalmú dokumentumokkal (Dobos, 2015; Vinnai, 2018; Tóth, 2019). Az USA-ból a múlt században indult Plain Language Movement célja ezen átláthatóság előmozdítása a nem szakértő címzettek számára azáltal, hogy a hivatalos dokumentumok megszövegezéséhez a közérthetőséget segítő (nyelvi) javaslatokat tesz.

A gyakorlatban azonban egy effajta kontroll rendkívül nehéz feladat, amelyet csak az adott területen kompetens szakértők képesek végrehajtani. A jelen tanulmány célja kettős; egyrésztől kísérleti jelleggel választ keres arra a kérdésre, hogy a megfelelő tanító adatok birtokában értelmezhetjük-e a közérthető megfogalmazást osztályozási problémaként; erre vonatkozóan az elért eredmények adhatnak támpontot. Másrésztől, amennyiben egy ilyen értelmezés kivitelezhetőnek bizonyul, olyan gépi tanult modell készítése, amellyel lehetőség nyílik a szakértői munka támogatására a különös figyelmet igénylő mondatok címkézésével az átfogalmazás előtt álló szövegekben.

A tanulmány a következők szerint épül fel; a 2 fejezetben röviden áttekintjük a "közérthetőség" főbb történeti megközelítéseivel foglalkozó fontosabb szakirodalmat. A 3 és a 4. fejezet ismerteti a kísérletünk alapjául szolgáló korpusz tulajdonságait keletkezését. Ezt követően az 5. fejezet ismerteti az első gépi tanulási eredményeket, amelyeken a 6. fejezet során hiperparaméter hangolással végzünk optimalizálást. A tanulmányt ezt követően rövid konklúzió zárja.

## 2. Kapcsolódó irodalom

A jogi / hivatalos szövegek közérthetőségét az elmúlt évtizedekben sokan sok különböző nézőpontból vizsgálták. A teljesség igénye nélkül, a történetileg legkorábbi módszerek közül érdemes megemlíteni például az olvashatósági formulák (*readability formulas*) alkalmazását, amelyek a szavak átlagos szótagszámának, valamint a mondatok átlagos szószámának arányából vonnak le következtetéseket a szöveg befogadhatóságára vonatkozóan (Edgar és Jeanne S., 1948; Dubay, 2004; Üveges, 2020). Ezek virágkorukat a múlt század közepén élték, de elterjedtségüket jól jelzi, hogy a MS Word szövegszerkesztőjében (az angol verzió esetében) a mai napig megtalálhatók a szöveget minősítő opcionális indikátorként.

Kognitív szempontból az érthetőség kérdésével a nyelvtudományon belül a pszicholingvisztika foglalkozik behatóan. A pszicholingvisztikai szakirodalom (Pléh és Lukács, 2014; Kas és Lukács, 2012; Pléh, 2013) állásfoglalása sokkal kevésbé mechanikus, egyben nem is feltétlenül jól automatizálható megoldást kínál. Az érthetőséget gátló tényezőket nem a mondat, hanem sokkal inkább a pragmatikai kontextus és a szöveg, mint egység "globális" szintjén helyezi el, és csak kevesebb olyan aspektust azonosít, amelyek a mondat határán belül detektálhatók (ilyen például a főmondat megszakíttósága).

Ebbe a sodorba illeszkedik be a jelen tanulmány keretét adó irányvonal, az USA-ban a '70-es években indult Plain Language Movement (PLM) is, amely főleg a kommunikáció hatékonyságának javítása felől közelíti meg a kérdést, és amelynek nyomán az Egyesült Államokban több mint egy évtizede törvény<sup>1</sup> is szabályozza az ottani hivatalok tájékoztató anyagaiban érvényesítendő közérthetőségi szempont betartását (Felsenfeld és mtsai, 1981; Cutts, 1999; Garner, 2001; Willerton, 2015). A dokumentum a közérthetőség feltételeit a következők szerint definiálja: "Világos, tömör, jól szervezett és a témának vagy területnek és a célközönségnek megfelelő egyéb bevált gyakorlatokat követő írás"<sup>2</sup>.

Egy másik közkeletű megfogalmazásban egy szöveg akkor közérthető, ha az olvasóját képessé teszi arra, hogy:

- megtalálja, amire szüksége van,
- megértse, amit talál, amikor először olvassa vagy hallja a szöveget,

<sup>1</sup> Public Law 111 - 274 - Plain Writing Act of 2010

<sup>2</sup> Writing that is clear, concise, well-organized, and follows other best practices appropriate to the subject or field and intended audience.

- felhasználja azt, amit talál a szükségletei érdekében<sup>3</sup>.

A dominánsan pragmatikai központú meghatározás ellenére a PLM számos gyakorlati megoldást is szorgalmaz, amelyek stiláris-, szintaktikai vagy lexikai átalakítások útján segít(het)ik az olvasót a szöveg könnyebb befogadásában.

Az amerikai kormányzat 2011-ben kiadott szövetségi közérthetőségi iránymutatása, a Federal Plain Language Guidelines<sup>4</sup> talán a legteljesebb összefoglalója ezen javaslatoknak, amely többek között ajánlja cselekvő igék használatát passzív szerkesztés helyett, a nominalizáció kerülését, a funkcióigék túlhasználatának mellőzését vagy éppen az átlagos mondathossz lehetőség szerinti csökkentését.

A magyar szakirodalomban az általános értelemben laikusok nézőpontja főként a "jog és nyelv" kutatások kapcsán került előtérbe. Középpontjában jellemzően az a gondolat áll, miszerint a joghoz való egyenlő hozzáférés alappillére (egyebek mellett) a mindenki számára egyenlően érhető megfogalmazás követelményének érvényre jutása is (Vinnai, 2014; Szabó és Vinnai, 2018a; Minya és Vinnai, 2018).

A jelen tanulmány keretében a vizsgálódás elvi alapját főként a PLM keretrendszerében lefektetett elveknek és ajánlásoknak a széles körben elérhetővé és alkalmazhatóvá tétele motiválja.

### 3. Gépi reprodukálhatóság

A Plain Language által elvárt stiláris, lexikai jellegzetességeknek, preferált és diszpreferált fogalmazási módoknak ugyanakkor csak egy korlátozott része lehet az, amely a mondat szintjén belül szintaktikai és / vagy lexikai jegyek összességére könnyen lefordítható. Nagyobb részüik olyan, a teljes szöveget, vagy éppen az olvasó szempontjait tekintetbe vevő elveket fogalmaz meg, amelyek a hagyományos NLP eszközökkel nem, vagy csak nehezen megragadhatók.

A 2. fejezetben említett elvek az algoritmikus megvalósítás szempontjából sok esetben közvetlenül túlságosan absztraktak, ezek alkalmazása szakértői feladat, amelyhez szükséges a címzettek nézőpontjának és vélhetőleges kontextuális tudásának beható ismerete is. Különösen igaz ez, ha tekintetbe vesszük, hogy más-más hivatali közegeben a gyakorlatban eltérő elvek vezethetnek érthetőbb megfogalmazáshoz.

#### 3.1. Felügyelt gépi tanulási módszerek

A felügyelt gépi tanulási módszerek, mint amilyen a Naive Bayes modell (NB), a Support Vector Machine (SVM) vagy éppen a Logistic Regression (LR) számos

<sup>3</sup> "Material is in plain language if your audience can: find what they need, understand what they find the first time they read or hear it, use what they find to meet their needs." Forrás: <https://www.plainlanguage.gov/about/definitions/> (Elérés: 2021.12.27.)

<sup>4</sup> Federal Plain Language Guidelines, US Government, March 2011, online: <https://www.plainlanguage.gov/media/FederalPLGuidelines.pdf> (Elérés: 2021.12.27.)

nyelvtudományi probléma esetében szolgáltatott már hatékony megoldásokat, amennyiben például az adott kérdés lefordítható volt osztályozási problémára például a szentimentelemzésben (Chauhan, 2017) vagy a spam-detekció területén (Sun és mtsai, 2020).

Bár szükségszerű korlátokkal, de hasonló osztályozás elképzelhető lehet a közérthetőség kérdésében is, amennyiben rendelkezünk elegendő mennyiségű tanítóadattal, amelyet szakértők címkéztek fel az adott szakterületnek és címzetti körnek megfelelően "közérthető" és "nem közérthető" címkékkel.

### 3.2. Nemzeti Adó- és Vámhivatal - Közérthetőségi Program

A Nemzeti Adó- és Vámhivatal Kommunikációs Főosztályának Médiaosztálya immár három éve működtet Közérthetőségi Programot, amelynek keretében három szakértő a hivatal valamennyi kommunikációs anyagát ellenőrzi, és azt javaslatokkal korrektúrázza. Az átvizsgált dokumentumok ezt követően még egy végső szakmai ellenőrzés után kerülnek publikálásra. Ezekben a változatokon ismételt ellenőrzés a Közérthetőségi Program munkatársai részéről már nem történik.

A hivatal munkatársai a jelen kutatáshoz rendelkezésre bocsátották a teljes 2021-ben korrektúrázott anyagot, amelyben követhetők az újra fogalmazás előtti és a már átírt anyagok. Ennek köszönhetően a fent említett gépi tanulási kísérlet elvben megvalósíthatóvá vált. Fontos kiemelni, hogy a kísérletben felhasznált dokumentumverziók tehát a záró szakmai szempontú ellenőrzés előtti szöveget tartalmazzák.

## 4. A korpusz rövid bemutatása

Az átírt dokumentumokból minden esetben rendelkezésre állt tehát két változat; egy eredeti és egy szakértők által korrektúrázott változat; előbbire a továbbiakban **eredeti** -ként, míg utóbbira **átfogalmazott**-ként utalunk. Mivel a jelen kutatás nagyban kapcsolódik a PLM által propagált elvekhez, így érdemes megemlíteni, hogy a szakértőkkel folytatott beszélgetés, valamint az általuk a NAV munkatársai számára készített segédletek, belső kommunikációs anyagok alapján az átfogalmazáskor alkalmazott elvek túlnyomó többsége metszetet képez a PLM elvárásaival (pl. az olvasó szempontjainak középpontba helyezése, a funkcióigék kerülése stb.).

A fentieket alapul véve a keletkezett korpusz ideális választásnak tűnik annak vizsgálatára, hogy egy olyan absztrakt és nehezen körülhatárolható fogalom, mint a "közérthetőség" megragadható-e valamilyen szinten korpusznyelvészeti és gépi tanulási módszerek alkalmazásával is?

Az eredeti korpuszt a következők szerint osztottuk részekre:

- minden dokumentumból rendelkezésre állt **eredeti**, és **átfogalmazott** verzió is. Az összetartozást a fájlok elnevezési konvenciója kódolta; a közös, korpuszban egyedi prefix után "A" jelölte az **eredeti**, "B" pedig az **átfogalmazott** szöveget (pl.: A44A és A44B),

- a dokumentum-párokat a spaCy (Honnibal és Montani, 2017) természetesnyelvi elemző segítségével és az Orosz György által készített nyelvmodell<sup>5</sup> alkalmazva mondatokra szegmentáltuk,
- az így kapott mondatok közül eltávolítottuk azokat, amelyek a dokumentumpár mindkét tagjában azonosan fellelhetők voltak.

A megmaradt mondatok a szülő fájlok fájlneveinek posztfixe ("A" vagy "B") alapján kerültek besorolásra vagy az *eredeti*, vagy az *átfogalmazott* szövegek részkorpuszába.

A válogatás eredményképpen 5710 mondat került az *eredeti*, és 3010 mondat az *átfogalmazott* részkorpuszba, összesen több mint 270 ezer token terjedelemben (az adatokat részletesen az 1. táblázat szemlélteti<sup>6</sup>).

Szófaj	Átfogalmazott	Átfogalmazás előtti	Arány	
token	85754	186900	<b>0,3145</b>	<b>0,6854</b>
NOUN	25649	57792	0,2991	0,3092
ADJ	14293	32716	0,1667	0,175
PRON	2634	5929	0,0307	0,0317
CONJ	3386	6784	0,0395	0,0363
NUM	1502	3763	0,0175	0,0201
VERB	5543	11579	0,0646	0,062
ADV	3521	7289	0,0411	0,039
PROPN	2826	5245	0,033	0,0281
ADP	1238	3086	0,0144	0,0165
AUX	0	0	0	0
DET	9874	20578	0,1151	0,1101
INTJ	13	42	0,0002	0,0002
PART	241	558	0,0028	0,003
PUNCT	13681	28545	0,1595	0,1527
SCONJ	1151	2566	0,0134	0,0137
SYM	60	97	0,0007	0,0005
X	142	331	0,0017	0,0018

1. táblázat. Az egyes részkorpuszok szófaji statisztikái

A két részkorpusz méretének eltérését főként az a jelenség okozza, hogy az *eredeti* szövegek mondatai első lépésben sok esetben jelentősen rövidültek (például a jogszabály hivatkozások lábjegyzetbe utalása, vagy a nem elengedhetetlen kifejezések eltávolítása miatt), amely által több, korábban különálló mondat

<sup>5</sup> <https://github.com/spacy-hu/spacy-hungarian-models>

<sup>6</sup> token - tokenszám, ADJ – melléknév, ADV – határozószó, ADP – névutó, AUX – segédige, CONJ – mellérendelő kötőszó, DET – névelő, INTJ – indulatszó, NOUN – főnév, NUM – számnév, PART – igekötő, PRON – névmás, PROPN – tulajdonnév, PUNCT – központozás, SCONJ – alárendelő kötőszó, SYM – szimbólum, VERB – ige, X – egyéb

összeolvasztására volt lehetőség az átalakítás során. Érdeemes megemlíteni, hogy az **átfogalmazott** mondatok átlagos tokenszáma még ezzel együtt is alacsonyabb (28,48), mint az **eredeti** mondatoké (32,73) amely jól harmonizál a 2. fejezetben megfogalmazottakkal.

Az átalakításokra néhány példát az (1) - (2), valamint a (3) - (4) párok szolgáltatnak, melyek esetében a pár első tagja az **eredeti**, a második pedig az **átfogalmazott** változat.

- (1) A Bevezető felületen található tájékoztató szövegek segítik a felhasználót az alkalmazás megismerésében. Az alkalmazás megismerését a felhasználó a Bevezető felületen található Tovább funkciógombra való kattintással tudja elkezdni, valamint folytatni.
- (2) A Bevezető tájékoztató szövegei segítik a felhasználót az alkalmazás megismerésében. Ezt a felhasználó a Bevezető felületen található Tovább funkciógombra kattintva tudja elkezdni.

A (3) - (4) jó példa arra, amikor a szövegből a funkcióige ("kerül") eltávolításával a szerkezet egyszerűsíthető. A funkcióigés konstrukciók esetében a jelentést voltaképpen a kifejezés névszói tagja hordozza, a szerkezet egésze pedig (kivéve terminus technicusok esetében) helyettesíthető egyetlen igével, mint a "bemutatásra kerül" (=bemutatják), vagy a "ellenállást tanúsít" (=ellenáll) esetében (Lanstyák, 2020).

- (3) A felugró ablakban a Mégsem funkciógombra való kattintás hatására az eredetileg elmentett összekapcsolás megmarad és **nem kerül felülírásra** az új összerendeléssel.
- (4) A felugró ablakban a Mégsem funkciógombra kattintva az eredetileg elmentett összekapcsolás megmarad és az új összerendelés **nem írja felül**.

A korpusz jellemzői kapcsán lényeges információ lehet annak hasonlósága más jogi / hivatalos doménbeli szövegekhez. Ehhez az összevetéshez a Miskolc Jogi Korpusz (MJK) részkorpuszainak szöveganyagát használtuk fel, a hasonlóság kifejezésére pedig a szókincs alapú Jaccard-távolságot (Hancock, 2004) alkalmaztuk. A részkorpuszonkénti összevetés amiatt lehet szükséges, mivel a MJK, bár domén tekintetében közös szövegeket tartalmaz, mégsem tekinthető egységessnek; a jogi fórumok szövegei például inkább a köznyelv, míg a törvényszövegek a formális jogi nyelvhasználat prototipikus esetei felé tendálnak (Szabó és Vinnai, 2018b).

Jaccard távolság szerint az **eredeti**, és a már **átfogalmazott** szövegek közötti különbség azonnal szembeötlő; a metrika 0,61-es értéke a szókincs jelentős eltérését mutatja. A MJK részkorpuszaival vett hasonlóságokat a 2. táblázat mutatja be (a feltüntetett összevetések: fórum - jogi fórumok szövegei, átirat - bírósági tárgyalásokon és rendőrségi kihallgatásokon készített átiratok, jogszabályok - jogszabályok szövegrészletei, ítéletek - bírósági és törvényszéki ítéletek, metanyelv - jogi tankönyvek, miniszteri indoklások szövege, kódexjog - törvények szövegei).

	Átfogalmazott	Eredeti
<b>fórum</b>	<b>0,85</b>	0,83
<b>átirat</b>	0,83	0,83
<b>jogszabályok</b>	0,82	0,79
<b>ítéletek</b>	0,82	0,79
<b>metanyelv</b>	0,81	0,77
<b>kódexjog</b>	0,76	0,74

2. táblázat. Az *átfogalmazott* / *eredeti* részkorpuszok lexikai hasonlósága a MJK egyes részkorpuszaival.

Az eredmények alapján mindamellet, hogy az *átfogalmazott* szövegek szó-kincse jelentősen eltávolodott az *eredeti* verziótól, eközben egy kivétellel a MJK valamennyi részkorpuszához minimálisan közeledett (kivételt ez alól az átíratok szövegei képeznek, amelyek azonban beszélt nyelvi közlések leiratai, ilyenformán maguk is valamelyest önálló csoportot alkotnak).

Fontos azonban kiemelni, hogy mind az *eredeti*, mind az *átfogalmazott* szövegek a MJK leginkább "köznyelvi"-nek tekinthető részeihez a leghasonlóbbak, vagyis a fórumok, és az átíratok szövegeihez.

## 5. Gépi tanítási kísérlet

Az előfeldolgozás standard lépéseket foglalt magában: kiszűrtük a stopszavakat, eltávolítottuk a mondatokból a számjeggyel írt számokat, kisbetűsítettük az egyes szavakat, valamint a spaCy segítségével lemmatizáltuk azokat, majd az így kapott, normalizált mondatokat (amely a szótárat és az IDF súlyokat a tanító / validációs halmazon tanulta) TF-IDF vektorizáltuk (uni- és bigramokat is megengedve).

A továbbiakban három gépi tanulási algoritmust alkalmaztunk; a korábban már említett (Bernoulli) Naive Bayes (NB), Support Vector Machine (SVM) és Logistic Regression (LR) módszereket. Első közelítésben alapértelmezett paraméterezéssel kíséreltük meg a tanítást. A kapott eredményeket a 3. táblázat szemlélteti, ahol az első oszlop a választott módszert, a második pedig a szövegosztályt mutatja. A tanító és teszhalmazok felosztásakor a 90% - 10% arányt választottuk.

Mindhárom módszer esetén az *eredeti* mondatok felismerése volt hatékonyabb, a legjobb eredményt e tekintetben a LR hozta; itt az F értéke 0,77 volt. A leginkább kiegyensúlyozott eredmény ezzel szemben a SVM esetében volt megfigyelhető; itt volt a legkisebb eltérés a két kategória F-értékei között.

## 6. Hiperparaméter-optimalizálás

A fenti eredmények arra engedtek következtetni, hogy a korpuszban rendelkezésre állhat elegendő tanítóadat jobb eredmények eléréséhez is. A hatékonyság

		Pontosság	Fedés	F-érték
<b>NB</b>	Eredeti	0,64	0,81	0,71
	Átfogalmazott	0,25	0,12	0,16
<b>SVM</b>	Eredeti	0,67	0,63	0,65
	Átfogalmazott	0,37	0,42	0,39
<b>LR</b>	Eredeti	0,68	0,87	0,77
	Átfogalmazott	0,49	0,23	0,31

3. táblázat. Az kipróbált gépi tanuló algoritmusok teljesítménye alapbeállításokkal.

maximalizálása érdekében kísérletet tettünk rá, hogy az egyes modellek hiperparamétereit hangoljuk.

A rendelkezésre álló korpuszt a későbbiekben tanító-, validációs- és tesztalmanazra bontottuk 80%-10%-10% arányban. Az optimalizálást a validációs halmazon végeztük, illetve az adatokon 10-szeres keresztvalidációt alkalmaztunk. A korpusz címkéinek kiegyensúlyozatlansága miatt a modelleken minden esetben a `class_weight='balanced'` beállítást használtuk.

### 6.1. SVM

A gépi tanuló algoritmusok általánosan kétféle paraméter készlettel rendelkeznek; a modell paraméterek tanulása (vagy becslése) a gépi tanulási folyamat része, a hiperparaméterek viszont közvetlenül nem tanulhatók a rendelkezésre álló adatokból, emiatt azok optimalizálása kézzel történhet meg. Ez utóbbi tulajdonságuk ellenére optimális beállításuk nagyban befolyásolhatja az algoritmus teljesítményét.

SVM esetében ilyen hiperparaméterek lehetnek például a  $C$  (regularizáció - amely az egyes osztályokat elválasztó hipersík margin-jának nagyságát szabályozza), a  $\gamma$  (amely arra hat ki, hogy a potenciális szeparáló határ megválasztásakor attól mennyire távoli pontok játszanak még szerepet a döntésben) illetve a kernel, avagy magfüggvény (amely a hipersík kiszámítási módjának egyenletét határozza meg). A jelen kísérletben a következő lehetséges értékek valamennyi (összesen  $4^3 = 64$ ) kombinációjával tanítottunk modelleket, majd ezeket értékeltük ki:

- $C$ : [0,1, 1, 10, 100]
- $\gamma$ : [1, 0,1, 0,01, 0,001]
- $kernel$ : ['rbf', 'poly', 'sigmoid', 'linear'].

Tekintettel arra, hogy az esetleges gyakorlati alkalmazás során fontosabbnak tartottuk biztosítani, hogy a modell által problémásnak ítélt mondatok valóban megvizsgálandók legyenek a szakértők által, mint azt, hogy nagy számú mondatot jelöljünk ellenőrzésre, ezért az optimalizálás során az **eredeti** szövegosztály predikcióinak pontosság (Precision) értékét igyekeztünk maximalizálni.



Mindezen feltételek mellett a legjobb eredményt  $\{C=100, \textit{Gamma}=0.001, \textit{kernel}='rbf'\}$  választással értük el (a legjobb 5 modell eredményeit a 4. táblázat mutatja be).

Paraméterek: [C,Gamma,kernel]	(Átlagos-) pontosság	Szórás
100, 0,001, rbf	<b>0,767</b>	0,02
10, 0,01, rbf	0,765	0,03
1, 1, sigmoid	0,755	0,02
1, 0,1, rbf	0,752	0,03
0,1, 1, sigmoid	0,736	0,03

4. táblázat. A legjobb eredményt (átfogalmazandó mondatok - pontosság) adó beállítások (SVM).

## 6.2. LR

A SVM-mel ellentétben Logisztikus Regresszió esetén kevésbé ismert, hogy a lehetséges hiperparaméterek változtatása garantáltan javítaná a predikciós teljesítményt. Ennek ellenére (tekintettel arra, hogy a kiinduló modellek közül alapbeállításokkal ez érte el a legígéretesebb eredményt) itt is kísérletet tettünk a következő beállítások használatával:

- $C$ : [0,01, 0,1, 1, 10, 100, 1000]
- $solver$ : ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'].

ahol  $C$  ismét a regularizációért felelt, a  $solver$  pedig a probléma megoldásáért felelős algoritmus kiválasztását tette lehetővé.

Az eredmények értékelése során megfigyeltük, hogy a  $C$  érték növelésével (ahogy az várható) az átfogalmazandó szövegosztály predikciói esetében a pontosság valamelyest növekedett. A különböző solverek közül a tapasztalatok alapján,  $C$  bármely választása mellett teljesítmény tekintetében az alábbi (az 5. táblázatban is megfigyelhető) sorrend alakult ki:

'lbfgs' > 'newton-cg' > 'liblinear' > 'sag' > 'saga'

A LR esetében, továbbra is az **eredeti** szövegosztály predikcióinak pontosságát tekintve elsődlegesnek a  $\{C=100, solver='newton-cg'\}$  paraméter választás bizonyult a leghatékonyabbnak.

A kapott adatsor alapján általánosan is igaz, hogy a  $C$  változása a LR esetében függött össze jelentősen a magasabb pontosság értékek elérésével, míg a SVM esetében inkább a megfelelő kernel megválasztása bizonyult döntő fontosságúnak. Az adatokból emellett az látszik, hogy a SVM esetében az 'rbf', valamint a 'sigmoid' kernelek teljesítettek a legjobban.

Paraméterek: [C,solver]	(Átlagos-) pontosság	Szórás
1000, lbfgs	<b>0,705</b>	0,02
1000, newton-cg	0,703	0,02
1000, liblinear	0,703	0,02
1000, sag	0,701	0,02
1000, saga	0,7	0,02

5. táblázat. Metrikák az optimálisnak tekintett paraméterek mellett (LR).

### 6.3. Kiértékelés

Az ilyen módon optimalizált modell-paramétereket ezt követően az eredeti teszt halmazon ismét kiértékeljük. Ennek kapcsán a következő eredmények születtek:

		Pontosság	Fedés	F-érték
<b>SVM</b>	Eredeti	<b>0,78</b>	0,60	0,68
	Átfogalmazott	0,48	0,68	0,56
<b>LR</b>	Eredeti	0,70	0,76	0,73
	Átfogalmazott	0,46	0,39	0,42

6. táblázat. Az optimalizált paraméterek mellett tanított modellek teljesítménye az eredeti teszt halmazon kiértékelve.

Összességében elmondható, hogy a megfelelő beállításokkal mintegy 11% nyerhető az **eredeti** szövegosztály optimalizálás nélküli pontosság értékeihez képest SVM használatával. A LR a teszhalmazon 2%-kal jobban teljesített mint eredetileg, ez azonban még a keresztvalidáció során megfigyelt szóráson belül van. Érdekes tendencia, hogy míg a hiperparaméter hangolás SVM esetében egyértelműen pozitívan hatott a teljes osztályozás sikerességére (mindkét szövegosztály F-értéke nőtt), addig a LR esetében ez a hatás aszimmetrikusan jelent meg; az **eredeti** szövegosztály predikcióinak pontossága ugyan nőtt, de F-értéke csökkent, míg a már **átfogalmazott** szövegek esetében a fedés nagy mértékű javulása magával húzta az F-érték pozitív elmozdulását is.

A tévesen klasszifikált mondatok esetében még további vizsgálatot igényel, hogy milyen jellemző okok állnak azok hátterében. Néhány jellemző példa a tévesen átfogalmazandónak ítélt mondatokra<sup>7</sup>:

- (5) Az Ön által képviselt adózótól 2021. augusztus 26. és szeptember 1. között beérkezett adatok azt valószínűsítik, hogy egyes számlával (egyszerűsített

<sup>7</sup> Tekintettel arra, hogy az algoritmus által kapott normalizált és lemmatizált alak kevésbé jól olvasható, itt a megfelelő korpuszbeli "eredeti" mondatot közöljük.

számlával) bizonylatolt értékesítéseiről pénztárgépes nyugtát is állított ki.

- (6) Ha nem indul el a telepítőprogram, akkor a JAR állományok futtatása és hozzárendelése a java futtatási környezethez című dokumentáció nyújthat segítséget a hiba megoldásában.

A második mondat esetében megjegyzendő, hogy az instrukció az ÁNYK (Általános Nyomtatványkitöltő) alkalmazás telepítéséhez ad segítséget.

A következő példák két (tévesen) **átfogalmazott**-nak ítélt mondatot szemléltetnek:

- (7) Abban az esetben, ha az utas nem nyilatkozik illetve hamis, pontatlan vagy hiányos információkat ad meg, a vámhatóság a készpénzt lefoglalhatja vagy visszatarthatja illetve az utas büntetéssel sújtható.
- (8) A személyes közreműködés módját és ellentételezését a szövetkezet tagjának a szövetkezettel kötött tagsági megállapodása tartalmazza.

A tévesen osztályozott mondatok jól rávilágít(hat)anak a tanulmányban tárgyalt módszerek korlátaira és arra a körülményre, hogy a "közérthetőség" jelentős mértékben nem csak a dokumentum szóhasználatán keresztül, de a szakértők által ismert és tekintetbe vett pragmatikai kontextusban érhető tetten. Ezekben a konkrét esetekben például a téves osztályozás magyarázata az lehet, hogy a mondatok stílusukban valóban párhuzamba állíthatók a már **átfogalmazott** dokumentumokkal.

## 7. Konklúzió

Cikkünkben kísérletet tettünk arra, hogy kézzel készített korpusz felhasználása mellett felügyelt gépi tanulással kísérjünk meg szövegeket osztályozni azok közérthető jellege szerint. A szakértők által korábban **átfogalmazott** mondatok jellegzetességeit a TF-IDF vektorizálással alakítottuk a gépi tanuló algoritmusok számára értelmezhető bemenetté. A cél egy olyan modell készítése volt, amely alkalmas lehet a szakértők figyelmét felhívni azon mondatokra egy-egy hivatalos szövegben, amely további megfontolást érdemel a szöveg közérthetőre alakítása során, ezzel gyorsítva a szakértői munkát.

A közérthetőség nehezen definiálható mibenléte ellenére az eredmények alapján úgy tűnik, hogy megfelelő mennyiségű és minőségű tanítóadattal a probléma valamelyest kezelhető gépi tanulás segítségével.

További kutatást igényel, hogy az elért eredmények csak egy szűk szövegrétegen belül (a NAV tájékoztató anyagai) vagy általánosságban, a laikusoknak címzett hivatalos szövegek esetében is alkalmazható-e. A jelen cikkben (annak pilot jellege miatt) nem tettünk kísérletet fejlettebb módszerek (LSTM, BERT stb.) alkalmazására, azonban az eredmények tükrében ilyen kísérlet is indokolt lehet.

Szintén megoldandó feladat, hogy a TF-IDF vektorizáció korlátain kívül milyen egyéb jegyek figyelembevétele (szórend, szintaktikai sajátosságok stb.) lehet fontos egy ilyen jellegű osztályozási probléma esetén, illetve egy ehhez illeszkedő, összetettebb, a nyelvi megformáltságot jobban tekintetbe vevő vektorizálási forma kidolgozása is indokolt lehet.

## Köszönetnyilvánítás

Az Innovációs és Technológiai Minisztérium ÚNKP-21-3 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.

## Hivatkozások

- Chauhan, P.: Sentiment analysis: A comparative study of supervised machine learning algorithms using rapid miner. *International Journal for Research in Applied Science and Engineering Technology* pp. 80–89 (2017)
- Cutts, M.: *The Plain English Guide*. Oxford University Press (1999)
- Dobos, C.: Nyelven belüli fordítás és tisztességes jogi eljárás. In: Szabó, M. (szerk.) *A jog nyelvi dimenziója*, pp. 215–226. Bíbor Kiadó, Miskolc, Magyarország (2015)
- Dubay, W.H.: *The Principles of Readability*. Costa Mesta: Impact Information (2004)
- Edgar, D., Jeanne S., C.: A formula for predicting readability. *Educational Research Bulletin* 27, 11–20 (1948)
- Felsenfeld, C., Cohen, D.S., Fingerhut, M.: The plain english movement in the united states: Comments. *Canadian Business Law Journal* 6, 408–452 (1981)
- Garner, B.A.: *Legal Writing in Plain English*. University of Chicago Press: Chicago (2001)
- Hancock, J.: Jaccard Distance (Jaccard Index, Jaccard Similarity Coefficient) (10 2004)
- Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
- Kas, B., Lukács, A.: Processing relative clauses by hungarian typically developing children. *Language and Cognitive Processes* 27, 500–538 (2012)
- Lanstyák, I.: Lanstyák istvÁn a funkcióigés szerkezetek néhány általános kérdéséről istvÁn lanstyák on some general questions of light verb constructions 21, 61–91 (3 2020)
- Minya, K., Vinnai, E.: Hogyan írjunk érthetően? kilendülés a jogi szaknyelv komfortzónájából. *Magyar Jogi Nyelv* pp. 13–18 (2018)
- Pléh, C.: *A lélek és a nyelv*. Akadémiai Kiadó: Budapest (2013)
- Pléh, C., Lukács, A.: *Pszicholingvisztika*. Akadémiai Kiadó: Budapest (2014)
- Sun, N., Lin, G., Qiu, J., Rimba, P.: Near real-time twitter spam detection with machine learning techniques. *International Journal of Computers and Applications* 0(0), 1–11 (2020)

- Szabó, M., Vinnai, E.: A törvény szavai. Miskolc, Bíbor Kiadó (2018a)
- Szabó, M., Vinnai, E.: A törvény szavai: Az OTKA-112172 kutatási zárókonferencia anyaga Miskolc, ME – MAB, 2018. május 25. Miskolc, Magyarország : Bíbor Kiadó (2018b)
- Tóth, J.: Tudnak-e a jogászok érthetően fogalmazni, avagy nem is kell azt tudni? Magyar Jogi Nyelv pp. 31–37 (2019)
- Vinnai, E.: A magyar jogi nyelv kutatása. *Glossa Iuridica* p. 29–48 (2014)
- Vinnai, E.: Megértette a figyelmeztetést? a figyelmeztetések és tájékoztatások közlése a büntetőeljárásokban. In: Szabó, M., Vinnai, E. (szerk.) A törvény szavai : Az OTKA-112172 kutatási zárókonferencia anyaga Miskolc, ME – MAB, 2018. május 25., pp. 281–295. Bíbor Kiadó, Miskolc, Magyarország (2018)
- Willerton, R.: *Plain Language and Ethical Action - A Dialogic Approach to Technical Content in the 21st Century*. Routledge: New York (2015)
- Üveges, I.: Automatizálható a közérthető megfogalmazás? Jog, számítógépes nyelvészet és pszicholingvisztika találkozása. *Magyar Jogi Nyelv* 4, 1–8 (2020)



## Magyar nyelvű időpont-egyeztető dialógusrendszer v2

Nagy Soma Bálint<sup>1</sup>, Herdinai Viktor<sup>1</sup>, Pálfi Gellért<sup>1</sup>, Fegyó Tibor<sup>2,3</sup>,  
Mihajlik Péter<sup>3</sup>, Farkas Richárd<sup>1</sup>

<sup>1</sup> Szegedi Tudományegyetem, Informatikai Intézet

<sup>2</sup> SpeechTex Kft.

<sup>3</sup> Budapesti Műszaki és Gazdaságtudományi Egyetem,

Távközlési és Médiainformatikai Tanszék

nagysomabalint@gmail.com, rfarkas@inf.u-szeged.hu,

tfegyoo@speecheatex.com

**Kivonat:** A dialógusrendszerek fejlesztése napjaink fontos kutatási területévé nőtte ki magát. Munkánkban a korábban bemutatott időpont-egyeztetésre szolgáló chatbotunk felhasználói visszajelzésekből származó tapasztalatait tárgyaljuk, valamint részletezzük, hogy az így megtalált hibákat/hiányosságokat milyen módon javítottuk ki. A legfontosabb fejlesztésünk, hogy a dialógus állapotainak leírására egy teljes új reprezentációt vezetünk be, ami az időpont-egyeztetés, mint tárgyalási feladatot általánosan tudja támogatni. Célunk kezdetektől fogva a telefonos időpont-egyeztetés volt. Bemutatjuk chatbotunk beszéd-alapú interfészre való adaptálásának első lépéseit, egy kis mintás kiértékelésen keresztül.

### 1. Bevezetés

A chatbotok elképesztő popularitásra tettek szert az elmúlt évtizedben. Míg korábban elsősorban cégek fejlesztettek, egyszerű kulcsszókeresés-alapú rendszereket, az elmúlt néhány évben a számítógépes nyelvészetet is elérte a chatbot láz. Amennyiben a magyar üzleti chatbotok bonyolultabb és természetes emberi interakciókat akarnak kezelni, akkor szükségük van magyar nyelvi jelenségek kezelésére alkalmas algoritmusokra. Azért, hogy a jelenlegi chatbotok határait és továbbfejlesztési igényét megértsük, egy chatbot alkalmazás fejlesztési lépéseit mutattuk be Nagy és mtsai. (2021) cikkünkben.

Az alkalmazás üzleti/társadalmi célja, hogy a felhasználó magyar nyelven időpontot egyeztessen bármilyen környezetben. Médiaként elsősorban beszéd-alapú (telefonos) ügyfélszolgálatok mögötti dialógus rendszert képzelünk el, ezért csak folyó szöveges kommunikációra koncentrálnak (és például nem vizsgáljuk a chatbotok által gyakran használt kattintással listaelem választó lehetőségét).

Jelen cikkünkben bemutatjuk az időpont-egyeztető chatbotunk első verziójára kapott visszajelzéseket, és a visszajelzések alapján végzett fejlesztéseket. A jelen rendszert tekinthetjük kettes verziónak (v2). A chatbot forráskódja, illetve a kulcsfontosságú részét alkotó dátum- és időpontentitások felismerésére szolgáló könyvtár elérhető a <https://github.com/szegedai/hun-appointment-chatbot> és

<https://github.com/szegedai/hun-date-parser> oldalakon, illetve működése tesztelhető a publikusan elérhető az <https://www.inf.u-szeged.hu/algmi/chatbot/> oldalon.

## 2. Kapcsolódó munkák

Számos chatbotokkal és dialógus rendszerekkel foglalkozó nemzetközi publikáció érhető el, köztük néhány, amelyben időpontegyeztető rendszert mutatnak be a szerzők. A francia nyelvű időpont egyeztetést célul kitűző CLASSiC projekt keretében többféle megoldás született a problémára, köztük a részlegesen megfigyelhető Markov döntési folyamatokra épülő ún. “System 2” (Jurcicek és mtsai., 2010). A NASTIA rendszer (L. El Asri, 2014) olyan módon fejlesztette tovább a korábban publikált megvalósításokat, hogy a dialógus egyes, előre meghatározott állapotaiban megerősítéses tanulás segítségével választ a rendszer a rendelkezésre álló stratégiák közül. Az ilyen tanulórendszerek esetén szükséges jutalmat az egyes beszélgetések után a felhasználó által adott elégedettségi pont jelentette. A DINASTI (L. El Asri, 2014) publikációban a NASTIA rendszer alkotói közzétették a tanításra, valamint a kiértékelésére készített 1734 francia nyelvű dialógust tartalmazó korpuszt.

Az utóbbi időszakban a mélytanulás térnyerésével a dialógusrendszerek megvalósítására a szabályalapú rendszerek helyett a teljes egészében adatvezérelt, “end-to-end” tanított dialógusrendszerek kerültek az előtérbe. Eric és Manning (2017) a neurális dialógusmodellek feladatorientált dialógusrendszerek esetén való alkalmazási nehézségeit tárgyalja, és az ún. kulcs-érték kinyerő háló (key-value retrieval network) architektúrát javasolja a problémák megoldására. A szerzők által tárgyalt három scenárió egyike az időpontegyeztetés, a cikkben tárgyalt rendszer tanítása során használt időpontegyeztetési dialógusokat is tartalmazó angol nyelvű adathalmazt közzétették.

A magyar nyelvű chatbotokról szóló publikációk száma csekély. Kemény és Recski (2018) Tervezz Velem utazásszervező Facebook Messenger chatbotjának célja a BKK API lekérdezésével a felhasználók utazást érintő kérdéseit megválaszolni. Chatbotjuk képes volt a nyelvileg töredékes, rövid üzenetekből is pontos felhasználói szándék kinyerésére. A Lara könyvtár (Nagyfi, 2018) egy célzottan magyar nyelvű chatbotok fejlesztésére készült eszköz, felhasználói szándék meghatározását, entitások kinyerését teszi lehetővé, emellett pedig számos nyelvi feldolgozó eszközt tartalmaz magyar nyelvre.

Korábbi munkánkban (Nagy és mtsai., 2021) bemutattuk az időpont-egyeztető chatbotunk első verzióját, amely képes volt a felhasználó és a bot között több körben, a felhasználó szöveges üzeneteit felhasználva, nagyon egyszerű dialóguslogikát követve időpont-egyeztetést lefolytatni. Chatbotunk a gépi tanulásra alapuló, nyílt forráskódú Rasa keretrendszert használja (Bocklisch és mtsai., 2017), amit a spaCy-hez



készült magyar nyelvi feldolgozó eszközök, illetve CBOW szóbeágyazás (Orosz, 2019) segítségével alkalmaztunk a magyar nyelvre.

Korábbi cikkünkben bemutattuk továbbá a chatbotok fejlesztésére használható nyílt forráskódú keretrendszereket, valamint tárgyaltuk a chatbotunk szempontjából kritikus jelentőségű természetes szövegből dátum-, illetve időpontentitások kinyerésére használható eszközöket. Részleteztük az általunk fejlesztett és közzétett dátum-, időpont- és időintervallum-kinyerő program célját és működését. Jelen cikkünkben bemutatjuk az időpont-egyeztető chatbotunkon végzett fejlesztéseket.

### 3. Felhasználói felület a chatbot tesztelésre

Annak érdekében, hogy felhasználói visszajelzések segítségével javíthassunk a chatbotunkon, megértjük a valós emberi beszélgetés során elkövetett jellemző hibákat, a chatbothoz egy publikusan elérhető webes felhasználói felületet fejlesztettünk: <https://www.inf.u-szeged.hu/algmi/chatbot>. Külön figyelmet szenteltünk a visszajelzési lehetőségekre, és minden felhasználói interakciót tároltunk (anonim módon), hiszen a célunk az ember-gép diskurzusok mélyebb megértése volt.

Fontos, hogy minden beszélgetést eltároljunk, mivel ezek mutatják meg, hogy hol tudunk javítani a chatbot működésén, ehhez használtuk a Rasa TrackerStore nevű funkcióját. A TrackerStore lehetőséget nyújt egy adatbázis-szolgáltatás (például MongoDB, PostgreSQL, Redis) egyszerű integrálásához. Esetünkben a MongoDB NoSQL adatbázisra esett a választás. A dokumentumokban el van tárolva a beszélgetés azonosítója, a felhasználó üzenetei, a chatbot válaszai, a felismert intent és megannyi fontos adat. A dokumentumokat lehet szűrni valamennyi mezője alapján, illetve mivel JSON-szerű formátumban (BSON) vannak tárolva, egyszerűen lehet őket JSON fájlba exportálni, ezen fájlokból pedig könnyedén tudjuk rekonstruálni a felhasználó és bot közötti beszélgetést olvasható formátumban.

Lehetőséget teremtettünk a beszélgetéssel kapcsolatos azonnali visszajelzésre is. A szöveges visszajelzéseket a beszélgetés azonosítójával és időpontjával szintén egy MongoDB adatbázisban tároljuk, az azonosító alapján pedig rálinkeljük a szóban forgó beszélgetés logjára. Egy kérdőíves visszajelzést is feltettünk, de erre egyelőre még statisztikailag értelmezhetetlenül kevés kitöltés érkezett.

### 4 Visszajelzések az első verziós időpont-egyeztetőről

Az elkészült webes felületen 2021. szeptembere és novembere között 80 beszélgetés zajlott a chatbottal. Két fő alapos tesztelőnk volt (senior product managerek), rajtuk kívül körülbelül 15<sup>1</sup> programozó szakos egyetemi hallgató próbálta ki a chatbotot.

---

<sup>1</sup> Mivel a teljes anonimitásra törekedtünk, nem tárolunk cookiekat és semmilyen információt, amiből vissza lehetne következtetni, hogy melyik beszélgetést ki folytatta.

A chatbotról visszajelzést adó fent bemutatott felhasználó felület, illetve a bot forráskódjának Github oldalán kapott visszajelzésekből és beszélgetések átolvasásából a következő problémacsoportokat azonosítottuk.

#### **4.1. Felhasználói szándék felismeréséhez köthető hibák**

Chatbotunkban a felhasználói szándék felismerését egy transzformer-alapú, neurális háló architektúra, a DIET (Bunk és mtsai., 2020) végzi. Mivel mi a dátum és időentitások megállapítására a saját, szabályalapú entitáskinyerőnket használtuk, így csak a felhasználói szándék kinyerésére alkalmaztuk a DIET-et. A Rasa-ba a szöveg előfeldolgozására a magyar nyelvű spaCy-t integráltuk (Orosz, 2019).

A visszajelzésekből világossá vált, hogy a chatbot felhasználói szándék felismerésére szolgáló komponense nem eléggé robusztus, érzékeny a felhasználói üzenetekben történő kis változásokra is. Például többször megjelenő hiba a felhasználói üzenetekben megjelenő különféle írásjelek, speciális karakterek okozta felhasználói szándék tévesztés volt. A bot például az “Üdv” üzenetet helyesen osztályozta, azonban az “Üdv!” esetben hibásan állapította meg a felhasználó szándékát (amely mindkét példa esetén a bot kezdeti üdvözlése lett volna a felhasználó részéről).

Több lehetséges megoldás vizsgálata után végül amellel döntöttünk, hogy megvárjuk az új magyar spaCy pipeline elkészültét és nem kezdünk bele az előfeldolgozási lépések javításába. Gyors megoldásként egyszerűen eltávolítjuk a speciális karaktereket az előfeldolgozás során, ami a hibák egy részét orvosolja. Eddig ez nem okozott újabb problémákat (pedig a mondatvégi írásjel fontos lehetne a szándék azonosításában). Az NLU pipeline-ban, a kezdeti tokenizáló lépés elé (miután meggyőződünk arról, hogy az nem konfigurálható úgy, hogy a speciális karaktereket eltávolítsa) egy általunk implementált lépést helyeztünk el, amely eltávolítja a speciális karaktereket a szövegből. Ezt a modult a Rasa pipeline architektúrájával kompatibilisen kellett implementálnunk.

#### **4.2. A dátum és időpont entitás kinyerésének hibái**

A dátum és időpont entitások kinyerését a saját, erre a célra implementált könyvtár végzi. Ennek jellemzője, hogy szabályalapú és tiszta szöveges bemenetet feltételez. A felhasználói bemenetet 16 lehetséges komplex reguláris kifejezés illesztése után, számos heurisztika segítségével kíséreljük meg értelmezni, ennek köszönhetően az entitás kinyerő tapasztalataink alapján megbízhatóan működik

- számos formában megadott dátum és időpont,
- ékezetek, kis-nagybetű különbségek, minimális elírások, egyes gyakori rövidítések,
- számok szöveges formában történő kiírása,
- illetve hiányos dátum vagy időpont kifejezések esetén.

Mivel azonban a jelenleg használt módszer azt igényli, hogy minden felismert scenáriót előre manuálisan kezeljünk, így elkerülhetetlen, hogy a rendszer hibázzon, amikor egy olyan formában kifejezett dátum- vagy időentitást kap bemenetül, amely felismerését megelőzőleg nem implementáltuk. A közzétett chatbottal folytatott beszélgetéseket tekintve az ilyen esetek súlya nem túlnyomó, azonban a beszélgetésre való hatása miatt mindenképp kezelni kell ezeket a hiányosságokat. Erre a típusú hibára néhány példa:

- Találkozunk *két hét múlva*. (A könyvtárban nem volt megvalósítva az *n időperiódus múlva* forma értelmezése.)
- Ráérek *17-18h* között. (A könyvtár nem volt képes az SMS-nyelvben az óra kifejezésére használt *h* értelmezésére.)

A fenti és ehhez hasonló hibákat teljesen megszüntetni ugyan nem lehetséges, de a felismert szerkezetek bővítésével képesek voltunk számos korábban gyakran előforduló hiba orvoslására.

Szintén az entitáskinyerőnk számlájára írható hiba volt, hogy amikor ugyan az egyes dátum és időelemeket helyesen azonosítjuk a bemeneti szövegben, azonban a heurisztika, ami a végső kimenetet alkotja meg a felismert elemek ismeretében, hibás vagy hiányos. Erre példa:

- *Nyolckor* kezdődik a vacsora. (Mivel a szövegből explicit módon nem derül ki a napszak, így hibásan reggel 8 órát nyert ki a szövegből a program.)

Erre a konkrét hibára részleges megoldást jelentett, hogy lehetővé tettük az időpontok kinyerését a bemeneti szövegből már ismert, tágabb időtartam megadásával kiegészítve, a fenti példa esetén, ha előre meghatározzuk, hogy a kinyerni kívánt időpont esti, akkor a program helyesen 20:00-t nyer ki a szövegből. Ez tette lehetővé, hogy a bottal folytatott többkörös időpont egyeztetés során, ha a felhasználó és a bot már megállapodott egy tágabb időtartamban, akkor a felhasználó azt anélkül tudja tovább specifikálni, hogy teljesen pontosan meg kelljen határozni a kért, szűkebb intervallumot. Jelenleg a könyvtár rendelkezik olyan függvénnyel, ami paraméterül várja a tágabb intervallumot, illetve a szöveget, amelyből a tágabb intervallum ismeretében kívánunk időpontot kinyerni. Például, ha a felhasználó és a bot korábban már megegyezett abban, hogy nem az aktuális, hanem a soron következő hétre keresnek időpontot, akkor megadható a következő hét, mint tágabb intervallum, így a könyvtár a *péntek* szöveghez az e hét péntek helyett a jövő pénteket rendeli hozzá.

### 4.3. A dialógus limitált állapotmenedzsmentje okozta hibák

Egy dialógus állapota alatt a felhasználóval folytatott üzenetváltás korábbi üzeneteiből kinyert, felhasználható formában eltárolt, a beszélgetés folyamatát később befolyásolni képes információkat értjük. Első verziós chatbotunk az ún. szlot kitöltési (slot filling) dialógus menedzsment keretben volt implementálva, amelyet a Rasa - és minden egyéb chatbot fejlesztő keretrendszer - megvalósít. Első verziós chatbotunk a dialógus

irányítása során két szlot kitöltésre koncentrált: legyen egy olyan dátum (nap), amiben megegyeznek a felhasználóval a találkoztól illetően, illetve a találkozó konkrét időpontja a napon belül. Ezt a két információt szöveges formában (ISO dátum formátum) tároltuk, amiket a bot a felhasználóval folytatott beszélgetés során olvasni-írni tudott.

Több felhasználói beszélgetésben megfigyelhető, hogy az előzetes feltételezésünkkel ellentétben két szlot nem elegendő a beszélgetés állapotának reprezentálásához. Gyakran fordult elő, hogy a felhasználó visszautalt egy korábban említett lehetséges dátumra, ha az éppen tárgyalt dátum nem volt megfelelő. Két szlittal ilyen sok információ eltárolása nem lehetséges, további szöveges szlotok felvétele átláthatatlanná tette volna a chatbothoz írt vezérlést. Erre a problémára megoldást az 5. fejezetben tárgyalt új dialógus állapotrepresentáció implementálása jelentett.

#### **4.4. Felhasználói élménnyel kapcsolatos problémák**

A beszélgetések átolvasásából szembetűnő volt, hogy a chatbot elmulasztotta a kezdeményezést, a beszélgetést a felhasználónak kellett kezdeményeznie, a bot nem adott semmilyen segítséget a felhasználónak azzal kapcsolatban, hogy milyen típusú probléma megoldásában képes asszisztálni. Az új verzióban a bot küldi az első üzenetet, bemutatkozik, elmondja, hogy milyen célt szolgál és konkrét kérdést tesz fel a felhasználónak.

Korábban az asszisztensnél magánál lehetett időpontot foglalni, a felhasználó arról érdeklődhetett az asszisztenssel, hogy ő, azaz a bot mikor ér rá. A természetesség és a hitelesség növelése érdekében a jelenlegi verzióban a bot egy képzeletbeli karakter, Főnök úr személyi asszisztense, így a felhasználó nem a bot szabad időpontjairól, hanem Főnök úr naptárával kapcsolatban érdeklődik.

Megjegyezzük, hogy erre a módosításra, nem csak a felhasználói élmény növelése, hanem a 2021. áprilisában megjelent EU rendelet (EB, 2021) miatt is szükség volt, hiszen az kimondja, hogy “Bizonyos MI-rendszerek, amelyek rendeltetése a természetes személyekkel való interakció ... a természetes személyeket figyelmeztetni kell, hogy MI-rendszerrel lépnek kapcsolatba”. Tehát megtévesztő lehetett a korábbi verzióknak, mert esetleg valaki azt hihette, hogy emberrel, és nem géppel beszél.

## **5. Új dialógus állapotrepresentáció**

A legnagyobb limitációt az jelentett, hogy az állapotrepresentációt két darab, szöveg típusú szlittal valósítottuk a chatbotunk első verziójában. A Rasa keretrendszerben a szlotok kulcs-érték párok, a beszélgetés folyamán a bot memóriájaként szolgálnak. A szlotok száma és típusa előre meghatározott. A két szlot feladata a dátum, illetve az adott napon belül kölcsönösen megfelelő időpont eltárolása volt.

Azt, hogy ez milyen esetekben jelent limitációt, a következő annotált beszélgetés mutatja be:

Felhasználó:	<b>Időpontot szeretnék jövő héten</b>	<i>(“jövő hét” entitás felismerése, két jövő heti szabad nap ajánlása.)</i>
Chatbot:	<b>A legközelebbi két nap amikor a Főnök úr ráér a kért időszakban jövő hét szerdán és jövő hét csütörtökön lesz.</b>	
Felhasználó:	<b>Esetleg csütörtökön?</b>	<i>(“csütörtök” entitás felismerése, adott hét csütörtöki napjaként értelmezi a chatbot)</i>

A példában a szövegekörnyezetből kiderül, hogy a felhasználó csütörtöki időpont kérésekor valószínűleg továbbra is a jövő hétre keres időpontot, azonban, mivel ez az információ már nem része a bot memóriájának, így a bot képtelen rájönni, hogy jövő hét csütörtökről van szó.

A probléma igazi gyökere az, hogy az időpont-egyeztetési feladat fél úton helyezkedik el a szlot kitöltési feladat és a tárgyalási célú dialógusok között (El Asri és mtsai., 2014). Egyrészt időpont szlotokat akarunk kitölteni a felhasználóval, de az nem a klasszikus módon történik, mint mondjuk egy éttermi rendelő chatbotnál, hanem meg kell egyezni a mindkét fél számára alkalmas időpontban. A szlotok számának növelésével nem tudtuk volna a probléma igazi gyökerét orvosolni.

Ahhoz, hogy a komplexebb időpont-egyeztetési dialógusokat is meg tudjuk valósítani, újra kellett gondolnunk a dialógus állapotának reprezentációját, azt, hogy milyen adattípus választása lenne célszerű a bot memóriájaként való használatra, mi lenne az a forma, amelyben minden releváns korábban elhangzott információ tömören, strukturáltan tárolható, a későbbiekben kellően absztrakt módon lekérdezhető. Ez a probléma annak a vizsgálatával egészült ki, hogy milyen technikai korlátai vannak egy tetszőlegesen komplex adattípus szlotban való tárolására a Rasa részéről. A korábbi választás célszerűnek tűnt, mivel az általunk használt Rasa keretrendszer csupán Python által szerIALIZÁLNI képes információkat tud szlotban tárolni, kizárva ezzel a nem automatikusan szerIALIZÁLHATÓ típusok (például az általunk is használt datetime, illetve DateTimeRange objektumok) triviális használatát.

Az új dialógusállapot-reprezentációt úgy alkottuk meg, hogy a következő szempontoknak feleljen meg:

1. Alkalmas legyen a bot (előre ismert) és a felhasználó (a dialógus folyamán kifejtett) szabad időpontjainak tárolására közös idővonalon, olyan módon, hogy ezek átfedései egyszerűen lekérdezhetőek legyenek
2. Tárolható legyen a korábban már kizárt (sikertelen végkifejlettel tárgyalt) időtartamok listája
3. Amennyiben a bot és a felhasználó több körös egyeztetést követően egy rövidebb konszenzuális időtartamra jut, ám azt az egyik fél végül elutasítja, a botnak legyen emléke az elutasított időpont előtti körben megegyezett időtartamról.

Az első pont szükségességét az indokolja, hogy szükség lehet a felhasználó által korábban említett lehetséges, azonban a beszélgetés során nem kifejtett időpont későbbi felidézésére. Például:

Felhasználó: **Időpontot szeretnék kedden vagy szerdán** (*“kedd, szerda” entitások elmentése, mint a felhasználó számára lehetséges elfogadható időintervallumok.*)  
Chatbot: **Főnök úr kedden délután 2 és 4 óra között ér rá.**

A fenti beszélgetés esetén, noha a bot a kedd tárgyalásába kezd bele a felhasználóval, érdemes eltárolni, hogy a felhasználó egy másik lehetséges napot is megjelölt, ha a keddi napon végül nem sikerül kölcsönösen elfogadható időpontot találni.

A második pontot az indokolja, hogy nem szeretnénk a felhasználónak kétszer ugyanazt az időpontot ajánlani, ha korábban már elutasította a felajánlott időpontot.

A harmadik pont szükségességét a következő szcenárió illusztrálja:

Felhasználó: **Időpontot szeretnék jövő héten** (*“jövő hét” entitás felismerése, két jövő heti szabad nap ajánlása.*)  
Chatbot: **Melyik nap lenne alkalmas?**  
Felhasználó: **Kedd** (*“jövő hét” ismeretében a “jövő kedd” megjegyzése*)  
Chatbot: **Főnök úr ráér jövő hét kedden délután 2 és 4 óra között.**  
Felhasználó: **Esetleg 3 órakor?** (*“délután 3 óra” entitás felismerése*)  
Chatbot: **Főnök úr ráér jövő hét kedden délután 3 órakor. Beírom a naptárjába a találkozót.**  
Felhasználó: **Lehetne inkább csütörtökön?**

Az aktuális üzenetben észlelt időintervallumot a korábbi üzenetekben megtalált időintervallumok fényében szeretnénk értelmezni, tehát, ha tudjuk, hogy a legutóbbi üzenetben tárgyalt intervallum a [jövő kedd 14:00; jövő kedd 16:00], akkor az aktuális üzenet délután 3 óráját a korábbi intervallum a fényében jövő kedd 15:00-ként szeretnénk értelmezni. Azonban, ha egy olyan entitást nyerünk ki, mint ebben az esetben a csütörtök, amit a korábbi intervallumok a jövő kedd 14:00, [jövő kedd 14:00; jövő kedd 16:00] és a jövő kedd fényében nem értelmes, addig szeretnénk visszalépkedni, amíg olyan intervallummal találkozunk, aminek tekintetében már értelmezhető a csütörtök, ez esetben ez a beszélgetés kezdetén említett jövő hét (ha nincs ilyen, akkor természetesen a teljes jövőn, korlátozás nélkül értelmezzük az aktuális dátum- és időentitást).

A feltételeinket kielégítő adatszerkezet tetszőleges számú idővonalból áll. Jelenleg három, a bot számára alkalmas, a felhasználó számára alkalmas, illetve a már kizárt

időintervallumokat tartalmazza. Ezeket a beszélgetés során bejövő információk alapján frissítettük. A három idővonal már jóval komplexebb dialógusok lebonyolítását teszi lehetővé, és ha szükséges, a jövőben újabb idővonalakkal bővíthetjük a reprezentációt (például ‘nem a legjobb, de ha kell megoldom’).

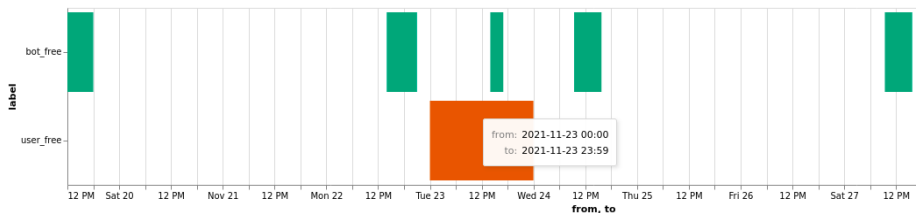
A dialógus stratégiánk célja, hogy az éppen tárgyalt időpontot “tölcseyszerűen”, egy tágabb időintervallumból a szűkebb felé terelje, de megteremtve annak a lehetőségét, hogy visszalépjünk egy fentebbi szintre, amennyiben szükséges.

A technikai akadályokon úgy léptünk át, hogy a saját adatszerkezetünkhöz megírjuk a szerializáló és deszerializáló eljárásokat, és a Rasa ezt így már képes kezelni, mint egyetlen speciális szlot.

Az új adatszerkezet előnye volt továbbá, hogy lehetővé tette, hogy az osztály felett, annak részeként az adattagokra vonatkozó komplex lekérdezéseket, módosításokat implementáljunk, így a dialógusvezérlés során megvalósított kódot letisztultá, olvashatóvá tettük.

Az időpont-egyeztetési folyamat követéséhez, a jelentkező hibák megértéséhez az idővonalakat, valamint az aktuálisan tárgyalt időintervallumhoz tartozó “tölcse” vizualizáló komponens is készítettünk. Ennek célja az volt, hogy a beszélgetés minden pillanatában látható legyen, hogy az adott pontig milyen információ áll a bot rendelkezésére, milyen emlékek alapján határozza meg a válaszát. Például:

Felhasználó: **jövő kedden ráér?**  
Chatbot: **Főnök úr ráér a kért időszakon belül, délután 2 és este 9 között.  
Mikor lenne jó pontosan?**



1. ábra: A vizualizációs komponens kimenete.

## 6. Adaptáció audio-interfészre

Célunk a kezdetektől az volt, hogy az asszisztensünk hangalapon használható legyen. Ez volt az oka, hogy nem vizsgáltuk a lehetőségét olyan multimédiás opciók használatának, amelyek más szöveges chatbotokban gyakoriak, például gombok használata.

Ahhoz, hogy a chatbotunk hangalapon használható legyen, elkezdtük vizsgálni a magyar beszédleírató (speech-to-text, STT) chatbotunk pontosságára gyakorolt

hatását. Elkezdődött a SpeechTex Kft. beszédleiratózó rendszere (Mihajlik és mtsai, 2021) és az időpont-egyeztető chatbotunk technológiai integrációja. Az előbb említett publikációhoz képest abban különbözik a dialógusrendszerünkben alkalmazott beszédfelismerő megoldás, hogy a nyelvmodelljét a (Tarján és mtsai, 2016) szerinti közéleti, televíziós adatokkal tanított nyelvmoddellel interpoláltuk.

Kezdeti lépésként arra voltunk kíváncsiak, hogyan reagál chatbotunk szándékfelismerési, valamint a dátumintervallum-felismerő komponense arra a potenciális zajra, amelyet az jelent, hogy nem ember által gépelt, hanem automatikusan leiratozott szöveg képzí a bemenetet. E két modul kiértékelése azért lényeges, mert ennek a két komponensnek a pontos működése kritikus a beszélgetés helyes alakulása szempontjából, bármelyik teljesítménye is romlik szignifikánsan, az a felhasználók beszélgetésselményére közvetlenül rossz hatással lesz. A kis mintás kiértékelési kísérletünkben a 4. fejezetben bemutatott valós beszélgetésekből válogattunk 70 darab felhasználói közlést, és azt 6 fő (mind férfi, egy 41 éves, a többiek 21-24 évesek) felolvasta. A felolvasást számítógépes mikrofonnal végeztük el, mert a hangalapú időpont-egyeztetőnk első verziója demózási célokkal nem, csak mobil eszközön futó webes felületen keresztül lesz elérhető.

### 6.1. Szándékfelismerés kiértékelése

A szándékfelismerő komponens tanítása (*felhasználói szándék címke, felhasználói szándékot jelző mondat*) párok alapján történik. Kiértékeléséhez ilyen párokat gyűjtöttünk. A mondatok forrása a botot tesztelő felhasználók múltbeli üzenetei, az üzenetekhez tartozó címkéket többségi szavazással határoztuk meg.

A kiértékelést elvégeztük az eredeti mondatokon, valamint a beszédleiratózó modellt futtatva a 6 darab felolvasott változaton. A 6 darab felhasználói szándékot (osztályt) összesen 50 darab tesztmondattal kiértékelve az alábbi táblázat tartalmazza a pontosságot. A STT modell által leiratozott mondatok esetében a beszélőkre átlagolt pontosságot és szórást tüntetjük fel.

	Szándékfelismerés pontossága
Szöveg	77,0
Felolvasott majd automatán leiratozott	72,4 ± 4,3

1. táblázat. Szándékfelismerés kiértékelése.

### 6.2. Dátumfelismerő kiértékelése

A dátumfelismerő modul, lévén, hogy reguláris kifejezésekre alapul nem rendelkezik jó hibatoleranciával a bemeneti szövegben található zajra nézve, ezért különösen fontos



volt kiértékelni, hogy milyen hibára számíthatunk az STT modellünk által leiratozott szövegeken. Ehhez egy 20 dátumra vagy időre utaló kifejezésből álló listát gyűjtöttünk. A kezdeti listát úgy állítottuk össze, hogy a dátumfelismerő által ismert és kezelni képes eseteket tartalmazzon (azaz itt 100% a szövegből mért pontosság). Ezt követően a lista mondatait 6 olvasó olvasta fel, amit a STT modell segítségével leiratoztunk, majd az így képzett listákon is elvégeztük a kiértékelést. Az olvasónként átlagolt pontosság  $90,84 \pm 5,4\%$  volt.

### 6.3. Beszédleiratozó típushibái

Ahogy az a fenti kiértékelési metrikák is alátámasztják, az általános beszédleiratozó is keveset hibázik. Néhány hiba:

Eredeti szöveg	Felolvasott majd leírt	Hiba
<i>ma vagy holnapután</i>	<i>ma a vagy holnapután</i>	hosszú <i>maaa</i>
<i>az túl korai nekem</i>	<i>hát az túl korai nekem</i>	extra <i>hát</i> (spontán beszéd)
<i>a főnökkel szeretnék beszélni</i>	<i>a főnöke szeretnék beszélni</i>	nyelvi modell hibája
<i>jövő hét</i>	<i>jövő 7</i>	speciális domain

2. táblázat. Beszédleiratozás hibái.

Ezeket a hibákat egy speciális nyelvi modellel, vagy akár a dátumfelismerőnk kiegészítésével is javítani lehet a jövőben.

## 7. Összegzés

Munkánk során a felhasználói visszajelzésekre, és a használat során megfigyelt gyakori problémákra reagálva több aspektusból is újra gondoltuk a korábban fejlesztett időpont-egyeztető chatbotunkat. Tapasztalataink alapján a dialógus új állapotrepresentációja gördülékenyebb, természetesebb dialógusokat eredményez. A továbbfejlesztett bot közzétételével, felhasználói visszajelések alapján kívánjuk megállapítani, hogy ez valóban beigazolódik-e. A chatbotot az <https://www.inf.u-szeged.hu/algmi/chatbot/> oldalon tesszük elérhetővé, a kipróbálóknak lehetősége van a hibák jelentésén túl egy kérdőív kitöltésére, ami a felhasználói élményükkel kapcsolatos kérdéseket tartalmaz.

Jövőbeli munkánk fókuszában a chatbot hangalapú verziójának elérhetővé tétele, és spontán beszéd leiratozása által keletkező új kihívások kezelése áll.

## Köszönetnyilvánítás

A publikációban szereplő kutatást az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- T. Bocklisch, J. Faulkner, N. Pawlowski, A. Nichol: Rasa: Open Source Language Understanding and Dialogue Management. In Proceedings of NIPS workshop on Conversational AI (2017)
- T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol: DIET: Lightweight Language Understanding for Dialogue Systems. [arxiv.org/abs/2004.09936](https://arxiv.org/abs/2004.09936) (2020)
- F. Jurcicek, S. Keizer, F. Mairesse, K. Yu: D5. 4: Proof-of-concept CLASSIC Appointment Scheduling system (“System 2”). (2010)
- L. El Asri, R. Lemonnier, R. Laroche, O. Pietquin, H. Khouzaimi: NASTIA: Negotiating Appointment Setting Interface. In Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC’14 (2014)
- L. El Asri, R. Lemonnier, R. Laroche, O. Pietquin: DINASTI: Dialogues with a Negotiating Appointment Setting Interface. Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC’14 (2014)
- M. Eric, C. D. Manning: Key-Value Retrieval Networks for Task-Oriented Dialogue (2017)Európai Bizottság: A mesterséges intelligenciára vonatkozó harmonizált szabályok (mesterséges intelligenciáról szóló jogszabály) megállapításáról és egyes uniós jogalkotási aktusok módosításáról. AZ EURÓPAI PARLAMENT ÉS A TANÁCS RENDELETE. Brüsszel, 2021.4.21.
- Kemény Boldizsár, Recski Gábor: Természetes nyelvi interfész menetrend- és utazástervező szolgáltatásokhoz. In XIV. Magyar Számítógépes Nyelvészeti konferencia, (2018)
- Mihajlik Péter, Balog András, Tarján Balázs, Fegyó Tibor: End-to-end és hibrid mélyneuronháló alapú gépi leiratozás magyar nyelvű telefonos ügyfélszolgálati beszélgetésekre, In: XVII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 139-145. (2021)
- Nagyfi R: Lara. <https://github.com/sedthh/lara-hungarian-nlp>. (2018)
- Orosz Gy.: Hungarian models for spaCy. <https://github.com/spacy-hu/spacy-hungarian-models>. (2019)
- Tarján Balázs, Varga Ádám, Tobler Zoltán, Szaszák György, Fegyó Tibor, Bordás Csaba, Mihajlik Péter: Magyar nyelvű, élő közéleti- és hírműsorok gépi feliratozása, In: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016), pp. 89-99. (2016)

## Szerzői index, névmutató

- Ács, Judit, 343  
Agócs, Ádám, 241  
Alberti, Gábor, 449  
Arthur, Frigyes Viktor, 185
- Bajzát, Tímea Borbála, 93, 375  
Balog, András, 199  
Balogh, Réka, 175  
Beke, András, 271  
Berend, Gábor, 3, 45, 227  
Berkecz, Péter, 59  
Bial, Bence, 343  
Bóna, Judit, 151
- Csanády, Bálint, 549  
Csányi, Gergely Márk, 521, 587  
Csapó, Tamás Gábor, 185  
Cserháti, Réka, 45, 227
- Egas-López, José Vicente, 175
- Farkas, Richárd, 59, 633  
Fegyó, Tibor, 199, 633  
Fejes, Attila, 271  
Feldmann, Ádám, 603  
Fellegi, Zsófia, 375  
Ferenczi, Gergő, 431  
Ferenczi, Zsanett, 213  
Ficsor, Tamás, 45  
Frey, Balázs Konrád, 343  
Friedszám, Attila, 449
- Gadó, Krisztián, 587  
Gosztolya, Gábor, 151, 175  
Grácsi, Tekla, 199  
Guba, Csenge, 325
- Hagymási, Judit, 449
- Héja, Enikő, 431  
Herdinai, Viktor, 633  
Hoffmann, Ildikó, 151, 175  
Horváth, Péter, 375
- Imre, Nóra, 175  
Indig, Balázs, 31, 375, 505  
Ivaskó, Lívia, 161
- Jelencsik-Mátyus, Kinga, 431  
Jenei, Attila Zoltán, 137
- Kahla, Mram, 281  
Kalivoda, Ágnes, 77  
Kálmán, János, 175  
Karabulut, Aslihan, 375  
Kárpáti, Laura, 449  
Kelemen, Sára Bernadett, 573  
Kenyeres, Ádám, 257  
Kicsi, András, 227  
Kleiber, Judit, 449  
Kmetty, Zoltán, 573  
Kohári, Anna, 199  
Kolláth, István, 227  
Kőrös, András, 417  
Kósa, Karolina, 311  
Kovács, György, 257  
Kovács, Viktória, 561  
Kundráth, Péter, 375
- Laki, László János, 295, 357, 417, 431  
Lévai, Dániel, 31, 403  
Ligeti-Nagy, Noémi, 417, 431  
Lipp, Veronika, 77  
Lukács, András, 549
- Mády, Katalin, 199

Makrai, Márton, 105, 505  
Mihajlik, Péter, 45, 199, 633

Nagy, Attila, 343  
Nagy, Dániel, 587  
Nagy, Soma Bálint, 633  
Nanys, Patrick, 343  
Naszódi, Mátyás, 477  
Nemeskey, Dávid Márk, 121, 403  
Novák, Attila, 45, 389  
Novák, Borbála, 389  
Nyéki, Bence, 417

Orosz, György, 59, 403  
Orosz, Tamás, 521, 587  
Osváth, Mátyás, 311

Pákáski, Magdolna, 175  
Pálfi, Gellért, 633  
Palkó, Gábor, 375  
Pethő, Gergely, 77

Ring, Orsolya, 325, 491, 521

Sárközi-Lindner, Zsófia, 375  
Sass, Bálint, 77  
Simon, Eszter, 403  
Simon, László, 77  
Svindt, Veronika, 151  
Szabó, Gergő, 59  
Szabó, Martina Katalin, 325

Szántó, Zsolt, 59, 403  
Szaszák, György, 271, 505  
Szécsényi, Tibor, 535, 561  
Szemes, Botond, 93  
Szeteli, Anna, 449  
Szeteli, Ádám, 449  
Szlávich, Eszter, 93, 375  
Sztahó, Dávid, 137, 271

Terbe, Dániel, 161  
Timári, Mária, 375  
Tóth, László, 151, 161, 175  
Tündik, Ákos Máté, 505

Üveges, István, 491, 521, 587, 619

Vadász, Noémi, 403, 417, 431  
Vadász, János Pál, 587  
Váradi, Tamás, 417, 431, 603  
Vetráb, Mercedes, 175  
Vida, Bence, 375  
Vincze, Veronika, 325  
Virág, Nándor, 535  
Vági, Renáto, 587

Yang, Zijian Győző, 15, 241, 281,  
295, 311, 357, 417, 431,  
463, 573, 603

Zainkó, Csaba, 45