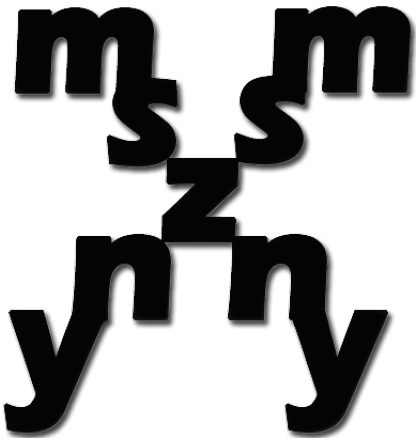
# XV. Magyar Számítógépes Nyelvészeti Konferencia



Szerkesztette:
Berend Gábor
Gosztolya Gábor
Vincze Veronika

Szeged, 2019. január 24–25.

**Szerkesztette**[1]**:**
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.

**Nyomtatta:**
JATEPress
6722 Szeged, Petőfi Sándor sugárút 30–34.

Szeged, 2019. január

**Az MSZNY 2019 konferencia szervezője:**
MTA-SZTE Mesterséges Intelligencia Kutatócsoport

[1] a LaTeX's 'confproc' csomagjára támaszkodva

# Előszó

2019. január 24-25-én tizenötödik alkalommal rendezzük meg Szegeden a Magyar Számítógépes Nyelvészeti Konferenciát. 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. Nagy örömet jelent számunkra, hogy a hagyományokat követve a konferencia idén is nagyfokú érdeklődést váltott ki az ország nyelv- és beszédtechnológiai szakembereinek körében.

Az évek során hagyománnyá vált az is, hogy a mesterséges intelligencia vagy a számítógépes nyelvészet egy-egy kiemelkedő alakja plenáris előadást tart a konferencián. Az idei évben Turán György (MTA-SZTE Mesterséges Intelligencia Kutatócsoport és University of Illinois at Chicago) előadásában az interpretálhatóságról és annak számítógépes nyelvészeti vonatkozásairól lesz szó.

Az idei évben is szeretnénk különdíjjal jutalmazni a konferencia legjobb cikkét, mely a legkiemelkedőbb eredményekkel járul hozzá a magyarországi nyelv-és beszédtechnológiai kutatásokhoz. Továbbá idén először tervezzük bevezetni a "Legjobb Bírálók Díját" is, így elismerve a bírálók fáradságos, ámde nélkülözhetetlen munkáját. A konferenciához idén is kapcsolódni fog egy kerekasztal-megbeszélés, ahol a főbb szakmai kérdéseket, a szakterület jelenlegi helyzetét és várható haladási irányát, valamint a konferenciához közvetlenül kapcsolódó kérdéseket vitatják meg a résztvevők.

Köszönettel tartozunk a LogMeIn-nek, a Neumann János Számítógéptudományi Társaságnak, valamint a Clementine-nak is, akik anyagi támogatásukkal járultak hozzá a konferencia sikeres lebonyolításához. Az előzőeken felül hálásak vagyunk az MTA-SZTE Mesterséges Intelligencia Kutatócsoportján és a Szegedi Tudományegyetem Informatikai Intézetének Szoftverfejlesztés Tanszékén dolgozó azon kollégáknak, akik a helyi szervezésben segédkeztek. 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
Novák Attila
Simon Eszter
Sztahó Dávid
Vincze Veronika

# Tartalomjegyzék

# Beszédtechnológia I.

# Beszélőinvariáns akusztikus modellek létrehozása mély neuronhálók ellenséges multi-taszk tanításával

Tóth László[1], Gosztolya Gábor[2]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
{tothl, ggabor}@inf.u-szeged.hu

**Kivonat** Bár a mély neuronhálós technológia bevezetésével a beszédfelismerő rendszerek pontossága rengeteget javult, a környezeti tényezőkkel szembeni robusztusságuk növelése továbbra is az egyik legfontosabb kutatási terület. Cikkünkben egy nemrégiben javasolt eljárást, a neuronhálók ellenséges multi-taszk tanítását próbáltuk bevetni a beszélő személyére való érzékenység csökkentésére. Ehhez olyan tanító adatbázisra van szükség, ami a szöveges átirat mellett a beszélő személyére vonatkozó annotációt is tartalmaz. Bár a kiindulási alapként szolgáló cikkhez képest jóval több beszélővel, valamint teljesen kapcsolt neuronháló helyett konvolúciós hálóval dolgoztunk, ennek ellenére minden konfigurációban konzisztens 2-3% körüli relatív hibacsökkenést kaptunk. A módszert beszélőklaszterezéssel kiterjesztve arra az esetre is adunk egy megoldási javaslatot, amikor nem áll rendelkezésre beszélőannotáció. A kezdeti eredmények bíztatóak, ebben a felügyelet nélküli esetben is hibacsökkenést mértünk, habár a felügyelt esethez képest szerényebb mértékűt.
**Kulcsszavak:** beszédfelismerés, mély neuronhálók, multi-taszk tanulás, ellenséges tanulás

## 1. Bevezetés

A mély neuronhálókon alapuló beszédfelismerési technológia ma már széles körben elfogadott és elterjedt [1]. Azonban továbbra is kihívás, hogy ezeket a rendszereket robusztussá tegyük, azaz hatékonyságuk ne romoljon a legkülönfélébb felhasználási körülmények között sem. Sajnos ilyen zavaró tényező rengeteg létezik, a beszélő személy hangjának egyedi sajátosságaitól a mikrofonok eltérő átviteli karakterisztikáján át a beszűrődő háttérzajig. A neuronhálók általánosító képességének növelésére az egyik lehetőség a regularizációs módszerek használata a betanítás során. Általánosan megfogalmazva, a regularizáció célja, hogy a háló ne tanuljon rá nagyon specifikusan az aktuális tanítóadatokra, mert ez az ún. túltanulás az új adatokra való általánosítási képesség csökkenését okozhatja. A túltanulás csökkentésének egy lehetséges módja, ha a hálónak több feladatot kell megtanulnia egyszerre, ez az ún. multi-taszk tanítás [2]. Megfigyelték ugyanis, hogy ha ezek a feladatok kicsit eltérnek, de hasonló jellegűek, azaz hasonló

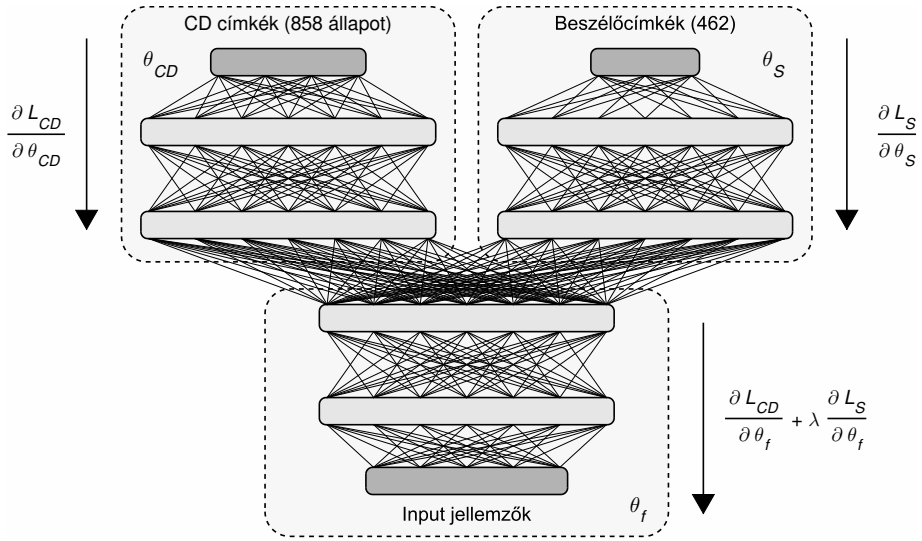
belső reprezentáció kialakítását igénylik, akkor a két feladat egyidejű tanulásának köszönhetően a háló robusztusabbá válik, és gyakran mindkét feladaton jobb pontosságot ér el, mint külön-külön tanításnál. A multi-taszk tanítást a beszédfelismerésben is többen sikeresen alkalmazták már [3,4].

Míg a sztenderd multi-taszk tanításnál arra törekszünk, hogy a háló a másodlagos feladaton is kis hibát érjen el, létezik a módszernek egy ellenséges (adversarial) multi-taszk tanítás nevű változata is, ahol a másodlagos feladat hibáját nem minimalizálni, hanem *maximalizálni* próbáljuk [5]. Ettől azt várjuk, hogy a háló olyan belső reprezentációt alakítson ki, amely a másodlagos feladatra nézve invariáns. A beszédtechnológiában az ellenséges multi-taszk tanítást eddig leginkább az akusztikus modellek felvételi környezetre, például a háttérzajra való robusztussá tevésére alkalmazták [6,7], de az akcentussal [8], illetve legújabban a beszélő személyével szemben való függetlenítésre is találunk példát [9]. Mi ez utóbbival fogunk itt próbálkozni, azaz az ellenséges multi-taszk tanítástól a modell beszélőinvariánssá, de legalábbis a beszélő személyére kevésbé érzékennyé válását reméljük. Cikkünkben bemutatjuk az ellenséges tanítás módszerét, és a kapott eredmények alapján kielemezzük a megoldás előnyeit-hátrányait. A módszer egyik hátránya az lesz, hogy beszélőket azonosító annotációt igényel, ezért a kiértékelést nem magyar adatbázison, hanem az angol TIMIT adatbázison végezzük, amelynek tanító része egyenletes eloszlásban 462 beszélőtől tartalmaz mintát (Meng és társai cikkükben jóval kevesebb beszélővel dolgoztak [9]). A kiindulási cikkhez képest további lényeges eltérés lesz, hogy teljesen kapcsolt háló háló helyett konvolúciós hálót fogunk használni. Mivel a konvolúció célja eleve a beszélő személyére való érzékenység csökkentése, kérdéses, hogy konvolúciós háló esetén is segít-e az ellenséges tanítás.

## 2. Multi-taszk és ellenséges multi-taszk tanítás

A multi-taszk neuronháló sematikus felépítését szemlélteti az 1. ábra. A hálózatnak mindkét (vagy esetleg több) feladathoz van egy-egy dedikált kimenőrétege, illetve opcionálisan lehetnek feladatspecifikus rejtett rétegei is. Az ábrán a két ág hibafüggvényét $L_{CD}$ és $L_S$, az ágak paramétereit (súlyait) pedig $\theta_{CD}$ és $\theta_S$ jelölik (CD a környezetfüggő (context-dependent) állapotokat, S a beszélőket (speaker) kódolja). A hálózat inputja, valamint alsó rétegei közösek, ami technikailag annyi nehézséget okoz, hogy a hiba visszaterjesztése során a közös rétegekhez érve a két ágból érkező hibát össze kell adni (azaz az ábrán $\lambda = 1$). Ez arra kényszeríti a hálózatot, hogy ezekben a közös rétegekben olyan reprezentációt alakítson ki, amely mindkét feladat megoldását segíti.

Sajnos tudomásunk szerint jelenleg csak empirikus úton lehet kideríteni, hogy egy konkrét másodlagos feladat felvétele segíteni fogja-e vagy sem az eredeti feladat megoldását, az észszerűség azonban azt diktálja, hogy hasonló jellegű, de a fő feladattól némiképp eltérő másodlagos feladatot érdemes választani. Az is csak kísérleti úton deríthető ki, hogy mely rétegnél érdemes a hálózatot elágaztatni. A logika és a tapasztalat is azt mondja azonban, hogy minél eltérőbb a




1. ábra: Az (ellenséges) multi-taszk neuronháló struktúrája.

két feladat, annál kevesebb közös, és annál több feladatspecifikus rétegre lesz szükség [10].

Tudomásunk szerint a multi-taszk tanítást beszédtechnológiában elsőként Green és társai alkalmazták, ahol a felismerés mellett a másodlagos feladat a beszéd háttérzajtól való megtisztítása volt [11]. A mély neuronhálós világban a multi-taszk tanítás Seltzer és Droppo munkájában bukkan fel újra, akik az aktuális beszédhang felismerése mellé a kontextus, azaz a szomszédos hangok felismerését vették fel második feladatnak [3]. Nagyon hasonló ehhez Bell és Renals megoldása, akik a környezetfüggő állapotcímkék mellé a környezetfüggetlen címkék megtanulását tekintették másodlagos feladatnak [4]. Lényegében ezt a megoldást ismételtük meg korábban magyar nyelvre, és a korábban említett munkákkal egybevágóan néhány százalékos relatív hibacsökkenést értünk el [12].

Bár logikusan hangzik, hogy a közös reprezentáció egy másodlagos feladatra való érzékenyítése segíthet, ennek épp az ellenkezője, azaz a reprezentáció valamilyen szempontból invariánssá tételének hasznossága is éppen annyira indokolható. Ez utóbbi a célja az ún. ellenséges (adversarial) multi-taszk tanításnak [5], ami a beszédtechnológiában tudomásunk szerint 2016-ban bukkant fel először [6]. Ellenséges tanítás esetén a multi-taszk háló struktúrája ugyanaz marad, viszont a tanítás során a másodlagos feladathoz tartozó hibát nem minimalizálni, hanem *maximalizálni* próbáljuk. Technikailag ezt úgy oldjuk meg, hogy a másodlagos feladathoz tartozó feladatspecifikus ágban továbbra is minimalizálást végzünk; azonban a hibavisszaterjesztési folyamat során a közös jellemzőkinyerő rétegekhez érve az $\lambda$ paraméternek *negatív* értéket adunk. Ennek hatására a hálózat olyan közös reprezentáció kialakítására fog törekedni, amely alapján a feladat-

specifikus ágak a elsődleges feladatot minél pontosabban, a másodlagos feladatot viszont minél kevésbé tudják megoldani. Az így kialakított közös reprezentáció optimális esetben tehát nem fog a második feladat megoldását segítő információt tartalmazni, azaz invariáns lesz arra. A módszert a beszédtechnológiában eddig főleg arra próbálták használni, hogy a neuronhálót az aktuális környezetre érzéketlenné, "domain-invariánssá" tegyék, ahol a környezeten alapvetően a különféle háttérzajok értendők, de van példa az akcentussal szembeni robusztusság növelésére is [8]. Vizsgálatainkban Meng és társai "beszélőinvariáns" modellt ígérő módszertanát próbáltuk reprodukálni, ahol a másodlagos feladatot a beszélő felismerése képezte [9].

Shinohara cikkében azt javasolja, hogy az ellenséges tanítást csak fokozatosan vezessük be, azaz az $\lambda$ paraméter értékét fokozatosan növeljük a tanítási iterációk során [6]. Tanácsát követve az $k$-adik iterácóban a paraméter értékét az alábbi képlet szerint állítottuk be:

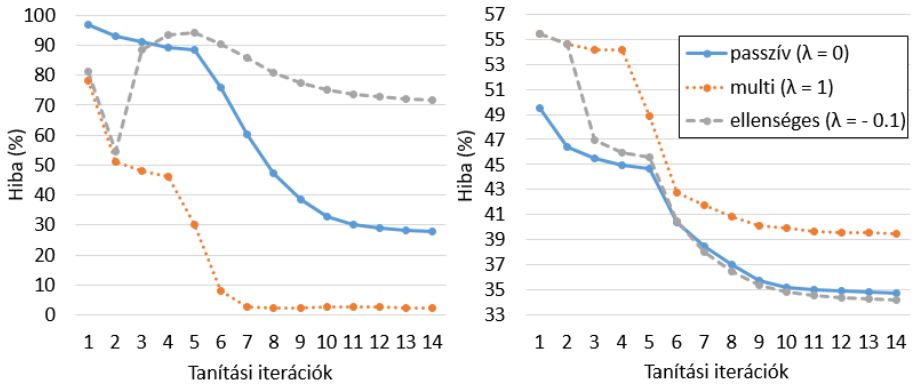
$$\lambda_k = min\Big(\frac{k}{c}, 1\Big) \cdot \lambda,$$

azaz $\lambda$ a végleges értékét $c$ iteráció után veszi fel. Shinohara cikkében $c = 10$ szerepel, de mi a $c = 7$ értékkel is kísérleteztünk, mivel tapasztalatunk szerint a tanulási ráta felezése tipikusan 6-7 iteráció után indul be.

## 3. Kísérleti beállítások

Vizsgálatainkat az angol nyelvű TIMIT beszédadatbázison végeztük. Míg ez az adatbázis beszédfelismerési szempontból már nagyon kicsinek számít, beszélőfelismerési kísérletekre ideális, mivel sok beszélőtől tartalmaz mintákat egyenletes eloszlásban. A tanító mintahalmazban 462 beszélőtől szerepel 8-8 mondat, míg a "core" teszthalmaz 24, a tanító adatoktól független beszélőből áll. Fejlesztési (development) mintaként a tanító adatokból véletlenszerűen kivett 10% mintát használtunk, így erre a halmazra a beszélőfüggetlenség nem teljesült.

Kísérleteinkben egy olyan neuronhálót alkalmaztunk, amely a legalsó rétegében konvolúciós neuronokat tartalmaz, melyek a frekvenciatengely mentén végeznek konvolúciót [13]. Megjegyezzük, hogy egy bonyolultabb hálóstruktúrával a konvolúciót az időtengelyre is kiterjeszthetjük, amivel kicsit jobb eredményeket kaphatnánk [14], de itt most a célunk nem a maximális teljesítmény elérése volt, hanem az ellenséges tanítási módszer működésének elemzése. Konvolúciós neuronhálónk a legalsó, konvolúciós rétegen kívül még két további teljesen kapcsolt réteget tartalmazott a legalsó, közös blokkban, míg a feladatspecifikus blokkok mindkét ágon 1-1 rejtett réteget használtak. A rejtett rétegek mindegyike 2000 darab "egyenirányított" (ReLU) neuronból állt. A kimentő réteg a beszédfelismerési feladathoz tartozó ágon a hibrid rejtett Markov-modelles beszédfelismerő 858 állapotának megfelelően 858 kimenő neuront tartalmazott, míg a másodlagos, beszélőfelismerési ágon a beszélők számának megfelelő 462 neuron került a kimenő rétegbe. A hálót az adatvektor-szintű keresztentrópia hibafüggvény minimalizálásával tanítottuk mindkét ágon.




2. ábra: A másodlagos feladat hibájának alakulása a tanítás során a tanítóhalmazon (bal oldal), illetve az elsődleges feladat hibája a development halmazon (jobb oldal).

## 4. Eredmények és diszkusszió

A módszer működésének megértéséhez első lépésben elvégeztünk egy kísérletet, amely a multi-taszk és az ellenséges multi-taszk tanítás hatását hasonlítja össze. Az 2. ábra bal oldali része szemlélteti, hogy tipikusan hogyan alakul a másodlagos (beszélőfelismerési) feladat hibája a tanítóhalmazon a tanítási iterációk függvényében. Elsőként $\lambda$ értékét nullára állítottuk. Ez azt jelenti, hogy a fő feladat mellett a másodlagos ág is tud ugyan tanulni, de a közös rétegekben kialakuló rejtett reprezentációba nem szólhat bele (ezért címkéztük ezt az esetet 'passzív' tanulásként). A főágon kapott eredmény fog viszonyítási alapként szolgálni, hiszen ilyenkor ez az ág ugyanazt az eredményt adja, mint egy egyfeladatos hálókonfiguráció. Az ábrán azt láthatjuk, hogy a másodlagos ág ilyenkor is tud tanulni, 30% körüli pontosságot ér el a beszélők felismerésében. Második lépésben hagyományos multi-taszk tanítást futtatunk, azaz $\lambda$ értéke 1 volt. Az ábra azt mutatja, hogy ilyenkor a beszédfelismerő ág 3% körüli pontossággal képes azonosítani a train halmaz beszélőit. Végezetül, $\lambda$ értékét $-0,1$-re, azaz ellenséges tanulásra állítottuk, és a látványosabb hatás kedvéért két iteráció multi-taszk tanulás után váltottunk át ellenséges multi-taszk tanulásra. A másodlagos ág hibája ekkor gyorsan felszalad 90% fölé, és végig 70% fölött marad.

A 2. ábra jobb oldala mutatja ezzel párhuzamosan az elsődleges, beszédfelismerési ágon kapott hibaértékeket (ezúttal a development halmazon, mert itt már az általánosítási képeség is fontos, hiszen ezt a kimenetet fogjuk felhasználni a beszédfelismerőben). Azt láthatjuk, hogy az alaprendszerhez képest a multi-taszk esetben lényegesen megnövekszik a hiba, míg ellenséges tanítás mellett ha szerény mértékben is, de csökken.

Az eredmények számszerű kiértékelése során az $\lambda$ (és részben a $c$) paraméterek optimális értékét igyekeztünk megtalálni. A kezdeti próbálkozások alapján $c$ ér-

| Paraméterek | | Keretszintű hiba | | Felism. hiba |
|---|---|---|---|---|
| $\lambda$ | $c$ | 1. ág (dev) | 2. ág (train) | (teszthalmaz) |
| 0 (passzív) | – | 34,7% | 36% | 18,6% |
| -0,03 | 7 | 34,3% | 57% | 18,3% |
| -0,06 | 7 | **34,1%** | 73% | 18,1% |
| -0,10 | 7 | 34,3% | 82% | 18,1% |
| -0,10 | 10 | 34,2% | 79% | 17,9% |
| -0,15 | 10 | 34,4% | 85% | **17,8%** |
| -0,20 | 10 | 34,6% | 90% | 18,1% |

1. táblázat. Beszédhang-felismerési hibaarányok különböző paraméterértékekkel.

tékét 7-re állítottuk, $\lambda$-t pedig 0,03 és 0,1 között változtattuk. Az 1. táblázat mutatja a kapott hibaértékeket – az összehasonlítás alapjául szolgáló 'passzív' konfigurációt az első sorban tüntettük fel. Az első eredményoszlop a neuronháló keretszintű hibáját mutatja a development halmazon, a másodikban érdekességképp a másodlagos feladat keretszintű hibáját tüntettük fel (ezt csak a tanítóhalmazon mértük), végül a felismerő lefuttatása után a teszthalmazon kapott beszédhang-felismerési hibaarányokat az utolsó oszlop mutatja. Rögzített $c = 7$ érték mellett szépen látszik, hogy $\lambda$ növelésével a másodlagos feladat hibája is nő, miközben a fő feladat hibája konzisztensen alatta marad az alaprendszerének. Az is az elvártnak megfelelő viselkedés, hogy $c$ értékét 10-re növelve $\lambda$ értékét is növelni lehetett. A development halmazon a keretszintű hiba $c = 7, \lambda = -0,06$ esetén érte el a minimumát, míg a teszthalmazon a felismerési hiba $c = 10, \lambda = -0,15$ mellett. Ennek az lehet az oka, hogy a development halmazunk beszélői a tanítópéldák között is szerepeltek. A megbízhatóbb kiértékeléshez meg kell majd ismételnünk a kísérletet a development halmaz beszélőfüggetlen újratervezésével. Azt a tanulságot azonban mindenképpen le tudtuk vonni, hogy a módszer valóban segít, hiszen konzisztensen minden esetben hibacsökkenést tapasztaltunk. A csökkenés mértéke átlagosan 3% körüli volt, a teszthalmazon kapott legjobb érték 3,8% relatív hibacsökkenésnek felel meg. Összevetésképp, Meng és társai 5%-os javulásról számoltak be [9]. Az eltérés oka az lehet, hogy mi konvolúciós hálót használtunk, ami eleve csökkenti a háló beszélő személyére való érzékenységét. Korábbi méréseink szerint a TIMIT adatbázison a felismerési eredmények beszélők szerinti szórását a konvolúció bevezetése 5,7%-kal csökkentette [15].

A javulás szerény mértéke miatt az is felvetődött bennünk, hogy az eredmények esetleg pusztán a tanulásba bevezetett 'zajnak' köszönhetően javultak – ismert ugyanis, hogy némi zaj hozzáadása a tanításhoz javítani tudja a neuronhálók általánosítási képességét. Ennek ellentmond azonban, hogy $\lambda$ előjelét megfordítva egyértelmű romlást tapasztaltunk ($\lambda = 0,1$ esetén is). A biztonság kedvéért kiszámoltuk a felismerési hiba beszélőkre nézve vett szórását is. Azt találtuk, hogy a 17,8%-ot elérő modell szórása az alaprendszeréhez képest kb. 10%-kal alacsonyabb. Ez igazolja, hogy az ellenséges tanításnak valóban olyan

| Paraméterek | | Keretszintű hiba | | Felism. hiba |
|---|---|---|---|---|
| $\lambda$ | $c$ | 1. ág (dev) | 2. ág (train) | (teszthalmaz) |
| 0 (passzív) | – | 34,7% | 36% | 18,6% |
| -0,10 | 10 | 34,1% | 70% | 18,4% |
| -0,06 | 7 | 34,3% | 65% | 18,3% |

2. táblázat. Beszédhang-felismerési hibaarányok beszélőklaszterezéssel.

hatása volt, mint amit vártunk tőle. Ennek ellenére a kapott modellt beszélőinvariánsnak nevezni erős túlzás – például Meng és társai további komoly javulást kaptak a modellen beszélőadaptációt alkalmazva [9].

### 4.1. Felügyelet nélküli eset

A Meng és társai által javasolt módszer komoly hátulütője, hogy beszélők szerint annotált adatbázist igényel. Bár a TIMIT esetén rendelkezésre áll ilyen annotáció, a legtöbb, beszédfelismerők betanításához összeállított korpusz nem tartalmaz ilyen információt. Az ilyen esetek kezelésére valamilyen felügyelet nélküli tanítási módszert kell bevetnünk. Mi azzal próbálkoztunk, hogy a tanító adatbázis fájljait klaszterezés segítségével csoportokra bontottuk. A klaszterezésre egy hierarchikus beszélőklaszterezési módszert alkalmaztunk [16,17,18]. A klaszterek számát 50-re állítottuk, $\lambda$-t pedig a korábban legjobb eredményeket adó értékre állítottuk be. A 2. táblázatban látható kezdeti eredmények biztatóak, mivel a keretszintű hiba a validációs halmazon a korábbiakhoz hasonló módon csökkent; a teszthalmazon kapott felismerési eredmények azonban szerényebb javulást mutatnak, mint a valódi beszélőcímkék használata esetén. Ezért további, alaposabb kiértékelést tervezünk a klaszterméret változtatásával, valamint más klaszterező algoritmusok kipróbálásával.

## 5. Összegzés

Cikkünkben egy nemrégiben javasolt gépi tanulási technikát, a mély neuronhálók ellenséges multi-taszk tanítását vizsgáltuk, a módszerrel a gépi beszédfelismerők akusztikus modelljének beszélőkre való érzékenységét akartuk csökkenteni. Kísérleteinkben a módszer konzisztensen 2-3% körüli relatív hibacsökkenést hozott. Ez kisebb, mint a kiindulási alapként felhasznált publikációban szereplő 5%, aminek oka az lehet, hogy az eredeti cikkel szemben mi konvolúciós hálót használtunk, ami eleve kevésbé érzékeny a beszélők közti eltérésekre. A módszert kiterjesztve egy megoldási lehetőséget javasoltunk arra az esetre is, amikor a tanítókorpuszhoz nem áll rendelkezésre beszélőkre vonatkozó annotáció. A módszer ebben az esetben is működni látszik, bár a kapott hibacsökkenés szerényebb. A jövőben ennek a felügyelet nélküli megoldásnak az alaposabb kivizsgálását tervezzük, a klaszterméretek és a klaszterezési algoritmusok széles körű vizsgálatával.

## Köszönetnyilvánítás

Tóth Lászlót az MTA Bolyai János Kutatási Ösztöndíja, valamint az Emberi Erőforrások Minisztériuma ÚNKP-18-4 kódszámú Új Nemzeti Kiválóság Programja támogatta. A kutatást az Emberi Erőforrások Minisztériuma Emberi Erőforrások Minisztériuma 20391-3/2018/FEKUSTRAT kódjelű pályázata támogatta. A kutatáshoz használt grafikus kártyát az NVIDIA Corporation ajándékozta csoportunknak.

## Hivatkozások

1. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine **29**(6) (2012) 82–97
2. Caruana, R.: Multitask learning. Journal of Machine Learning Research **17**(1) (1997) 41–75
3. Seltzer, M., Droppo, J.: Multi-task learning in deep neural networks for improved phoneme recognition. In: Proc. ICASSP. (2013) 6965–6969
4. Bell, P., Renals, S.: Regularization of deep neural networks with context-independent multi-task training. In: Proc. ICASSP. (2015) 4290–4294
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, H., Larochelle, H., Laviolette, F., Marchand, M., Lempitzky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**(59) (2016) 1–35
6. Shinohara, Y.: Adversarial multi-task learning of deep neural networks for robust speech recognition. In: Proc. Interspeech. (2016) 2369–2372
7. Denisov, P., Vu, N., Font, F.: Unsupervised domain adaptation by adversarial learning for robust speech recognition. In: Proc. ITG Conference of Speech Communication. (2018)
8. Sun, S., Yeh, C., Hwang, M., Ostendorf, M., Xie, L.: Domain-adversarial training for accented speech recognition. In: Proc. ICASSP. (2018) 4854–4858
9. Meng, Z., Li, J., Chen, Z., Zhao, Y., Mazalov, V., Gong, Y., Juang, B.: Speaker-invariant training via adversarial learning. In: Proc. ICASSP. (2018) 5969–5973
10. Tóth, L., Grósz, T., Markó, A., Csapó, T.: Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces. In: Proc. Interspeech. (2018) 3172–3176
11. Lou, Y., Lu, Y., Seghal, S., Gupta, S., Du, J., Tham, C., Green, P., Vincent, W.: Multitask learning in connectionist speech recognition. In: Proc. Australian International Conference on Speech Science and Technology. (2004)
12. Tóth, L., Gosztolya, G.: Adaptation of DNN acoustic models using KL-divergence regularization and multi-task training. In: Proc. SPECOM. (2016) 108–115
13. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.: Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition. In: Proc. ICASSP. (2012) 4277 – 4280
14. Tóth, L.: Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition. In: Proceedings of ICASSP. (2014) 190–194
15. Tóth, L.: Phone recognition with hierarchical convolutional deep maxout networks. EURASIP Journal on Audio, Speech and Music Processing **25** (2015)

16. Han, K.J., Kim, S., Narayanan, S.S.: Strategies to improve the robustness of Agglomerative Hierarchical Clustering under data source variation for speaker diarization. IEEE Transactions on Audio, Speech and Language Processing **16**(8) (2008) 1590–1601
17. Wang, W., Lu, P., Yan, Y.: An improved hierarchical speaker clustering. Acta Acustica **33**(1) (2008) 9–14
18. Kaya, H., Karpov, A., Salah, A.: Fisher Vectors with cascaded normalization for paralinguistic analysis. In: Proceedings of Interspeech. (2015) 909–913

# Autoenkóderen alapuló jellemzőreprezentáció mély neuronhálós, ultrahang-alapú némabeszéd-interfészekben

Pintér Ádám[1], Gosztolya Gábor[1,2], Tóth László[1], Grósz Tamás[1], Csapó Tamás Gábor[3,5], Markó Alexandra[4,5]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
[3]Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
[4]Eötvös Loránd Tudományegyetem, Fonetikai Tanszék
[5]MTA-ELTE Lendület Lingvális Artikuláció Kutatócsoport
{ ggabor, tothl, groszt } @ inf.u-szeged.hu
csapot @ tmit.bme.hu, marko.alexandra @ btk.elte.hu

**Kivonat** A neurális hálón alapuló némabeszéd-interfészek általában a teljes ultrahangkép alapján becslik meg a spektrális paramétereket, melyekből a vokóder aztán beszédet generál. Habár ez a megközelítés igen kézenfekvő, és tapasztalataink szerint érthető beszédet képes generálni, több hátránya is van: egyrészt nehezen ragadja meg az egymáshoz közel eső területek (gyakorlatilag a pixelek) közötti összefüggéseket, másrészt igen pazarló. Könnyen belátható, hogy a képpontok egy jelentős része irreleváns a spektrális paraméterek becslése szempontjából, a szomszédos képpontok által tárolt információ nagyon rendundáns, a mély háló mérete pedig nagy a sok jellemző miatt. Jelen cikkünkben ezen problémák kezelésére egy autoenkóder neurális hálót tanítunk az ultrahangképre, és a szintézishez szükséges spektrális paraméterek becslését az autoenkóder háló rejtett bottleneck rétegében található neuronok aktivációi alapján végezzük egy második mély hálóval. Kísérleti eredményeink alapján a javasolt eljárás hatékonyabb, mint a hagyományos megközelítés: a kapott átlagos négyzetes hibák minden esetben alacsonyabbak, a korrelációértékek pedig magasabbak voltak, mint a standard technikával kapottak. További előnye az eljárásnak, hogy, a bottleneck réteg (relatíve) alacsony neuronszáma miatt több szomszédos kép felhasználása a becslés során nem jár a paraméterszám lényeges növekedésével, miközben szignifikánsan javítja a paraméterbecslés pontosságát.
**Kulcsszavak:** némabeszéd-interfész, mély neuronháló, autoenkóder

## 1. Bevezetés

Az utóbbi évtizedben megnőtt az érdeklődés a beszédjel artikulációs jellemzőkből való helyreállítása iránt, ami az ún némabeszéd-interfészek (Silent Speech Interface, SSI) alapját képezi [1]. Ezen a területen a feladat a beszédjel rekonstruálása az artikulációs szervek (pl. nyelv vagy ajkak) mozgásából anélkül, hogy az

alany valóban beszédjelet produkálna. A némabeszéd-interfészeknek kézenfekvő alkalmazási területeik lehetnek a beszédképzésben sérültek (pl. gégeeltávolításon átesett betegek) életminőségének javításában, illetve a beszéd továbbításában extrémen zajos környezetben (pl. katonai alkalmazásokban). Az artikulációs adatok rögzítése történhet ultrahangos képalkotással (ultrasound tongue imaging, UTI) [2,3,4,5,6], elektromágneses artikulográffal (electromagnetic articulography, EMA) [7,8], állandó mágneses artikulográffal (permanent magnetic articulography, PMA) [9], elektromiográfiával (electromyography, EMG) [10], avagy a fentieket keverő multimodális megoldásokkal [11].

A jelenlegi legkorszerűbb SSI rendszerek a „közvetlen szintézis" alapelvét alkalmazzák, vagyis a beszédjelet közbeeső átalakítások (pl. beszédhangok felismerése) nélkül, közvetlenül az artikulációs szervek mozgásából kinyert jellemzőkből állítják elő, vokóder használatával [3,4,5,8,9]. Ebben a folyamatban egy hangsúlyos gépi tanulási lépés az artikulációs jellemzők (pl. ultrahangképből nyert vektorok) alapján a vokóder (spektrális) paramétereinek becslése, melyre általában mély neurális hálót (Deep Neural Network, DNN, pl. [6,8,9]) vagy Gauss keverékmodellt (Gaussian Mixture Model, GMM, pl. [12,13]) szokás használni.

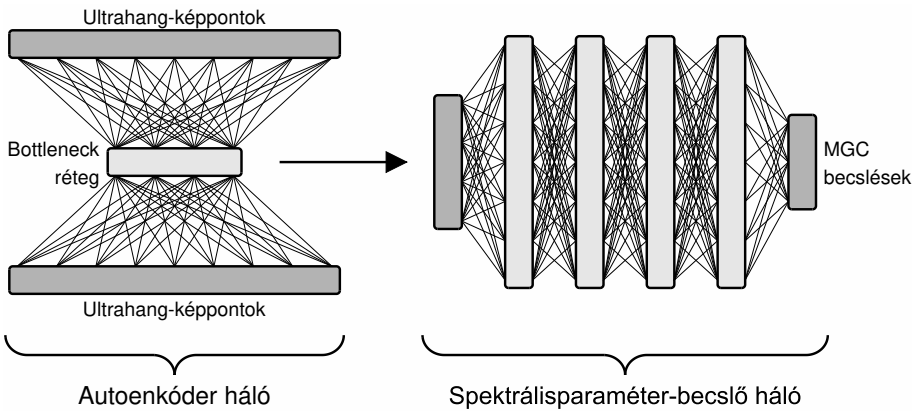
Az ultrahangkép-alapú SSI esetében a gépi tanuló eljárás bemenetét egy képkocka pixelei jelentik. Könnyen látható, hogy ez a megközelítés, bár kézenfekvő és korábbi tapasztalataink (ld. pl. [6,14,15,16]) alapján érthető beszéd szintetizálását teszi lehetővé, több tekintetben is szuboptimális. A bemenetként használt, képenként több ezer képpont (pl. a teljes nyers képkocka $64 \times 842$ méretű, azaz 53 888 képpontból áll) nagymértékben redundáns, valamint sok irreleváns jellemzőt is tartalmaz (bár ezen jellemzőkiválasztással lehet segíteni [14]). A túl sok jellemző az alkalmazott mély háló hatékonyságára (tanítási és kiértékelési idők, tárolt súlyok száma) egyértelműen negatív hatással van, és a spektrális paraméterek becslését is ronthatja. Egy hatékony tömörítési eljárással mindkét területen javíthatunk.

Jelen cikkünkben a bemenetként használt ultrahangképet egy autoenkóder hálózat segítségével tömörítjük, és a beszédszintézis spektrális paramétereit a bottleneck réteg aktivációit mint jellemzőket használva becsüljük egy második mély neurális hálóval. Kísérleti eredményeink alapján a javasolt megközelítés pontosabb paraméterbecslést tesz lehetővé, miközben a DNN mérete jelentősen csökken.

## 2. Némabeszéd-interfész spektrális paramétereinek becslése autoenkóder hálók használatával

### 2.1. Autoenkóder neurális hálók

Az autoenkóder neurális hálózat tanítására egy olyan felügyelet nélküli gépi tanulási eljárást alkalmazunk, melynek eredményeképpen a háló a rejtett rétegeiben az eredeti információ egy tömörebb változatát állítja elő, majd ezt a kimeneti rétegig visszafejti [17]. Célja, hogy bejövő paraméterekből egy identitásfüggvényhez hasonló leképezést tanuljon meg egy kompaktabb reprezentáción keresztül.




1. ábra: A javasolt kétlépéses DNN-alapú MGC-paraméterbecslő eljárás működési sémája.

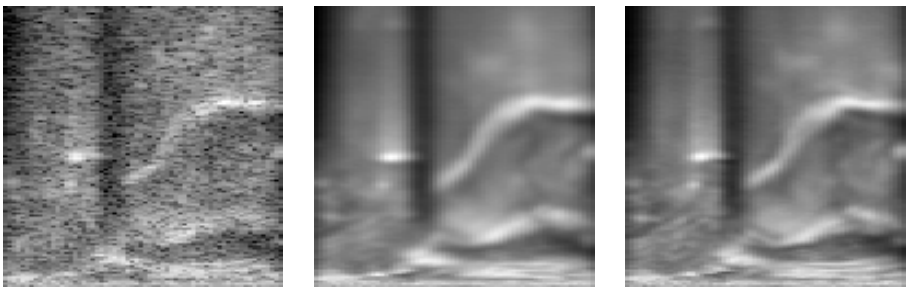
Technikailag általában egy olyan neurális hálóval valósítják meg, melynek a tanítás során elvárt kimenete megegyezik a bemenettel. Tömörítéskor az egyik rejtett rétegnek a bemenő jellemzők számánál lényegesen kevesebb neuronból kell állnia (*bottleneck* réteg). Korábbi kísérletek megmutatták, hogy ez a módszer alkalmas az egyes bemenetek közötti kapcsolatok feltárására [18], zajszűrésre [19], tömörítésre [20] vagy éppen új példák generálására a korábbi adatok alapján [21]. Az autoenkóder hálókat használják többek között képfeldolgozási [20,22], hangfeldolgozási [18] és természetes nyelvi feldolgozási [23] területeken.

Egy autoenkóder háló struktúráját tekintve két fő részből áll: az enkóder rész felelős a tömör reprezentáció előállításáért, a dekóder pedig a tömör információ alapján a bemenet visszaállításáért. A korábban említett bottleneck réteg a két rész metszetében található, ebben a rétegben számítódik/alakul ki a bemenet kódolt változata.

### 2.2. A spektrális paraméterek becslése autoenkóder hálók használatával

Jelen dolgozatunkban a beszédszintézis spektrális paramétereinek becslésére egy kétlépéses eljárást javaslunk, mindkét lépésben valamilyen mély neurális hálót alkalmazva. Az első lépésben egy autoenkóder hálót tanítunk egy-egy ultrahangkép pixeleinek rekonstruálására. A második lépésben egy újabb mély neurális hálót tanítunk, az autoenkóder háló bottleneck rétegében található neuronok aktivációit használva jellemzőként. Ennek a második hálónak a feladata már a beszédszintézis lépés paramétereinek predikciója (ld. 1. ábra).

Véleményünk szerint ennek a megközelítésnek több előnye is van. Az egyik pozitívum, hogy az autoenkóder háló észleli a szomszédos képpontok redundanciáját és képes az egymástól távolabb eső pixelek közti kapcsolatok felfedezésére is. Egy másik lehetséges előnye a javasolt megoldásnak azzal van kapcsolatban,



2. ábra: Egy szájüreg-ultrahangkép eredeti felvétele (balra), valamint az autoenkóder hálóval visszaállítva $N = 64$ (középen) és $N = 512$ (jobbra) neuront használva a bottleneck rétegben.

hogy az ultrahangkép természeténél fogva zajos. Reményeink szerint az autoenkóder háló azzal, hogy csak a tendenciaszerű változásokat kódolja a bottleneck rétegében, automatikusan elvégez egy zajszűrési lépést is. A harmadik előny, mellyel megközelítésünk rendelkezik, a tömörítéssel kapcsolatos. Egy általunk használt, standard felépítésű háló súlyainak számát nagymértékben határozza meg a bemeneti jellemzők száma; például a teljes, bár $64 \times 128$-ra átméretezett ultrahangkép pixeleinek megfelelő 8 192 bemeneti neuron és az első rejtett réteg 1 024 neuronja között kb. 8,4 millió kapcsolat van. Mivel a bottleneck réteg természetszerűleg (relatíve) kevés neuronból áll, ennek aktivációit használva bemenetként a végső hálónk jóval kevesebb kapcsolatból, így kevesebb súlyból állhat, amely mind tárolási szempontból, mind a predikció időigénye szempontjából előnyös. Amennyiben pedig, korábbi kísérleteinket követve (ld. pl. [6,14,15]), a szomszédos ultrahangképeket is felhasználjuk az aktuális keret MGC értékeinek megbecslésére, lehetőségünk nyílik lényegesen több szomszédos „kép” használatára úgy, hogy a háló súlyainak száma nem lesz nagyobb, mint az eredeti hálóé.

A 2. ábrán egy eredeti szájüreg-ultrahang kép látható (bal oldal), valamint ennek autoenkóder háló által visszaállított két változata; a középső kép esetén a bottleneck réteg 64 neuronból állt, míg a jobb oldali képnél 512 neuront tartalmazott. Látható, hogy az eredeti kép igen zajos, míg a visszaállított képek sokkal simábbak. A több rejtett neuront tartalmazó háló láthatólag több apró részletet őrzött meg az eredeti ultrahang-felvételből, mint a csupán 64 rejtett neuronnal rendelkező: utóbbi esetben a kép sokkal homályosabb, ugyanakkor a nyelv kontúrja itt is jól kivehető. Természetesen nem egyértelmű, hogy a konkrét feladat esetén legalább hány neuron szükséges optimális vagy közel optimális teljesítményhez.

## 3. Kísérletek

A következőkben bemutatjuk az elvégzett kísérletek technikai körülményeit: az alkalmazott adatbázist, a neurális hálók paramétereit és a kiértékeléskor használt metrikákat.

### 3.1. A felvételek rögzítése

A kísérletekhez használt felvételeket egy (42 éves) magyar anyanyelvű, beszédképzési problémával nem rendelkező nő segítségével rögzítettük, aki összesen 438 mondatot olvasott fel. Eközben a nyelv mozgását az Articulate Instruments Ltd. által gyártott „Micro" típusú ultrahang-berendezéssel rögzítettük 82 kép/másodperc sebességgel. Ezzel párhuzamosan a beszédjelet is felvettük egy Audio-Technica – ATR 3350 típusú kondenzátormikrofonnal (további részletekért lásd [6,14]). A továbbiakban ismertetett kísérletek inputját a nyers ultrahangfelvételek képezték. A 438 felvételt szétosztottuk tanító (310 felvétel), fejlesztési (41 felvétel) és teszthalmazra (87 felvétel).

### 3.2. Előfeldolgozás és szintetizálás

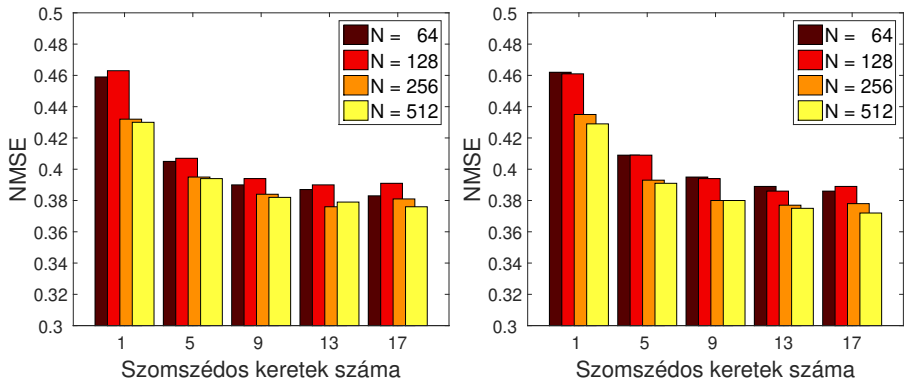
Az ultrahangképeket feldolgozás előtt az eredeti 64×946 felbontásról 64×128 pixelre méreteztük át. Az eredetileg $[0, 255]$ skálát használó pixelértékeket a képfeldolgozásban megszokott módon (ld. pl. [24]) elosztottuk 255-tel, így $[0, 1]$ skálára konvertálva azokat. A beszédjel elemzésére és szintetizálására a nyílt forrású SPTK eszköztár egyik vokóderét használtuk (http://sp-tk.sourceforge.net). A beszédjelet újramintavételeztük 22 050 Hz-en. A spektrális burkológörbét 24 MGC-LSP együtthatóval, valamint az energiaértékkel reprezentáltuk, ami összességében egy 25-dimenziós vektort eredményezett. A paramétereket az ultrahangképekkel szinkronban, 12 ms kereteltolással nyertük ki. A mély neuronhálók tanítása során az előbbi vektor standardizált változata képezte a megtanulandó célvektort.

### 3.3. A neurális háló paraméterei

A neurális hálók megvalósításához a Tensorflow [25] keretrendszert használtuk; a rejtett rétegekben minden esetben Swish aktivációs függvényt alkalmazó neuronokat alkalmaztunk [26], míg a beszédszintézis-paraméterek becslését szolgáltató 25 neuronnál lineáris aktivációt használtunk. A Swish neuronok $\alpha$ paraméterét 1.0 értéken rögzítettük.

A viszonyítási alapként szolgáló mély háló esetében a bemeneti réteg megfelelt az ultrahangkép képpontjainak, így 8 192 neuront tartalmazott, míg az öt rejtett réteg 1 024-1 024 neuronból állt. A súlyok kordában tartása érdekében L2 regularizációt alkalmaztunk. Korábbi kísérleteink (ld. pl. [6,14]) alapján tudtuk, hogy a szomszédos ultrahangképek használata segíthet az MGC paraméterek becslésében, így egy olyan hálót is tanítottunk, amely öt egymás utáni ultrahangképet kapott bemenetként (így ennek bemeneti rétege 40 960 neuronból állt). A tanítási célértékek a középső képkockához tartozó MGC paraméterek voltak. A két háló paramétereinek száma 12,6 millió (egy ultrahangkép esetén), illetve 46,2 millió (öt szomszédos ultrahangkép használata esetén) volt.

Az autoenkóder háló bottleneck rétegében $N = 64, 128, 256$ és 512 neuronnal kísérleteztünk, melyek közvetlenül (tehát további rejtett rétegek nélkül) voltak összekötve a bemeneti és kimeneti rétegekkel. (Ezek egy utrahangképnek voltak




3. ábra: A fejlesztési halmazon (balra) és a teszthalmazon (jobbra) mért átlagos normalizált hibaértékek az autoenkóder háló bottleneck rétegének neuronszáma ($N$) és a használt szomszédos keretek számának függvényében.

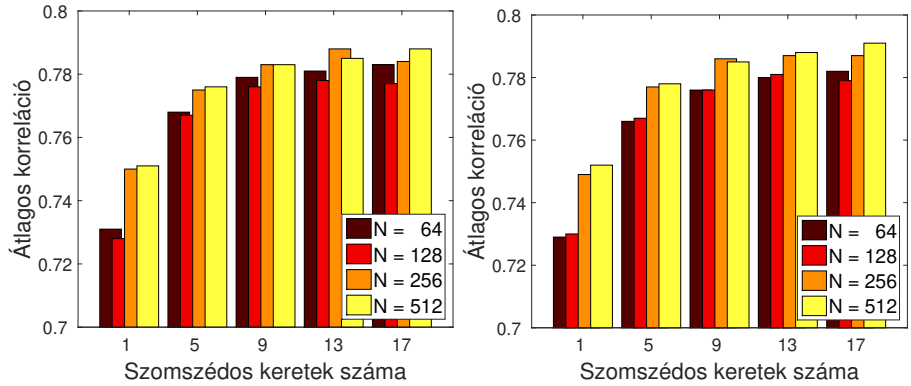
megfeleltetve, tehát 8 192 neuronból álltak.) A bottleneck réteg aktivációira tanított, MGC paraméterbecslő mély háló az előzőekhez hasonlóan egy-egy öt rejtett rétegből álló, mindegyikben 1 024 Swish neuront tartalmazó DNN volt. A teljes képhez viszonyítva lényegesen alacsonyabb jellemzőszám azt is lehetővé tette, hogy még több szomszédos „ultrahangképet” használjunk, így ebben az esetben kísérleteinket (összesen) $m = 1$, 5, 9, 13 és 17 szomszédos keret felhasználásával végeztük.

### 3.4. Kiértékelés

Mivel az MGC spektrális paraméterek becslése egy regressziós probléma, az egyes modellek kiértékelésére standard regressziós metrikákat alkalmaztunk. Az egyik lehetőség a négyzetes hiba használata; mivel 25 paramétert becsültünk, így kézenfekvő megközelítés az egyes spektrális paraméterekre kapott négyzetes hiba kiátlagolása. Ugyanakkor azt is érdemes figyelembe vennünk, hogy az egyes kimeneti értékek eltérő skálán mozoghatnak; ennek orvoslására inkább a normalizált négyzetes hibát használtuk. Egy másik lehetséges metrika az eredeti és a becsült értékek korrelációjának kiszámítása; a 25 korreláció-értéket egyszerű átlagszámítással összegeztük.

## 4. Eredmények

A 3. ábra bal oldala mutatja a mért átlagos normalizált négyzetes hibaértékeket a fejlesztési halmazon a különböző, autoenkóder-alapú konfigurációk esetén. Látható, hogy $m = 1$, illetve $m = 5$ (2-2) szomszédos keretet használva a becslések még lényegesen pontatlanabbak, mint akár $m = 9$ keret esetében; efölött viszont a javulás csak minimális, vagy egyenesen nincs is. A bottleneck réteg neuronszámát vizsgálva azt találtuk, hogy az $N = 64$ és $N = 128$ méretű hálók




4. ábra: A fejlesztési halmazon (balra) és a teszthalmazon (jobbra) mért átlagos korrelációértékek az autoenkóder háló bottleneck rétegének neuronszáma ($N$) és a használt szomszédos keretek számának függvényében.

valamivel pontatlanabb paraméterbecslést adtak, mint az $N = 256$ és $N = 512$ variációk, ugyanakkor a különbség csak akkor volt számottevő, mikor egyáltalán nem használtunk szomszédos kereteket ($m = 1$ eset). A teszthalmazon mért átlagos normalizált négyzetes hibaértékek (ld. 3. ábra jobb oldala) tendenciái szinte tökéletesen megegyeznek a fejlesztési halmazon tapasztaltakkal.

Az átlagos korrelációértékek a fejlesztési és a teszthalmazon (ld. 4. ábra) is nagyon hasonlóan alakultak: $m = 9$ szomszédos jellemzővektort használva optimális vagy aközeli értékeket kaptunk. Az autoenkóder háló bottleneck rétegében, tapasztalataink szerint, érdemes volt legalább 256 neuront használni, habár a különbség általában nem volt jelentős az egyes modellek teljesítménye között (legalább 9 szomszédos képet használva).

A konkrét értékeket (ld. 1. táblázat) megvizsgálva szembeszökő, hogy a teljes képet használva a szomszédos ultrahangképek használata, valamilyen oknál fogva, most nem javított a predikción. Az autoenkóder-alapú modellek esetén a legjobb teljesítményt az $N = 256$ eset hozta 13 (6-6) szomszédot használva mindkét metrika szerint és mindkét halmazon, de az is látható, hogy 9 szomszédot használva is csak kevéssel maradnak el az eredmények ettől a szinttől. A teszthalmazon mért 0,376-0,394 átlagos normalizált négyzetes hibaértékek 25-29%-os relatív hibacsökkenésnek felelnek meg, míg a 0,680-as átlagos korrelációértékekhez viszonyított 0,776-0,787-es értékek 30-33%-os hibacsökkentést jelentenek, melyeket bízvást nevezhetünk szignifikánsnak.

A táblázatban feltüntettük az egyes DNN-alapú modellek méretét (azaz a hálók összes súlyának számát) is. Mivel az autoenkóder-alapú konfigurációk esetében első lépésként az ultrahangkép kódolását kell elvégezni, ezekben az esetekben a feltüntetett értékek tartalmazzák az autoenkóder háló kódolásért felelős részének súlyszámait is. (Ezek 0,5 milliónak ($N = 64$), 1,0 milliónak ($N = 128$), 2,1 milliónak ($N = 256$) és 4,2 milliónak ($N = 512$) adódtak.) Látható, hogy az autoenkóder-alapú konfigurációk összesített súlyszáma csak néhány esetben

| Megközelítés | Szomsz. száma | Param. száma | NMSE | | Korreláció | |
|---|---|---|---|---|---|---|
| | | | Fejl. | Teszt | Fejl. | Teszt |
| Standard | 1 | 12,6M | 0,529 | 0,534 | 0,680 | 0,676 |
| | 5 | 46,2M | 0,523 | 0,530 | 0,684 | 0,680 |
| Autoenkóder, N = 64 | 1 | 4,8M | 0,459 | 0,462 | 0,731 | 0,729 |
| | 9 | 5,3M | 0,390 | 0,395 | 0,779 | 0,776 |
| Autoenkóder, N = 256 | 1 | 6,6M | 0,432 | 0,435 | 0,750 | 0,749 |
| | 9 | 8,7M | 0,384 | 0,380 | 0,783 | 0,786 |
| | 13 | 9,7M | 0,376 | 0,377 | 0,788 | 0,787 |
| Autoenkóder, N = 512 | 1 | 8,9M | 0,430 | 0,429 | 0,751 | 0,752 |
| | 5 | 11,0M | 0,394 | 0,391 | 0,776 | 0,778 |
| | 9 | 13,1M | 0,382 | 0,380 | 0,783 | 0,785 |

1. táblázat. A fejlesztési és a teszthalmazon mért átlagos normalizált négyzetes hibaértékek (NMSE) és átlagos korrelációértékek, valamint az egyes hálók súlyainak száma

haladta meg a viszonyítási alapként szolgáló, közvetlenül a teljes képet feldolgozó hálóét, azonban az öt egymást követő ultrahangképre tanított DNN méretétől jelentős mértékben elmaradtak. Ezen értékek alapján kijelenthetjük, hogy a javasolt, autoenkóder-alapú eljárás nemcsak pontosabb szintézisparaméter-becslésekhez vezet, hanem még számításilag is kedvezőbb.

## 5. Összegzés

Jelen cikkünkben az ultrahang-alapú némabeszéd-interfészek területén vizsgáltuk az autoenkóder neurális hálók alkalmazhatóságát. Megközelítésünkben a teljes szájüreg-ultrahangképre tanított autoenkóder háló bottleneck rétegének aktivációit mint jellemzőket használtuk, és a beszédszintézis spektrális paramétereit egy második mély hálóval becsültük. Kísérleti eredményeink alapján a javasolt eljárás a viszonyítási alapként szolgáló, pixelalapú megoldásnál hatékonyabbnak bizonyult: a becslések minden esetben pontosabbnak adódtak, és a háló súlyainak száma is csökkent. Véleményünk szerint ez több dolognak tudható be: az autoenkóder háló zajszűrési képességén kívül azt is ki tudtuk használni, hogy így az eredeti kép egy sokkal tömörebb reprezentációját állítottuk elő.

Az elvégzett kísérletek folytatására több kézenfekvő lehetőség is adódik. Az autoenkóder hálót kombinálhatjuk konvolúció alkalmazásával, mely remélhetőleg tovább növeli az eljárás hatékonyságát. Az autoenkóder-alapú reprezentációnak várhatóan nagyobb a robosztussága az ultrahang-készülék esetleges elmozdulásával szemben is, mint annak, amelyben minden képpontot a többi pixeltől független jellemzőként kezelünk. Emiatt megközelítésünk akár még a némabeszéd-interfészek beszélőfüggetlen működésének elérésében is segíthet. A közeljövőben tervezzük ilyen kísérletek elvégzését is.

## Köszönetnyilvánítás

A kutatást részben a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta (FK 124584). Tóth László munkáját az MTA Bolyai János Kutatási Ösztöndíja, valamint az Emberi Erőforrások Minisztériuma ÚNKP-18-4 kódszámú Új Nemzeti Kiválóság Programja támogatta. Grósz Tamás munkáját a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal Mesterséges Intelligencia Nemzeti Kiválósági Programja támogatta a 2018-1.2.1-NKP-2018-00008 azonosítójú projekt keretében. A cikk elkészítéséhez használt Titan-X grafikus kártyát az NVIDIA Corporation adományozta.

## Hivatkozások

1. Denby, B., Schultz, T., Honda, K., Hueber, T., Gilbert, J.M., Brumberg, J.S.: Silent speech interfaces. Speech Communication **52**(4) (2010) 270–287
2. Denby, B., Stone, M.: Speech synthesis from real time ultrasound images of the tongue. In: ICASSP, Montreal, Kanada (2004) 685–688
3. Hueber, T., Benaroya, E.l., Denby, B., Chollet, G.: Statistical mapping between articulatory and acoustic data for an ultrasound-based silent speech interface. In: Interspeech, Florence, Olaszország (2011) 593–596
4. Hueber, T., Bailly, G., Denby, B.: Continuous articulatory-to-acoustic mapping using phone-based trajectory HMM for a silent speech interface. In: Interspeech, Portland, USA (2012) 723–726
5. Jaumard-Hakoun, A., Xu, K., Leboullenger, C., Roussel-Ragot, P., Denby, B.: An articulatory-based singing voice synthesis using tongue and lips imaging. In: Interspeech, San Francisco, USA (2016) 1467–1471
6. Csapó, T.G., Grósz, T., Tóth, L., Markó, A.: Beszédszintézis ultrahangos artikulációs felvételekből mély neuronhálók segítségével. In: MSZNY 2017, Szeged (2017) 181–192
7. Wang, J., Samal, A., Green, J.: Preliminary test of a real-time, interactive silent speech interface based on electromagnetic articulograph. In: SPLAT, Baltimore, USA (2014) 38–45
8. Bocquelet, F., Hueber, T., Girin, L., Savariaux, C., Yvert, B.: Real-time control of an articulatory-based speech synthesizer for brain computer interfaces. PLOS Computational Biology **12**(11) (2016) e1005119
9. Gonzalez, J.A., Cheah, L.A., Green, P.D., Gilbert, J.M., Ell, S.R., Moore, R.K., Holdsworth, E.: Evaluation of a silent speech interface based on magnetic sensing and deep learning for a phonetically rich vocabulary. In: Interspeech, Stockholm, Svédország (2017) 3986–3990
10. Nakamura, K., Janke, M., Wand, M., Schultz, T.: Estimation of fundamental frequency from surface electromyographic data: EMG-to-F0. In: ICASSP, Prága, Csehország (2011) 573–576
11. Freitas, J., Ferreira, A.J., Figueiredo, M.A.T., Teixeira, A.J.S., Dias, M.S.: Enhancing multimodal silent speech interfaces with feature selection. In: Interspeech, Szingapúr (2014) 1169–1173
12. Janke, M., Wand, M., Nakamura, K., Schultz, T.: Further investigations on EMG-to-speech conversion. In: ICASSP, Kiotó, Japán (2012) 365–368

13. Gonzalez, J.A., Cheah, L.A., Gomez, A.M., Green, P.D., Gilbert, J.M., Ell, S.R., Moore, R.K., Holdsworth, E.: Direct speech reconstruction from articulatory sensor data by machine learning. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**(12) (2017) 2362–2374
14. Csapó, T.G., Grósz, T., Gosztolya, G., Tóth, L., Markó, A.: DNN-based ultrasound-to-speech conversion for a silent speech interface. In: Interspeech, Stockholm, Svédország (2017) 3672–3676
15. Grósz, T., Tóth, L., Gosztolya, G., Csapó, T.G., Markó, A.: Kísérletek az alapfrekvencia becslésére mély neuronhálós, ultrahang-alapú némabeszéd-interfészekben (in Hungarian). In: MSZNY, Szeged (2018) 196–205
16. Tóth, L., Gosztolya, G., Grósz, T., Markó, A., Csapó, T.G.: Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces. In: Interspeech, Hyderabad, India (2018) 3172–3176
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press, Cambridge, MA (1986) 318–362
18. Lattner, S., Grachten, M., Widmer, G.: Learning transformations of musical material using Gated Autoencoders. In: CSMC, Milton Keynes, Nagy-Britannia (2017) 1–16
19. Geras, K.J., Sutton, C.: Scheduled denoising autoencoders. In: ICLR, San Diego, USA (2015) 365–368
20. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Deep convolutional autoencoder-based lossy image compression. In: PCS, San Francisco, USA (2018) 253–257
21. Zhao, S., Song, J., Ermon, S.: Learning hierarchical features from generative models. In: ICML, Sydney, Ausztrália (2017) 4091–4099
22. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: StyleBank: An explicit representation for neural image style transfer. In: CVPR, Honolulu, Hawaii (2017)
23. Andrews, M.: Compressing word embeddings. In: ICONIP, Kiotó, Japán (2016) 413–422
24. Varga, L.: Information Content of Projections and Reconstruction of Objects in Discrete Tomography. PhD thesis, Doctoral School of Computer Science, University of Szeged (2013)
25. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.
26. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions (2018)

# Ügyfélszolgálati beszélgetések nyelvmodellezése rekurrens neurális hálózatokkal

Tarján Balázs[1,3], Fegyó Tibor[1,3], Mihajlik Péter[1,2]

[1] Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
tarjanb@tmit.bme.hu
[2] THINKTech Kutatási Központ Nonprofit Kft.
mihajlik@thinktech.hu
[3] SpeechTex Kft.
tfegyo@speechtex.com

**Kivonat:** A spontán, társalgási beszéd leírása a mai napig komoly kihívás elé állítja a gépi beszédfelismerő rendszereket. A témák sokszínűsége és a kevés tanítóadat különösen megnehezíti a nyelvi modellek tanítását. Cikkünkben telefonos ügyfélszolgálati beszélgetéseket modellezük rekurrens LSTM neurális hálózat segítségével, mellyel közel felére sikerült csökkentenünk a perplexitást a hagyományos, count n-gram modellhez képest. Azt találtuk, hogy a rekurrens LSTM akkor is felülmúlja a count modell pontosságát, ha memóriája hosszát alacsonyra korlátozzuk (LSTM n-gram). 10 vagy annál nagyobb fokszámú LSTM n-grammal pedig a korlátozás nélküli LSTM nyelvi modell teljesítménye is megközelíthető. Ez alapján arra következtetünk, hogy a rekurrens neurális nyelvi modellek pontosságának titka a hatékony simításban rejlik, nem a hosszú távú memóriában. Az új, neurális nyelvmodell segítségével nem csak a perplexitást sikerült csökkentenünk, hanem a kapcsolódó beszédfelismerési feladaton a szóhiba-arányt is relatív 4%-kal.

## 1 Bevezetés

A statisztikai nyelvmodellek számos természetes nyelvfeldolgozási feladatban játszanak kulcsszerepet. Nem kivétel ez alól a gépi beszédfelismerés sem, ahol hosszú időn át a szó n-gram count statisztikák alapján, maximum likelihood becsléssel tanított ún. **count n-gram** nyelvi modellek [1] voltak az egyeduralkodók. Az utóbbi évek során azonban először az előrecsatolt neurális hálózatokra épülő [2], majd a rekurrens neurális nyelvmodellek [3] megtörték ezt a dominanciát. A rekurrens modellek felépítésükből fakadóan jól modellezik a szövegben található hosszú távú függőségeket, mely képességet elsősorban Long Short-Term Memory (LSTM) [4] egységek alkalmazásával sikerült kiaknázni [5].

Cikkünkben egy kísérletsorozat első állomását mutatjuk be, melynek keretében a neurális nyelvmodellek gyakorlati alkalmazhatóságát kívánjuk feltérképezni magyar, majd terveink szerint idegen nyelvű beszédfelismerési feladatokban. Az itt bemutatott első kísérleti eredmények egy telefonos ügyfélszolgálati beszélgetések kézi leiratait tartalmazó adatbázison születtek. Azért erre az adatbázisra esett a választásunk, mert

aránylag kis mérete gyors tanítást tesz lehetővé, miközben a beszélgetések spontán jellege, illetve a kevés tanítóadat kellően megnehezíti a hagyományos count n-gram modellek dolgát.

A **rekurrens LSTM** nyelvmodellekkel bár valóban nagyon alacsony perplexitás érhető el, valós időben működő beszédfelismerő rendszerbe nehéz integrálni őket. A gyors dekódolás egyik feltétele ugyanis, hogy kellően kis méretűre tudjuk csökkenti a keresési teret, melyet megakadályoz, hogy a rekurrens LSTM nyelvi modell rengeteg belső állapotot vehet fel. A probléma megoldására született az **LSTM n-gram**ok koncepciója [6], melyben a count modellekhez hasonlóan korlátozzuk a valószínűségbecslés során figyelembe vett korábbi szavak számát. Az LSTM n-gramok angol és német nyelven is sikeresnek bizonyultak [6, 7], ezért úgy döntöttük, hogy a hagyományos LSTM nyelvmodell mellett ezt az új struktúrát is kiértékeljük és összevetjük a count modellek teljesítményével.

Fontosnak tartottuk, hogy már a kísérletsorozatunk elején szülessenek beszédfelismerési eredmények is rekurrens LSTM nyelvmodell felhasználásával. A rekurrens modellben tárolt tudás kezdeti kinyerésére egy egyszerű megoldást alkalmaztunk [8]: nagy mennyiségű szöveget generáltunk a neurális modell segítségével, melyből aztán count n-gram modellt tanítottunk és interpoláltuk az eredeti nyelvi modellel. Legjobb tudomásunk szerint ezek az első, publikált, magyar nyelvű beszédfelismerési eredmények, melyek neurális nyelvmodell felhasználásával jöttek létre.

A következő fejezetben a kísérleteinkhez használt tanító- és tesztadatbázisokat, majd utána a cikkünk fő témáját képező nyelvmodellezési módszereket mutatjuk be. A negyedik fejezetben ismertetjük a különböző eljárásokkal kapott szöveges és beszédfelismerési eredményeket, majd az utolsó fejezetben összefoglalását adjuk vizsgálataink legfontosabb eredményeinek.

## 2 Tanító- és tesztadatbázisok

### 2.1 Tanító-adatbázisok

A nyelvi modellek tanításához magyar nyelvű, telefonos ügyfélszolgálati beszélgetések anonimizált kézi leiratait tartalmazó adatbázist használtunk, melyre a továbbiakban **MTUBA** (Magyar Telefonos Ügyfélszolgálati Beszédadatbázis) néven fogunk hivatkozni. A normalizálás során eltávolítottuk az egyértelmű akusztikai megfeleltetéssel nem rendelkező tokeneket (írásjelek). Megtartottuk azonban a kézi leiratok eredeti mondathatárait, mely a feladat párbeszédes, spontán jellegéből fakadóan a szokásosnál rövidebb, átlagosan 6,9 szót tartalmazó mondatokat eredményezett.

A tanítókorpusz 290 órányi felvétel kézi leiratát, összesen **3,4 millió tokent** és **100 ezer egyedi szóalakot** tartalmazott. A tanítás gyorsítása és szótáron kívüli szavak modellezése céljából a végleges tanítószövegben csak a **leggyakoribb 50 ezer szóalak**ot tartottuk meg, a többit szótáron kívüli szóként modelleztük és <unk> szimbólummal helyettesítettük.

### 2.2 Tesztadatbázisok

A kísérleteinkben szereplő nyelvi modellek teszteléséhez az MTUBA e célokra kijelölt részét használtuk. A tesztszövegekben csak a tanítószöveg szűkített szótárában szereplő szóalakokat tartottuk meg, az egyéb szavakat a szótáron kívüli szavak szimbólumával (<unk>) helyettesítettük. A tesztadatbázisok vonatkozó statisztikákat a **1. táblázat** tartalmazza.

A tesztadatadatbázist két független részre bontottuk. Az első ún. **validációs tesztszöveget** a modellek hiperparaméter-optimalizálása során használtuk (learning rate szabályozás, early stopping), míg a második ún. **kiértékelő tesztadatbázist** a kész modellek szöveges és beszédfelismerési kiértékelésére. A kiértékelő tesztadatbázist további részekre osztottuk (lásd 1. táblázat). A sztereo módon rögzített felvételeken külön tudtuk vizsgálni az ügyfélszolgálatos (MTUBA sztereo 1) és az ügyfél (MTUBA sztereo 2) oldalt. Ezzel szemben a kiértékelő tesztadatbázis mono felvételein a két oldal hanganyaga egy sávra lett keverve, így csak egybe tudjuk őket kezelni (MTUBA mono).

| | **Validációs teszt** | **Kiértékelő teszt** | | | |
|---|---|---|---|---|---|
| | Σ | MTUBA sztereo 1 | MTUBA sztereo 2 | MTUBA mono | Σ |
| Tokenek száma | 45773 | 10599 | 4792 | 50921 | 66312 |
| Tesztfelvétel hossza [perc] | - | 127 | 127 | 478 | 732 |
| OOV arány [%] | 2,7 | 1,4 | 1,5 | 2,8 | 2,5 |

1. táblázat. A tesztadatbázisok jellemzői
(OOV (Out of Vocabulary) arány: szótáron kívüli szavak aránya)

## 3 Nyelvi modellezés

Cikkünk célja, hogy különböző típusú nyelvmodellezési módszereket összehasonlítsunk egy valós életből származó beszédfelismerési feladaton. Ennek érdekében hagyományos count-alapú és neurális nyelvi modelleket is alkalmaztunk. A tanítási folyamatot és az alkalmazott módszereket mutatjuk be ebben a fejezetben.

### 3.1 Count n-gram nyelvi modell

A hagyományos, count-alapú, Kneser-Ney eljárással simított [9] nyelvi modelleket az SRI nyelvi modellező eszköz segítségével [10] tanítottuk. Az SRI toolkit jellemzője, hogy alapértelmezésben a 3 és annál nagyobb fokszámú, csak egyszer előforduló n-gram-okat nem veszi figyelembe a tanítás során. Kísérleteinkben ezt a funkciót kikapcsoltuk, így a szokásosnál jóval több n-gramot tartalmazó nyelvi modelleket jöttek

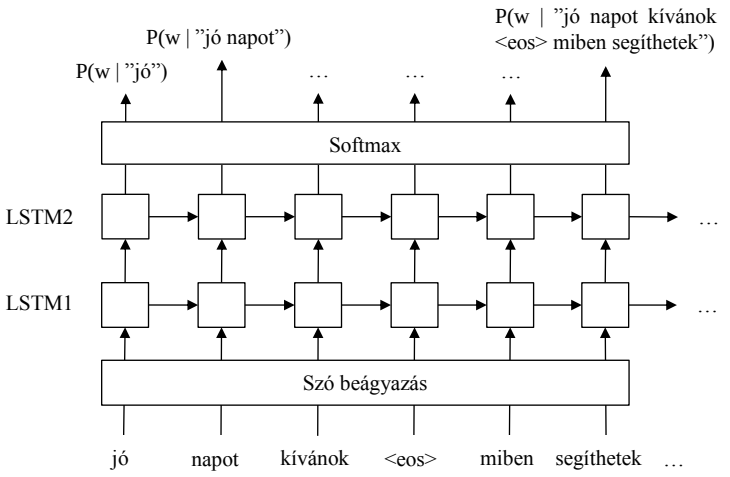
létre. Célunk ugyanis az volt, hogy egy adott fokszám mellett mindig a lehető legpontosabb count n-gram modellt tanítsuk.

Kétféle módon modelleztük a sortörést. Az első a hagyományos ún. **mondatonkénti** modellezés, melynél a sor elejére egy mondatkezdő (<s>), végére pedig egy mondatzáró (</s>) szimbólumot helyezünk, és nem engedjük meg, hogy az így létrejött mondatokon átíveljenek az n-gramok. A második, ún. **mondatösszefűzéses** módszer esetén az n-gram statisztika összeállítása során megengedjük a sorok között átívelő n-gramokat. A mondathatárok visszaállíthatósága érdekében azonban ennél a módszernél is jelöljük a sorok végét, melyre egy normál szóként modellezett speciális szimbólum szolgál (<eos>).

### 3.2 LSTM nyelvi modell

Az egyik nyelvi modell típus, mellyel a count n-gram nyelvi modelleket összevetjük cikkünkben egy 2 rétegű, Long Short-Term Memory (LSTM) [4] egységet tartalmazó, rekurrens neurális hálózat. Ezzel a típusú hálózattal korábban sikerült jelentős perplexitás csökkenést elérni a Penn Tree Bank (PTB) adatbázison [6, 11]. A hálózat felépítését az **1. ábra** szemlélteti.

A szavakat először egy szóbeágyazó mátrix segítségével vektorrá alakítjuk. A tanítás során ezután a szóvektorokat átvezetjük egy dropout [12] rétegen. A szóvektorok innen az első LSTM rétegre kerülnek, melynek kimenete egy dropout rétegen keresztül a következő LSTM réteg bemenetére van kötve. A második LSTM réteg kimenete egy újabb dropout réteg alkalmazása után kerül a softmax rétegre, melynek mérete megegyezett az alkalmazott szótár méretével. A softmax kimenetén a következő szóra vonatkozó valószínűségi eloszlást kapjuk, melyet úgy érünk el, hogy a tanítás folyamán mindig a következő szó a target, melyhez képest a hibát mérjük (cross entropy).




1. ábra: A kísérleteink során használt rekurrens LSTM nyelvi modell struktúra
( P(w | ”history”), a history után becsült szóeloszlást jelöli )

Keras [13] implementációnk[1] alapjául a Tensorflow minta nyelvi modell kódja szolgált[2], mely a [11]-ban bemutatott PTB modelleket valósítja meg. A fenti megvalósításból a hiperparaméterek egy részét is átvettük, így 650 dimenziós szóvektorokat, 650 dimenziós kimeneti réteggel rendelkező LSTM-eket, 35 hosszú szekvenciákat és 0,5-ös eldobási rátájú dropout-ot alkalmaztunk. Több optimalizáló függvény kipróbálása után végül a momentummal gyorsított, hagyományos Stochastic Gradient Descent (SGD) mellett döntöttünk. A tanulási ráta kezdeti értékének 1-et állítottunk be, és minden epoch végén feleztük, amennyiben hibanövekedést tapasztaltunk a validációs teszthalmazon. A tanítást akkor fejeztük be, ha három egymást követő epoch után sem regisztráltunk javulást a validációs teszten.

A **mondatösszefűzéses** LSTM nyelvi modellnél az egyes batch-ek között megőrizzük az LSTM állapotokat, azaz Tensorflow terminológiával élve „stateful" hálózatot tanítunk. **Mondatonkénti** modelleknél a sorok végén töröltük az LSTM állapotokat. A batch mérete mindkét modelltípusnál 32 volt. Többféle előre betanított szóbeágyazó mátrixszal [14, 15] próbáltuk modelljeink pontosságát javítani, de legtöbb esetben semekkora, vagy csak marginális mértékű javulást tapasztaltunk.

### 3.3 LSTM n-gram nyelvi modell

A hagyományos LSTM nyelvi modellek egyik hátránya, hogy a „stateful" felépítésből fakadóan a tanítás során nem lehet a tanítómintákat véletlenszerűen keverni, hanem azoknak mindig előre meghatározott sorrendben kell érkezniük. Ezen felül a batch méret növelése sincs jó hatással a pontosságukra, mivel csökken az egy egységként modellezett szöveghossz. A nehézkes tanítás mellett a gyakorlatban az is gondot okoz, hogy az LSTM modellek nagyon sok belső állapotot vehetnek fel ($H \in \mathbb{R}^h$, ahol h az LSTM réteg mérete).

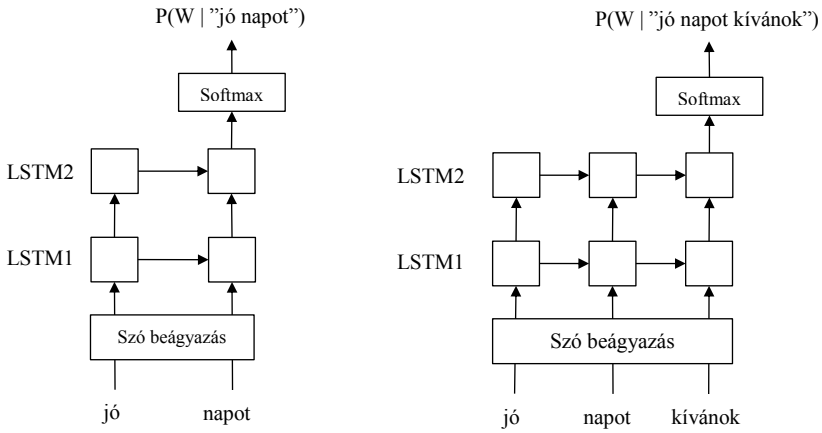
A fenti hátrányok kiküszöbölése érdekében merült fel, hogy érdemes lehet a becsléshez használt mintákat n-gramokba szervezik, és így tanítani rekurrens LSTM nyelvi modelleket [6]. Az ún. **LSTM n-gramok** használatakor tehát a count n-gramokhoz hasonlóan korlátozzuk a becslés során figyelembe vett előtörténet hosszát. Első közelítésben ez egy meglepő döntés, hiszen a rekurrens LSTM hálózatok előnyének tradicionálisan a hosszú távú függőségek jó modellezését tekintik, azonban angol és német nyelveken végzett korábbi vizsgálatok kimutatták, hogy a modellezés pontossága nem feltétlenül csökken drasztikus mértékben [6, 7]. Egy 3- és egy 4-gram rekurrens LSTM nyelvi modell felépítését szemléltetjük a **2. ábrán**.

Az LSTM n-gram implementációnkban a folyamat a hagyományos LSTM nyelvi modellhez hasonlóan a szavak vektorizálásával kezdődik. Ezután egy dropout rétegre kerülnek a szóvektorok, ahonnan az út az első, majd a második rekurrens LSTM rétegre vezet. Az LSTM n-gram modellek esetén nem alkalmaztunk dropout-ot a két LSTM réteg között. Fontos különbség a hagyományos LSTM modellhez képest, hogy az LSTM n-gram-nál csak a teljes előtörténet beolvasása után adunk becslést az azt követő szavak eloszlására.

[1] https://github.com/btarjan/stateful-LSTM-LM

[2] https://www.tensorflow.org/tutorials/sequences/recurrent

Az LSTM n-gramok tanítása során alkalmazott hiperparaméterek és optimalizálás megegyezett a hagyományos LSTM nyelvi modell bemutatása során ismertetett értékekkel. A kivételt két paraméter képezi: a szekvencia hossza, mely természetesen illeszkedik az aktuális n-gram fokszám értékéhez (n-1), másrészt a batch mérete, melynek optimális értékét 512-ben állapítottuk meg. A **mondatösszefűzéses** és **mondatonkénti** modellek itt csak annyiban térnek el, hogy előbbieknél az n-gramok a sorokon átívelhetnek, míg az utóbbinál a sor végével lezáródnak.




2. ábra: Két példa (3- és 4-gram) a kísérleteink során használt rekurrens LSTM n-gram nyelvi modellek felépítésére

# 4 Eredmények

A következőkben a telefonos ügyfélszolgálati beszélgetések korpusza alapján, különböző módszerekkel tanított nyelvi modellek kiértékelését ismertetjük. Az előző fejezetben bemutatott három modelltípust vetjük össze: a klasszikus count n-gram-okat, az új state-of-the-art LSTM modelleket, illetve ez utóbbiak a gyakorlatban jobban használható, egyszerűsített változatát, az LSTM n-gramokat. A modellek kiértékelését a 2.2 fejezetben ismertetett, független teszthalmazon hajtottuk végre.

## 4.1 Nyelvi modellek szöveges kiértékelése

A nyelvi modellek szöveges kiértékelése során a tesztanyaghoz való illeszkedésük mértékét vizsgáltuk **perplexitás** (PPL) segítségével. A 3. fejezetben ismertetett count és LSTM n-gram nyelvi modellek mondatonkénti és mondatösszefűzéses változatainak perplexitását a fokszám növekedésének függvényében vizsgáltuk. A validációs és kiértékelő tesztszövegen kapott eredményeket a **2. táblázatban** foglaltuk össze.

A táblázat alapján az első, ami szembetűnő, hogy a mondatösszefűzéses modellek lényegesen jobban teljesítettek, mint a mondatonkénti modellek. Ez az eredmény azért is lehet elsőre meglepő, mert [6]-ban a PTB korpuszon vizsgálva nem volt szignifikáns különbség a két modellezés között. A fenti cikkel ellentétben azonban itt egy rövid sorokból álló (átlagosan kb. 7 szó soronként, míg PTB kb. 21 szó soronként), spontán beszélgetések leiratai alapján készült korpuszt vizsgáltunk, mely egyben azt is jelenti, hogy nagyobb a sorok közötti összefüggés.

| | **Validációs teszt** | | | | **Kiértékelő teszt** | | | |
|---|---|---|---|---|---|---|---|---|
| | Mondatonkénti | | Mondatösszefűzés | | Mondatonkénti | | Mondatösszefűzés | |
| Fokszám | Count | LSTM | Count | LSTM | Count | LSTM | Count | LSTM |
| 2 | 154,7 | 148,5 | 159,5 | 148,6 | 128,4 | 124,1 | 131,8 | 124,4 |
| 3 | 123,6 | 107,9 | 121,0 | 99,6 | 97,3 | 86,4 | 93,1 | 77,8 |
| 4 | 119,4 | 96,5 | 114,8 | 83,8 | 92,8 | 75,4 | 86,3 | 64,2 |
| 5 | 118,7 | 91,3 | 113,6 | 76,2 | 92,1 | 71,7 | 85,0 | 58,3 |
| 6 | 118,6 | 88,4 | 113,2 | 72,5 | 91,9 | 69,4 | 84,7 | 54,9 |
| 8 | - | 86,7 | - | 69,3 | - | 68,1 | - | 52,4 |
| 10 | - | 86,6 | - | 65,8 | - | 67,3 | - | 49,5 |
| 12 | - | 85,7 | - | 63,9 | - | 67,0 | - | 48,0 |
| 14 | - | 86,0 | - | 62,8 | - | 67,3 | - | 47,1 |
| ∞ | - | **76,9** | - | **60,1** | - | **61,2** | - | **44,6** |

2. táblázat. Nyelvi modellek perplexitása az n-gram fokszám függvényében

Érdekes továbbá megfigyelni, hogy a rekurrens neurális hálózatok perplexitása milyen sokáig mutat csökkenést a fokszám növekedésével, míg a count n-gram modell hamar telítődik. Amíg egyetlen korábbi szó alapján becsüljük a következő valószínűségét (n=2), addig nincs jelentős különbség a count és LSTM modell között. Amint azonban több szót is figyelembe veszünk a becsléshez (n>2), jelentős előnyre tesz szert a rekurrens modell. Ez azzal magyarázható, hogy a neurális modell a szóvektorok és a visszacsatolt felépítés segítségével sokkal pontosabb becslést tud nyújtani a tanítás során meg nem figyelt n-gramokra is.

A táblázatban ∞-nel jelölt sorban találhatóak a hagyományos LSTM eredmények. Mondatösszefűzéses modellezés esetén ez arra utal, hogy minden korábbi szót figyelembe veszünk a valószínűség becsléséhez. Mondatonkénti modellezés esetén a mondat hossza természetesen korlátozza a figyelembe vett szavak számát. Mondatösszefűzéses modellezésnél a fokszám növekedésével egyre közelebb kerülünk a hagyományos LSTM perplexitásához. 10-gram fölött a különbség 10% alá csökken, melyből arra következtethetünk, hogy az LSTM nyelvi modellek legfőbb előnye nem az, hogy nagyon hosszú függőségeket képesek modellezni, hanem hogy jobb általánosító képességekkel bírnak, mint a count modellek, így azoknál lényegesebben robusztusabb becslést szolgáltatnak.

### 4.2 Beszédfelismerési kísérletek

Kísérletsorozatunk végső célja, hogy a neurális nyelvi modellek segítségével pontosabb gépi beszédleiratozás váljék lehetővé magyar nyelven is. Ezért döntöttünk úgy, hogy a szöveges kiértékelésen túl beszédfelismerési kísérleteket is végzünk. A rekurrens neurális nyelvi modellek alkalmazása azonban nem triviális a beszédfelismerésben.

#### 4.2.1 Neurális nyelvmodell mintavételezése

Leggyakrabban úgy hasznosítjuk a neurális nyelvi modelleket, hogy a beszédfelismerés első köre során egy count n-gram nyelvi modellel ún. lattice-t hozunk létre a felismerési hipotézisekből, majd egy második körben újra súlyozzuk a felismerési hipotéziseket a lattice-ben immáron a neurális modell segítségével. Ez a kétkörös futtatás azonban időigényes, így nem támogatja a valós idejű beszédátírást. Cikkünkben ezért egy másik, [8]-ban ismertetett módszert alkalmaztunk. Ennek lényege, hogy a betanított neurális nyelvmodell felhasználásával szöveget generálunk, melyből hagyományos count n-gram modellt tanítunk, amit utána interpolálunk az eredeti tanítószöveg modelljével. A módszer mögött az a logika, hogy ha kellően sok szöveget generálunk a neurális modellel, akkor az a szöveg jól fogja reprezentálni a neurális modell által megtanult szókapcsolati eloszlásokat.

A lehető legjobb eredmény elérése érdekében a szöveggeneráláshoz új, rekurrens LSTM nyelvmodellt tanítottunk, melynek megemeltük a szótárméretét 50000-ről 95000-re. Ennek hatására lényegesen lassabb lett a modell tanítása, de 2,5%-ról 1,9%-ra tudtuk csökkenteni az kiértékelő halmazon mért OOV arányt. Az új LSTM modell segítségével generáltunk egy közel 125 millió szavas tanítókorpuszt. Mivel a count n-gram modellek teljesítménye 4-es fokszám fölött már nem javult érdemben, 4-gram modellt tanítottunk a generált szövegből, melyet utána az eredeti tanítószöveg 4-gram modelljével (KM-2 a 3. táblázatban) interpoláltunk egy a validációs halmazon meghatározott súlyozás alapján. A kapott interpolált, 4-gram count modell nagyon nagy méretűnek adódott (100 millió n-gram), így entrópia-alapú metszés [16] segítségével négy lépésben csökkentettük a méretét (BM-4, BM-3, BM-2, BM-1).

#### 4.2.2 Beszédfelismerési eredmények

A tesztfelvételeket a nyelvi modellek és a VoXerver nevű, WFST-alapú beszédfelismerő dekóder [17] segítségével szöveggé alakítottuk. A dekódolás során HMM-DNN hibrid megközelítésben egy három rejtett rétegű, rétegenként 2500 neuront tartalmazó, 4907 kimeneti állapottal rendelkező, előrecsatolt, mély neuronhálót alkalmaztunk. Összesen 290 órányi 8 kHz mintavételi frekvenciájú telefonos beszélgetésen tanítottuk az akusztikus modellt a KALDI toolkit [18] segítségével. Az akusztikus jellemzővektorok 13 dimenziós MFCC paraméterekre épültek, melyet LDA és MLLT lineáris transzformáció követett. Osztott, három állapotú, környezetfüggő beszédhangmodelleket használtunk. A kiértékelő teszthalmazon mért szóhiba-arányokat az **3. táblázatban** ismertetjük.

A 3. táblázatban kezdeti modellként (KM) az eredeti tanítószöveg alapján tanított 3-gram (KM-1) és 4-gram (KM-2) modellre hivatkozunk. Látható, hogy hiába növeljük a modell fokszámát 3-ról 4-re, a szóhiba-arány csak minimális mértékben csök-

ken, míg a modell mérete drasztikusan megnő. Ez a tipikus esete annak, hogy hiába tanul a modell rengeteg hosszabb n-gramot a tanítószövegből, azoknak csak kis százaléka hasznosul a tesztelés során (alacsony hit rate).

Ezzel szemben a rekurrens LSTM nyelvi modell alapján tanult n-gramok sokkal jobban hasznosíthatóak. A BM-1-es modell például méretében nagyjából megegyezik a KM-1 modellel mégis relatív 2%-kal jobb szóhiba-aránnyal rendelkezik. Ha nagyobb modellméretet is megengedünk, tovább tudjuk csökkenteni a hibát. A KM-2-vel nagyjából megegyező méretű BM-3 modell relatív 3%-kal csökkenti a szóhiba-arányt míg, ha 3GB-os memóriafoglalás is megengedett, akkor összesen 4%-os relatív szóhiba-arány csökkenést mérhetünk.

A fenti hibaarány csökkenések bizakodásra adnak okot, de természetesen messze nem tekinthetjük a problémát megoldottnak. A generált szöveg alapján történő count n-gram mintavételezéssel 90 körüli értékre sikerül csökkentenünk a nyelvi modell perplexitását (BM-4). A generáláshoz használt eredeti rekurrens LSTM nyelvi modellel azonban 56-os perplexitást mértünk a kiértékelő tesztanyagon, így látható, hogy maradt még bőven lehetőség a beszédfelismerő nyelvi modelljét javítani.

| **Modell** | **n-gramok száma [millió]** | **Modell mérete [GB]** | **PPL [-]** | **Szóhiba-arány [%]** | | | |
|---|---|---|---|---|---|---|---|
| | | | Σ | MTUBA sztereo 1 | MTUBA sztereo 2 | MTUBA mono | Σ |
| KM-1 | 1,2 | 0,2 | 110,2 | 10,7 | 33,0 | 32,8 | **29,3** |
| KM-2 | 5,0 | 1,3 | 103,0 | 10,5 | 33,3 | 32,7 | **29,2** |
| BM-1 | 1,2 | 0,3 | 100,8 | 10,5 | 32,4 | 32,2 | **28,7** |
| BM-2 | 4,2 | 0,9 | 93,0 | 10,3 | 31,6 | 31,7 | **28,3** |
| BM-3 | 8,8 | 1,6 | 91,4 | 10,1 | 31,7 | 31,7 | **28,3** |
| BM-4 | 18,5 | 3,1 | 90,5 | 10,0 | 31,8 | 31,5 | **28,1** |

3. táblázat. Beszédfelismerési eredmények a kiértékelő teszthalmazon
(KM: kezdeti modell, BM: bővített modell)

## 5 Összefoglalás

Cikkünk egy kísérletsorozat első állomása, melyben a neurális nyelvi modellek alkalmazását vizsgáljuk beszédfelismerő rendszerben. Kíséreteinkben egy telefonos ügyfélszolgálati beszélgetéseket tartalmazó adatbázison tanítottunk hagyományos count n-gram és rekurrens LSTM neurális nyelvi modelleket. Az LSTM nyelvi modell segítségével **közel felére tudtuk csökkenteni a kiértékelő szövegen a perplexitást**. Az LSTM nyelvi modellnek egy gyakorlatban jobban alkalmazható változata az ún. LSTM n-gram, ahol n-gramok alapján tanítjuk a rekurrens LSTM modellt, így egyben korlátozzuk a becsléshez felhasználható korábbi szavak számát.

Az LSTM n-gramok minden fokszám mellett jobbnak bizonyultak, mint a count n-gram modellek. Sőt a korlátozás nélküli LSTM modell teljesítményét is megközelítik aránylag kis fokszám mellett. Mindez arra utal, hogy a rekurrens LSTM nyelvi modellek elsősorban **a fejlettebb simítás és nem a hosszú távú memóriájuk miatt pontosabbak**, mint a count n-gram modellek.

Az LSTM nyelvi modellek szöveges kiértékelésén túl arra is kísérletet tettünk, hogy a beszédfelismerés minőségét is javítsuk vele. Erre jelenlegi cikkünkben egy egyszerű módszer vetettünk be: nagy mennyiségű szöveget generáltunk a neurális modellel, majd az ebből tanított count n-gram modellt adaptáltuk az eredeti n-gram modellel. Az eredmények azt igazolják, hogy a generált szöveg hasznos n-gramokat tartalmaz, mivel belőle épített count modellel relatív **4%-kal sikerült csökkentenünk a kiértékelő teszt szóhiba-arányát**. A 4%-os hibacsökkenést eredményező bővített count modell perplexitása 90 volt, ami ha figyelembe vesszük, hogy a kezdeti modell perplexitása 103, míg a generálást végző LSTM modell perplexitása 56 volt, azt jelenti, hogy a potenciális perplexitás javulásnak a 28%-át sikerült egyelőre a beszédfelismerő rendszerben is hasznosítani.

A jövőbeli terveink között szerepel a kísérleteink kiterjesztése más, nagyobb méretű adatbázisokra. Ezen kívül újabb nyelvi modellezési technikákat és céljainak jobban megfelelő szóbeágyazási modelleket is ki kívánunk próbálni. Nyelvünk gazdag morfológiáját közvetlenül nem modellezik a jelenlegi módszerek, melyen a jövőben szintén változtatni szeretnénk.

## Köszönetnyilvánítás

Kutatásunk az EUREKA_15-1-2016-0019 azonosító számú DANSPLAT projekt támogatásával készült.

## Bibliográfia

1. Jelinek, F., Mercer, R.I.: Interpolated estimation of Markov source parameters from sparse data. In: Pattern recognition in practice. Proc. workshop Amsterdam, May 1980. p. 381–397,401 (1980).
2. Arisoy, E., Chen, S.F., Ramabhadran, B., Sethy, A.: Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition. IEEE Trans. Audio, Speech Lang. Process. 22, 184–192 (2014).
3. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association. pp. 1045–1048 (2010).
4. Hochreiter, S., Schmidhuber, J.J.: Long short-term memory. Neural Comput. 9, 1–32 (1997).
5. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association (2012).
6. Chelba, C., Norouzi, M., Bengio, S.: N-gram Language Modeling using Recurrent Neural Network Estimation. CoRR. 1703.10724, (2017).

7. Tüske, Z., Schlüter, R., Ney, H.: Investigation on LSTM Recurrent N-gram Language Models for Speech Recognition. In: Interspeech 2018. pp. 3358–3362. ISCA, ISCA (2018).
8. Deoras, A., Mikolov, T., Kombrink, S., Karafiát, M., Khudanpur, S.: Variational approximation of long-span language models for LVCSR. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 5532–5535 (2011).
9. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Comput. Speech Lang. 13, 359–393 (1999).
10. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings International Conference on Spoken Language Processing. pp. 901–904. , Denver, US (2002).
11. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent Neural Network Regularization. CoRR. 1409.2329, (2014).
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 15, 1929–1958 (2014).
13. Chollet, F., others: Keras, (2015).
14. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist. 5, 135–146 (2017).
15. Makrai, M.: Filtering Wiktionary Triangles by Linear Mapping between Distributed Word Models. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). pp. 2766–2770. European Language esources Association (ELRA), Portorož, Slovenia (2016).
16. Stolcke, A.: Entropy-based pruning of backoff language models. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. pp. 270–274 (2000).
17. Tarján, B., Mihajlik, P., Balog, A., Fegyó, T.: Evaluation of lexical models for Hungarian Broadcast speech transcription and spoken term detection. In: 2nd International Conference on Cognitive Infocommunications (CogInfoCom). pp. 1–5. , Budapest, Hungary (2011).
18. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: IEEE Workshop on Automatic Speech Recognition and Understanding. pp. 1–4 (2011).

# SZEMANTIKA

# CBOW/A: módosított CBOW algoritmus annotált szövegekből készített vektortérmodellek létrehozására

Novák Attila, Laki László János, Novák Borbála

Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Kar
MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
Budapest, Práter u. 50/a.
{vezetéknév.keresztnév}@itk.ppke.hu

**Kivonat** Cikkünkben a szóbeágyazási modellek készítésére alkalmas fastText könyvtár CBOW algoritmusának egy olyan módosított változatát mutatjuk be, amellyel a felszíni szóalakok és az azokhoz tartozó annotációk reprezentációját egyszerre tartalmazó vektortérmodell hozható létre. Bemutatunk egy konkrét modellt is, amelyet morfológiai és szintaktikai függőségi annotációt tartalmazó angol nyelvű korpuszon tanítottunk be, és amely alkalmas olyan lekérdezések hatékony megválaszolására, mint hogy *mit eszünk, mit csinálunk egy csontvázzal, mit csinálunk még azzal, amit eszünk,* stb.

## 1. Bevezetés

A szakirodalomból ismert, hogy számos alkalmazásban hasznos lehet grammatikai annotációt tartalmazó korpuszból épített szóbeágyazási modelleket használni, mert ezek bizonyos feladatokban jobban teljesítenek, mint az annotálatlan felszíni szóalakokból épített beágyazási modellek [1,2]. Ugyanakkor a legtöbb gyakorlati nyelvtechnológiai feladatban szükség van a felszíni szóalakok vektor reprezentációjára. Ebben a cikkben egy olyan vektortérmodellt mutatunk be, amely egyszerre tartalmazza a felszíni szóalakok, a lemmák és a szavak közötti grammatikai viszonyok reprezentációját, amelyben tehát triviális módon értelmezhető az ilyen különböző típusú objektumok közötti távolság, és így használható olyan jellegű kérdések megválaszolására, hogy tipikusan milyen elemek állnak egymással adott típusú kapcsolatban. Például, az *eat* ige tárgyaként ételek listáját várjuk eredményül. A cikkben bemutatott modellt egy angol nyelvű korpuszból hoztuk létre, de az alkalmazott módszer nyelvfüggetlen.

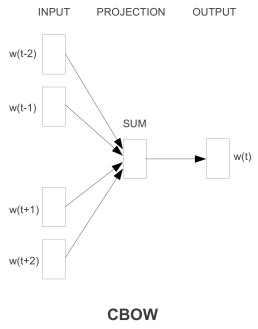
## 2. Folytonos disztribúciós szemantikai modellek

A disztribúciós szemantika lényege, hogy a szavak jelentése szorosan összefügg azzal, hogy milyen kontextusban használjuk őket. A hagyományos disztribúciós

szemantikai modellek létrehozásakor az egyes szavak előre meghatározott méretű környezetét az azokban előforduló szavak nagy korpuszból számított előfordulási statisztikái alapján határozzuk meg.

Ezzel szemben a nyelvtechnológiai kutatások egyik kurrens módszere a folytonos vektoros reprezentációk alkalmazása (*word embedding*), melyek nyers szöveges korpuszból szemantikai információk kinyerésére alkalmazhatók. Ebben a rendszerben a lexikai elemek egy valós vektortér egyes pontjai, melyek konzisztensen helyezkednek el az adott térben, azaz, az egymáshoz szemantikailag és/vagy morfológiailag hasonló szavak egymáshoz közel, a jelentésben eltérő elemek egymástól távol esnek. Mindemellett, a vektoralgebrai műveletek is alkalmazhatók ebben a térben, tehát két elem szemantikai hasonlósága a két vektor távolságaként meghatározható, illetve a lexikai elemek pozícióját reprezentáló vektorok összege, azok jelentésbeli összegét határozzák meg [3,4].

Ennek a modellnek a tanítása során is az egyes szavak fix méretű környezetét vesszük figyelembe, az ezekből álló vektor azonban egy neurális hálózat bemenete. A környezetet reprezentáló vektorok együttesét használja a hálózat arra, hogy megjósolja a célszót. A tanítás során a hiba visszaterjesztésével és ennek megfelelően a környezetet reprezentáló vektorok frissítésével jön létre a célszót helyesen megjósoló súlyvektor, ami a neurális hálózat megfelelő rétegéből közvetlenül kinyerhető. Mivel a hasonló szavak hasonló környezetben fordulnak elő, ezért a szövegkörnyezetre optimalizált vektorok a hasonló jelentésű szavak esetén hasonlóak lesznek. Az erre a feladatra felépített neurális hálózat a CBOW (*continous bag-of-words*) modellt implementálja, ami az 1. ábrán látható és az egyik legnépszerűbb implementációja a word2vec[1]. Egy másik lehetőség az ú.n. skip-gram modell alkalmazása, amikor a hálózat bemenete a célszó, az optimalizálás célja pedig e szó környezetének megjósolása.




1. ábra. A CBOW (*continous bag-of-words*) modell

[1] `https://code.google.com/archive/p/word2vec/`

A fastText algoritmus [5] a word2vec implementációját elsősorban azzal egészítette ki, hogy a szavak mellett az azokat alkotó karakter n-gramok reprezentációit is létrehozza, illetve a szó környezetében szereplő szavak karakter n-gramjait is szövegkörnyezetnek tekinti.

Ebben a cikkben a fastText algoritmus CBOW modelljének egy olyan módosított változatát mutatjuk be, amely egy modellen belül egyszerre hozza létre egy elemzett korpusz alapján a felszíni szóalakok és a hozzájuk rendelt akár több különböző típusú annotáció vektorreprezentációját.

## 3. A korpusz előkészítése

A jelen cikkben bemutatott kísérletek kiinduló anyagául a 2,25 milliárd token méretű angol Wikipedia korpusz[2] szolgált. A korpuszt a SpaCy keretrendszerbe[3] integrált angol neurális taggerrel és függőségi elemzővel elemeztük, amely lemmát, szófajcímkét és a szavak közötti függőségi viszonyokat rendelt az egyes szóalakokhoz. A SpaCy elemzéseit a feldolgozás első lépésében a CONNL-U formátum módosított változatában íratjuk ki (l.2. ábra). Majd további feldolgozás után egy olyan reprezentáció születik, amelyben a felszíni szóalakot annotációs címkék sorozata követi, melyek közül az első a lemma és a szófaj, és ezt az igevonzatok és a szabad határozók esetében az igei fej és az adott összetevőt a fejhez kapcsoló reláció címkéje követi. Az utóbbi típusú címkéből több is lehet, ha az adott szó több predikátummal is vonzatviszonyban áll (l.3. ábra).

A függőségi fa szerkezetű alapelemzéseket kiterjesztett függőségi reprezentációvá alakítjuk át. Az átalakítás során számos transzformációt végzünk, illetve számos új függőségi viszonyt veszünk fel. Azonos reprezentációt kap például egy adott aktív igealak tárgya, ugyanazon ige passzív változatának alanya, illetve egy befejezett melléknévi igenév vagy egy passzív vonatkozó mellékmondat által módosított főnév. A tagmondatok fejéhez kapcsolódó tartalmas szavak így olyan annotációt is kapnak a lemmájuk és a szófajcímkéjük mellett, amely explicit módon tartalmazza, hogy milyen igékhez milyen függőségi viszony kapcsolja őket. Ebben az annotációban az ú.n. *phrasal verb*-ök, a prepozíciós vonzatok és a kopulás szerkezetek összevontan tartalmazzák az igét és a prepozíciót, illetve a kopulát és a névszói állítmányt.

A 2. ábrán a *Bryozoa* egyrészt a *phylum* névszói állítmány alanya, másrészt egy olyan jelzői mellékmondat módosítja, amelynek feje a *know* 'ismer' ige. A *know* ige *as* prepozíciós vonzata pedig a *Polyzoa*, illetve az *animals* 'állatok'. A 3. ábrán látható, hogy a feldolgozás második lépése után már az eredetileg a jelzői mellékmondat által módosított *Bryozoa* a *know* ige tárgyaként szerepel. Bár a feldolgozás során a koordinált elemekre átterjesztjük a koordináció „fejének" vonzatviszonyait, egy elemzési hiba folytán az *Ectoprocta* a példában szereplő annotációban nem lett a *know* prepozíciós vonzata. Ugyan ebben a mondatban ez csak hibás elemzés eredménye, de egyébként feltehetőleg érdemes

[2] A `https://dumps.wikimedia.org/` linkről letölthető 2016. májusi verzió
[3] `https://spacy.io/`

lenne az appozitív szerkezetekre is elvégezni a koordinációra alkalmazott műveletet. Ugyancsak érdemes lenne a `compound` viszony mentén az angol összetett szavakat is egy elemmé összevonni.

```
#The Bryozoa, also known as the Polyzoa, Ectoprocta or commonly as moss animals, are a phylum
of aquatic invertebrate animals.
0       The     the     DET     DT      det     1       bryozoa PROPN
1       Bryozoa bryozoa PROPN   NNP     nsubj   16      be      VERB    _know#VB@<acl
                                                                        _be_phylum#VB@nsubj
2       ,       ,       PUNCT   ,       punct   1       bryozoa PROPN
3       also    also    ADV     RB      advmod  4       know    VERB    _know#VB@advmod
4       known   know    VERB    VBN     acl     1       bryozoa PROPN
5       as      as      ADP     IN      prep    4       know    VERB    _know#VB@prep
6       the     the     DET     DT      det     7       polyzoa PROPN
7       Polyzoa polyzoa PROPN   NNP     pobj    5       as      ADP     _know#VB@prep_as@pobj
8       ,       ,       PUNCT   ,       punct   7       polyzoa PROPN
9       Ectoprocta      ectoprocta      PROPN   NNP     appos   7       polyzoa PROPN
10      or      or      CCONJ   CC      cc      9       ectoprocta      PROPN
11      commonly        commonly        ADV     RB      advmod  12      as      ADP
                                                                        _know#VB@prep_as@advmod
12      as      as      ADP     IN      prep    4       know    VERB    _know#VB@prep
13      moss    moss    NOUN    NN      compound        14      animal  NOUN
14      animals animal  NOUN    NNS     pobj    12      as      ADP     _know#VB@prep_as@pobj
15      ,       ,       PUNCT   ,       punct   1       bryozoa PROPN
16      are     be      VERB    VBP     ROOT    16      be      VERB
17      a       a       DET     DT      det     18      phylum  NOUN
18      phylum  phylum  NOUN    NN      attr    16      be      VERB    _be_phylum#VB@attr
19      of      of      ADP     IN      prep    18      phylum  NOUN
20      aquatic aquatic ADJ     JJ      amod    22      animal  NOUN
21      invertebrate    invertebrate    ADJ     JJ      amod    22      animal  NOUN
22      animals animal  NOUN    NNS     pobj    19      of      ADP     _of@pobj
23      .       .       PUNCT   .       punct   16      be      VERB
```

2. ábra. A felhasznált korpusz egy mondatának annotációja a kiegészített CONLL-U formátumban a feldolgozás első lépése után

## 4. A módosított CBOW algoritmus

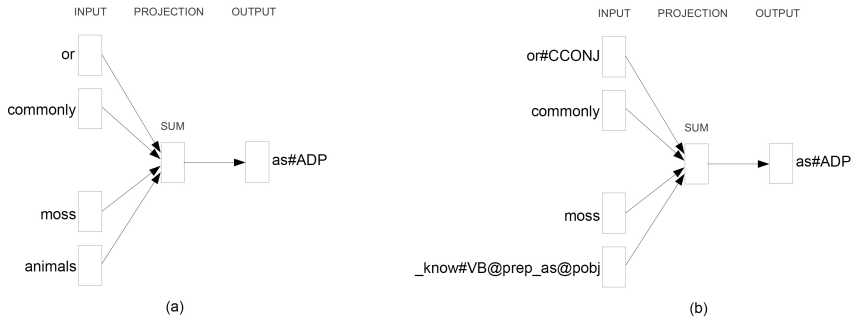
A szóbeágyazási modell építéséhez használt annotált korpuszban a szó (illetve írásjel) típusú tokeneket tetszőleges számú speciális, a 3. ábrán szereplő példában ■ jellel kezdődő címke típusú token követi. A fastText könyvtár CBOW algoritmusát úgy módosítottuk, hogy az ismertetett formájú bemeneti korpuszból olyan modellt építsen, amelyben a felszíni szóalakok és a hozzájuk tartozó annotációs címkék egyszerre vannak reprezentálva.

Az algoritmus első változatában a modell építése során csak a felszíni szóalakokat használtuk a betanítandó neurális hálózat bemeneteként megjelenő szövegkörnyezetként. Célszóként azonban a felszíni alakok és a hozzájuk tartozó címkék is megjelentek (4.(a) ábra). Ez a konfiguráció azonban olyan modellt hozott létre, amely a legcsekélyebb mértékben sem hasonlított ahhoz, amit kapni szerettünk volna. A címkék modell által generált vektorreprezentációja nemhogy hasonlított volna az adott címkével annotált szó reprezentációjához, hanem éppen ellenkezőleg, a lehető legnagyobb mértékben különbözött tőle (gyakorlatilag

```
The ■the#DET Bryozoa ■bryozoa#PROPN ■_know#VB@dobj ■_be_phylum#VB@nsubj
, ■,#PUNCT also ■also#ADV known ■know#VERB as ■as#ADP the ■the#DET
Polyzoa ■polyzoa#PROPN ■_know#VB@prep_as@pobj , ■,#PUNCT Ectoprocta
■ectoprocta#PROPN or ■or#CCONJ commonly ■commonly#ADV as ■as#ADP moss
■moss#NOUN animals ■animal#NOUN ■_know#VB@prep_as@pobj , ■,#PUNCT are
■be#VERB a ■a#DET phylum ■phylum#NOUN of ■of#ADP aquatic ■aquatic#ADJ
invertebrate ■invertebrate#ADJ animals ■animal#NOUN . ■.#PUNCT
```

3. ábra. A felhasznált korpusz egy mondatának annotációja a feldolgozás második lépése után

merőleges volt rá). Ennek az volt az oka, hogy a negative sampling algoritmus kizárólag negatív példaként látta bármely szó szövegkörnyezetében a címkéket és ezért a hálózat minden címke reprezentációját igyekezett a lehető legmesszebb juttatni a pozitív szövegkörnyezetként is előforduló felszíni szóalakok reprezentációjától (minden címke minden szótól a lehető legtávolabb helyezkedett el). Hogy valóban ez történik, azt úgy tettük egyértelművé, hogy egy olyan korpuszon tanítottuk be a modellt, amelyben minden szóalaknak pontosan egy címkéje volt, amely azonos volt magával a szóalakkal. Az így betanított modellben gyakorlatilag minden szó és a hozzá tartozó címke koszinusz távolsága (hasonlósága) lényegében nulla volt.




4. ábra. A módosított CBOW modell architektúrák

Ezt az anomáliát úgy küszöböltük ki, hogy a szövegkörnyezetben egyenletes eloszlással mintavételeztük a felszíni alakokat és címkéiket (4.(b) ábra). Így a címkék és a felszíni alakok egyaránt megjelentek pozitív és negatív tanítópéldaként, és így a kapott modell már sokkal inkább hasonlított ahhoz, amit vártunk.

A modell tanítása során 300 dimenziós vektorokat építettünk, és nem használtuk a fastText karakter-n-gram alapú modelljét (a -minn 0 -maxn 0 kapcsolókat használtuk). Egyénként az alapbeállításokkal futtattuk a tanítást: 5 token suga-

rú ablak, min. 5 előfordulás a szavakra és a címkékre, negatív mintavételezés 5 példával, stb.

## 5. Mire jó ez a modell

A modell egyszerre tárolja rendkívül kompakt formában a szavak felszíni alakjára, azok lemmájára és szófajára, illetve a közöttük tipikusan fennálló függőségi viszonyokra jellemző reprezentációkat. Az, hogy egyetlen modellen belül jelennek meg ezek az információk, lehetőséget ad arra, hogy a modellnek olyan kérdéseket tegyünk fel, hogy például *Mit isznak?, Mit bányásznak?, Miben hiszünk? Ki eszik? Mit csinálunk egy csontvázzal?*, stb. Annak kiértékelésére, hogy a modell az ilyen jellegű kérdésekre mennyire jó választ ad, sajnos nem állt rendelkezésünkre megfelelő gold standard erőforrás. A modellt jellemző átfogó kvantitatív kiértékelés helyett ezért kénytelenek vagyunk egy viszonylag szűk lexikai elemkészletre vonatkozó lekérdezéslista eredményeként kapott válaszok kézi kiértékelésére, illetve azokra a megfigyelésekre szorítkozni, amelyeket a [6] cikkben leírt szóbeágyazási modellek vizualizációjára szolgáló felületen keresztül a modellel kapcsolatban tettünk.

A felület képes arra, hogy a lekérdezésként megadott szóhoz megjelenítse a vektortérben hozzá legközelebb álló elemeket azok koszinuszhasonlóságával és gyakoriságával együtt. Lehetőség van arra, hogy szűrőket definiáljunk az így megjelenített legközelebbi szomszédok alakjára vonatkozóan. Ez ad lehetőséget például arra, hogy a *Mit isznak?* jellegű kérdésekre a rendszer által adott választ megkaphassuk. Ehhez egy olyan lekérdezést fogalmazunk meg, amelyben a „*drink* ‘iszik’ ige tárgya” objektum legközelebbi szomszédait keressük azzal a feltétellel, hogy a modellben szereplő elemeket megszűrjük, és csak a `NOUN` szófajcímkét tartalmazóakat tartjuk meg. Egy ilyen lekérdezés eredménye látható az 5. ábrán.

Ha címkét nem tartalmazó elemhez (felszíni szóalakhoz) indítunk lekérdezést, akkor a rendszer automatikusan olyan lekérdezésekkel egészíti ki az eredeti lekérdezést, amelyben az adott szót lemmának feltételezve hozzáfűzi ahhoz a korpuszban az adott lemmával előforduló szófajcímkéket. Így például a *can* lekérdezéshez megkapjuk válaszként egyrészt a *can* szóalak, másrészt a *can* ’tud, képes’ segédige, harmadrészt a *can* ‘konzerv’ főnév mint lemma, illetve az annotációhoz használt SpaCy tagger által hibásan más szófajúként címkézett elemek reprezentációjához legközelebbi elemeket (6. ábra). Valamely lemmával indítva a lekérdezést, a válaszban általában az első találatok között megkapjuk a szó ragozott alakjait, ragozott alakhoz pedig valahol az első találatok között lesz a szó lemmája.

### 5.1. Az elemzőrendszer hibái

A lekérdezőrendszer által a modellből visszaadott válaszok viszonylag koncentrált módon elénk tárják, hogy a korpusz annotációjához használt elemzőrendszer milyen változatos jellegű hibákat vezet be az annotációba. Ez már azon a szinten is megjelenik, hogy a generált kimenetben látunk olyan szófajcímkéket, illetve

| 0 | _drink#VB@dobj | 1 | 18814 |
|---|---|---|---|
| 1 | drink#NOUN | 0.7048 | 36413 |
| 2 | drinking#NOUN | 0.6033 | 27063 |
| 3 | juice#NOUN | 0.5734 | 14046 |
| 4 | beer#NOUN | 0.5699 | 41032 |
| 5 | bottle#NOUN | 0.5634 | 32186 |
| 6 | drinker#NOUN | 0.5593 | 3204 |
| 7 | brandy#NOUN | 0.5588 | 2957 |
| 8 | champagne#NOUN | 0.5379 | 3691 |
| 9 | alcohol#NOUN | 0.5374 | 54060 |
| 10 | pint#NOUN | 0.5339 | 2581 |
| 11 | vodka#NOUN | 0.5320 | 3045 |

5. ábra. A „*drink* ige tárgya” objektum legközelebbi szomszédai

| 0 | can | 1 | 2176270 |
|---|---|---|---|
| 1 | can | 1.00000001471 | 2176270 |
| 2 | can#VERB | 0.9967 | 2314044 |
| 3 | may | 0.8407 | 1367560 |
| 4 | may#VERB | 0.8396 | 1624916 |
| 5 | could#VERB | 0.8212 | 963212 |
| 6 | could | 0.8167 | 943772 |
| 7 | must | 0.8164 | 389083 |
| 8 | must#VERB | 0.8126 | 395927 |
| 9 | will | 0.7895 | 1220730 |
| 10 | will#VERB | 0.7882 | 1303870 |
| 11 | should#VERB | 0.7329 | 594608 |

| 0 | can#VERB | 1 | 2314044 |
|---|---|---|---|
| 1 | can#VERB | 1.00000009725 | 2314044 |
| 2 | can | 0.9967 | 2176270 |
| 3 | may#VERB | 0.8300 | 1624916 |
| 4 | may | 0.8295 | 1367560 |
| 5 | could#VERB | 0.8230 | 963212 |
| 6 | could | 0.8172 | 943772 |
| 7 | must | 0.8057 | 389083 |
| 8 | must#VERB | 0.8041 | 395927 |
| 9 | will#VERB | 0.7876 | 1303870 |
| 10 | will | 0.7840 | 1220730 |
| 11 | should#VERB | 0.7292 | 594608 |

| 0 | can#NOUN | 1 | 9460 |
|---|---|---|---|
| 1 | can#NOUN | 1.00000004784 | 9460 |
| 2 | cans | 0.9386 | 5678 |
| 3 | bottle#NOUN | 0.7667 | 32186 |
| 4 | bottles | 0.7545 | 12914 |
| 5 | bag#NOUN | 0.6997 | 33140 |
| 6 | bags | 0.6971 | 12483 |
| 7 | bottle | 0.6969 | 19000 |
| 8 | tins | 0.6909 | 783 |
| 9 | carton#NOUN | 0.6866 | 1506 |
| 10 | bag | 0.6566 | 19570 |
| 11 | containers | 0.6511 | 10963 |

| 0 | can#PROPN | 1 | 1980 |
|---|---|---|---|
| 1 | can#PROPN | 1.0000001992 | 1980 |
| 2 | CAN | 0.6573 | 2764 |
| 3 | Can | 0.4954 | 54105 |
| 4 | Llauder | 0.4206 | 6 |
| 5 | tap#PROPN | 0.3849 | 1660 |
| 6 | SPAM | 0.3815 | 416 |
| 7 | be#PROPN | 0.3780 | 1724 |
| 8 | jerrycan#PROPN | 0.3636 | 11 |
| 9 | llauder#PROPN | 0.3602 | 6 |
| 10 | pan#PROPN | 0.3542 | 40251 |
| 11 | Pan | 0.3481 | 38327 |

6. ábra. A *can* különböző előfordulásainak legközelebbi szomszédai

olyan függőségi relációkat a kérdésként megadott szóhoz mint lemmához rendelve, amelyről tudjuk, hogy hibás.

### 5.2. Szemantikailag jól behatárolható vonzatú igék – mit eszünk

Az elemzőrendszer által bevezetett hibák ellenére a modell válaszainak túlnyomó része elég meggyőző, különösen azokban az esetekben, ahol például az adott ige adott vonzatviszonyában szemantikailag jól behatárolható körbe tartozó lexikai elemek jelennek meg. A 7. ábrán az „*eat* ‘eszik’ ige tárgya” viszonylatában például valóban azt látjuk, hogy a listában túlnyomórészt ételek jelennek meg, köztük viszonylag előkelő helyen számos olyan különleges étel is, amely csak néhány alkalommal fordul elő a Wikipedia-korpuszban, ugyanakkor minden esetben mint az étkezés tárgya. Ezek az elemek így a modellben szorosabban kapcsolódnak

az evéshez, mint sok számunkra talán prototipikusabbnak tűnő étel, amelyeknek azonban számos egyéb aspektusával például az elkészítésük vagy feldolgozásuk módjával kapcsolatban rengeteg információ fordul elő a korpuszban, és így a reprezentációjuk távolabb esik az evés tárgya objektum vektorreprezentációjától. Az *eat* alanyára vonatkozóan nem jön létre a modellben ennyire jól körülhatárolható reprezentáció. A főleg enciklopédikus ismereteket tartalmazó korpuszban eleve nagyságrendekkel ritkábban jelenik meg az evés alanya testes lexikai elemmel kitöltve. Ugyanakkor a *leak* 'szivárog' ige lehetséges alanyai jobban körülhatárolható jelentésű csoportokba tagolódnak. Mint „a *leak* ige alanya" objektum 100 legközelebbi főnévi szomszédját klaszterezve ábrázoló 8. ábrán látható, a modell erre jól vissza is adja, hogy folyadékok, gázok, azok szállítására, tárolására, az áramlás és a nyomás szabályozására stb. szolgáló eszközök és konténerek, valamint információ (titkok, feljegyzések stb.) szokott (ki)szivárogni.

| 0 | _eat#VB@dobj | 1 | 78427 |
|---|---|---|---|
| 1 | meat#NOUN | 0.5810 | 50211 |
| 2 | meal#NOUN | 0.5749 | 33003 |
| 3 | eating#NOUN | 0.5535 | 3159 |
| 4 | food#NOUN | 0.5519 | 254592 |
| 5 | manjuu#NOUN | 0.5519 | 5 |
| 6 | flesh#NOUN | 0.5492 | 15264 |
| 7 | carrot#NOUN | 0.5461 | 4247 |
| 8 | gebrocht#NOUN | 0.5435 | 21 |
| 9 | diet#NOUN | 0.5392 | 35798 |
| 10 | φαγεινν#NOUN | 0.5374 | 6 |
| 11 | taiyaki#NOUN | 0.5181 | 28 |

7. ábra. Az „*eat* ige tárgya" objektum legközelebbi főnévként elemzett szomszédai

### 5.3. Más igék hasonló vonzatai – mit csinálunk még azzal, amit eszük

Ugyan arra a kérdésre, hogy *Mit eszünk?*, a választ megkaphatnánk pusztán az elemzett korpuszban az *eat* ige tárgyaként megjelölt lexikai elemek lekérdezésével is, a modell természetes módon lehetőséget ad az olyan kérdések megfogalmazására és megválaszolására is, hogy például *Milyen igék tárgya, vagy milyen igék valamilyen prepozíciós tárgya szokott olyasmi lenni, mint az eat 'eszik' ige tárgya?*. (9. ábra) Ha az így kapott igék listáját összevetjük a pusztán az *eat* igéhez közeli igék listájával, akkor jól látható, hogy az első kérdésre válaszként kapott

| | |
|---|---|
| 0 | dilbit#NOUN dispersant#NOUN downflow#NOUN overfilling#NOUN csf#NOUN permeate#NOUN filtrate#NOUN fluid#NOUN liquid#NOUN leachate#NOUN slurry#NOUN seawater#NOUN coolant#NOUN |
| 1 | turbopump#NOUN oxidizer#NOUN oxidiser#NOUN antifreeze#NOUN pressurant#NOUN freon#NOUN |
| 2 | naphtha#NOUN flammable#NOUN combustible#NOUN gallon#NOUN avgas#NOUN gas#NOUN fuel#NOUN |
| 3 | methane#NOUN ammonia#NOUN fume#NOUN vapor#NOUN vapour#NOUN |
| 4 | protodermis#NOUN ampule#NOUN tetrafluoroethylene#NOUN styrol#NOUN hexafluoride#NOUN sulphide#NOUN trichloroethene#NOUN |
| 5 | caulking#NOUN drywall#NOUN penetrant#NOUN gasket#NOUN sealant#NOUN lubricant#NOUN |
| 6 | envelope#NOUN duct#NOUN pump#NOUN injector#NOUN stokehold#NOUN compartment#NOUN bilge#NOUN boiler#NOUN feedwater#NOUN pipework#NOUN pipe#NOUN |
| 7 | bubbler#NOUN diverter#NOUN deaerator#NOUN aspirator#NOUN thermosiphon#NOUN scrubber#NOUN dehumidifier#NOUN |
| 8 | manhole#NOUN downpipe#NOUN standpipe#NOUN |
| 9 | venting#NOUN pressurisation#NOUN umbilical#NOUN ductwork#NOUN ducting#NOUN |
| 10 | ballonet#NOUN gasbag#NOUN arcing#NOUN corium#NOUN preventer#NOUN calandria#NOUN drywell#NOUN pressurizer#NOUN |
| 11 | anthrax#NOUN sarin#NOUN secret#NOUN memo#NOUN leaker#NOUN bugging#NOUN wikileak#NOUN |
| 12 | slick#NOUN _leak#VB@nsubj leaking#NOUN leak#NOUN seepage#NOUN leakage#NOUN spill#NOUN spillage#NOUN |
| 13 | malfunction#NOUN overheating#NOUN shutoff#NOUN scram#NOUN firedamp#NOUN blackdamp#NOUN rupture#NOUN explosion#NOUN bursting#NOUN |

8. ábra. A „*leak* ige alanya” objektum 100 legközelebbi főnévi szomszédja klaszterezve

listáról hiányoznak az *eat* ige lexikai reprezentációjához egyébként közeli, az étkezéstől különböző testi szükségletekre és élvezetekre vonatkozó igék, ugyanakkor megjelennek a pusztítással kapcsolatos igék, amely jól szemlélteti, hogy amit megeszünk, azt elpusztítjuk. A prepozíciós vonzatokra vonatkozó kérdés pedig egészen új evés- és fogyasztásigéket hoz be a listára, amelyek a szintaktikai disztribúció különbözősége miatt nem jelentek meg az előbbi halmazokban.

| 0 | _eat#VB@dobj | 1 | 78427 |
|---|---|---|---|
| 1 | _eat#VB@dobj | 1.0000 | 78427 |
| 2 | _consume#VB@dobj | 0.6521 | 34141 |
| 3 | _drink#VB@dobj | 0.6255 | 18814 |
| 4 | _ingest#VB@dobj | 0.5984 | 3965 |
| 5 | _cook#VB@dobj | 0.5980 | 10631 |
| 6 | _swallow#VB@dobj | 0.5414 | 5390 |
| 7 | _eat_out#VB@dobj | 0.5316 | 116 |
| 8 | _devour#VB@dobj | 0.5242 | 3080 |
| 9 | _digest#VB@dobj | 0.4930 | 2547 |
| 10 | _eat_up#VB@dobj | 0.4925 | 472 |

| 0 | eat#VERB | 1 | 109537 |
|---|---|---|---|
| 2 | drink#VERB | 0.6370 | 39160 |
| 3 | consume#VERB | 0.6151 | 44994 |
| 4 | cook#VERB | 0.6138 | 25875 |
| 5 | devour#VERB | 0.6011 | 4366 |
| 6 | chew#VERB | 0.5581 | 6142 |
| 7 | vomit#VERB | 0.5555 | 3111 |
| 8 | ingest#VERB | 0.5551 | 5736 |
| 9 | sleep#VERB | 0.5460 | 45645 |
| 10 | swallow#VERB | 0.5455 | 10402 |
| 11 | munch#VERB | 0.5348 | 210 |

| 0 | _eat#VB@dobj | 1 | 78427 |
|---|---|---|---|
| 1 | _feed#VB@prep_on@pobj | 0.5829 | 40039 |
| 2 | _feast#VB@prep_on@pobj | 0.5103 | 588 |
| 3 | _taste#VB@prep_like@pobj | 0.4741 | 577 |
| 4 | _subsist#VB@prep_on@pobj | 0.4582 | 1393 |
| 5 | _regurgitate#VB@prep_by@pobj | 0.4516 | 9 |
| 6 | _dine#VB@prep_on@pobj | 0.4512 | 163 |
| 7 | _consume#VB@prep_as@pobj | 0.4512 | 1050 |
| 8 | _cook#VB@prep_like@pobj | 0.4508 | 145 |
| 9 | _be_rice#VB@prep_along@prep_with@pobj | 0.4491 | 11 |
| 10 | _forage#VB@prep_for@pobj | 0.4439 | 1648 |

9. ábra. Az (1) *eat* ige tárgyához leghasonlóbb tárggyal rendelkező igék listája, (2) az *eat* ige tárgyához leghasonlóbb főnevek listája, és az (3) *eat* ige tárgyához leghasonlóbb prepozícióval rendelkező igék listája

### 5.4. A vonzatok irányából induló lekérdezések

Ha nem az igék, hanem a vonzatok irányából indulva teszünk fel kérdéseket a rendszernek, akkor arra kaphatunk választ, hogy egy-egy főnév tipikusan mi-

lyen igékkel áll valamilyen meghatározott viszonyban, például mi szokott történni vagy miket szoktunk csinálni az adott dologgal. Ha például megkérdezzük a rendszertől, hogy milyen igék tárgya a *skeleton* 'csontváz' főnév (10. ábra), akkor a számtalan ásatással, temetéssel, ravatalozással, rekonstrukcióval kapcsolatos ige mellett megjelenik a *char* 'elszenesedik' ige is, amelynek nem tárgya, hanem alanya az, ami elszenesedik. Ezt a hibát az annotálórendszerünkben alkalmazott azon feltételezés vezeti be, hogy a befejezett melléknévi igenevek által módosított főnév eredetileg az ige tárgya, azonban a páciens alanyú igékből is képezhető befejezett melléknévi igenév. Hasonlóképpen kerül a *Mit eszünk?* kérdésre kapott válasz elemei közé néhány idegen nyelvű 'enni' jelentésű szó, pl. az ógörög $\phi\alpha\gamma\epsilon\iota\nu$ vagy a finn *syödä*, amelyek az angol Wikipediában szereplő etimológiai fejtegetéseknek az elemzőrendszer általi félreelemzéséből jöttek létre (pl. a többször előforduló *Greek "*$\phi\alpha\gamma\epsilon\iota\nu$*" to eat* olyan alakú szerkezet, mint a *some food to eat*) (7. ábra).

| 0 | skeleton#NOUN | 1 | 18934 |
|---|---|---|---|
| 1 | _unearth#VB@dobj | 0.4950 | 5492 |
| 2 | _discover#VB@dobj | 0.4943 | 156506 |
| 3 | _excavate#VB@dobj | 0.4924 | 12528 |
| 4 | _find#VB@dobj | 0.4643 | 882721 |
| 5 | _disarticulate#VB@dobj | 0.4513 | 12 |
| 6 | _fossilize#VB@dobj | 0.4504 | 68 |
| 7 | _uncover#VB@dobj | 0.4426 | 18932 |
| 8 | _derive#VB@dobj_that@prep_of@pobj | 0.4402 | 12 |
| 9 | _excavate_up#VB@dobj | 0.4284 | 6 |
| 10 | _mummify#VB@dobj | 0.4098 | 144 |

10. ábra. A *skeleton* tárgyú igék listája

### 5.5. Eredmények

Idő és gold standard adatok hiányában sajnos csak egy viszonylag szűk lekérdezéslista eredményeként kapott válaszok pontosságának kiértékelésére volt módunk. Ötféle lekérdezés eredményét teszteltük:

1. adott ige tárgyaként milyen főnevek jelennek meg
2. az adott ige tárgyaként megjelenő főnevek milyen más igék tárgyaként jelennek meg
3. az adott ige tárgyaként megjelenő főnevek milyen más igék prepozíciós vonzataként jelennek meg
4. adott ige alanyaként milyen főnevek jelennek meg
5. adott főnév milyen igék tárgyaként jelenik meg

Az tárgyra vonatkozó első három lekérdezés eredményét a következő igékre értékeltük ki: *eat* 'eszik', drink 'iszik', mine 'bányászik' prove 'bizonyít' build

‘épít’ excavate ‘kiás, ásatásokat folytat’ terminate ‘megszüntet, befejez’ expect ‘vár’. Az alanyra vonatkozó lekérdezést a következő igékre: eat ‘eszik’, leak szivárog, kiszivárogtat’, explode ‘felrobban’, prove ‘bizonyít’, flow ‘folyik’, dry szárít’. Az utolsó „milyen igék tárgya” lekérdezést pedig a következő főnevekre futtattuk: skeleton ‘(csont)váz’, rice ‘rizs’, toy ‘játék’, key ‘kulcs’, lamp lámpa’, paper ‘papír, cikk’, lamb ‘bárány’.

Minden lekérdezéshez az első 40 jelöltet értékeltük ki. Helyesnek értékeltünk egy választ, ha az adott szó az adott viszonylatban helyes (evéshez étel, bányászathoz ásvány vagy ahonnan bányásznak – ez is lehet a *mine* ige tárgya, csontvázhoz kiásás, elásás stb.), illetve ha van olyan helyes és tipikus vonzat, amivel a másik ige által megnevezett tevékenységet tényleg szokták csinálni. Ha nem a megfelelő vonzatviszonyban jelent meg egy szó, azt nem fogadtuk el (pl. a *folyó-kanyarulatban, kanyonban* stb. *folyik víz*, de nem maga a *kanyon* folyik). Nem anyanyelvi beszélőként egyébként a válaszok első ránézésre zajosabbnak tűntek, mint amilyennek végül bizonyultak: utánanézve az eredményül kapott igéknek és főneveknek, a gyanús és számunkra ismeretlen szavak nagyobb részéről az derült ki, hogy valóban jó találat.

Az 1 táblázatban látható, hogy mit kaptunk az egyes lekérdezések eredményének illetve az összes lekérdezés aggregált eredményének pontosságára. Látható, hogy az első benyomásoknak megfelelően az alany viszonylatában kaptuk a leggyengébb eredményt, legjobban pedig a „melyik másik igéknek vannak hasonló tárgyai” kérdésre válaszolt a rendszer.

| típus | pontosság |
|---|---|
| tárgy>főnév | 0.85 |
| tárgy>másik ige tárgya | 0.95 |
| tárgy>másik ige prep. vonzata | 0.76 |
| alany>főnév | 0.71 |
| főnév>milyen ige tárgya | 0.82 |
| all | 0.83 |

1. táblázat. A rendszer válaszainak pontossága a tesztelt lekérdezéstípusokra

## 6. Konklúzió

A cikkben bemutattunk egy algoritmust és egy azzal generált konkrét modellt, amely egy közös vektortérmodellben ábrázolja felszíni szóalakok és az azokhoz rendelt annotációk disztribúcióalapú reprezentációját. A bemutatott modellben morfoszintaktikai és részleges szintaktikai függőségi annotációt használtunk. Mivel a reprezentáció nagyon kompakt, a modelltől olyan viszonylag komplex kérdésekre, mint hogy *Mit szoktunk még azokkal a dolgokkal csinálni, amit inni szoktunk?* is nagyon egyszerű formában és rendkívül gyorsan értelmes választ

kapunk. Az algoritmus természetesen bármilyen más annotáció és az annotált elemek közös disztribúciós modellbe gyúrását és az annotáció formai jegyeire alkalmazott szűrők segítségével a különbözőféleképpen annotált elemek, illetve az annotálatlan nyers adatok közötti disztribúciós hasonlóságok feltárását is lehetővé teszi.

## Köszönetnyilvánítás

Jelen kutatás az FK 125217 és a PD 125216 számú projekt keretében az FK 17 és a PD 17 pályázati program finanszírozásában a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással valósult meg.

## Hivatkozások

1. Ebert, S., Müller, T., Schütze, H.: LAMB: A good shepherd of morphologically rich languages. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, USA (2016)
2. Novák, A., Novák, B.: POS, ANA and LEM: Word embeddings built from annotated corpora perform better. In Gelbukh, A., ed.: Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2018, Hanoi, Vietnam, Springer International Publishing, Cham. (2018)
3. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, Association for Computational Linguistics (2013) 746–751
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. CoRR **abs/1607.04606** (2016)
6. Novák, A., Siklósi, B., Wenszky, N.: Szóbeágyazási modellek vizualizációjára és böngészésére szolgáló webes felület. In Tanács, A., Varga, V., Vincze, V., eds.: XIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem, Informatikai Tanszékcsoport (2017) 355–362

# Interpretability of Hungarian embedding spaces using a knowledge base

Vanda Balogh[1], Gábor Berend[1,2], Dimitrios I. Diochnos[3], György Turán[2,4], Richárd Farkas[1]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
[3]Department of Computer Science, University of Virginia
[4] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago
{bvanda, berendg, rfarkas}@inf.u-szeged.hu diochnos@virginia.edu gyt@uic.edu

**Abstract.** While word embeddings have proven to be highly useful in many NLP tasks, they are difficult to interpret for humans. Sparse word embeddings are reminiscent of knowledge bases containing words that are already characterized in sparse forms. In our work, we investigate to what extent sparse word representations convey knowledge about the words in knowledge bases. We utilize Hungarian sparse word embeddings and ConceptNet, a knowledge base that supports Hungarian.
**Keywords:** sparse word embedding, interpretability, knowledge base, ConceptNet

## 1 Introduction

Word embeddings generate low dimensional word representations from large corpora. Each word is represented as a vector of real numbers. Word vectors that are similar to each other tend to be semantically related which makes word embeddings effective in a variety of natural language processing tasks. Even though word embeddings perform well in these tasks, they are difficult to interpret for humans. Word embeddings employ dense representations of words, while natural language phenomena are extremely sparse by their nature. Motivated by this sparse behaviour, the construction of word embeddings that were made sparse is getting popular recently [1,2,3,4,5].

Knowledge bases already include words in sparse forms implicitly. The relations in these knowledge bases can describe how the words are related to each other by their lexical definition, and also how they are related through commonsense knowledge. Thus, we have human interpretable features at hand. This appealing characteristic of human assembled knowledge representation has already inspired others to create non-distributional word representations [6].

In our work, we would like to explore the interpretability of the dimensions of distributed word embeddings. As our first experiment, we examine Hungarian sparse embedding matrices by assigning each dimension one concept extracted

from ConceptNet[7], a multilingual knowledge base. One potential application of these assignments can be knowledge graph expanding.

## 2 Related Work

The explanatory power of distributional semantic models (DSMs) in terms of meaning is not clear as they often provide a quite coarse representation of semantic content [8]. There have been proposals for the semantic evaluation of DSMs, e.g., QVEC[9] and BLESS[10]. The QVEC evaluation measure aims to score the interpretability of word embeddings, a topic close to our research. Dimensions of the word embeddings are aligned with interpretable dimensions – corresponding to linguistic properties extracted from SemCor [11] – to maximize the cumulative correlation of the alignment. BLESS is a dataset designed for the semantic evaluation of DSMs. It contains semantic relations connecting (target and relatum) concepts as tuples. Thus, BLESS allows the evaluation of models by their ability to extract related words given a target concept. The method called the THING RECOGNIZER [12] attempts to make Hungarian embedding spaces interpretable by assigning semantic features to words in a language-independent manner.

In the following, we briefly introduce ConceptNet, a semantic multilingual knowledge base. In our work, we extract interpetable features (concepts) from ConceptNet in order to help exploring the interpretability of the dimensions of word embeddings.

### 2.1 ConceptNet

| Relation | Symmetric | Example Assertion |
|---|---|---|
| ANTONYM | ✓ | deep ↔ shallow |
| HASCONTEXT | ✗ | gurl → slang |
| HASPROPERTY | ✗ | marsh → muddy and moist |
| ISA | ✗ | eagle → bird |
| MADEOF | ✗ | ice → water |
| SYNONYM | ✓ | bright ↔ sunny |
| RELATEDTO | ✓ | torture ↔ pain |
| USEDFOR | ✗ | science → understand life |

Table 1: Extract of relations from ConceptNet 5.

ConceptNet is a semantic multilingual knowledge base describing general human knowledge collected from a variety of resources including WordNet, Wiktionary and Open Mind Common Sense. ConceptNet can be perceived as a graph whose nodes correspond to words and phrases. The nodes of the semantic network are called *concepts* and the (directed) edges connecting pairs of nodes are called *relations*. The records of the knowledge base are called *assertions*. Each assertion associates two concepts – *start* and *end* nodes – with a relation in the

semantic network and has additional satellite information beyond these three objects; for example, the *dataset* from where the assertion was obtained (e.g., WordNet). Figure 1 provides an example of an assertion found in ConceptNet 5 – the latest iteration of ConceptNet. Relations can be symmetrical, e.g., SYNONYM and RELATEDTO, or asymmetrical, e.g., HASPROPERTY and ISA. An incomplete list of relations present in ConceptNet 5 can be found in Table 1.


```
{
  "dataset": "/d/wikitionary/en",
  "license": "cc:by-sa/4.0",
  "sources": [{"contributor": "/s/resource/wikitionary/en"}],
  "weight": 1.0,
  "uri": "/a/[/r/HasContext/,/c/hu/poligon/n/,/c/en/geometry/]",
  "rel": "/r/HasContext",
  "start": "/c/hu/poligon/n",
  "end": "/c/en/geometry"
}
```


Fig. 1: Example assertion from ConceptNet 5. The *start* and *end* nodes are connected by an edge labelled *rel* corresponding to the relation between the nodes. Assertions feature additional information like *dataset* which represents the source of the assertion and *weight*, the strength of the assertion which is a positive value.

## 3 Experiments

Our aim is to explore the interpretability of the dimensions of Hungarian embedding matrices by assigning each dimension a human interpretable feature. A somewhat unnatural characteristic of standardly applied word embeddings (e.g. word2vec [13] or Glove [14]) is that the learned vectors have non-zero coefficients everywhere, implying that every word can be characterized with every dimension at least to a tiny extent. From a human perception point of view this dense behavior is quite undesired, because for most features we would not like to see any relation to hold. To approximate the sparse behaviour of natural language phenomena, we employ embeddings that are turned sparse as a post-processing step suggested in [4]. Results from other studies and literature argue that sparse word representations are more interpretable by humans (e.g. word intrusion) and perform well on downstream tasks (e.g. sentiment analysis) [1,3,2,5,15,4].

The rows of a sparse embedding matrix $S$, correspond to sparse word vectors representing words. We call the columns (dimensions) of the sparse embedding matrix *bases*. As human interpretable features, we take concepts extracted from a semantic knowledge base, ConceptNet, and the sparse embedding we employ is derived from the dense Numberbatch [16] vectors. This way, our goal reformulates to designating a concept to each base.

We basically deal with a tripartite graph (see Figure 2) with words connected to bases – corresponding to the columns of the embedding matrix – and concepts,

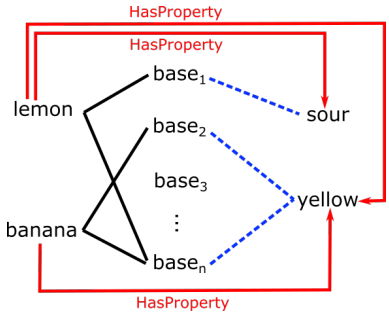



Fig. 2: Tripartite graph presenting the connections between embedded words, bases and concepts. Connections denoted by solid lines are present, our aim is to recover the relations between bases and concepts (dashed lines).

respectively. A word, $w$ is connected to $base_i$ if the $i$th coordinate of the sparse word vector corresponding to $w$ is nonzero. Also, $w$ is connected to a concept $c$ with label $l$ if there exists an assertion in ConceptNet that associates $w$ and $c$ with the relation $l$. We are interested in the relations between concepts and bases (dotted lines).

### 3.1 Hungarian sparse word embeddings

Numberbatch [16] is an embedding approach combining distributional semantics and ConceptNet 5.5 using a variation on retrofitting [17]. The Hungarian sparse word embeddings are derived from dense Numberbatch embeddings related to Hungarian concepts (i.e., concepts prefixed with `/c/hu/`). As a side note, the words present in ConceptNet align much better with the vocabulary provided by Numberbatch than with other embeddings' vocabulary. This is because Numberbatch implicitly makes use of words (and their specific forms) from ConceptNet and any arbitrary embedding would include a vast amount of forms of a single word since Hungarian is a morphologically rich language.

Sparse embeddings $\mathbf{s}_i$ are derived from dense embeddings $\mathbf{x}_i$ according to the objective function

$$\min_{D \in \mathcal{C}, s} \frac{1}{2n} \sum_{i=1}^{n} \left( \|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 + \lambda \|\mathbf{s}_i\|_1 \right),$$

where $D$ is a dictionary matrix of basis vectors with length not exceeding 1. The regularization constant, $\lambda$ controls the sparsity of the resulting embeddings $s_i$. As $\lambda$ increases, the density of the nonzero coefficients in $s_i$ decreases. In total, we use four sparsity levels according to $\lambda$s from $\{0.2, 0.3, 0.4, 0.5\}$. Table 2 shows sparsity of each sparse embedding matrix. We have a vocabulary of 17k words which are embedded into a vector space of 1000 dimensions.

| $\lambda$ | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|
| sparsity | 99.66% | 99.81% | 99.88% | 99.92% |

Table 2: The ratio of zero elements to all the elements from sparse embedding matrices with $\lambda$ regularization coefficient.

### 3.2 Hungarian ConceptNet

We utilize the Hungarian part of ConceptNet 5.5 and ConceptNet 5.6. in our experiments. Every assertion has a *start* and *end* node, which are connected by a directed labeled edge where the label is specified by the relation between the nodes. Basically, an assertion is a triplet of (start node, relation, end node). If the relation is symmetric, the connecting edge is bidirectional. In the following, we refer to start nodes as (embedded) *words* and end nodes are regarded as *concepts* (which should not be confused with the concepts mentioned in Section 2.1). These end nodes – seen as concepts – will be assigned to the bases of the sparse embedding matrix.

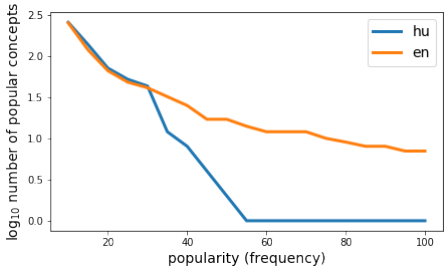



Fig. 3: Comparison on the number of English and Hungarian concepts that appear frequently (above frequency) as end nodes in assertions. The y axis shows the $\log_{10}$ of the number of concepts that are frequent.

For our experiments, we produce the subgraph of ConceptNet which encodes useful information on Hungarian (embedded) words. It is important to note that English is a core language of ConceptNet (i.e. the language is admittedly well supported) while Hungarian is not. First, we take the assertions associating two Hungarian nodes and to further diversify them, we adopt assertions associating a Hungarian start node with an English end node. It is worth to expand the set of concepts with English concepts, because there are significantly more English concepts that appear a lot as end nodes of assertions i.e. among the popular concepts there are more English ones (see Figure 3). To avoid redundancy, the assertions including symmetric relations that connect two Hungarian concepts are dropped. Instead, these groups of Hungarian words defined by symmetric relations (eg. synsets via the SYNONYM relation) are represented by English end nodes. In other words, the assertions including symmetric relations between Hungarian and English are kept in order to group together Hungarian concepts connected by symmetric relations according to their English equivalent. As an

example the Hungarian synonyms "ronda", "csúnya" and "ocsmány" are all connected to the English "ugly" through a SYNONYM relation. Instead of working with the complete graph of these Hungarian words that contains unnecessary information, we simply make use of the information that they can be grouped together by the English "ugly".

| assertion \ version | **ConceptNet 5.5** | **ConceptNet 5.6** |
|---|---|---|
| any → any | 28 million | 32 million |
| hu → hu | 31984 | 51819 |
| hu → en | 57941 | 61666 |
| hu → (hu ∨ en) | 89925 | 113485 |
| *filtered* hu → hu | 23844 | 42403 |
| *filtered* hu → (hu ∨ en) | 81785 | 104069 |

Table 3: Summary on the number of assertions in ConceptNet 5.5 and ConceptNet 5.6. The assertion types are listed according to the languages of the connected nodes. The filtered assertions disregard possible assertions associating two Hungarian nodes with a symmetric relation.

Altogether, we have a result of 81k and 104k assertions from ConceptNet 5.5 and 5.6, respectively. Further on, we will refer to the resulting subsets of ConceptNet globally as *Hungarian Conceptnet* (HCN) and use it in our experiments. The version 5.5 or 5.6 (of HCN) is always specified if required. Table 3 summarizes the number of assertions present in HCN 5.5 and 5.6. For further purposes, we experiment with end nodes that are with the connecting relation to reflect the meaning of assertions. We call this approach *augmented* in terms of the representation of end nodes – seen as concepts. So the assertion associating "eb" and "dog" with the SYNONYM relation has its start node "eb" and its end node is "dog/SYNONYM".

Basically, HCN 5.5 is used for association of concepts to bases and HCN 5.6 is used for evaluation. All in all, there are 48k and 58k distinct end nodes included in HCN 5.5 and HCN 5.6., respectively. If we ignore the relations by which end nodes were augmented we get 44k and 53k different relations. Although relations may be important in terms of meaning, we may resort to ignoring them to be able to further group together words according to their connecting concepts. Ignoring relations is also motivated by assertions like (a_fiók, SYNONYM, jacket), which presents a case where probably the relation SYNONYM is wrong between the word "a_fiók" and "jacket". A relation like ATLOCATION or RELATEDTO would fit better. In general, we use two types representation for concepts (ends nodes): the ones ignoring relations and the augmented approach.

Overall, there are 26 types of relations present in HCN. Surprisingly, there are relations for which there are substantially fewer assertions in HCN 5.6 than HCN 5.5 (see Figure 4). A lot of relations have the same number of assertions in both versions of HCN. The relation ETYMOLOGICALLYDERIVEDFROM is not present in HCN 5.5 and some of the richest relations include DERIVEDFROM, FORMOF, RELATEDTO and SYNONYM.

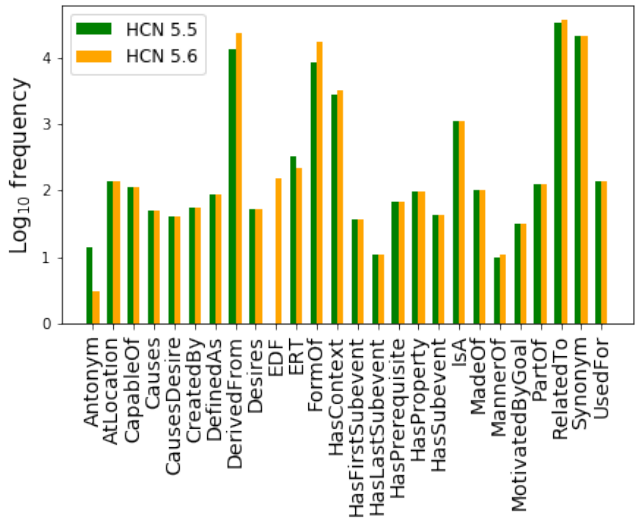



Fig. 4: $\text{Log}_{10}$ frequency of the assertions in HCN 5.5 and 5.6 according to relations. The relation EDF is short for EtymologicallyDerivedFrom and ERT refers to EtymologicallyRefersTo.

### 3.3 Phases of association

The process of associating a base with a concept is divided into four phases. First, we produce an adjacency matrix based on HCN 5.5, then we multiply its transpose with the sparse word embedding matrix. Afterwards, the positive pointwise mutual information (PPMI) values of the resulting matrix are computed and finally, the association takes place by taking the argmax of the matrix containing PPMI values. The four phases are detailed below. Figure 5 provides an overview of the four phases.

***I. Produce ConceptNet matrix.*** Given HCN 5.5 (described in §3.2), we consider it as a bipartite graph whose two sets of vertices correspond to two ordered sets containing the start and end nodes of assertions, respectively. The start nodes are regarded as (possibly embedded) words and the end nodes as concepts. The bipartite graph is represented as a biadjacency matrix $C$ (which simply discards the redundant parts of a bipartite graph's adjacency matrix). Every word $w$ corresponding to a start node is associated with an indicator vector $v_w$ where the $i$th coordinate of $v_w$ is 1 if $w$ is associated to the $i$th end node, 0 otherwise. At this point, words can have two sparse representations: the vectors coming from sparse word embeddings and the binary vectors provided by HCN. To differentiate them, we call the former ones *embedded vectors* and the latter ones *ConceptNet vectors*. It is important to note, that it is possible for an embedded word to lack its ConceptNet vector representation if the word itself is not present in the set of start nodes. On another note, there are words

in HCN 5.5 that are not presented in the vocabulary of the embedding; that is, they do not have embedded vector representations.

***II. Compute product.*** We binarize the nonnegative sparse embedding matrix $S$ by thresholding it at 0, then we take the product of the transpose of $C$ and the binarized version of $S$. The result is a dense matrix $A$, whose element at the $i$th row and $j$th column equals the number of words the $i$th concept and the $j$th base (from the sparse embedding matrix) appear together.

***III. Compute PPMI.*** To generate a sparse matrix from the dense $A$ matrix, we compute its positive pointwise mutual information (PPMI) for every element. PPMI for the $i$th concept $c_i$ and $j$th base $b_j$ is computed as

$$\mathrm{PPMI}(c_i, b_j) = \max\left(0, \ln \frac{\mathrm{P}(c_i, b_j)}{\mathrm{P}(c_i)\mathrm{P}(b_j)}\right),$$

where probabilities are approximated as relative frequencies of words as follows: $\mathrm{P}(c_i)$ is the relative frequency of words connected to the $i$th concept, $\mathrm{P}(b_j)$ takes the relative frequency of words whose $j$th coefficient in their embedded vector representation is nonzero and $\mathrm{P}(c_i, b_j)$ is the relative frequency of the co-occurrences of the words mentioned above. The result is a sparse matrix $P$ whose columns correspond to bases, and its rows correspond to concepts.

***IV. Take argmax.*** By taking the arguments of the maximum vales of every column in $P$ we can associate a base with a concept.

```
Association(sparse_word_embedding, conceptnet){
    nodes = {(start, end) in conceptnet}
    C = biadjacency(nodes)
    A = transpose(C) * binarize(sparse_word_embedding)
    P = PPMI(A)
    max_concepts = argmax(P, max_by=columns)
    //the ith element is the concept associated with the ith base
    return max_concepts
}
```

Fig. 5: The process of associating concepts to bases summarized in pseudocode.

## 4 Evaluation metrics

To evaluate the associations between bases and concepts, we employ HCN 5.6. We are interested if the new assertions compared to HCN 5.5 are presented in the associations (which can be perceived as a link prediction task). We would like to measure if the prominent words of the $i$th base (i.e., the words whose $i$th embedded coordinate is nonzero) are in relation with the concept associated

to the $i$th base according to HCN 5.6 (only new assertions are considered). We define the set $D_{b_i}$ as the set that contains the prominent words of base $b_i$, i.e.

$$D_{b_i} = \{w_j | w_j(i) > 0, 1 \leq j \leq n\}$$

where $w_j(i)$ is the $i$th coordinate in the embedded vector representation of $w_j$ and $n$ is the size of the vocabulary of the sparse embedding. It is worth to mention that $D_{b_i}$ only depends on the embedding itself. Furthermore, we define $F_{b_i}$ as the subset of $D_{b_i}$ that contains words which are present in the new assertions as start nodes and have a connecting end node to the concept that is associated to $b_i$, formally

$$F_{b_i} = \{w_j | w_j \in D_{b_i} \text{ and } (w_j, \mathit{concept}(b_j)) \in N\},$$

where $\mathit{concept}(b_j)$ is the concept associated to $b_i$ and $N$ contains *(start node, end node)* pairs that make a new assertion to HCN 5.6. The following information retrieval measures are used for evaluation:

***Mean Reciprocal Rank*** The reciprocal rank (RR) of the $i$th base, $b_i$ is

$$\mathrm{RR}(b_i) = \frac{1}{\mathrm{rank}(w_{F_{b_i}})},$$

where $w_{F_{b_i}}$ is the word from $F_{b_i}$ with the highest coefficient in $b_i$, and for a word, $w$, $\mathrm{rank}(w)$ is the rank of $w$ among $D_{b_i}$, so that the word with the largest coefficient in $D_{b_i}$ has a rank of 1, and the word with the smallest coefficient has a rank of $|D_{b_i}|$. Mean Reciprocal Rank (MRR) is the mean of the reciprocal ranks over all the bases.

***Mean Precision*** The precision of base $b_i$ is computed as

$$\mathrm{Prec}(b_i) = \frac{|F_{b_i}|}{|D_{b_i}|}.$$

This way, mean precision (MR) equals $\frac{1}{m}\sum_{i=1}^{m}\mathrm{Prec}(b_i)$, where $m$ is the number of bases.

***Mean Average Precision*** The average precision of base $b_i$ is the following:

$$\mathrm{AvgPrec}(b_i) = \frac{1}{|D_{b_i}|}\sum_{k=1}^{n}\frac{|F_{b_i}^k|}{|D_{b_i}^k|},$$

where both $F_{b_i}^k$ and $D_{b_i}^k$ are cutoffs of $F_{b_i}$ and $D_{b_i}$, respectively, so that the words are restricted to the first $k$ words coming from the embedding vocabulary (of size $n$). Mean average precision is the mean of average precisions over the bases.

In addition to all the above, we examine these metrics in the light of all the assertions in HCN 5.5 and 5.6. In this case we only have to alter the definition of the set $F_{b_i}$ to

$$\hat{F_{b_i}} = \{w_j | w_j \in D_{b_i} \text{ and } (w_j, \mathit{concept}(b_j)) \in \hat{N}\},$$

where $\hat{N}$ contains *(start node, end node)* pairs that make an assertion in HCN 5.5 or 5.6, i.e., we let the assertions of both HCN versions to overlap.

## 5 Results and Discussion

Based on PPMI values, we map a concept to each base. Some associations can be inspected in Table 4. The first thing we notice is that English concepts are much more frequent than Hungarian ones. The proportion of English concepts is ranging between 98.9% and 100% for all $\lambda$s. One reason behind this is definitely that within HCN 5.5 the number of different English concepts is 35k (38k augmented), while the number of Hungarian concepts is 9k (9k augmented). Also, English concepts are more popular (see Figure 3).

Some concepts associated with bases reflect the dominant words of the bases (the words that had the highest coefficient in the specific base). However, some concepts seem to have nothing in common with the dominant words of the associated base. This might be because some of the less dominant words of the base contribute to the PPMI. For example, the concept "en/accident/Synonym" is associated with the 10th base of the sparse embedding matrix ($\lambda$=0.2) whose most dominant words include "személygépkocsi", "automobil", "autós", "kocsi", "autó", however, there are words – like "baleset", "mentőautó", "gázol", "ütközés", "gyorshajtás", "ittas vezetés" – with smaller coefficients in the base that have more in common with the concept itself.

| base | concept | PPMI | most dominant words of base |
|---|---|---|---|
| 875 | en/dehydrated/RelatedTo | 7.978 | aszalt szilva, dunyha, birsalma, birs, birskörte |
| 533 | en/hard_disk/RelatedTo | 7.824 | szerves, szervesen, farész, merisztéma, háncsrész |
| 927 | en/audacity/RelatedTo | 7.690 | habar, malter, habarcs, vakolat, mozsár |
| 243 | en/beach/Synonym | 7.285 | mamusz, pacsker, papucs, vietnami papucs, strandpapucs |
| 593 | en/adult/RelatedTo | 7.285 | érett, andragógia, felnő, felnőtt, felnőttképzés |
| 327 | en/absinthe/RelatedTo | 4.549 | odavisz, odaad,idead, átad, ad |
| 15 | en/about_cardinality/HasContext | 4.481 | irracionális szám, háromszögszám, numerikus analízis, másodfokú függvény, logaritmusfüggvény |
| 431 | en/about_cardinality/HasContext | 4.413 | ezik, eszt, aszt, lél, lál |
| 709 | en/agape/Synonym | 4.248 | többé kevésbé, a volánnál, mihelyt, dagály, egymás |
| 814 | en/abscess/Synonym | 3.856 | spongyabob kockanadrág, szemérem, rosszban, sárgavállú amazon, flerovium |

Table 4: Association pairs scoring the highest and lowest PPMI values at sparse embedding with $\lambda = 0.2$ using concepts augmented with relation type.

The associations are evaluated on four sparse word embeddings (based on their regularization constants) with two types of concept sets (with or without relations). Three evaluation measures connected to information retrieval (MRR,

MP, MAP) are used which can focus on either the new assertions to HCN 5.6 or all the assertions present in HCN 5.5 and 5.6. Table 5 shows the evaluation scores. The results of new assertions to HCN 5.6 naturally have very low scores. This is because, out of the 48k concepts in HCN 5.5, at most 1k is used for the associations and most of the concepts (end nodes) present in the new assertions come from the remaining 47k concepts. On average, there is only 35 common concepts between the concepts coming from the associations and the new assertions to HCN 5.6. Also, we know that there are around 20k more assertions in HCN 5.6, however this version of HCN introduces 5k assertions with concepts (end nodes) not present in HCN 5.5.

| $\lambda$ | aug | MRR | MP | MAP |
|---|---|---|---|---|
| 0.2 | ✗ | 0.00009 | 0 | 0 |
| | ✓ | 0.00026 | 0 | 0 |
| 0.3 | ✗ | 0.00117 | 0 | 0 |
| | ✓ | 0.00126 | 0 | 0 |
| 0.4 | ✗ | 0.00013 | 0 | 0 |
| | ✓ | 0.00121 | 0 | 0 |
| 0.5 | ✗ | 0.00113 | 0.00021 | 0.00005 |
| | ✓ | 0.00165 | 0.00017 | 0.00003 |

(a) Evaluation in terms of new assertions to HCN 5.6.

| $\lambda$ | aug | MRR | MP | MAP |
|---|---|---|---|---|
| 0.2 | ✗ | 0.02093 | 0.00333 | 0.00487 |
| | ✓ | 0.02197 | 0.00334 | 0.00485 |
| 0.3 | ✗ | 0.03780 | 0.00812 | 0.01178 |
| | ✓ | 0.04086 | 0.00829 | 0.01228 |
| 0.4 | ✗ | 0.05440 | 0.01559 | 0.01858 |
| | ✓ | 0.05661 | 0.01553 | 0.01963 |
| 0.5 | ✗ | 0.06731 | 0.02797 | 0.03090 |
| | ✓ | 0.06917 | 0.02573 | 0.02960 |

(b) Evaluation on all assertions available to HCN 5.5 and 5.6

Table 5: Evaluation scores for associations using sparse embeddings with $\lambda$ regularization constant, and concepts from HCN 5.5 either including relations or ignoring them.

We can observe that sparser embeddings (with higher $\lambda$ values) perform significantly better in terms of all evaluation metrics. Moreover, embeddings with lower $\lambda$ values miss most of the new assertions to HCN 5.6, which results in near zero scores. A reason for that can be that "less sparse" embeddings contain too much noise. Ignoring relations of concepts definitely helps the new assertions in terms of MP and MAP, although this is not true for all the assertions, generally. However, the MRR values are consistently better at augmented representations of concepts: on average the 16th most dominant word of each base (of the sparse embedding with $\lambda = 0.5$) is connected to the associated concept in either HCN5.5 or 5.6. The augmented representation of concepts restricts associations, but gives more precise results.

Some highlights from the best performing associations can be seen in Table 6. We can notice that some of the dominant words of specific bases do not actually form assertions in HCN together with the associated concepts. Thus, the power of the resulting associations resides in the ability to augment existing knowledge bases or improve their quality.

| base | concept | dominant words from base |
|---|---|---|
| 46 | en/association_football/RelatedTo | **labdarúgó**, **futball**, labdarúgás, **foci**, **focilabda** |
| 198 | en/athletics/HasContext | **súlygolyó**, **súlyemelés**, **súlyemelő**, **súlyú**, súly |
| 773 | en/dustman/RelatedTo | szemetes, **hulladék**, **szemeteskonténer**, **szemét**, **szemetet** |
| 139 | en/bespectacled/RelatedTo | **optika**, **optikus**, **lencse**, **szemlencse**, **szemüveg** |
| 43 | en/building_material/HasContext | **kőkemény**, **mészkő**, **kőbánya**, **homokkő**, **kő** |

Table 6: Example for remarkable associations between concepts and bases with some of the dominant words in the base. Words in **bold** do not form an assertion in HCN with the associated concept, making the resulting associations applicable in knowledge base expanding.

## 6 Conclusion

The general theme of our study was the interpretability of Hungarian word embeddings. We experimented with associating a property (concept) for each column of the embedding matrix. Motivated by the sparse behaviour language phenomena, we employed sparse word embeddings which provide sparse vectorial representation for words.

We utilized four Hungarian sparse embeddings provided by Numberbatch. The concepts were extracted from ConceptNet 5 with Hungarian language in focus. English concepts were adopted to enrich the set of concepts and because they were more popular among assertions. The concepts could either be augmented with relations or not. The strategy of association was based on PPMI values. To measure how the associations reflect the assertions, we introduced four metrics, namely Mean Reciprocal Rank, Mean Average Reciprocal Rank, Mean Precision and Mean Average Precision.

Overall, the results confirm the results of [10,9] that word embeddings have semantic content related to word meaning, and provide a further step towards identifying such word meanings explicitly. We can conclude that sparser representations (with higher $\lambda$) perform better in terms of all evaluation metrics, probably because they contain less noise. The augmented approach of concept representations seems to have more precise results in terms of ranking, but it is subject to noise. On the other hand, the non-augmented approach may be more comprehensive. Also, the results indicate that associations may provide help in expanding knowledge bases, especially ConceptNet. In our ongoing work we explore more general forms of word meaning.

## Acknowledgments

This work was supported by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008).

## References

1. Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T., Hovy, E.H.: SPINE: sparse interpretable neural embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. (2018) 4921–4928
2. Murphy, B., Talukdar, P.P., Mitchell, T.M.: Learning effective and interpretable semantic models using non-negative sparse embedding. In Kay, M., Boitet, C., eds.: COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India, Indian Institute of Technology Bombay (2012) 1933–1950
3. Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., Smith, N.A.: Sparse overcomplete word vector representations. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics (2015) 1491–1500
4. Berend, G.: Sparse coding of neural word embeddings for multilingual sequence labeling. Transactions of the Association for Computational Linguistics **5** (2017) 247–261
5. Sun, F., Guo, J., Lan, Y., Xu, J., Cheng, X.: Sparse word embeddings using l1 regularized online learning. In: IJCAI, IJCAI/AAAI Press (2016) 2915–2921
6. Faruqui, M., Dyer, C.: Non-distributional word vector representations. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics (2015) 464–469
7. Speer, R., Havasi, C.: Representing general relational knowledge in conceptnet 5. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), European Language Resources Association (ELRA) (2012)
8. Lenci, A.: Distributional models of word meaning. Annual Review of Linguistics **4**(1) (2018) 151–171
9. Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., Dyer, C.: Evaluation of word vector representations by subspace alignment. In: EMNLP, The Association for Computational Linguistics (2015) 2049–2054
10. Baroni, M., Lenci, A.: How we BLESSed distributional semantic evaluation. In: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics. GEMS '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 1–10
11. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.: A semantic concordance. In: HLT, Morgan Kaufmann (1993)

12. Novák, A., Novák, B.: Cross-lingual generation and evaluation of a wide-coverage lexical semantic resource. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), European Language Resource Association (2018) 45–51
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
14. 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), Doha, Qatar, Association for Computational Linguistics (2014) 1532–1543
15. Vyas, Y., Carpuat, M.: Sparse bilingual word representations for cross-lingual lexical entailment. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (2016) 1187–1197
16. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge (2017)
17. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2015) 1606–1615

# Mit hozott édesapám? Döntést – Idiomatikus és félig kompozicionális magyar igei szerkezetek azonosítása párhuzamos korpuszból

Novák Attila, Laki László János, Novák Borbála

Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Kar
MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
Budapest, Práter u. 50/a.
{vezetéknév.keresztnév}@itk.ppke.hu

**Kivonat** Cikkünkben egy olyan algoritmust mutatunk be, amelynek segítségével elemzett angol-magyar párhuzamos korpuszból gyűjtöttünk idiomatikus és félig kompozicionális igei szerkezeteket a szómegfeleltetések felhasználásával. Mivel a kutatás kontextusát egy kérdések megfogalmazását lehetővé tevő elemző megalkotása jelentette, ezek a szerkezetek elsősorban abból a szempontból érdekeltek minket, amennyiben a szerkezet egyes elemeire a kompozicionalitás hiányából kifolyólag nem lehet kérdezni. Az eredményeket összevetettük egy ilyen szerkezeteket tartalmazó létező erőforrással, és azt találtuk, hogy algoritmusunk sok új számunkra érdekes nem vagy részben kompozicionális igei szerkezetet azonosított.

## 1. Bevezetés

A nem kompozicionális igei szerkezetek olyan igéből és főnévből álló kifejezések, ahol a kifejezés jelentése nem kiszámítható a szerkezet tagjainak jelentéséből. Ezek lehetnek teljesen idiomatikus kifejezések, de olyan szerkezetek is, ahol ugyan a kifejezés egyik vagy akár mindkét tagjának a jelentése fontos mozzanatot visz a komplex kifejezés jelentésébe, az utóbbi mégsem egészen a szokásos kompozicionális jelentéskombinációs műveletek eredményeként áll elő, hanem vagy tartalmaz valami pluszt, vagy valamelyik elem (rendszerint az ige) jelentéséből legfeljebb valamilyen mozzanat jelenik meg. A nem teljesen idiomatikus kifejezéseket szokás félig kompozicionális igei szerkezeteknek hívni. Bár a ma egyre inkább teret hódító neurális architektúrák kevésbé érzékenyek ezekre a szerkzetekre, kezelésük szinte minden nyelvtechnológiai feladatban külön figyelmet igényel.

Az ebben a cikkben bemutatott módszerrel angol-magyar párhuzamos korpuszból gyűjtöttünk ki félig vagy még kevésbé kompozicionális igei szerkezeteket. Ezekkel az volt a célunk, hogy egy fejlesztés alatt álló elemzőrendszerhez készített igei vonzatkeret-adatbázis építését és ezeknek a korpuszhoz való illesztését támogassuk [1].

## 2. A félig kompozicionális igei szerkezetek

Formailag a félig kompozicionális igei szerkezetekben egy ige és egy vagy több (ragozott, prepozíciós vagy névutós) névszói vonzat szerepel. A nyelvészetben számos csoportosítása ismert a félig kompozicionális, illetve idiomatikus szerkezeteknek olyan paraméterek mentén, hogy a kifejezés egyes tagjainak önálló szintaktikai és szemantikai szerepe mennyiben és hogyan járul hozzá a kifejezés egészének jelentéséhez.

Az egyik ilyen csoportosítás [2]-ben szerepel, ahol a szerző négy csoportot definiál:

- idiómák: ahol a kifejezés tagjainak jelentéséből egyáltalán nem számítható ki az egész jelentése, pl. *feldobja a talpát* 'meghal'
- mind az ige, mind a főnév eredeti jelentésükben járul hozzá a kifejezés jelentéséhez, de a kifejezés hordoz valami pluszt, pl. *iskolába jár* 'ott tanul'[1]
- idiómaszerű, de az idiómáknál szabadabb kifejezések, ahol az egyik tagra nem lexikális, hanem egy szemantikai kategóriát meghatározó megkötés van, pl. *főbe, hasba, lábon lő*
- azok az állandó fordulatok, ahol az ige jelentése nem jelenik meg a kifejezés jelentésében a maga teljességében, a vonzatszerű névszói elem apportálja a szemantikai tőke javát, az ige szinte pusztán a grammatikai kategóriáját viszi vásárra, jelentéstartalmából legfeljebb valamilyen mozzanat jelenik meg, pl. *lehetőség nyílik valamire*

A nemzetközi szakirodalomban is hasonló csoportosítások jelennek meg. Sag szerint [3] például a többszavas kifejezések két fő csoportot alkotnak: lexikalizálódott és intézményesült kifejezések. Lexikalizálódott kifejezéseknek azokat a kifejezéseket tekintik, amiknek a szintaktikai vagy szemantikai felépítése legalább részben idioszinkratikus, vagy olyan szavakat tartalmaznak, amik önmagukban nem fordulnak elő azzal a jelentéssel. Ezek a fajta kifejezések a lexikai kötöttségük szempontjából további alcsoportra bonthatók: teljesen kötött kifejezések, félig kötött kifejezések és szintaktikailag rugalmas kifejezések. A Sag rendszerezésében szereplő intézményesült kifejezések szintaktikailag és szemantikailag ugyan kompozicionálisak, de adott kontextusban az átlagosnál gyakrabban jelennek meg együtt.

Célunk nem az volt, hogy ezeknek az előre definiált csoportosításoknak megfelelő kifejezéseket gyűjtsünk, hanem egy saját kritériumrendszert állítottunk fel. Mivel a célunk egy olyan elemzőrendszer létrehozása, amely ténylegesen alkalmas arra, hogy releváns kérdéseket tegyen fel azzal a szöveggel kapcsolatban, amit feldolgoz [1], ezért a félig kompozicionális szerkezetek azonosítása során is ez volt a fő szempont.

Az idiomatikus és félig kompozicionális szerkezetek azonosításakor is azt a célt tartottuk tehát szem előtt, hogy egy kifejezés az arra vonatkozó releváns kérdés megfogalmazása szempontjából hogyan viselkedik. A *döntést hoz* kifejezés esetén nem jó kérdés a *Mit hoz?*, hacsak nem viccet szeretnénk csinálni belőle, pl.:

[1] A portás vagy a tanári kar hiába jár szintén oda, ők munkába járnak, nem iskolába.

```
- Mit hozott Édesapám?
- Döntést.
```

A kérdezés szempontjából ugyanakkor például az egyébként szintén nem kompozicionális *csinálja a fesztivált* kifejezés kevésbé tűnik érdekesnek, mert a minden ágenses igére használható *Mit csinál?* kérdést lehet ezzel kapcsolatban is feltenni. Az utóbbi esetben is megfigyelhető ugyanakkor, hogy némileg humoros hatást kelt a csak a tárgyat megnevező válasz: *Mit csinál? A fesztivált.*, ami egy kompozicionális ige-vonzat kapcsolat esetében teljesen normális lenne. Mindazonáltal a *csinál* ige a mi szempontunkból kevésbé tűnik „veszélyesnek", mint más idiomatikus vagy félig kompozicionális szerkezeteket alkotó igék.

Szintén nem érdekesek a mi szempontunkból az *iskolába jár, fát vág* típusú szerkezetek, ahol ugyan van valami jelentéstöbblet, mind az ige, mind a névszói vonzat jelentése viszonylag csorbítatlanul jelen van a kifejezés jelentésében, ezért nem ostobaság és nem vicc sem a vonzatra *(Mit vág?)* sem az igére *(Mit csinál a fával?)* kérdezni.

A szemantikailag kötött vonzatú igék esetében változatos a kép a kérdezés szempontjából. A *főbe/fejbe/hasba/ülepen/fenékbe/lábon lőtték* esetében nem lehet azt kérdezni, hogy *Mibe/min/hova lőtték?*, hanem csak a *Hol/melyik testrészén lőtték meg?* kérdés lehetséges. Tehát van két alternatív minta, mindkettőben testrészekre korlátozódik az egyik argumentumként megjelenő elemek köre, de az egyik minta (ahol egyébként a rag is változik az adott testrész függvényében) nem teszi lehetővé a testrészre kérdezést, a másik viszont (ahol a rag fixen a szuperesszívusz), lehetővé teszi azt. Látjuk tehát, hogy a szemantikai/lexikai kötöttség hol nyitva hagyja, hogy pedig nem teszi lehetővé az adott vonzatra kérdezést.

## 3. Módszer

A statisztikai gépi fordítás fénykorában számos, nem feltétlenül gépi fordítási feladatra, elkezdték alkalmazni a statisztikai gépi fordító rendszerek egyes alkotóelemeit. Az egyik ilyen alkotóelem a szómegfeleltetési modell (word alignment), ami a rendszer tanításához használt párhuzamos korpusz mondatain belüli szavakat megfelelteti egymással. Ez a megfeleltetés lehet $n : m$ vagy $m : n$, ahol $n \geq m$. A [4] cikkben a többszavas kifejezések szómegfeleltetési modell alapján történő azonosításának a következő definíciója szerepel:

Mivel a két nyelv közötti automatikus szómegfeleltetési modell a forrásnyelvi mondat szavainak ekvivalensét keresi a célnyelvi mondatban, ezért ha egy $S$ forrásnyelvi szósorozat (ahol $S = s_1...s_n, n \geq 2$) megfeleltethető egy $T$ célnyelvi szósorozatnak (ahol $T = t_1...t_m, m \geq 1$), azaz $S$ és $T$ egymás megfeleltetései, akkor feltételezhetjük, hogy $S$ és $T$ szemantikai tartalma legalább részben hasonló, és hogy $S$ egy potenciális többszavas kifejezés. Más szóval ez azt jelenti, hogy az $S$ szósorozat egy kifejezésjelölt, ha egy egy vagy több szóból álló $T$ sorozatnak feleltethető meg a célnyelven (azaz egy $n : m$ típusú megfeleltetésről van szó, ahol $n \geq 2, m \geq 1$). Fontos tehát, hogy a forrásnyelvi kifejezés több szóból

áll, ami a célnyelven egy vagy több szónak felel meg. Az általunk megvalósított algoritmus ebből a definícióból indul ki, de az eredeti célunk érdekében további megszorításokat tettünk, amiket a továbbiakban részletezünk.

### 3.1. A korpusz előkészítése

A keresett félig kompozicionális illetve idiomatikus igei kifejezéseket tehát egy párhuzamos korpuszból szerettük volna összegyűjteni. Ehhez egy 644,5 millió token méretű angol-magyar párhuzamos korpuszt [5] használtunk. Mivel a célunk az volt, hogy nem vagy félig kompozicionális igei szerkezeteket gyűjtsünk egy igei vonzatkeret-adatbázis építéséhez, ezért csak azokra a kifejezésekre koncentráltunk, amikben igék szerepelnek. Ehhez először a párhuzamos korpusz mindkét oldalának elemzésére volt szükség. Az angol oldalt a Stanford taggerrel [6] szófaji egyértelműsítettük és a morpha [7] lemmatizálóval lemmatizáltuk. A magyar oldalon a PurePos [8] szófaji egyértelműsítőt és lemmatizálót alkalmaztuk, amely a Humor morfológiai elemző [9,10] elemzéseit használja. Ezek után mindkét oldalon úgy alakítottuk át az elemzett szöveget, hogy minden eredeti tokent két token reprezentál: (1) a lemma a fő szófajcímkével és (2) a szóhoz tartozó további morfoszintaktikai címkék.

Az alábbi példa a *Szeretlek, kedvesem. – I love you, dear.* mondatpár így előfeldolgozott változatát mutatja:

```
szeret[IGE]  [Ie1] ,[PUNCT] kedves[FN] [PSe1][NOM]
I#PRP love#VB [P] you#PRP ,#, dear#RB
```

Ez az átalakítás veszteségmentesen megőrzi az eredeti szóalakokra vonatkozó információkat, így a végeredményben visszaalakíthatóak a szóalakok. Ez azért is fontos, mert sok esetben a félig kompozicionális kifejezésekben egy-egy szónak csak bizonyos alakja(i) megengedett(ek), tehát nem lenne elegendő a lemma azonosítása. Ugyanakkor, a kifejezések ilyen formában nem kötött részeire robosztusabb statisztikát kapunk a lemmák egységes kezelése miatt. Fontos szempont volt továbbá, ahogy már említettük, az igék azonosítása, amit ez az átalakítás szintén lehetővé tett.

### 3.2. A szómegfeleltetési modell létrehozása

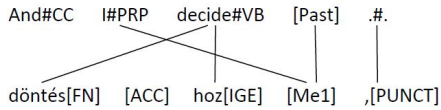
Az átalakított párhuzamos korpuszból a szómegfeleltetési modell létrehozásához a fast align programot [11] használtuk. Ennek kimenetéből a Moses SMT-készlet [12] frázistábla-építőjével készítettünk maximum 7-gram méretű frázispárokat. Bár a Moses különböző pontszámokat is rendel a megfeleltetett kifejezésekhez, amik többek között az adott fordítás feltételes valószínűségét jelzik, ezeket a pontszámokat végül nem használtuk fel, csupán magát a megfeleltetést, hogy melyik szavakat melyikhez rendelte hozzá.

Ezen kívül további szűréseket alkalmaztunk. A kapott listából kidobtuk azokat a frázispárokat, amikben nem szerepelt ige, vagy több ige szerepelt valamelyik oldalon, illetve azokat is, amikben mondat- vagy tagmondathatárra utaló jel (írásjelek) szerepeltek. Mivel a megfeleltetésben szereplő frázisok n-gram

alapúak, ezért ezek a szűrések nem jártak információveszteséggel, hiszen még a kidobott frázisok releváns részletei is megmaradtak másik frázis részeként. A szűrésnek köszönhetően viszont nem kerültek az eredménybe olyan hamis pozitív kifejezések, amik átlépnek például tagmondathatárokon. Hasonlóképpen, azoknak a frázisoknak az igéit tartalmazó részfrázisok, amikben több ige szerepelt, megjelentek másik frázisokban.

### 3.3. A félig kompozicionális szerkezetek kigyűjtése

A kifejezésjelölteknek a szűrés után megmaradt frázispárlistából való kiszűrése során azzal a feltételezéssel éltünk, hogy a magyar oldalon szereplő félig kompozicionális szerkezetek minden eleme az angol oldalon szereplő igéhez van kötve. Például ha az angol oldalon a *decide* ige szerepel, a magyar oldalon pedig a *döntést hoz* kifejezés, akkor ezek kötött áll fenn a megfeleltetés (1. ábra).




1. ábra. Példa egy szómegfeleltetésre az angol és magyar elemzett frázisok elemei között.

Az ilyen típusú megfeleltetéseket magyar igénként külön-külön kigyűjtöttük és az egyes gyűjteményekben szereplő angol igékhez összesítettük, hogy melyik angol igéhez melyik magyar ige milyen főnévvel szerepel. Ekkor már a magyar főneveket egyesítettük az utánuk külön tokenként szereplő morfoszintaktikai címkéjükkel és visszaalakítottuk az eredeti formájukra a Humor generátor funkciójával [9,10]. Ez a lista már önmagában érdekes abból a szempontból, hogy az angol igék a magyar felsorolásban szereplő kifejezéseknek tulajdonképpen mint egyszavas definíciói jelennek meg. Az 1. táblázatban a magyar *hoz* igéhez összegyűlt néhány angol ige és a szómegfeleltetés alapján a *hoz* mellett az adott igéhez kötött főnév visszaalakított alakjai láthatók. Így például a *fix* angol ige tulajdonképpen definiálja a *rendbe hoz, helyre hoz, működésbe hoz* magyar kifejezéseket. Ebbe a sorba bekerült a *dolgokat* vonzat is, ami a *rendbe hozza a dolgokat* kifejezésből ered, de mivel a szintaktikai viszonyokat nem vizsgáltuk, ezért nem tudunk különbséget tenni a különböző típusú vonzatok között. Az is látható a példából, hogy az algoritmusnak ezen a pontján olyan angol igékhez tartozó listák is létrejönnek, ahol az angol oldalon is félig kompozicionális igei szerkezet szerepel. Például a *make* ige esetén a *make a decision* a *döntést hoz* párja. Látható továbbá még az is, hogy a korpuszban megjelenő elírások, nem sztenderd szóalakok is megjelentek a listán.

Az algoritmus további lépéseiben az ilyen formán definícióként szereplő angol igékkel nem foglalkozunk, de a megfelelő tisztítás után jó alapanyaga lehet egy olyan szótárnak, amiben az összegyűlt többszavas kifejezések szerepelnek.

| angol ige | magyar főnevek |
|---|---|
| fix | rendbe, helyre, dolgokat, működésbe, stb. |
| scare | frászt, szívbajt, szívinfarktust, fraszt, szivbajt, stb. |
| make | döntést, változást, nyilvánosságra, hasznot, áttörést, világra, stb. |
| embarrass | zavarba, helyzetbe, szégyent, szégyenbe, stb. |
| freak | frászt, szívbajt, szivbajt, fraszt, stb. |
| connect | kapcsolatba, összefüggésbe, összeköttetésbe, stb. |

1. táblázat. A *hoz* igéhez tartozó kifejezésjelöltek listájára néhány példa az angol igék szerint csoportosítva

### 3.4. A kifejezések rangsorolása

Ahogy az előző fejezetben látható volt, sok olyan kifejezés is megjelent a listán, amik nem feltétlenül alkotnak félig kompozicionális kifejezést az éppen vizsgált magyar igével (pl. *dolgokat hoz*). Ezért különböző statisztikai mérőszámok lineáris kombinációjával meghatároztunk minden frázisjelölthöz egy pontszámot. A pontszámításhoz használt mérőszámok a következők voltak:

- az angol és magyar igepár közös előfordulásainak a száma, azaz hogy hányszor voltak összekötve egymással a szómegfeleltetési modellben
- hányszor volt a vizsgált igepár úgy összekötve egymással, hogy a magyar oldalon egy főnév is az angol igéhez volt kötve
- a különböző magyar főnevek száma, amivel az angol ige megfeleltetésben állt egy adott magyar ige esetén
- az adott magyar-angol ige megfeleltetés esetén, az angol igéhez kötött magyar főnevek mindegyikénél meghatároztuk azt, hogy hányszor fordult elő az adott igével, majd ezt elosztottuk az összes főnevek számával, ami az adott igével megfeleltetésben állt (normalizált gyakoriságérték)

A vágási küszöbértéket két szempont mentén határoztuk meg a fenti értékek alapján. Először az igepárokra vonatkozó szűrést végeztünk. Ekkor minden egyes angol-magyar igepárhoz a megfeleltetett magyar főnevek közül a legnagyobb normalizált gyakoriságértékkel rendelkező főnévhez tartozó értéket megszoroztuk a második paraméterrel, azaz azzal az értékkel, hogy hányszor volt a vizsgált igepár úgy összekötve egymással, hogy a magyar oldalon egy főnév is az angol igéhez volt kötve. Ez biztosította azt, hogy a szómegfeleltetési modellben csak nagyon ritkán egymáshoz rendelt igék, illetve főnevek ne kerüljenek a listába. Ez helyettesítette a frázistáblában eredetileg szereplő, a megfeleltetés valószínűségét tükröző pontszámokat is.

Az egyes igékhez tartozó főnevek listájában is meghatároztunk egy küszöbértéket az alapján, hogy a normalizált gyakoriságértékek szerint rendezett listában hol van hirtelen nagy esés. Erre azért volt szükség, mert ezekben a listákba sokszor bekerültek olyan szavak, amik ugyan tényleg szinte mindig az adott igével

voltak összekötve, de ez nem azért volt, mert annak magyar megfelelőjével valamilyen kifejezést alkotnának, hanem csupán azért, mert a korpuszban szereplő néhány előfordulásuk mindig azzal az igével szerepelt.

A két vágás segítségével tehát eliminálni tudtuk a nem megfelelő ige-ige és a nem megfelelő főnév-ige párosításokat.

## 4. Eredmények

Az algoritmus kiértékeléséhez a Szeged Korpuszból és a SzegedParalell korpuszból készült félig kompozicionális igei szerkezeteket tartalmazó listát használtuk [13]. Az ebben a listában félig kompozicionális kifejezések részeként szereplő igékre futtattuk le a fenti algoritmust.

A számunkra érdekes igei kifejezések (illetve a megfelelő ige-vonzat párok) azonosításához, és egyben az algoritmus kiértékeléséhez a korábban ismertetett kérdeztesztet alkalmaztuk. Azaz azokat a kifejezéseket tekintettük helyes találatnak, ahol az adott igének az adott névszó valóban vonzata, és a névszóra vonatkozó *kit/mit/hol/hova* stb. típusú kérdés az adott igével nem lehetséges, vagy vicces hatást kelt.

Az algoritmus 309 igére adott eredményt, ezekhez összesen 6531 névszójelöltet generált. Meglepően sok új a kérdezés szempontjából számunkra érdekes idiomatikus illetve félig kompozicionális kifejezést hozott felszínre, amelyek a Szeged Korpuszból és a SzegedParalell korpuszból készült listán nem szerepeltek. Ugyanakkor az utóbbi listán szereplő kifejezések egy része a mi tesztünk szerint nem volt problematikus. A cikk beadási határidejéig a lista 1/4-ét sikerült feldolgozni. Ezen az anyagon a Szeged Korpuszból és a SzegedParalell korpuszból készült, illetve a saját algoritmusunk által generált listán szereplő összes számunkra érdekes igei kifejezést alapul véve (ebből számoltunk fedést) a szegedi lista pontossága 83,6%-osra, fedése 32,2%-osra, a sajátunk pontossága 28,6%-osra, fedése 84,2%-osra adódott. A végeredményként előállt lista elemeinek 2/3-a tehát az itt leírt eljárás eredményeként került horogra, ami nagyon jó eredmény. A viszonylag alacsony pontosság miatt ugyanakkor mindenképpen az eredmények alapos kézi átvizsgálására van szükség. A pontosságot rontja többek között az is, hogy az igéhez tartozó igekötő nem minden esetben jelenik meg a frázistábla építésekor meghatározott 7 tokenes ablakban. Ez a hiba azonban kézi javítás során általában viszonylag könnyen orvosolható. Mivel eleve csak a kérdezés szempontjából problematikus idiomatikus és félig kompozicionális ige–névszó szerkezetek azonosítását tűztük ki célul, az algoritmus nem azonosítja ezeknek a szerkezeteknek a teljes vonzatkeretét sem (a lexikálisan kötött névszói elem melletti egyéb vonzatokat). Ezeket a tematikus szerepükkel együtt kézzel adjuk hozzá az algoritmus által azonosított szerkezetek leírásához.

## 5. Konklúzió

Cikkünkben egy elemzett angol-magyar párhuzamos korpuszból idiomatikus és félig kompozicionális igei szerkezeteket azonosító algoritmust mutattunk be. Az

itt bemutatott kutatás része egy kérdések megfogalmazását lehetővé tevő elemző megalkotására irányuló folyamatban levő projektnek, melynek részletesebb leírását l. a szintén jelen kötetben megjelent cikkünkben [1]. Az eredményeket egy ilyen szerkezeteket tartalmazó létező erőforrással összevetve azt találtuk, hogy algoritmusunk sok új számunkra érdekes nem vagy részben kompozicionális igei szerkezetet azonosított.

## Köszönetnyilvánítás

Jelen kutatás az FK 125217 és a PD 125216 számú projekt keretében az FK 17 és a PD 17 pályázati program finanszírozásában a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással valósult meg.

## Hivatkozások

1. Novák, A., Laki, L.J., Novák, B., Dömötör, A., Ligeti-Nagy, N., Kalivoda, A.: Egy magyar nyelvű kérdezőrendszer. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019), Szeged, SZTE (2019)
2. Komlósy, A.: Régensek és vonzatok. In Kiefer, F., ed.: Strukturális magyar nyelvtan 1. Akadémiai Kiadó (1992) 299–527
3. Sag, I.A., Baldwin, T., Bond, F., Copestake, A.A., Flickinger, D.: Multiword expressions: A pain in the neck for nlp. In: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing. CICLing '02, Berlin, Heidelberg, Springer-Verlag (2002) 1–15
4. de Medeiros Caseli, H., Ramisch, C., das Graças Volpe Nunes, M., Villavicencio, A.: Alignment-based extraction of multiword expressions. Language Resources and Evaluation **44**(1-2) (2010) 59–77
5. Lison, P., Tiedemann, J.: Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In Chair), N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
6. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. NAACL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 173–180
7. Minnen, G., Carroll, J.A., Pearce, D.: Applied morphological processing of english. Natural Language Engineering **7**(3) (2001) 207–223
8. Orosz, Gy., 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), Hissar, Bulgaria, Incoma Ltd. Shoumen, Bulgaria (2013) 539–545
9. Novák, A.: Milyen a jó Humor? [What is good Humor like?]. In: I. Magyar Számítógépes Nyelvészeti Konferencia [First Hungarian conference on computational linguistics], Szeged, SZTE (2003) 138–144

10. Novák, A.: A New Form of Humor – Mapping Constraint-Based Computational Morphologies to a Finite-State Representation. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (2014) 1068–1073 ACL Anthology Identifier: L14-1207.
11. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of ibm model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2013) 644–648
12. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: ACL, The Association for Computer Linguistics (2007)
13. Vincze, V.: Semi-Compositional Noun + Verb Constructions : Theoretical Questions and Computational Linguistic Analyses. PhD thesis (2012)

# Témaspecifikus gépi fordítórendszer minőségének javítása domain adaptáció segítségével

Laki László János[1,2,3]

[1] MorphoLogic Lokalizáció Kft.
1012 Budapest, Logodi utca 54
[2] MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport,
[3] Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar,
1083 Budapest, Práter utca 50/a
e-mail: laki.laszlo@itk.ppke.hu

**Kivonat** A mély tanulásos módszerek elterjedése napjainkban nagymértékben megváltoztatta a gépi fordítások emberi megítélését. A statisztikai gépi fordítórendszerekkel (SMT) szemben a neurálishálózat-alapon működő architektúrák (NMT) sokkal olvashatóbb fordításokat generálnak, melyek a hivatásos fordítók számára könnyebben és hatékonyabban javíthatók az utófeldolgozás során. Az új módszer nehézsége azonban, hogy a stabilan jó fodítási minőséget adó rendszerek tanításához nagy méretű tanítóanyagra van szükség. Ez azonban a legtöbb fordítócég vagy nyelvpár esetén nem áll rendelkezésre. Munkám során a kicsi és jó minőségű *in-domain* tanítóanyagokat adatszelekció segítségével feldúsítottam egy nagy méretű *out-of-domain* korpusz leginkább hasonló szegmenseivel. Az így létrehozott architektúrával sikerült statisztikailag szignifikáns mértékben javítanom a fordítórendszer minőségét az összes vizsgált esetben. Kutatásom során igyekeztem megtalálni a feladathoz leginkább alkalmas szelekciós módszert, illetve megvizsgáltam a rendszer működését több különböző nyelv- és domainpár kombinációval.
**Kulcsszavak:** NMT, domain adaptáció, adatszelekció

## 1. Bevezetés

Napjainkban a legtöbb tudományterületen teret hódítanak a neurálishálózat-alapú gépitanulásos módszerek, mivel segítségükkel jelentős javulást lehet elérni az eddig piacvezető statisztika-alapú módszerekhez képest. Ugyanez a tendencia figyelhető meg a gépi fordítás területén is. A neurálishálózat-alapú gépi fordító rendszerek (NMT) mára már nemcsak az emberi kiértékelés szempontjából, hanem az általánosan használt automatikus kiértékelő metrikák számai alapján is jobb minőséget produkálnak az eddig piacvezető SMT rendszerekhez (Statistical Machine Translation) képest [1]. Az NMT rendszerek előnye az SMT-vel szemben, hogy az emberi olvasó számára folyékonyabban olvasható fordításokat generálnak. Ennek köszönhetően sokkal nagyobb az elfogadottsága mind a hivatásos fordítók, mind a többi felhasználó körében. Hátránya azonban, hogy ehhez a stabil működéshez viszonylag nagy tanítóanyagra van szüksége. A tudományos

közösség jóvoltából a legtöbb nyelvpárra elérhetőek kisebb-nagyobb szabadon hozzáférhető párhuzamos korpuszok (lásd: OPUS párhuzamos korpusz gyűjtemény[4]). Ezek viszont nagyobb méretük mellett többnyire zajosak, és gyakoriak bennük a hibás, a nem odaillő, vagy a rosszul párosított fordítások.

A fordítással foglalkozó cégek, vagy a szabadúszó fordítók korábbi munkáikat fordítómemóriákba (TM – Translation Memory) gyűjtik. Általánosan igaz, hogy az esetek többségében ez a TM a kifejezetten jó minősége ellenére viszonylag kis méretű, így önmagában az NMT rendszer tanítására csak megkötésekkel alkalmas. Az adott domainbe tartozó szövegeket viszonylag magas minőséggel lehet velük fordítani, de amint a fordítandó szöveg eltérő domainből származik nagymértékben visszaesik a minőségük.

Munkám során az NMT fordítórendszer minőségének javítására tettem kísérletet olyan módon, hogy a jó minőségű *in-domain* tanítóanyagokat korpuszszelekció segítségével feldúsítottam *out-of-domain* anyagból kiválasztott szegmensekkel. A módszer lényege, hogy a kibővített anyaggal létrehozott fordítórendszerek robosztusabban képesek fordítani a tanítóanyaggal csak részben hasonló mondatokat, így javítva a rendszer minőségét. Megvizsgáltam több szelekciós módszer hatékonyságát, valamint összehasonlítottam a különböző szegmensszámú rendszerek minőségét.

A dolgozat tematikája a következő: Először röviden áttekintem a témához legközelebb álló publikációkat (2. fejezet), majd bemutatom az általam használt adatszelekciós modelleket (3. fejezet), végül ismertetem a futtatási környezeteimet (4. fejezet) és az elért eredményeimet (5. fejezet).

## 2. Kapcsolódó irodalom

A kutatók a domain adaptációval történő minőségjavítást már a statisztikai gépi fordító rendszereknél alkalmazták. Számos megoldás közül én a ModernMT [2] nevű szabadon hozzáférhető fordítórendszert szeretném kiemelni. A rendszer lényege, hogy a tanítóanyagot több részre klaszterezik és ezekből a részekből különkülön építenek modelleket. A módszernek köszönhetően minden mondatot a hozzá legjobban hasonló szegmensekből épített modellel lehet fordítani, ezzel érve el a legjobb fordítási minőséget.

A Chatterjee et al. [3] adaptálták a fenti technikát NMT rendszerre. Rendszerük egy előre tanított generikus engine-en alapul. Minden egyes fordítandó mondat alapján kikeresik a tanítóanyagból a hozzá leginkább hasonló szegmenseket, amikkel tovább tanítják az alap generikus engine-t, ezzel optimalizálva a rendszert az adott mondathoz. A módszer nehézsége, hogy minden fordítandó mondat előtt tanítási ciklust kell végezni, ami nagyban lelassítja a fordítási folyamatot.

A témával kapcsolatban az egyik legfrissebb publikációt Silva et al. [4] készítették. A legnagyobb különbség kettőnk módszere között a megvizsgált adatszelekciós módszerekben, valamint a rendszerek összeállításában figyelhető meg.

[4] http://opus.nlpl.eu/

Az általuk használt subword-alapú modell hátránya, hogy gyakran rontja el a tanítóanyagban ritkán szereplő szavak fordítását, vagy helyesírását ezért ebben a kutatásban szóalapú modellt használtam. További különbség a felhasznált NMT keretrendszer is, ahol ők a MarianNMT-t [5] használták.

## 3. Adatszelekciós módszerek

Annak érdekében, hogy egy jó minőségű *in-domain* NMT rendszert hozzunk létre célszerű a nagyméretű általános tanítóanyagból kiválogatni a domainhez leginkább hasonló szegmenseket. Fontos kérdés a megfelelő adatszelekciós módszer alkalmazása, mivel ez jelentősen befolyásolja a végleges rendszer minőségét. A megfelelő módszer kiválasztásánál fontos szempont volt a minőség mellett az adott módszer sebessége is, mivel ezt a technikát egy ipari célú rendszerbe integráltam. Annak érdekében, hogy a feladathoz leginkább alkalmas szelekciós módszert alkalmazzam, megvizsgáltam több különböző megközelítést is.

Kézenfekvő és viszonylag könnyen implementálható módszernek számít a **TF-IDF** módszer [6], amely a szövegfeldolgozás egyik gyakran alkalmazott algoritmusa. A módszer lényege, hogy az *in-domain* dokumentumban szereplő szegmensekből kigyűjti a legjellemzőbb szavakat (nem stopword-ök) és ezek segítségével osztályozza az *out-of-domain* szegmenseket. A metódus alkalmazásának több hátulütője ismert. Egyrészt nehéz hozzá erőforrás- és futásidőbarát implementációt készíteni. Másrészt pedig csak kis mértékben korrelál az emberi értékeléssel.

Napjainkban a TF-IDF módszer helyett a szakirodalomban főleg **szóbeágyazási modell-alapú** szelekciót javasolnak [7,8,9]. A módszer minősége nagymértékben meghaladja a TF-IDF technikát, mivel a dokumentumok/szegmensek osztályozásához nemcsak karakter szinten veszi figyelembe a szavakat, hanem a vektoros reprezentációnak köszönhetően az indexált szavak környezetből származó információit is tartalmazza. A metódus hátránya azonban, hogy nem nyelvfüggetlen; a modell betanításához egy viszonylag nagyméretű egynyelvű tanítóanyagra van szükség, ami a legtöbb nyelv esetén nem áll rendelkezés. Ebből kifolyólag ezzel a módszerrel nem végeztem méréseket ebben a dolgozatban.

Választásom a **perplexitás-alapú** hasonlóság vizsgálatra esett. A módszer lényege, hogy az *in-domain* anyagból nyelvenként létrehoz egy nyelvmodellt (LM), majd az elkészült nyelvmodellek alapján az *out-of-domain* korpusz szegmenseihez az 1. egyenletben szereplő képlet alapján kiszámolja a perplexitás értékeket

$$10^{-\frac{1}{N}\sum_{i=0}^{N}\log_{10} p(x_i)} \tag{1}$$

, ahol a $p(x_i)$ az $i$. szó nyelvodellből számolt valószínűsége. Tehát a párhuzamos korpusz szegmenspárjaihoz két perplexitás értéket rendel, a végső pontszámot a két perplexitás érték átlagából kapja meg. Ezen érték alapján rangsorolható az *out-of-domain* tanítóanyag szegmenspárjai. Munkám során a rangsorolt tanítóanyagból vágtam ki a vizsgált korpuszméreteket.

Munkám során két különböző nyelvmodell rendszert vizsgáltam. Elsőként a KenLM [10] nevű nyelvmodellező rendszert, ami szógyakoriság alapon épít fel egy

n-gram modellt. Az eszköz egy c++ nyelven írt szabad felhasználású program, mely mind időben mind erőforrásigényben erősen optimalizált, illetve tetszőleges méretű tanítóanyagból is képes jó minőségű modellt építeni. A KenLM segítségével egy 5-gram alapú modellt hoztam létre. A másik alkalmazott eszköz az RNNLM [11] volt, mellyel egy rekurrens neurálishálózat-alapú nyelvmodellt tanítottam be. Fontos kérdés, hogy a viszonylag kisméretű *in-domain* tanítóanyag elégséges-e a neurális hálózat betanítására, mivel a tanítás során nem használtam extra külső tanító anyagot.

## 4. Kísérleti környezet leírása

### 4.1. Tanítóanyag összetétele

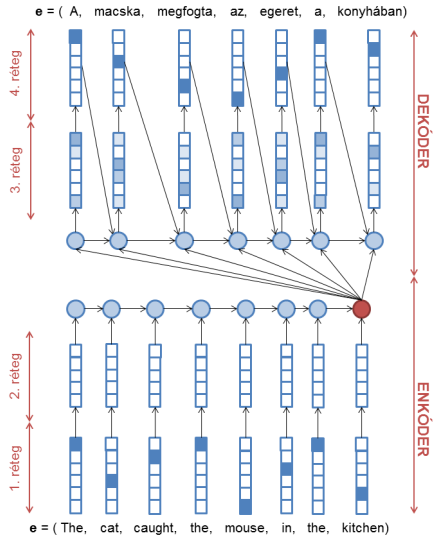
Mivel kutatásomban egy kereskedelmi fordítási környezet minőségének javítását tűztem ki célul, így a rendszerek betanításához fordítócégek témaspecifikus fordítómemóriáit használtam. A méréseket 3 különböző nyelvpáron végeztem el. Az angol-német és az angol-francia nyelpárok mellett a japán-angol nyelvpárral vizsgáltam a szelekciós módszer hatását nyelvtanilag távolabbi nyelvpár esetén is. A *in-domain* korpuszméret megváltoztatásával az *out-of-domain* korpusz méretéből fakadó dominanciájának hatása csökkenthető, valamint vizsgálható a szelekciós módszer tanulási minősége is. A mérések során három különböző méretű *in-domain* korpuszméretet alkalmaztam: 25K, 50K, 100K szegmenspárok. Mindegyik rendszer esetén a tanítóanyagból véletlenszerűen elkülönítettem 1000 szegmenst tesztelés és 3000 szegmenst validációs halmaz céljából. Mind a három esetben más domaint választottam: az angol-francia esetben informatikai, japán-angol esetben orvosi szöveg, míg angol-német nyelvpár esetén ipari dokumentáció témájú tanítóanyagokat alkalmaztam. A német és a francia esetben *out-of-domain* tanítóanyagként a jogi szövegeket tartalmazó Európai Parlamenti Jogszabályok Gyűjteményét[5] (DGT) használtam, míg japán esetben az ügyfél saját IT témájú fordító memóriáját. Annak ellenére, hogy az eredmények közvetlenül nem reprodukálhatóak, hasonló környezet előállítható szabadon hozzáférhető korpuszok segítségével, mint például az EMEA[6] (orvosi dokumentumok) vagy az OpenSubtitles[7] (filmfeliratok) korpuszok.

### 4.2. Gépi fordítórendszer bemutatása

Célom a gépi fordítás minőségének javítása volt, amihez az OpenNMT [12] keretrendszert használtam. Az OpenNMT a Harvard egyetem valamint a Systran cég közös munkája. Egy Lua nyelven íródott gépi fordító keretrendszer, melybe több modellt is implementáltak. Munkám során a figyelmi modellel kiegészített [13] RNN-alapú enkóder-dekóder architektúrájú modellt használtam [14,15]. A modell lényege, hogy kettéválasztja a fordítás folyamatát két elkülöníthető részre.

[5] http://opus.nlpl.eu/DGT.php
[6] http://opus.nlpl.eu/EMEA.php
[7] http://opus.nlpl.eu/OpenSubtitles2018.php

Az enkódolás során lényegében egy RNN-alapú seq2seq modellt hoz létre, tehát a szóbeágyazási modellhez hasonlóan a fordítandó modellekből egy n-dimenziós vektort készít. Az 1. ábrán ez a vektor felel meg az ábra közepén látható piros/sötét node-nak. A második fázis a dekódolás, ahol a mondatvektorból generálja ki a célnyelvi mondatot egy RNN réteg segítségével.




1. ábra: Enkóder-dekóder architektúra vázlatos rajza

Ez az architektúra a transformer-alapú modell megjelenéséig a piacvezető modellnek számított. Munkám során azért nem a transformer-alapú modellt használtam, mert az eddigi méréseim alapján nem sikerült mérhetően jobb minőséget produkálni vele. A jövőben szeretném figyelemmel kísérni ennek a technológiának a fejlődését is és megtalálni az optimális paraméter értékeket.

Méréseim során a tanítóanyagokon a gépi fordítás során általánosan használt előfeldolgozási lépéseken (tokenizálás, truecasing) kívül a szótárméret csökkentése érdekében a tanítóanyagban szereplő számokat és dátumokat placeholderekre cseréltem. További fontos különbség az általános architektúrához képest, hogy nem alkalmaztam a BPE (Byte pair encoding) technológiát [16], hanem 100 ezer elemben limitált szóalapú rendszert tanítottam be. Erre az aktuálisan rendszerben lévő fordítási környezet miatt volt szükség. Az általam használt neurálishálózat belső paraméterei megegyeznek az OpenNMT rendszer default paraméter értékeivel[8].

[8] http://opennmt.net/OpenNMT/options/train/

## 5. Eredmények és kiértékelés

Munkám során az általánosan alkalmazott automatikus kiértékelő metrikát a BLEU [17] módszert használtam. Munkám során a gépi fordítás során általánosan alkalmazott implementációt[9] használtam alapértelmezett paraméterértékek mellett. Annak ellenére, hogy köztudottan alacsonyabb a módszer korrelációja az emberi kiértékeléshez képest [18,19,20], továbbra is alkalmazzák, mivel eddig még nem sikerült ennél megbízhatóbb mérési módszert alkotni a fordítás kiértékeléséhez. Általánosan elfogadott vélemény, hogy a BLEU-ben mért statisztikailag szignifikáns különbségű rendszerek az emberi kiértékelés során is jobban teljesítenek.

| | Nincs válogatás | KenLM | KenLM +tuning | RNNLM | RNNLM +tuning |
|---|---|---|---|---|---|
| In-domain(25K) | 7,32% | | | | |
| Out-of-domain (3M) | 39,93% | | | | |
| Out-of-domain(3M)+tuning(25K) | 56,43% | | | | |
| In-domain(25K)+Out-of-domain(0,5M) | | 58,71% | **63,52%** | 58,52% | 63,24% |
| In-domain(25K)+Out-of-domain(1M) | | 58,59% | 62,60% | 58,43% | 62,57% |
| In-domain(25K)+Out-of-domain(2M) | | 58,58% | 62,32% | 58,37% | 62,25% |
| In-domain(25K)+Out-of-domain(3M) | | 58,58% | 61,32% | 58,20% | 61,09% |

1. táblázat. A táblázat az EN→FR (IT(25K)+DGT(3M) domain) fordítási irányba mért BLEU értékeit mutatja.

Az eredményeket az *in-domain* korpusz mérete alapján rendeztem és ez alapján fogom bemutatni. A legkisebb tanítóanyaggal az angol-francia nyelvpárú rendszer rendelkezik. Az 1. táblázatból látszik, hogy a pusztán $25K$ szegmensen tanított rendszer csupán $7,32\%$ BLEU pontosságot ért el. Ez annak tudható be, hogy a neurálishálózat-alapú modelleknek sokkal több tanítóanyagra van szüksége az optimális működéshez. Ebben az esetben ezt a baseline rendszert a csupán *out-of-domain* anyagon ($3M$) tanított rendszer messze túlhaladja ($\sim 40\%$). Ez a rendszer tekinthető egy általánosan használható generikus modellnek, amit tetszőleges szöveg fordítására lehet használni. Az eredmény tovább javul ($56,43\%$), ha az *out-of-domain* anyagból létrejött modellt a $25K$ *in-domain* anyaggal tovább tanítjuk. A továbbiakban ezt a lépést tuningnak fogom nevezni.

A táblázat második részében az *in-domain* anyag bővítésével létrehozott rendszerek eredményei olvashatók. Először a KenLM majd az RNNLM rendszerekkel tanított nyelvmodell-alapú osztályozók eredményei láthatók. Mind a két esetben tuningolást is végeztem. A táblázatokból kiolvasható, hogy a statisztikai módszerrel tanított nyelvmodell segítségével minden esetben jobb minőségű rendszer jött létre, mint a neurálishálózat-alapú módszer esetében. Ennek az lehet az oka, hogy a $25K$ tanítóanyag kevésnek bizonyul a neurális háló

[9] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl

tanításához. Ez a tendencia a továbbiakban is megmarad, ezért a későbbi táblázatokban ez az oszlop már nem fog szerepelni. A legmagasabb eredményt a $25K + 0,5M + tuning$ rendszer érte el messze túlszárnyalva a generikus rendszer $(3M + tuning)$ eredményét, ami azt jelenti, hogy jelentős javulás érhető el, ha a tanító halmazt az *in-domain* tanítóanyaghoz hasonló szegmensekkel egészítjük ki, majd a végén az *in-domain* anyaggal tuningolást végzünk. A BLEU-ben mért minőségjavulás mellett további nyereségnek tekinthető, hogy a generikus rendszerhez képest csökkentett tanítóanyagon tanult rendszer nagyságrendekkel kisebb futásidő alatt éri el a jobb minőséget.

| | Nincs válogatás | KenLM | KenLM +tuning |
|---|---|---|---|
| In-omain(44K) | 63,04% | | |
| Out-of-domain (3M) | 25,64% | | |
| Out-of-domain(3M)+tuning(44K) | 62,11% | | |
| In-domain(44K)+Out-of-domain(0,5M) | | 69,85% | 73,33% |
| In-domain(44K)+Out-of-domain(1M) | | 70,3% | **74,5%** |
| In-domain(44K)+Out-of-domain(2M) | | 69,80% | 73,84% |

2. táblázat. A táblázat a JA→EN (Medical(44K)+IT(3M) domain) fordítási irányba mért BLEU értékeit mutatja.

A 2. és a 3. táblázatokból is hasonló eredmények olvashatók ki. A legfontosabb különbség a generikus és a pusztán *in-domain* rendszerek eredményei között figyelhető meg. Ezekben az esetekben az *in-domain* anyag magasan túlszárnyalja a pusztán generikus modell eredményét, míg a tuningolt generikus rendszer is csak megközelíteni tudja ezt a minőséget. Ez annak tudható be, hogy az *in-domain* anyag hasonló és jó minőségű fordításokból áll, melynek köszönhetően az NMT rendszer az $50 - 100K$ méretű tanítóanyag segítségével is képes volt 50%-ot meghaladó fordítási minőséget produkálni. Mindkét esetben a válogatással kiegészített és tuningolt rendszerek statisztikailag szignifikáns minőségjavulást értek el.

| | Nincs válogatás | KenLM | KenLM +tuning |
|---|---|---|---|
| In-domain(100K) | 48,21% | | |
| Out-of-domain (3M) | 37,58% | | |
| Out-of-domain(3M)+tuning(100K) | 49,71% | | |
| In-domain(100K)+Out-of-domain(0,5M) | | 52,65% | **58,47%** |
| In-domain(100K)+Out-of-domain(1M) | | 51,88% | 57,32% |
| In-domain(100K)+Out-of-domain(2M) | | 50,75% | 56,98% |
| In-domain(100K)+Out-of-domain(3M) | | 49,71% | 56,12% |

3. táblázat. A táblázat az EN→DE (documentation(100K)+DGT(3M) domain) fordítási irányba mért BLEU értékeit mutatja.

A bemutatott eredmények tükrében a következő konklúziók vonhatóak le: 1.) Ha nem áll rendelkezésünkre jó minőségű *in-domain* tanítóanyag, akkor kénytelenek vagyunk a generikus *out-of-domain* anyagon tanított rendszert használni. 2.) Ha rendelkezésünkre áll bármekkora méretű *in-domain* tanítóanyag, a létező generikus modellünket tuning segítségével rá tudjuk hangolni erre a domainre, így sokkal jobb minőségű fordítás érhető el viszonylag rövid időn belül. 3.) A legjobb eredmény az *in-domain* tanítóanyag kiegészítésével és a tanítás végi tuninggal érhető el. Ezen architektúrák segítségével szignifikáns minőségjavulás érhető el a fordítás során.

A bemutatott eredményeket alátámasztják az ügyfeleink visszajelzései is, akik jelentős mértékben az *in-domain+out-of-domain+tuning* rendszer értékelték a legjobbnak és többször is megerősítették, hogy jelentősen jobb minőségű fordítást állítunk elő, mint a pusztán *out-of-domain* anyagon tanított generikus enginekkel értek el.

## 6. Összegzés

A fordítócégek többségére jellemző, hogy csupán kis méretű viszonylag jó minőségű fordítómemóriákkal rendelkeznek, melyek általában valamilyen speciális témakörből származnak. A korpusz méreténél fogva nem képes stabilan jó minőségű NMT fordítórendszer betanítására, mivel az nagyon érzékeny lesz a domaintől való eltérésre. Munkám során adatszelekció segítségével kiegészítettem a kisméretű *in-domain* tanítóanyagokat nagyobb *out-of-domain* tanítóanyagból válogatott szegmensekkel, így jelentősen sikerült javítani a fordítórendszer minőségét. Megállapítottam, hogy a túl kevés tanítóanyag esetén ajánlatos az elérhető összes *out-of-domain* anyaggal betanított rendszert az *in-domain* anyaggal továbbtanítani, míg valamivel nagyobb rendszer esetén az adatszelekcióval történő korpuszkiegészítés a célravezető.

## Köszönetnyilvánítás

Ezúton is szeretném megköszönni a Morphologic Lokalizáció Kft. támogatását, hogy biztosította korpuszainak használatát kutatásom elvégzéséhez.

## Hivatkozások

1. Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Monz, C.: Findings of the 2018 Conference on Machine Translation (WMT18). In: Proceedings of the Third Conference on Machine Translation, Belgium, Brussels, Association for Computational Linguistics (2018) 272–307
2. Nicola, B., Roldano, C., Mauro, C., Amin, F., Marcello, F., Davide, C., Luca, M., Andrea, R., Marco, T., Ulrich, G., David, M.: MMT: New open source MT for the translation industry. In: Proceedings of The 20th Annual Conference of the European Association for Machine Translation (EAMT), Copenhagen, Denmark, Association for Computational Linguistics (2017) 86–91

3. Chatterjee, R., Negri, M., Turchi, M., Federico, M., Specia, L., Blain, F.: Guiding Neural Machine Translation Decoding with External Knowledge. In: Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers, Copenhagen, Denmark, Association for Computational Linguistics (2017) 157–168
4. Silva, C.C., Liu, C.H., Poncelas, A., Way, A.: Extracting In-domain Training Corpora for Neural Machine Translation Using Data Selection Methods. In: Proceedings of the Third Conference on Machine Translation, Belgium, Brussels, Association for Computational Linguistics (2018) 224–231
5. 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, Melbourne, Australia, Association for Computational Linguistics (2018) 116–121
6. Salton, G., Yang, C.S.: On the specification of term values in automatic indexing. Journal of Documentation **29**(4) (1973) 351–372
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. CoRR **abs/1607.01759** (2016)
9. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and Word2vec for text classification with semantic features. In: 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC). (2015) 136–140
10. Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P.: Scalable Modified Kneser-Ney Language Model Estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria (2013) 690–696
11. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010 **2** (2010) 1045–1048
12. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. ArXiv e-prints (2017)
13. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014)
14. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR **abs/1406.1078** (2014)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR **abs/1409.3215** (2014)
16. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. CoRR **abs/1508.07909** (2015)
17. 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 on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 311–318
18. Tantug, A.C., Oflazer, K., El-Kahlout, I.D.: BLEU+: a tool for fine-grained BLEU computation. In: LREC 2008. (2008)
19. Callison-Burch, C., Osborne, M., Koehn, P.: Re-evaluating the role of BLEU in machine translation research. In: In EACL. (2006) 249–256
20. Banerjee, S., Lavie, A.: METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop

on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics (2005) 65–72

# Egy magyar nyelvű kérdezőrendszer

Novák Attila[1,2], Laki László János[1,2], Novák Borbála[1,2], Dömötör Andrea[2,3], Ligeti-Nagy Noémi[2,3], Kalivoda Ágnes[2,3]

[1]Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar
[2]MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
Budapest, Práter u. 50/a.
[3]Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar
2087 Piliscsaba, Egyetem u. 1.
{vezetéknév.keresztnév}@itk.ppke.hu

**Kivonat** Cikkünkben egy folyamatban lévő kutatásról számolunk be, amelynek keretében olyan korpuszannotációt hozunk létre, amely alkalmas a feldolgozott szöveggel kapcsolatban releváns kérdéseket megfogalmazni képes elemzőrendszer betanítására. A cikk terjedelmi korlátai által biztosított határok között röviden bemutatjuk a kutatás célkitűzéseit, a kiindulásul használt magyar UD korpusz javításával, a tematikus vonzatkeret-lexikon létrehozásával, a szabad határozók osztályozásával és a vonzatkeretek korpusz-előfordulásokra való illesztésével kapcsolatos eddigi erőfeszítéseinket.

## 1. Bevezetés

Az utóbbi években a korábbiakat meghaladó színvonalú eredményeket nyújtó módszernek bizonyult a neurális mélytanuló hálózatokon alapuló olyan ún. end-to-end rendszerek alkalmazása, amelyek semmilyen grammatikai elemzést nem tartalmaznak, ezért kétségek merültek fel azzal kapcsolatban, hogy van-e értelme egyáltalán grammatikai elemzéssel foglalkozni. Ugyanakkor az end-to-end rendszerek betanítása rendszerint hatalmas mennyiségű tanítóanyagot igényel, amely a legtöbb nyelven nem áll rendelkezésre. Ezért azt gondoljuk, hogy továbbra is lehet értelme egy grammatikai elemzést előállító rendszernek, amennyiben az elemzés eredménye közvetlenül felhasználható olyan feladatok végrehajtásához, amely a hétköznapi felhasználók számára is relevanciával bír.

Nem lehetünk elégedettek azonban egy olyan elemzéssel, amely olyan teljesen absztrakt kategóriákkal dolgozik, amelyeket nem lehet egyértelműen olyan fogalmakra lefordítani, ami hétköznapi emberek számára is érthető módon összefüggésbe hozható azzal, hogy mit jelent az adott szöveg. A szövegértés lényeges eleme, hogy képesek vagyunk értelmes kérdéseket feltenni az adott szöveggel kapcsolatban, és ez a képességünk szorosan összefügg azzal, hogy képesek vagyunk kérdésekre válaszolni is. Olyan elemzőrendszer létrehozását tűztük ki tehát célul, amely ténylegesen alkalmas arra, hogy releváns kérdéseket tegyen fel azzal a szöveggel kapcsolatban, amit feldolgoz. Ehhez számtalan olyan distinkcióra van

szükség, amiknek az eddigi elemzőrendszerekben nem láttuk nyomát. Jelen cikk ennek a munkálatnak az első fázisát mutatja be, amelyben célunk egy olyan annotált korpusz létrehozása, ahol az annotáció tartalmazza mindazokat a jegyeket, amik az adott szöveggel kapcsolatos kérdések generálásához szükségesek.

## 2. A hagyományos elemzés hiányosságai

Mivel olyan rendszer létrehozása a célunk, amely értelmes kérdéseket tud feltenni, ezért úgy döntöttünk, hogy az annotációban használt megkülönböztetések létjogosultságát alapvetően az határozza meg, hogy az adott konstrukcióval kapcsolatban milyen kérdéseket lehet föltenni. A **névszói csoportokra** vonatkozó kérdéseknél például alapvető a *ki?/mi?* megkülönböztetés, ezért a rendszernek pontosan meg kell tudnia különböztetni a személyeket a dolgoktól. Ugyanakkor a csoportokra vagy szervezetekre attól függően kérdezünk *ki?*-vel vagy *mi?*-vel, hogy milyen szerepet töltenek be az adott mondatban. Egy bank például nyelvileg személyként viselkedik, ha számlalevelet küld, de dologként, ha felszámolják. Az állítmányként használt névszói csoportokkal kapcsolatos kérdések generálásához pedig egy még ennél is jóval részletesebb osztályozásra van szükség. A *Lajos orvos* mondattal kapcsolatban a *Lajos ki?* kérdés nem túl kifinomult, a *Lajosnak mi a foglalkozása?* jóval pontosabban kérdez rá arra, ami a mondatban az állítás. A fogalmak foglalkozásként, állatként, eszközként, viselkedésként, stb. való osztályozása a névszói csoportok nem predikatív előfordulásaival kapcsolatban is jóval specifikusabb kérdések megfogalmazására ad lehetőséget: pl. *Milyen állatot láttál a kertben?* szemben a *Mit láttál a kertben?* kérdéssel. Különösen lényeges ez a koordinált frázisok esetében, ahol az egyik koordinált összetevőre csak akkor tudunk a kérdezett számára is azonosítható módon rákérdezni, ha a kérdés eléggé specifikus.

A **határozókkal** kapcsolatos kérdések megfogalmazásához is nagyságrendekkel részletesebb osztályozásra van szükség még a legminimálisabb szinten is, mint amivel a létező hagyományos elemzőrendszerek szolgálni tudnak. Az inesszívusz ragos szóalakok például rengeteg különböző funkciót tölthetnek be, és így különböző kérdés tartozik hozzájuk:

- *szeptemberben*: mikor?,
- *Londonban*: hol?,
- *fájdalmában*: mitől?,
- *magában (bízik)*: kiben?,
- *hármasban*: hányan?,
- *elemében (van)*: erre nem kérdezünk,
- stb.

Az **állítmánnyal kapcsolatos kérdések** megfogalmazása nemcsak a névszói állítmányok, hanem az igék esetében is olyan ismereteket igényel, amelyekkel a létező grammatikai leírások nem tudnak szolgálni. Hogy hogyan kérdezzünk az állítmányra annak egy adott vonzatát horgonyként használva, az attól függ, hogy az adott vonzat milyen tematikus szerepet tölt be az igei vonzatkeretben. A *Mit*

*csinált Jancsi Ferivel?* adekvát kérdés, ha *Jancsi* ágens és *Feri* páciens. Ugyanebben a helyzetben a *Mi történt Ferivel?* és a *Mit csinált Jancsi?* ugyanígy helyes kérdés.

A vonzatkeretek argumentumhelyeinek tematikus osztályozására szükség van az **oblikvuszi vonzatok és a szemantikailag tartalmas viszonyok** megkülönböztetéséhez is. Például: *bízik valamiben* szemben azzal, hogy *van valahol.*

Szükség van ugyanakkor a félig kompozicionális, illetve **idiomatikus szerkezetek** kompozicionális szerkezetektől való megkülönböztetésére is. Vicc lesz belőle, ha az előbbiekre kérdezünk:

- *Mit hozott Édesapám?*
- *Döntést.*

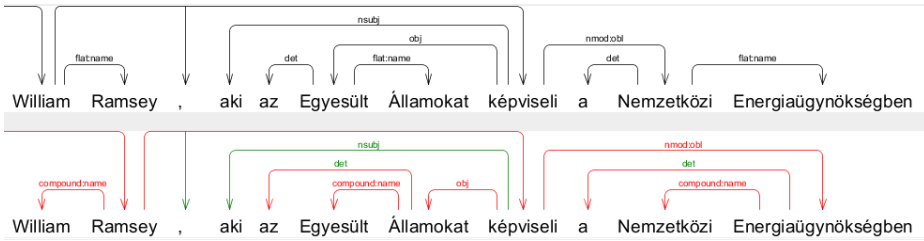
## 3. A korpusz

Kiindulási anyagként a Universal Dependencies (UD) korpusz [1] 1800 mondatból (42000 token) álló magyar alkorpuszát választottuk, hogy nemzetközi szinten is értelmezhető kontextusba helyezzük az általunk javasolt annotációs sémát. Az UD korpusz nagyjából egységes elvek és kategóriák felhasználásával sok nyelv szövegeire tartalmaz morfoszintaktikai és szintaktikai függőségi elemzést. Eredeti tervünk az volt, hogy a magyar UD korpuszban szereplő annotációt pusztán kiegészítjük, illetve finomítjuk a kérdések megfogalmazásához szükséges információkkal. Kiderült azonban, hogy a magyar alkorpuszban szereplő annotáció sok szempontból nem felel meg az érvényes UD specifikációnak, illetve sok véletlenszerű annotációs hibát tartalmaz, ezért a feladat része lett ezeknek a hibáknak a javítása.

Az UD 2.0 specifikációja[1] szerint a **több szavas kifejezések** belső szerkezetét `flat, fixed` vagy `compound` függőségi viszonyok alkalmazásával kell leírni. A `fixed` viszonyt kizárólag a teljesen megkövült funkciószó-szerű több szavas kifejezések leírására használják. A `compound` viszonyt kell használni azoknak a szerkezeteknek a leírására, amelyeknek van feje. Számos nyelvben, például az angolban, a több szavas neveket általában lapos endocentrikus szerkezeteknek tekintik, ezért a `flat` viszony használatát javasolják ezeknek a neveknek a leírására. Az UD 2.0 annotációs specifikációja azonban kategorikusan kizárja ennek a típusú elemzésnek a használatát azokban az esetekben, amikor a névnek szabályos szintaktikai szerkezete van (pl. címek, illetve az intézménynevek nagy része), ahol a szokásos szintaktikai viszonyok használatát írja elő, illetve az endocentrikus szerkezetű nevek esetében, ahol a `compound` viszonyt, illetve ennek valamelyik alváltozatát kell használni. A magyar névszói szerkezetek mindig endocentrikus szerkezetek, amelyek rendszerint jobb fejűek, ezért a nem szabályos szerkezetű és kompozicionális jelentésű nevek esetében a magyarban mindig a `compound` viszonyt kell használni. Ez biztosítja például, hogy a mindig a szerkezet fején megjelenő esetragok közvetlenül elérhetőek legyenek. Ezért a feldolgozás egyik lépéseként a korpuszban hibásan `flat` szerkezetűnek annotált több szavas

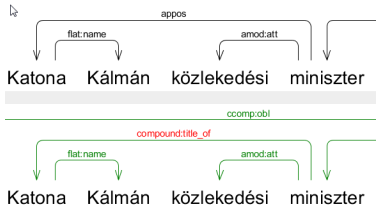
[1] `http://universaldependencies.org/guidelines.html`

neveket automatikusan `compound` szerkezetekké konvertáltuk. Egyelőre elmaradt a teljesen szabályos szerkezetű nevek konverziója, hiszen ezeket kézzel kellene kiválogatni és újraannotálni (1. ábra).




1. ábra. A nevek annotációjának javítása

A tévesen jobb fejű appozitív szerkezetként annotált *Katona Kálmán közlekedési minisztert*-típusú szerkezetekben[2] az UD 2.0 specifikációval kompatibilis módon `compound:title_of` viszonyt vettünk fel a név és a foglalkozás/funkció között (2. ábra).



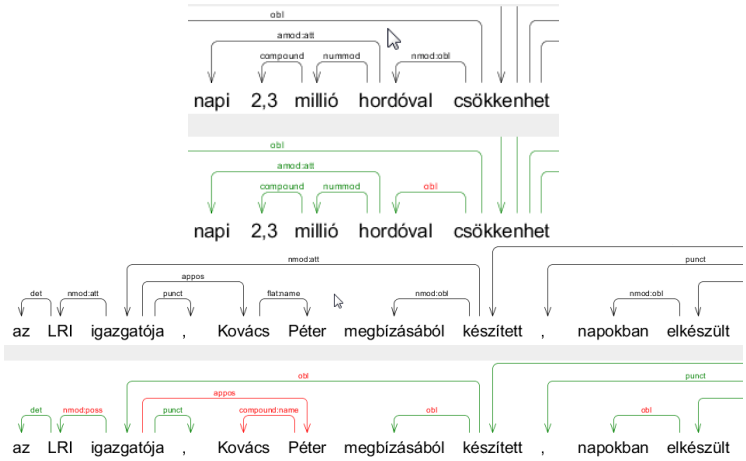

2. ábra. Név és foglalkozás javítása

Az alanyon, tárgyon és részeshatározón kívüli **névszói vonzatok** jelölésére az UD 2.0 specifikáció az `obl` relációt írja elő akkor is, ha a fej nem ige. Ez a korpuszban sokszor igei fejek esetén sem így szerepelt. Igei és igenévi fejek esetében tudtuk automatikusan javítani ezeket a annotációkat – amennyire lehetett (3. ábra).

Az **igekötős igék** lemmája nem tartalmazta az igekötőt azokban az esetekben, ahol az ige és az igekötő nem volt egybeírva. A vonzatok tematikus szerepeit tartalmazó lexikonban szereplő annotáció korpuszra vetítéséhez szükséges volt, hogy az igekötő része legyen ezekben az esetekben is a lemmának. Ezért ezt a hibát is kijavítottuk.

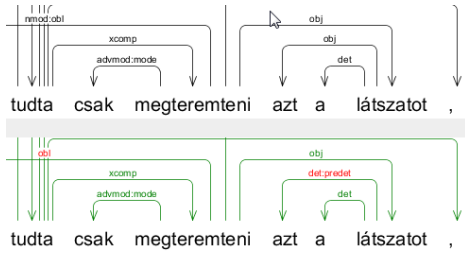
Az *azt a kutyát*-típusú **egyeztetett predeterminánst** tartalmazó szerkezetekben a mutató névmás sokszor tévesen ugyanazzal a címkével volt a névszói

[2] Az appozitív szerkezetekben esetegyeztetés van a két elem között, itt erről nincs szó.




3. ábra. Az `obl` reláció javítása igei és igenévi fejeknél – a második esetben az *igazgatója* szó rossz fejhez volt kötve, így az annotáció továbbra is hibás maradt

csoport fejéhez csatolva, mint amilyen funkciót a teljes NP betölt. Ezeket és az összes ilyen predeterminánsz címkéjét `det:predet` címkére cseréltük (4. ábra).

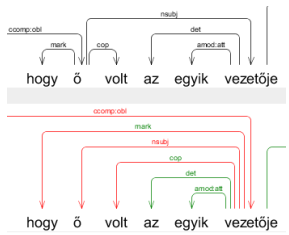



4. ábra. Hibásan annotált mutató névmás javítása

A **birtokos szerkezetekben** a birtokos annotációját `nmod:att`-ról `nmod:poss`-ra javítottuk (l. a 3. ábrán alul).

A **névutókat** egységesen `case` viszonnyal kapcsoltuk a névszói csoport fejéhez.

A harmadik személyű **névszói állítmányt** tartalmazó tagmondatok annotációjában az alany és az állítmány sok esetben meg volt cserélve, mert a fókuszt összetévesztették az állítmánnyal. A korábbiakban leírt javításokat programozottan végeztük. Ezeket a szerkezeteket azonban kénytelenek voltunk félig manuális módszerrel javítani: kézzel jelöltük meg azokat a mondatokat, ahol aztán az alany és állítmány annotációját programozottan javítottuk (5. ábra).




5. ábra. Felcserélt alany és állítmány javítása

## 4. Vonzatkeret-adatbázis

A magyar UD korpuszban szereplő összes ige és igenév tövét kigyűjtöttük, és a [2] cikkben leírt elemzett korpuszból épített szóbeágyazási modellben szereplő vektorreprezentációjuk alapján klasztereztük a [3,4] cikkekben leírt módon. A hasonló disztribúciójú (és vonzatkeretű) igék így egy-egy klaszterben gyűltek össze. A listát kiegészítettük minden egyes igéhez a *Magyar igei szerkezetek: a leggyakoribb vonzatok és szókapcsolatok szótára* magyar vonzatkeretszótárban [5] szereplő az adott igéhez tartozó leírással. Ezt a kiinduló reprezentációt ihletforrásként használva kézzel készítettük el az egyes igék lehetséges vonzatkereteinek leírását, amelyben az egyes vonzatok tematikus szerepe, formai jegyei (esetrag, névutó, birtokos végződés, stb.), esetleges opcionalitása, és a rájuk vonatkozó esetleges lexikai/szemantikai megszorítások szerepelnek.

Az igei vonzatkeretek leírásánál a fő szempont az volt, hogy minél több olyan információt adjunk meg, amelyek segítségével a lehető legjobb, legpontosabb kérdések tehetők fel. Éppen ezért a vonzatkeret-leírásokban használt tematikusszerep-készlet, bár azokból indul ki, legfőképpen abban követi az általánosan ismert tematikusszerep-hierarchiákat, hogy részleteiben éppen úgy különbözik azoktól, mint azok egymástól. Az igék leírása igyekszik minden lehetséges jelentést (vonzatkeretet) lefedni. Az, hogy a hasonló jelentésű és vonzatkeretű igék eleve összegyűjtve szerepeltek az adatbázisban, lehetővé tette, hogy több ige közös vonzatkeretét csak egyszer kelljen megadni, és az egy csoportba tartozó igék automatikusan öröklik az így megadott vonzatkereteket. Emellett természetesen az egyes igéknek egyéb csak rájuk jellemző vonzatkeretei is lehetnek, amelyek hozzáadódnak az igecsoportra jellemző vonzatkeretekhez.

Az igéhez tartozó vonzatokat és opcionális bővítményeket szerepek szerint vagy lexikálisan adtuk meg, minden esetben a szükséges esetragokkal vagy névutókkal kiegészítve. A szerepek meghatározása aszerint történt, hogy milyen kérdés tehető fel az adott mondatrészre, illetve a mondatrésszel az igére. Például az ágens kérdése a *mit csinál?*, a páciensé pedig a *mi történik vele?*.

Bizonyos szerepek egyúttal egyfajta szemantikai kategóriát is jelölnek, ilyen például a kontent (`CONT`), amely valamilyen kifejthető tartalomra, információra utal, vagy a cselekvést - elsősorban főnévi igenevet - jelölő `ACT`. Azok a vonzatok, amelyeknek nincs meghatározott tematikus szerepe, nem igazán lehet őket

horgonyként használva az állítmányra kérdezni, a semlegesnek tekinthető téma (TH) szerepet kapták.

Az idiomatikus vagy félig kompozicionális igei szerkezetek vonzatait nem szerep szerint, hanem lexikálisan, a szó vagy lexikális kategória megadásával jelöltük. Ahol indokolt volt, ezek a szerkezetek - önálló egységként értelmezve őket - külön vonzatkeret-leírást kaptak. Így például a *sor kerül* leírását nem a *kerül* igénél adtuk meg, hanem a kifejezéshez mint külön tételhez rendeltünk saját vonzatkeretet.

Az igék és igei szerkezetek vonzataihoz rendelt tematikus szerepeket az 1. táblázat foglalja össze.

A táblázatban felsoroltakon kívül külön jelet kaptak a mozgó szereplők, így például a mozgó ágens jele az AGMV lett. A leírásoknál alapvetően abból indultunk ki, hogy egy igéhez nem tartozhat több azonos szerepű vonzat, ahol erre mégis szükség volt, ott a co- prefixszel jelöltük a társszereplőt, így például a *sétál valakivel* jelölése AG_coAG-vAl.

Az előzőek szerint leírt vonzatkeretek speciális szemantikai besorolást is kaphattak, melyek segítségével a kérdések tovább finomíthatók. Az ehhez felhasznált kategóriák a következők:

- biotünet (pl.: *izzad*)
- érzékelés (pl.: *lát*)
- érzelem (pl.: *örül*)
- feltétel (pl.: *múlik valami valamin*)
- hang (pl.: *zeng*)
- helyzet (pl.: *szorít az idő*)
- kezdet (pl. *megalakul*)
- kognitív (pl.: *egyetért*)
- kommunikáció (pl. *értesít*)
- matematikai (pl. *összead*)
- nemverbális kommunikáció (pl. *int*)
- önjáró (a mozgáshoz nem használ eszközt, pl. *lép*)
- pénzügyi (pl. *utal*)
- pusztítás (pl. *szabotál*)
- pusztulás (pl. *kiszárad*)
- természeti (pl. *esik az eső*)
- transzformáció (pl. *felgyorsul*)
- viselkedés (pl. *kikezd valakivel*)
- viszony (pl. *támogat*)

Végül, a vonzatkeretekhez tartozik egy polaritásérték is, ami azt jelzi, hogy az adott esemény a páciensre vagy experiensre nézve pozitív, negatív vagy semleges.

A 6. ábrán a *sodródik, hull, zuhan, esik* igék leírása látható a vonzatkeret-adatbázisban. A részlet elején szereplő PATMV és PATMV_PATH keret, illetve a @.-tal jelölt semleges polaritás mindegyik igére vonatkozik, az egyes igéknél +-szal jelölt keretek ezekhez adódnak hozzá. A leírásokban szereplő kerek zárójelek az opcionalitást, a szögletes zárójelek pedig a valamilyen szemantikai kategóriát meghatározó példák felsorolását tartalmazzák.

```
PATMV PATMV_PATH
@.
sodródik[IGE] +CHAR_ár-vAl +PAT_TH-bA
hull[IGE] +AG_térd-rA_(CHAR~előtt) +hó +PAT~[haj|könny]-A +PAT@-pusztulás
zuhan[IGE] +EXP_álom-bA@.biotünet
esik[IGE] +[eső|hó]@.nature +szó_CONT-rŐl@.komm +PAT_[áldozat|fogoly]-Ul_TH-nAk
  +AGPAT_[késedelem|hiba|túlzás]-bA_(TH-bAn/-vAl~kapcsolatban/-t~illetően) +CHAR_tartomány-bA
  +csorba_PAT~[jóhír|hírnév|becsület|...]-A-n +PAT_fogság-bA +EXP_pánik-bA_(ST-tŐl)@-érzelem
  +PAT_has-rA_(CAU-tŐl) +választás_CHAR-rA +PAT-nAk_baj-A +EXP-nAk_nehéz-A-rA_ST
  +AGPAT_gondolkodó-bA_(TH-rŐl/-vAl~kapcsolatban/-t~illetően) +EXP_[kísértés|révület]-bA_(ST-tŐl)
  +szégyen_PAT-vAl +PAT_teher-bA_(TH-tŐl)
```

6. ábra. Részlet a vonzatkeret-adatbázisból

| Jel | Név | Kérdés az igére | Példa |
|---|---|---|---|
| AG | ágens | Mit csinál AG? | **Feri** felmászott a fára. |
| CHAR | jellemzett | Mi jellemző CHAR-ra? | A **szaktudás** előnyt jelent. |
| ATTR | attribútum | – | A szaktudás **előnyt** jelent. |
| EXP | experiens | Mit érez/érzékel EXP? | **Feri** szereti Julit.<br>**Feri** meglátott egy fecskét. |
| PAT | páciens | Mi történt PAT-tal? | Feri megcsókolta **Julit**. |
| PATDST | páciens-célpont | Mi történt PATDST-vel?<br>Hova került PAT? | A gyerek a **falra** kente a főzeléket. |
| TH | téma | – | Feri a **megérzéseire** hagyatkozik. |
| ST | stimulus | Milyen érzést kelt ST (EXP-ben)?<br>Milyen hatást vált ki ST (EXP-ben)? | Feri szereti **Julit**.<br>Feri megijedt az **árnyékától**. |
| CONT | információtartalom | – | Feri ismertette a **tervet** Lajossal. |
| REC | recipiens | – | Feri ismertette a tervet **Lajossal**.<br>Juli kapott egy **levelet**. |
| RES | eredmény | Honnan lett RES? | Feri hajtogatott egy **repülőt**. |
| INS | eszköz | Mire használta AG INS-t? | Feri **rollerrel** jár dolgozni. |
| CAU | okozó | Mit okozott CAU?<br>Mi lett CAU következménye? | Feri **baleset miatt** késett. |
| MOT | cél | – | Feri **mérnöknek** tanul. |
| LOC | hely | Mi történt LOC-ban/-n...? | Feri megcsókolta Julit a **moziban**. |
| SRC | forrás, kiindulópont | – | Feri kijött a **szobából**.<br>Feri megkérdezte **Lajostól** az állást.<br>Juli kapott egy levelet **Feritől**. |
| DST | célpont | – | Feri bement a **szobába**. |
| HOW | mód | – | Feri **ügyesen** felmászott a fára. |
| ASPECT | tekintet | – | Feri nem áll rosszul **anyagilag**. |
| ACT | cselekvés | – | Feri rollerrel jár **dolgozni**. |

1. táblázat. A vonzatkeretek leírásához használt tematikus szerepek

A vonzatkeret-adatbázis a cikk írásakor 1574 ige 5394 különböző vonzatkeretét tartalmazza valamennyi vonzat tematikus szerepével együtt. Bár az opcionális vonzatokat tartalmazó keretek (pl. `olvas AG_(HOW)_(PAT-t)_(REC-nAk)_(TH-rŐl)_(LOC-bAn)`) a gyakorlatban számtalan látszólag különböző szerkezetként jelennek meg, az előbbi számot úgy kaptuk, hogy az opcionális vonzatokat és az esetleges tematikusszerep-variánsokat tartalmazó kereteket egy keretnek számoltuk.

## 5. A szabad határozók szerepének azonosítása

Fontos feladat a mondatban hagyományosan „szabad határozóként” emlegetett esetragos névszók szerepének pontosabb meghatározása is. Ha ugyanis az esetragok felől közelítjük meg a kérdést, első közelítésben azt mondhatnánk, hogy az inesszívuszi esetragot magán viselő névszó valamilyen helyviszonyt jelöl, és a *Hol?* kérdésre válaszol. A *Hol diplomázott Fanni?* kérdésre azonban vicc az a válasz, hogy *Álmában.* Nyilvánvaló, hogy az irányhármasságot kifejező, *Hol?*, *Hová?* és *Honnan?* kérdésre válaszoló 3-3-3 esetrag (inesszívuszi *-bAn*, adesszívuszi *-nÁl*, szuperesszívuszi *-On*; illatívuszi *-bA*, allatívuszi *-hOz* és szublatívuszi *-rA*; illetve az elatívuszi *-bÓl*, ablatívuszi *-tÓl* és delatívuszi *-rÓl*) nem minden esetben a hely, a forrás vagy a cél megjelölésére szolgál. Ezért a szótő kategóriájának és az esetragnak a kombinációjával határoztuk meg az egyes szóalakok szerepét.

A feladat megfogalmazható úgy is, hogy határozókat csoportosítunk: vannak természetesen helyhatározók, mint a *sarkon*, vagy a *bankban*, vannak időhatározók, mint a *télen*, *decemberben*. De persze találkozunk időtartam-határozókkal is, mint az *Öt hónapra béreltük a lakást.* mondatban a *hónapra*. Összesen 31 főkategóriát állapítottunk meg, amelyek közül némelyik több alkategóriára osztható. Alkategóriákkal együtt 51 csoportba osztottuk a korpuszban található, helyhatározói esetraggal szabad bővítményi státuszban álló szótöveket. Az alkategóriák szemléltetésére a valóban helyhatározást szolgáló, *loc* kategóriába sorolt töveket hozzuk.

| kategória | példa | bAn | nÁl | On |
|---|---|---|---|---|
| loc all | *szekrény* | hol | hol | hol |
| loc ade | *Microsoft* | miben | hol | min |
| loc ine | *állam* | hol | minél | min |
| loc sup | *címoldal* | miben | minél | hol |
| loc ine-sup | *könyv* | hol | minél | hol |
| loc city-ine | *Altenkirchen* | hol | hol | melyik városon |
| loc city-sup | *Kaposvár* | melyik városban | hol | hol |
| loc country | *Afganisztán* | hol | hol | melyik országon |

A táblázat azt mutatja, hogy az adott főkategória (jelen esetben a *loc*) adott alkategóriájába (*all*, *ine*, *city-sup* stb.) tartozó szótövek adott esetrag (*-bAn*, *-nÁl*, *-On*) esetén milyen kérdést vonnak maguk után - azaz pontosan milyen szerepük van az adott mondatban. Az irányhármasság körébe tartozó esetragos határozók osztályozásával kapcsolatos eredményeinkről részletesebben is beszámolunk a jelen kötetben megjelent másik tanulmányunkban [6].

## 6. Félig kompozicionális szerkezetek automatikus azonosítása

Az idiomatikus és félig kompozicionális szerkezetek azonosításakor is azt a célt tartottuk szem előtt, hogy egy kifejezés az arra vonatkozó releváns kérdés megfogalmazása szempontjából hogyan viselkedik. A fent említett *döntést hoz* esetén nem jó kérdés a *Mit hoz?*, a *szóba hoz* esetében a *Hova/mibe hoz?*.

Az ilyen kifejezések összegyűjtésére saját algoritmust dolgoztunk ki. Ehhez először egy 644,5 millió token méretű angol-magyar párhuzamos korpusz [7] 7-gramjaira vonatkozó szómegfeleltetési (alignment) modellt hoztunk létre fast align programmal [8] úgy, hogy minden szót egy vagy két token reprezentált mind a magyar, mind az angol oldalon: a szótő a fő szófajcímkével és az esetleges egyéb morfoszintaktikai címkék. A párhuzamos korpuszból így kinyert frázispárokból azokat vettük figyelembe, amelyeknél mind az angol, mind a magyar oldalon pontosan egy ige szerepelt. Ezekből a frázispárokból minden magyar igéhez összegyűjtöttük az összes olyan főnevet a magyar oldalról, ami az angol oldalon szereplő, a magyar igéhez kötött igéhez volt kötve. Például a *döntést hoz* kifejezés esetén a vizsgált ige a *hoz*, és ha az angol oldalon a *decide* ige szerepel, akkor a *döntést* főnév szintén ehhez az igéhez van hozzárendelve, hiszen az angol oldalon nem szerepel külön szóként. Ezzel szemben például a *táskát hoz* esetén az angol oldalon a *bring* és a *bag* is szerepel, ezek megfelelően vannak hozzárendelve a magyar megfelelőikhez. Végül az egyes magyar igékhez összegyűjtött főnevek listáját gyakoriságuk és az adott igéhez tartozó homogenitás alapján normalizáltuk és sorba rendeztük. Az így kapott lista végét levágtuk (ahol már csak olyan kifejezések gyűltek össze, amik jelentése kompozicionális). Az algoritmus kiértékeléséhez a Szeged Korpuszból és a SzegedParalell korpuszból készült félig kompozicionális igei szerkezeteket tartalmazó listát [9] használtuk, illetve a saját algoritmusunk által nem azonosított, de ezen a listán szereplő és a kérdezőrendszer szempontjából valóban releváns kifejezéseket is felvettük a vonzatkeret-lexikonunkba kiegészítve azt a vonzatkeret kompozicionális elemeivel, illetve azok tematikus szerepeivel. Az idiomatikus és félig kompozicionális igei szerkezetek párhuzamos korpusz felhasználásával történő azonosításával kapcsolatos eredményeinkről a jelen kötetben megjelent másik tanulmányunkban [10] számolunk be részletesebben.

## 7. A vonzatkeretek korpuszbeli előfordulásokra való illesztése

A vonzatkereteket az UD korpuszbeli igeelőfordulásokra illesztő algoritmus első lépésben beolvassa és szintaktikailag ellenőrzi a vonzatkeret-leírásokat tartalmazó forrásfájlokat, és az öröklődési mechanizmust alkalmazva előállítja az egyes igék teljes vonzatkeret-leírását az igecsoporthoz tartozó vonzatkeretek és a csak az adott igére jellemző leírás összeolvasztásával.

A vonzatkeret-leírásokban szereplő explicit, illetve az egyes tematikus szerepek által implikált implicit formai megszorításokat (ragok, névutók, stb.) a ma-

gyar UD korpuszban használt morfológiai és szintaktikai annotációban szereplő jegyegyüttesekre fordítjuk le, és ezek felhasználásával illesztjük a vonzatkereteket az egyes igékhez a korpuszban. A hely (`LOC`), végpont (`DST`) és kiindulópont (`SRC`) szerepű kifejezések az irányhármasságra jellemző ragokat, névutókat és névmásokat tartalmazó névszói csoportokra, illetve a megfelelő határozószókra illeszkednek. Számos ige vonzatkeretében szerepel az útvonal (`PATH`) tematikus szerep, amely a végpont, a kiindulópont és érintett hely (`VIA`) szerepek tetszőleges kombinációjával helyettesíthető. A vonzatkeretlistában a könnyebb olvashatóság érdekében a ragok a mögöttes fonológiai alakjukban szerepelnek. Az illesztőalgoritmus ezeket a leírásokat alakítja át az UD korpuszban szereplő morfoszintaktikai jegyleírások formalizmusára.

Tekintettel a magyar pro drop jellegére, a hiányzó alanyokat és tárgyakat a megfelelő helyen implicit névmásokkal helyettesítjük, ha a vonzatkeret tartalmaz ilyen vonzatot és az adott tagmondatban nem jelenik meg testes alany, illetve tárgy. Az infinitívusz és az igenevek vonzatkereteit az adott igenévtípusra jellemző transzformációval hozzuk létre az alapige vonzatkereteiből.

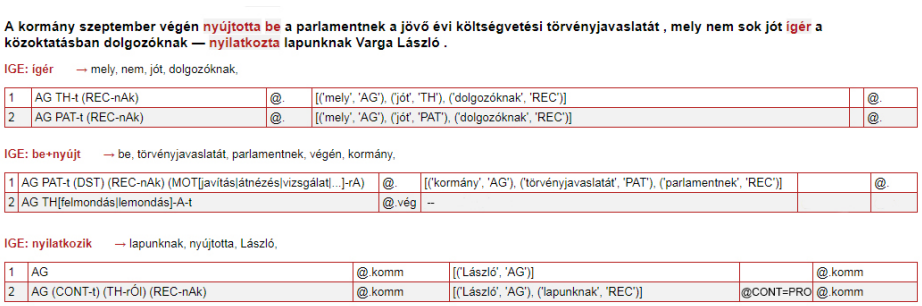
A félig kompozicionális szerkezetek egy része olyan formailag birtokos alakokat tartalmaz, amelyeknél nem a kifejezés fejét alkotó birtokjeles szóalak kapja a tényleges tematikus szerepet, hanem annak a birtokosa. Például: *a szomszédjának a nyakára küldte az adóhatóságot.* Ezeket a szerkezeteket a névutós szerkezetekhez hasonló alakúvá alakítjuk és a tényleges vonzat (*szomszédja*) lesz a módosított szerkezetben a vonzatként szereplő szerkezet feje. Ehhez már közvetlenül hozzárendelhető a megfelelő tematikus szerep.

Számos vonzatkeretben (az ige egy konkrét jelentése esetében) szemantikailag kötött típusú valamelyik argumentum. Például: `felkel [égitest]`, `átvesz [lábbeli|ruha]-A-t`. Az ilyen keretek illesztésénél a [11]-ben leírt módon morfológiailag elemzett korpuszból és lexikai szemantikai erőforrás felhasználásával épített szóbeágyazás alapú „Dologfelismerő” modellt használjuk. Ez a modell a szavakhoz lexikai szemantikai címkéket rendel. Ha az adott argumentum feje rendelkezik a vonzatkeretben meghatározott címkével, akkor a vonzatkeret illeszkedik. Például *felkel a nap*, *átveszi a tornacipőjét.*

A 7. ábrán egy minta látható arra, hogy egy adott mondat igéire milyen vonzatkeretek szerepeltek az adatbázisban, és ezek hogyan illeszkednek az adott mondatra.

## 8. Konklúzió

Cikkünkben egy olyan folyamatban lévő kutatásról számoltunk be, amelynek keretében létrehozott korpuszannotáció alkalmas a feldolgozott szöveggel kapcsolatban releváns kérdéseket megfogalmazni képes elemzőrendszer betanítására. A továbbiakban a lehetséges vonzatkeret-illeszkedések rangsorolása, a szabad határozók szerepének azonosítására szolgáló erőforrás rendszerbe illesztése, és ezek felhasználásával a kézi ellenőrzés alapjául szolgáló annotáció előállítása a célunk.



**A kormány szeptember végén nyújtotta be a parlamentnek a jövő évi költségvetési törvényjavaslatát , mely nem sok jót igér a közoktatásban dolgozóknak — nyilatkozta lapunknak Varga László .**

IGE: igér → mely, nem, jót, dolgozóknak,

| | | | | |
|---|---|---|---|---|
| 1 | AG TH-t (REC-nAk) | @. | [('mely', 'AG'), ('jót', 'TH'), ('dolgozóknak', 'REC')] | @. |
| 2 | AG PAT-t (REC-nAk) | @. | [('mely', 'AG'), ('jót', 'PAT'), ('dolgozóknak', 'REC')] | @. |

IGE: be+nyújt → be, törvényjavaslatát, parlamentnek, végén, kormány,

| | | | | | |
|---|---|---|---|---|---|
| 1 | AG PAT-t (DST) (REC-nAk) (MOT[javítás\|átnézés\|vizsgálat\|...]-rA) | @. | [('kormány', 'AG'), ('törvényjavaslatát', 'PAT'), ('parlamentnek', 'REC')] | | @. |
| 2 | AG TH[felmondás\|lemondás]-A-t | @.vég | -- | | |

IGE: nyilatkozik → lapunknak, nyújtotta, László,

| | | | | | |
|---|---|---|---|---|---|
| 1 | AG | @.komm | [('László', 'AG')] | | @.komm |
| 2 | AG (CONT-t) (TH-rÓl) (REC-nAk) | @.komm | [('László', 'AG'), ('lapunknak', 'REC')] | @CONT=PRO | @.komm |

7. ábra. Példa a vonzatok tematikus szerepeinek illesztésére a vonzatkeret-adatbázisból

## Köszönetnyilvánítás

Jelen kutatás az FK 125217 és a PD 125216 számú projekt keretében az FK 17 és a PD 17 pályázati program finanszírozásában a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással és az Emberi Erőforrások Minisztériuma ÚNKP-18–3-III-PPKE-26 kódszámú Új Nemzeti Kiválóság Programjának támogatásával valósult meg. Szeretnénk köszönetet mondani Fegyó Kingának és Bognár Ivettnek az igei vonzatkeretek és a vonzatok tematikus szerepeinek leírásában végzett munkájukért.

## Hivatkozások

1. Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: A multilingual treebank collection. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
2. Novák, A., Novák, B.: Pos, ana and lem: Word embeddings built from annotated corpora perform better. In Gelbukh, A., ed.: Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2018, Hanoi, Vietnam, Springer International Publishing, Cham. (2018)
3. Siklósi, B.: Using embedding models for lexical categorization in morphologically rich languages. In Gelbukh, A., ed.: Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, Springer International Publishing, Cham. (2016)
4. Siklósi, B., Novák, A.: Közeli rokonunk, az autó. XII. Magyar Számítógépes Nyelvészeti Konferencia (2016)
5. Sass, B., Váradi, T., Pajzs, J., Kiss, M.: Magyar igei szerkezetek: a leggyakoribb vonzatok és szókapcsolatok szótára. A magyar nyelv kézikönyvei. Tinta Könyvkiadó (2010)

6. Ligeti-Nagy, N., Novák, A.: Hol ugat a kutya? Örömében. helyhatározói esetragos névszók pontosabb annotációja. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019), Szeged, SZTE (2019)
7. Lison, P., Tiedemann, J.: Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
8. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of ibm model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2013) 644–648
9. Vincze, V.: Semi-Compositional Noun + Verb Constructions : Theoretical Questions and Computational Linguistic Analyses. PhD thesis, University of Szeged (2011)
10. Novák, A., Laki, L.J., Novák, B.: Mit hozott édesapám? döntést – idiomatikus és félig kompozicionális magyar igei szerkezetek azonosítása párhuzamos korpuszból. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019), Szeged, SZTE (2019)
11. Novák, A., Novák, B.: Cross-Lingual Generation and Evaluation of a Wide-Coverage Lexical Semantic Resource. In Calzolari, N., 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), Miyazaki, Japan, European Language Resources Association (ELRA) (2018)

# POSZTER, DEMÓ

# Konverterek magyar morfológiai címkekészletek között

Vadász Noémi, Simon Eszter

MTA Nyelvtudományi Intézet
E-mail: {vadasz.noemi, simon.eszter}@nytud.mta.hu

**Kivonat** A magyarra alkalmazott morfológiai annotációs sémák és címkekészletek sokszínűsége és eltérő dokumentáltsága ösztönzött minket abban a munkában, amelynek első lépéseit mutatja be ez a cikk. A munka két fő részből áll: egyrészt összegyűjtjük és közzétesszük a magyarra alkalmazott morfológiai annotációs sémákkal és címkekészletekkel kapcsolatos elérhető információkat, másrészt konvertereket írunk a címkekészletek között. Ebben a cikkben három konvertert ismertetünk.
**Kulcsszavak: magyar nyelv, morfológia, annotáció, címkekészlet, konverzió**

## 1. Bevezetés

Az elmúlt évtizedekben a magyar nyelvtechnológiai műhelyekben több morfológiai annotációs séma, valamint a hozzájuk tartozó kimeneti formalizmus és címkekészlet lett kifejlesztve. Közös vonásuk, hogy mindegyik a magyar nyelv morfológiáját kódolja, további számítógépes nyelvészeti feldolgozásra alkalmassá téve a szöveget. Olykor szükség van az egyes címkekészletek közötti konverzióra, például ha egy feldolgozó eszköz kimeneti formalizmusa nem egyezik meg egy következő feldolgozási lépés bemenetének formalizmusával. A konverzió egy plusz lépés beillesztése az elemzési láncba, így fennáll annak a veszélye, hogy nem várt hibák kerülnek a folyamatba. Ennek elkerülése érdekében törekedni kell a lehető legpontosabb konverzióra.

Kornai et al. (2004) [1] három fontos kritériumot támaszt, amelynek egy morfológiai elemző kimeneti formalizmusának meg kell felelnie: *informativitás*, *adekvátság* és *egyszerűség*. Az informativitás követelménye a címkekészletre vonatkozóan azt jelenti, hogy pontosan és a lehető legteljesebben tükrözze a szóalakban szereplő morfológiai információkat; az adekvátságé azt, hogy nyelvészetileg megalapozott kategóriákat tartalmazzon; az egyszerűségé pedig azt, hogy kézi és automatikus feldolgozásra is könnyen használható legyen. Ezek a kritériumok azonban gyakran ellentmondanak egymásnak, az ebből fakadó elméleti és formai különbségek nehezítik a címkekészletek közötti pontos konverziót.

A formai különbségek viszonylag könnyen áthidalhatók, azonban az elméleti különbségek már több problémát okoznak. Egyes annotációs sémák a szóalakban található összes morféma kódolására törekszenek, míg mások csupán az inflexiós

morfémákat kódolják. Eltérések lehetnek a szófajkészletben, bizonyos alkategóriák használatában, valamint az egyes nyelvi jelenségek kezelésének finomságában is. Az ideális cél a veszteségmentes konverzió, amihez a működő megoldást a leginkább közelíteni kell.

A használatban lévő morfológiai annotációs sémákat és címkekészleteket vizsgálva azzal szembesültünk, hogy sok esetben kevéssé dokumentáltak, valamint hogy a közöttük működő konverterek jellemzően csak saját, belső használatra készültek. Ezért a jelen cikkben ismertetett konvertereket nyílt forráskóddal és dokumentációval szabadon elérhetővé tesszük. A címkekészletek eltérő dokumentáltságát egy nyilvános GitHub repozitórium[1] létrehozásával orvosoljuk, amely tartalmazza az egyes annotációk által alkalmazott címkék teljes listáját, valamint az általunk fejlesztett konvertereket. A tárhely könnyen bővíthető más, eddig nem vizsgált vagy újonnan létrejövő címkekészletek ismertetésével, illetve az ezekre fejlesztett konverterekkel.

Jelen munkánkban először feltérképeztük a használatban lévő morfológiai címkekészleteket, erről lásd a 2. fejezetet. Emellett három konvertert készítettünk, amelyek a kurrens emMorph morfológiai elemző [2] kimeneti kódkészletét konvertálják egyrészt a magyarlánc 3.0 [3] által is használt Universal Dependencies (UD) kódkészletre, másrészt a magyarlánc 2.0 által is használt MSD-re, illetve annak egy jegy–érték párokban megfogalmazott verziójára, amelyre CoNLL-ként fogunk hivatkozni. A konvertereket a 3. fejezetben ismertetjük részletesen. A konverterek teljesítményét többféleképpen is kiértékeltük, amit a 4. fejezetben mutatunk be. A cikket összegzés és a jövőbeli tervek leírása zárja az 5. fejezetben.

## 2. Magyar morfológiai annotációs sémák

Ebben a fejezetben a jelenleg forgalomban levő magyar morfológiai annotációs sémákat ismertetjük – az általunk jelen fejlesztés kereteiben vizsgált formalizmusokra nagyobb hangsúlyt fektetve. Elsősorban azokra a formalizmusokra koncentrálunk, amelyek legalább egy széles körben használt és valamilyen formában elérhető korpuszban vagy egy hasonló tulajdonságokkal rendelkező elemző kimeneteként léteznek.

Az egyik ilyen annotáció az *MSD* (Morphosyntactic Description) [4], amely a magyarral együtt tíz nyelv részletes morfoszintaktikai reprezentációjára alkalmas. Különlegessége, hogy pozícióalapú kódolást valósít meg, vagyis a kód rögzített hosszúságú, és minden pozíciójához egy-egy morfoszintaktikai jegy van hozzárendelve, az egyes pozíciókat betöltő karakterek pedig a jegyekhez rendelt értékek. Az első pozíció mindig a szófaji kategóriáé, a többi pedig további morfoszintaktikai információkat kódol – például egy kijelentő módú, múlt idejű, egyes szám második személyű, tárgyas ragozású főige MSD-kódolásban így fest:

```
adtad ad Vmis2s---y
```

[1] https://github.com/dlt-rilmta/panmorph

Ez a szisztéma nem hierarchikus, vagyis nem tükrözi az egyes értékek közötti összefüggéseket, valamint a morfológiai jelöltséget sem, ám az alapos dokumentációból[2] kiderül, hogy melyek azok a kombinációk, amelyek előfordulhatnak az egyes címkékben, és melyek nem. Továbbá nem is sztringalapú, ami azt jelenti, hogy sem a lemma, sem a morfológiai szegmentumok, sem az allomorfok nem képezik részét a morfológiai elemzésnek. Nincsenek továbbá jelölve a derivációk sem, csak és kizárólag morfoszintaktikai kódok vannak.

A Szeged Korpusz és Treebank 1.0 [5] és 2.0 változata [6] MSD kódokat tartalmaz, valamint a magyarlánc 1.0 és 2.0 verziója is MSD kódokat adott ki. A magyarlánc 2.0-nak egy későbbi verziójában és a korpusz 2.5 változatában már a harmonizált MSD–KR kódkészlet található [7], amely néhány tulajdonságában eltér az eredeti MSD kódolástól. A továbbiakban erre a harmonizált változatra fogunk MSD-ként hivatkozni.

A Szeged Treebanknek létezik egy további verziója is, amely a 2009-es *Syntactic and Semantic Dependencies in Multiple Languages* című CoNLL shared task [8] követelményeinek megfelelő felépítésű – ezt hívjuk *CoNLL*-nek. Hangsúlyoznunk kell, hogy a CoNLL csak egy formátum, aminek a lényege, hogy a morfoszintaktikai információk linearizált jegy–érték párok formájában legyenek megfogalmazva, de az alkalmazott jegyek és lehetséges értékeik nem kötöttek. Ebben a változatban a CoNLL címkekészlet a Szeged Korpusz 2.0 MSD kódjaiból (tehát a még nem harmonizált MSD kódból) lett átkonvertálva.

A CoNLL kódolás az MSD kódot két részre osztja fel: az első pozícióban szereplő szófajkódot különválasztja, a további morfoszintaktikai információkat pedig a fent említett jegy–érték struktúrában jeleníti meg. Ebben a verzióban az egyes jegy–érték párok sorrendje kötött, az MSD pozícióit követi. Ha egy jegy nincs kitöltve értékkel, akkor 'none' értéket kell, hogy kapjon. Az MSD-hez hasonlóan ez az annotációs séma sem tükrözi a morfológiai jelöltséget, továbbá erre is igaz, hogy sem a lemma, sem a morfológiai szegmentumok, sem az allomorfok nem képezik részét a morfológiai elemzésnek. Nincsenek jelölve a derivációk sem, csak morfoszintaktikai kódokat tartalmaz. A fenti példa ebben a kódolásban így néz ki:

```
adtad ad V SubPOS=m|Mood=i|Tense=s|Per=2|Num=s|Def=y
```

A Szeged Dependency Treebanknek van egy olyan verziója is, amely a *UD* (Universal Dependencies and Morphology[3]) nevű nemzetközileg elterjedt, univerzálisnak szánt annotációs séma szabályait követi [9], valamint a magyarlánc 3.0 verziója is UD kódokat bocsát ki a morfológiai elemzés szintjén. A Szeged Dependency Treebank a UD 1. verziójának megfelelő címkéket tartalmazza. Azóta a UD 2. verziója is kijött már, de a magyar nyelvre és a Szeged Treebankre és így az azon alapuló eszközökre az újítások még nem lettek alkalmazva. A UD kódolás sokban hasonlít a CoNLL-hez: ez is egy linearizált jegy–érték struktúrát valósít meg, de itt a jegyek ábécésorrendben szerepelnek, és az értékkel nem kitöltött

[2] http://nl.ijs.si/ME/Vault/V3/msd/msd.pdf
[3] http://universaldependencies.org

jegyek nem jelennek meg. További tulajdonságaiban megegyezik a CoNLL fent ismertetett tulajdonságaival. A fenti példa ebben a kódolásban:

```
adtad ad VERB Definite=Def|Mood=Ind|Number=Sing|Person=2|
    Tense=Past|VerbForm=Fin|Voice=Act
```

A legújabb magyar morfológiai elemző az *emMorph*[2], amely az e-magyar [10] szövegfeldolgozó eszközlánc morfológiai moduljaként is funkcionál. Ennek az elemzőnek az annotációs sémája jelentősen eltér az eddig ismertetettekétől, ugyanis sztringalapú, vagyis a lemma, a morfológiai szegmentumok és bizonyos esetekben az allomorfok is az elemzés részét képezik. További eltérést jelent, hogy nemcsak morfoszintaktikai információkat kódol, hanem olyan derivációkat is kezel, amelyeknek nem feltétlenül van köze az adott szó mondatbeli szerepéhez. Annyiban viszont hasonlít az MSD-hez, hogy nem hierarchikus, valamint nem tükrözi a morfológiai jelöltséget sem. Az emMorph többféle módon képes megjeleníteni a kimenetet aszerint, hogy tartalmazza-e a szóalakhoz rendelt tövet és a szegmentumokat a szófajcímke és az elemzések mellett. Mi a tövet és a morfémákat nem tartalmazó morfológiai kódot konvertáljuk. A fenti példa ebben a rendszerben[4] ábrázolva:

```
adtad [/V][Pst.Def.2Sg]
```

Léteznek még további magyar morfológiai annotációs sémák is, amelyeket megemlítünk, de jelen cikkben részletes leírást nem adunk róluk, ugyanis a fejlesztés jelenlegi fázisában még nem tudunk kész konvertereket kiállítani ezekre a formalizmusokra. Az egyik ilyen a *Humor*, illetve annak több változata [11,12,13]. A Humornak egy verziója lett használva az MNSZ2 [14] és egy másik verziója az Ómagyar Korpusz [15] építésénél is, ezért a későbbiekben tervezzük az ebből az irányból induló konverterek fejlesztését is. Egy másik formalizmus a *KR* kód [16], amelyet a hunmorph [17] morfológiai elemző bocsát ki, és amelyre a jövőben szintén tervezünk konvertereket írni.

## 3. A konverterek

Legyen szó bármilyen formátumok közti konverzióról, többféle megközelítés létezik. Az egyik, ha a bemeneti címkekészletről a kimenetire egy közvetlen leképezést valósítunk meg. Egy másik lehetséges módszer, ha – a gépi fordítás egy fajtájánál használt *interlinguá*hoz hasonlóan – egy köztes metaformátumot találunk ki, amire le tudunk képezni minden bemeneti formátumot, és amiből elő tudunk állítani minden kimeneti formátumot. Ez a magyar nyelv morfológiája esetében egy minden eddiginél részletesebb, a szokásos vitás kérdésekben (főnév vs. melléknév, inflexió vs. derivació stb.) kötelezően döntést hozó, a morfológiai annotációk fent felsorolt tulajdonságait (hierarchikusság, sztringalapúság stb.)

[4] A címkék feloldása példákkal együtt az e-magyar honlapján (`https://e-magyar.hu/hu/textmodules/emmorph_codelist`) található.

egyszerre birtokló újabb morfológiai annotációt eredményezne, ami lehetetlen vállalkozásnak tűnik. Ezért az első megközelítés mellett döntöttünk, és közvetlen leképezést csináltunk három irányba, ahol a bemeneti oldalon mindig az emMorph kódja áll.

Az emMorph címkekészletről történő konvertálásnak több előnye is van. Egyrészt az emMorph formalizmusa összességében részletesebb, mint a célformalizmusok, ezért a konverzió viszonylag kis veszteséggel megoldható. Másrészt pedig a magyar nyelvre készült kurrens elemzőláncba, az e-magyarba is az emMorph elemző van beépítve, így az e-magyarral elemzett szöveg tetszőlegesen átalakítható a kezelt címkekészletek valamelyikére a felhasználó céljainak megfelelően. Az `emmorph2msd` konverter kimenete a magyarlánc 2.0 által is előállított MSD kód; az `emmorph2conll` konverter kimenete a 2. fejezetben ismertetett, az MSD kód átalakításával kialakított jegy–érték struktúrájú CoNLL kód; az `emmorph2ud` konverter kimenete pedig a magyarlánc 3.0 által is előállított UD kód.

A konverterek kidolgozásához megvizsgáltunk néhány elérhető konvertert, azok működéséből, felépítéséből levontuk a számunkra fontos tanulságokat. Az egyik ilyen konverter az e-magyarban működő DepTool.java[5], amely az emDep modul számára konvertálja az emMorph címkéket a fent ismertetett CoNLL formátumra, de egy belső, kevert címkekészletet használva. A magyarláncban is több konverter működik a címkekészletek között (pl. a harmonizált MSD és a UD között[6]).

Az `emmorph2ud` konverter az e-magyar elemzőlánc legfrissebb, `emtsv` elnevezésű verziójában [18] kiváltotta a DepTool.java konvertert. Az elemzőláncba illeszkedve az `emMorph` kimenetét konvertálja az `emDep` modul számára fogyasztható jegy–érték struktúrájú UD címkékre, valamint kimeneti formalizmusként lehetővé teszi, hogy a felhasználók az eddig elérhető `emMorph` kimenet mellett UD morfológiai címkéket is kaphassanak.

A konverterek elkészítésekor akkor volt a legkönnyebb dolgunk, amikor egy-az-egyhez megfeleltetés állt fenn a bemeneti és a kimeneti oldal között. Ugyanakkor sok esetben szükség volt a címkék megfeleltetésekor aleseteket és kivételeket megfogalmazni. Ennek oka a konverterek közötti elméleti különbségekben keresendő. Szemléltető példaként tekintsük a szófajok és az azokat reprezentáló címkék esetét. Az emMorph formalizmusában a szófajokat ábrázoló címkék megkülönböztetett formát kaptak a morfológiai jegyekhez képest (`[/Adj]`). Ugyanakkor a melléknevekhez és határozószókhoz járuló felsőfokot kifejező morféma is a szófajcímkékhez hasonló formátummal rendelkezik (`[/Supl]`), így külön figyelmet kellett fordítanunk arra, hogy a felsőfokban álló melléknevek és határozószók szófaját kinyerjük. Ráadásul az emMorph a kimeneti címkekészletekkel ellentétben a derivációkat is megjeleníti a címkékben. A helyes konverzióhoz a legkülső képzett alak szófaját és az arra rakódó inflexiós jegyeket kellett kinyernünk az

[5] `https://github.com/dlt-rilmta/hunlp-GATE/blob/master/Lang_Hungarian/src/hu/nytud/gate/util/DepTool.java`

[6] `https://github.com/zsibritajanos/magyarlanc/blob/master/magyarlanc/src/main/java/hu/u_szeged/converter/univ/Msd2UnivMorph.java`

emMorph címkéből, és ezeket a jegyeket kellett a kimeneti címkekészletek megfelelő jegyeire konvertálnunk.

Elkerülhetetlen volt, hogy egyes esetekben a lemma vagy a token felszíni tulajdonságaira is támaszkodjunk a konverzió során. Bár az emMorph címkekészlete tűnik a legrészletesebbnek, néhány nyelvi jelenség esetében mégsem tartalmazza a helyes kimeneti címkéhez szükséges morfoszintaktikai vagy lexikai információt. Például a kötőszavak bizonyos tulajdonságait nem kódolja az emMorph, míg a UD, a CoNLL és az MSD is külön jegyet ad a mellérendelő és az alárendelő kötőszóknak. Emellett az MSD és a CoNLL az egyes és a páros kötőszókat is külön jeggyel választja ketté, valamint azt is jelöli, hogy mondatok vagy szavak között állnak az aktuális mondatban. Mivel ezeket az információkat nem kódolja az emMorph, ezért a biztosan egy csoportba tartozó kötőszók felsorolásával oldottuk meg a megfelelő kimeneti címke előállítását.

A névmások kezelésében is alapvető különbségek vannak az emMorph és a kimeneti címkekészletek között. Az MSD, a CoNLL és a UD szófajcímkéi között szerepel a névmási címke, kiegészítve a névmás típusát (személyes, mutató, kölcsönös, visszaható, általános stb.) reprezentáló információval. Az emMorph a névmások esetében a szófajcímkében azt tünteti fel, hogy milyen szófajú szó (főnév, melléknév, számnév, determináns vagy határozószó) helyettesítője. A névmástípusok közül csak a kérdő és a vonatkozó névmást jelöli a szófajcímkében. A névmások és azok típusai zárt szóosztályt alkotnak, így felsorolhatóak. Az emMorph-fal nem kezelt névmástípusok tagjainak felsorolásával igyekeztünk megoldani a helyes kimeneti címkék kinyerését a konverzió során.

Az igekötők kezelésében is találunk különbségeket. A UD a dokumentációk alapján csak a *meg* igekötőt jelöli külön szófajjal, a többi igekötőt eredeti szófaja alapján címkézi, így az emMorph által igekötőnek címkézett *meg* kapja csak az igekötőhöz tartozó szófajcímkét a UD-ra való konvertáláskor. A másik két kimeneti címkekészlet a többi igekötőt is igekötőként jelöli, így azokkal nem kellett külön foglalkoznunk.

A UD nem csak az igekötők kezelésében tér el a többi készlettől, hanem a tulajdonneveket is külön szófajcímkével látja el. Ezért amikor a lemmatizáló nagybetűs tövet tulajdonít a szóhoz, akkor a kimeneti szófajcímke az emMorph kódról konvertált főnévi címke helyett tulajdonnév lesz. Ekkor a helyes átalakítás a megfelelő tövesítésen múlik.

Olyan jelenségek is akadnak, amelyek kimaradnak a konverzióból, vagyis hiába szerepelnek a kimeneti címkekészletben, a konverzió során nem tudnak előállni. Ez akkor fordul elő, ha a bemeneti oldalon nem szerepel egy jelenség, és a vizsgált szó felszíni tulajdonságaiból sem tudunk következtetni. Erre egy példa a birtokos eset címkéje. A magyarlánc a *-nAk* ragos névszók esetében mind a részesesetet, mind a birtokosesetet jelentő címkét tartalmazó címkesort kiadja, de az emMorph csak a datívuszi címkét ismeri, így a konverterünk is mindig csak ilyet fog kiadni. Egy hasonló példa a segédigék kezelése. A kimeneti címkekészletek megkülönböztetnek fő- és segédigéket, míg az emMorph nem. Mivel minden magyar igealakra igaz az, hogy kontextustól függően viselkedhet fő- és

segédigeként is, ennek a kérdésnek az eldöntését a szintaxis területére toljuk, és csak egy igei címkét alkalmazunk.

A konvertereket `Python3`-ban implementáltuk. A kódok szabadon elérhetőek és felhasználhatóak GNU GPLv3 licenc alatt, míg a kódkészleteket ismertető dokumentációt és táblázatokat CC-BY-SA-4.0 licenc alatt publikáljuk a `https://github.com/dlt-rilmta/panmorph` repozitóriumban.

## 4. Kiértékelés

A konverterek teljesítményét több mérőszámmal is szemléltetjük. A kiértékeléskor igyekeztük valóban a konverzió minőségét megítélni, azonban a címkekészletek alapvető elvi különbségei, valamint a címkekészletekkel dolgozó elemző eszközök eltérő minősége is okozhatnak hibapontokat az egyes címkék összevetésekor.

A három konverter fejlesztése és kiértékelése hasonló módon zajlott. Először létrehoztuk a fejlesztéshez és a teszteléshez szükséges elemzéseket. A címkekészletek dokumentációi alapján elkészítettük a konverterek első verzióját, majd azzal átkonvertáltuk a fejlesztőanyagban található összes emMorph címkét UD, MSD, illetve CoNLL címkére. A kimenetben szereplő hibatípusokat elemeztük, majd a feltárt hibák alapján javítottunk a konverteren. Végül a tesztanyagon kiértékeltük a konverterek teljesítményét.

### 4.1. `emmorph2msd` és `emmorph2ud`

Mind az emMorph, mind az MSD és a UD címke produktívan előállítható, előbbi az emMorph elemző, utóbbi a magyarlánc valamely verziójának kimeneteként, ezért az `emmorph2ud` és az `emmorph2msd` fejlesztéséhez is korlátlan mennyiségű elemzést tudtunk előállítani. A fejlesztéshez a Szeged Treebankből kinyert összes szóalakot használtuk, amely összesen 152 056 tokent tesz ki.

A fejlesztéshez a tokeneket leelemeztük az emMorph-fal, amely 195 416 elemzést eredményezett, majd ezeket az elemzéseket konvertáltuk UD és MSD kódra. A tokeneket a magyarlánc 2.0-val és 3.0-val is[7] megelemeztük – ezek számítottak a gold standard adatnak, amelyhez a konverter kimenetét hasonlítottuk.

A konverterek tesztelésekor nem az egyes tokenek számítanak egy tesztesetnek, hanem a token és egy hozzá tartozó emMorph elemzés. Ennek megfelelően a fejlesztőanyagban annyi teszteset van, ahány emMorph elemzés (195 416). Ez azt is jelenti, hogy azokban az esetekben, amikor az emMorph hibás elemzést ad egy szónak – úgy is, hogy mellette esetleg jó elemzést is ad, ami egy másik tesztesetet képez –, de a magyarlánc összes elemzése között nem szerepel egy ugyanolyan jelentésű hibás elemzés, akkor olyan hiba is a konverter rovására íródik, amely nem a konverzió, hanem az emMorph hibája.

[7] Bár a Szeged Treebank elérhető mind emMorph címkékkel, mind UD és MSD címkékkel, mi mégis az újraelemzés mellett döntöttünk. Egyrészt a Szeged Treebankben alkalmazott konverzió és a kézi javítás eredményezte esetleges formai hibákat akartuk ilyen módon kiküszöbölni, másrészt így több teszteset áll a rendelkezésünkre.

A két konverter végső kiértékelését egy másik teszthalmazon végeztük, amelyhez a Webcorpus 100 000 leggyakoribb szavának listáját használtuk fel [19]. Ezekkel a fent leírtak szerint jártunk el, vagyis a szavakat megelemeztük az emMorph-fal, valamint a magyarlánc 2.0 és 3.0 verzióival is. Mivel a konverterek az emMorph címkék konvertálását vállalják, a fejlesztő és a tesztadatból kivettük azokat a szavakat is, amelyekhez az emMorph nem tudott címkét rendelni (a kimenet `'None'` volt). Voltak olyan szavak is, amelyekkel egyik elemző sem birkózott meg. Jellemzően ezek a tokenek az elemző számára valamilyen speciális jelentéssel bíró karaktert tartalmaztak (pl. * karakterre végződtek) – ezekből összesen 6 388 darab volt. A végső tesztanyag 93 606 tokenjéből az emMorph elemzést követően 120 714 címke állt elő, amelyből kivettük a `'None'` címkéket, így összesen 105 545 tesztesetünk maradt a kiértékelés elvégzésére.

### 4.2. `emmorph2conll`

A magyarlánc 2.0 előállít ugyan CoNLL címkéket, de csak a szintaktikai elemzés előkészítő lépéseként, a már morfológiai egyértelműsítésen átesett MSD címke átalakításával. Ez azt jelenti, hogy egy tokenhez nem az összes lehetséges elemzés CoNLL címkéje áll a rendelkezésünkre, hanem minden tokenhez csak egy. Éppen ezért az `emmorph2conll` esetében a Szeged Treebank hasonló annotációval ellátott változatára támaszkodtunk a fejlesztéskor és a teszteléskor is. Legelső lépésként felosztottuk a Szeged Treebankből kinyert szólistát (152 056 token) két részre olyan arányban, ahogy a másik konverternél aránylott egymáshoz a Webcorpusból és a Szeged Treebankből kinyert fejlesztő- és tesztelőanyag mérete. Így a fejlesztésre 94 245 token állt rendelkezésünkre, amely az emMorph-fal megelemezve 120 714 címkét eredményezett. A végső tesztelésre 57 781 token maradt, a `'None'` címkék kivétele után összesen 74 702 teszteset állt rendelkezésünkre. A kiértékelés során ugyanazt a három tesztet végeztük el, mint a másik két konverter esetében.

Az `emmorph2conll` esetében szintén egy token és egy emMorph címke párosa képez egy tesztesetet, ugyanakkor azt sem szabad elfelejteni, hogy a teszteléskor a tokenekhez nem az összes elképzelhető elemzés áll rendelkezésre, hanem csak azok az egyértelműsített jelentések, amelyek valóban előfordultak a tesztanyagban.

### 4.3. A mérések

Bár többféle mérést végeztünk, minden esetben csak a valós pozitív (*true positive, TP*) találatokat számoltuk össze, hiszen a feladat kiértékelésekor a fedésnek nincs értelme (minden címkét konvertálunk). Ezért csak pontosságot (*accuracy*) számoltunk oly módon, hogy a helyesen konvertált esetek számát elosztottuk az összes teszteset számával.

Háromféle tesztet végeztünk el. Az első – legmegengedőbb – teszt során azt ellenőriztük, hogy a konvertált címke előfordult-e valaha a magyarlánccal elemzett tesztanyagban (tehát sem a tokent, sem az emMorph címkét nem párosítottuk hozzá). Bár feltételezhetjük, hogy a tesztanyag ugyan nem tartalmazza az összes

elképzelhető UD és MSD címkét, de a leggyakoribbakat biztosan, így ez a teszt annak a mérésére alkalmas, hogy valid címke jött-e létre a konverzió után. Vagyis ez csupán egy validitási kritériumot ellenőriz, önmagában nem elég mutatója a konverzió minőségének, elsősorban a fejlesztés során volt hasznos.

A második teszt volt a legszigorúbb, minden token esetében az ahhoz a tokenhez tartozó magyarlánc elemzésekkel vetettük össze a konvertált címkét. Emögött a mérőszám mögött az a feltételezés áll, hogy a kétféle elemző kimenetében szereplő címkék páronként megfeleltethetők egymásnak, mert ugyanaz a jelentésük. A valóságban azonban a két elemző sok jelenséget egészen eltérően kezel az annotációs sémák közötti elméleti különbségek miatt. Ráadásul az elemzők hibákat is vétenek, ami szintén nehezíti az összehasonlítást. Ezzel a szigorú mérőszámmal tehát nem pusztán a konverziót értékeljük ki, hanem a kétféle elemző különbségeit is kidomborítjuk, mert olyan esetek is hibásnak számítanak, amelyek a kétféle elemző eltérő minőségéből vagy megközelítéséből adódnak. Ezeket a hibákat nem válogattuk szét, így az eredményeket ennek tudatában kell értékelni.

A harmadik tesztben – a fenti torzító hatást kiküszöbölendő – úgy számoltuk a pontosságot, hogy a tokenhez tartozó emMorph címkéről konvertált kimenetet nem a tokenhez tartozó gold standard – UD, MSD vagy CoNLL – címkével vetettük össze, hanem az összes olyan címkével, amely bármely, ugyanolyan emMorph elemzéssel rendelkező tokenhez tartozik. Például a `[/N][Pl][Acc]` emMorph címkéből konvertált kimeneti címkét azokkal a gold standard címkékkel vetjük össze, amelyek olyan tokenekhez tartoznak, amelyeknek szintén van `[/N][Pl][Acc]` elemzése. Ez egy megengedőbb kiértékelés, ugyanakkor feltehetőleg kiszűri a kétféle elemző különbségeinek torzító hatását. A konvertálók teljesítménye szempontjából ezt a mérőszámot tartjuk a legfontosabbnak.

### 4.4. Eredmények és diszkusszió

Az első teszt tehát azt vizsgálta, hogy valid címkék jönnek-e létre a konverzió során. Az 1. táblázatban látható, hogy mindhárom konverter nagyon magas eredményeket ért el ezen a teszten, ám ez a magas szám alapvető elvárás, amely egy konverterrel szemben támasztható. Magyarázatra szorul azonban a tény, hogy egyik konverterrel sem sikerült elérni 100%-os eredményt. Mindhártom konverter esetében átnéztük a nem validnak ítélt címkék listáját, és ellenőriztük, hogy a rendelkezésünkre álló dokumentációk alapján hibásak-e. A leírások alapján megállapítottuk, hogy a nem validnak ítélt címkék valójában validak, csak egyszerűen hiányoztak a gold standard adatból.

A 2. táblázatban ismertetett eredmények a második tesztre vonatkoznak, így az elvárásoknak megfelelően ezek a leggyengébbek. A 4.3. fejezetben ismertetett kiinduló ötlet alapján ez lenne a megfelelő mérés a konverzió minőségére, ám szem előtt kell tartani a tesztek során tapasztalt torzító hatást, amelyet az egyes címkekészletek és az elemzők közötti alapvető elméleti különbségek okoznak. Gyakori például, hogy az egyik eszköz csak melléknévi, míg a másik csak főnévi címkét ad egy szónak. Még ha a többi jegyet sikeresen konvertálja is a konverter, és a konverzió valójában helyes kimeneti címkét eredményezett, amiatt, hogy az ennek megfelelő címke hiányzik a gold standard adatból, a konverzió

| | összes | TP | TN | Acc |
|---|---|---|---|---|
| `emmorph2ud` | 105 545 | 105 170 | 375 | 99,64% |
| `emmorph2msd` | 105 545 | 104 539 | 1 006 | 99,05% |
| `emmorph2conll` | 74 702 | 72 459 | 2 243 | 97,00% |

1. táblázat. A konverterek eredményei az első teszten.

is hibásnak számít. Ez a probléma akkor merül fel, ha az egyes emMorph elemzésekhez nem párosítható elemzés az összes magyarlánc kimenet közül, tehát amikor a magyarlánc fedése kisebb.

| | összes | TP | TN | Acc |
|---|---|---|---|---|
| `emmorph2ud` | 105 545 | 87 506 | 18 039 | 82,91% |
| `emmorph2msd` | 105 545 | 77 422 | 28 123 | 73,35% |
| `emmorph2conll` | 74 702 | 52 176 | 22 526 | 69,85% |

2. táblázat. A konverterek eredményei a második teszten.

Egy jellemző példa az anaforikus birtokos egyes és többes számú jelének előfordulása a fejlesztőanyagokban. A 3. táblázatból kiolvasható, hogy az emMorph szívesebben ad `[AnP]` és `[AnP.Pl]` címkéket a névszóknak, mint a magyarlánc különböző verziói. Természetesen az `emmorph2conll` kiértékelésekor ez a probléma fokozódik, mivel ott nem az összes lehetséges magyarlánc elemzés áll a rendelkezésünkre, hanem minden szóalakhoz csak egyetlen, a korpuszban lévő egyértelműsített elemzés.

| | anaforikus birtokosok |
|---|---|
| emMorph | 8 959 |
| MSD | 1 136 |
| UD | 5 804 |

3. táblázat. Az emMorph és a magyarlánc két verziója által eredményezett egyes és többes számú anaforikus birtokosok darabszáma a fejlesztőanyagban.

A harmadik tesztet tekintjük a legalkalmasabb mutatónak a konverzió minőségére vonatkozóan. Az eredményeket a 4. táblázat ismerteti. Az `emmorph2ud` és az `emmorph2msd` konverterek esetében 97% fölötti eredményt értünk el, az `emmorph2conll` azonban jóval gyengébben, bár 90% fölött teljesített a teszten.

| | összes | TP | TN | Acc |
|---|---|---|---|---|
| `emmorph2ud` | 105 545 | 103 489 | 2 056 | 98,05% |
| `emmorph2msd` | 105 545 | 102 693 | 2 852 | 97,30% |
| `emmorph2conll` | 74 702 | 68 691 | 6 011 | 92,00% |

4. táblázat. A konverterek eredményei a harmadik teszten.

Azt feltételezzük, hogy az `emmorph2conll` gyenge eredményének az oka a kiértékelés módszerében keresendő. Míg az egyes tokenekhez az emMorph többféle elemzést is eredményezhetett, addig az annotált korpuszban egy tokenhez természetesen jóval kevesebb elemzés tartozott. Az 5. táblázat a Szeged Korpusz szólistájának token/címke arányát mutatja az emMorph-fal és a magyarlánc 3.0 verziójával megelemezve, valamint a CoNLL címkékkel annotált korpuszban. Minél magasabb ez a szám, annál több gold standard címkével tudjuk összevetni a konvertált címkét. Ez azt jelenti, hogy a harmadik teszt eredményét az `emmorph2conll` konverter kiértékelése esetében hasonló fenntartásokkal kell kezelni, mint a második teszt eredményeit.

| | token | címke | címke/token |
|---|---|---|---|
| emMorph | 152 056 | 293 956 | 1,93 |
| UD | 152 056 | 242 477 | 1,59 |
| ConLL | 152 056 | 159 033 | 1,05 |

5. táblázat. A Szeged Treebank címke/token arányai az egyes címkék szerint.

A különböző morfológiai címkekészletek közötti konverzió során felmerül az eltérő tövesítés problémája is. Az emMorph mint derivációt is kezelő morfológiai elemző nyilván más tövet fog megállapítani egy képzett szó esetében, mint azok az elemzők, amelyek csak az inflexiós jegyeket kódolják. A tesztanyagokon kimértük, hogy az esetek mekkora részében jelenik meg az eltérő tövesítés. Ha egy szóhoz akár a bemeneti, akár a kimeneti oldalon több elemzés társul a tesztanyagban, akkor nehézkes a tövesítés összevetése. Ezért közvetlenül nem tudjuk összehasonlítani az elemzőket, csak közvetett módon. Azoknál a szavaknál hasonlítottuk össze a töveket, ahol a fenti kiértékelés alapján a 2. tesztben a konverter hibátlanul konvertált a címkék között.

A 6. táblázatban látható eredmények azt mutatják, hogy mindhárom címkekészletpárt tekintve az esetek legnagyobb részében nem különböznek a tövek a helyesnek ítélt konverziók között. Természetesen ez az eredmény nem jelenti azt, hogy a morfológiai kódok közötti konverziókor nem kell foglalkozni az eltérő tövesítéssel, az itt ismertetett konvertálók azonban egyelőre csak a morfológiai címkék közötti átváltást vállalják.

| | TP | egyező tő | különböző tő | accuracy |
|---|---|---|---|---|
| `emmorph2ud` | 87 506 | 80 237 | 7 269 | 91,69% |
| `emmorph2msd` | 77 422 | 70 299 | 7 123 | 90,80% |
| `emmorph2conll` | 52 176 | 48 021 | 4 155 | 92,04% |

6. táblázat. Az egyező lemmák a helyes konverziók esetében.

## 5. Összegzés

A konverterek elkészítésével lehetővé tesszük, hogy bárki könnyedén átalakíthassa az e-magyar elemzőlánc vagy az emMorph morfológiai elemző kimenetét egy általa választott morfológiai annotációs sémának megfelelően. Egyelőre az itt ismertetett három kódra tudunk konvertálni, de a jövőben tervezzük más be- és kimeneti kódkészletek közötti konverterek írását is. A fent bemutatott `emmorph2ud` konverter az e-magyar elemzőlánc új változatába is bekerült, ahol egyrészt egy közbülső láncszemként az emMorph kimenetét konvertálja az emDep modul számára fogyasztható jegy–érték struktúrájú UD címkére, másrészt pedig kimeneti formalizmusként lehetővé teszi, hogy az e-magyar elemzőláncot használók az eddig elérhető emMorph kimenet mellett UD címkéket is kaphassanak. Fontosnak tartjuk kiemelni, hogy az általunk készített konverterek forráskódja és a címkekészletek leírása szabadon elérhető a `https://github.com/dlt-rilmta/panmorph` nyilvános GitHub repozitóriumon keresztül.

## Hivatkozások

1. Kornai, A., Rebrus, P., Vajda, P., Halácsy, P., Rung, A., Trón, V.: Általános célú morfológiai elemző kimeneti formalizmusa. In: II. Magyar Számítógépes Nyelvészeti Konferencia. (2004)
2. Novák, A., Siklósi, B., Oravecz, Cs.: A New Integrated Open-source Morphological Analyzer for Hungarian. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA) (2016)
3. Zsibrita, J., Farkas, R., Vincze, V.: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: International Conference on Recent Advances in Natural Language Processing, Shoumen, Bulgária, INCOMA Ltd. (2013) 763–771
4. Erjavec, T.: MULTEXT-East Morphosyntactic Specifications. Version 3.0. (2004) http://nl.ijs.si/ME/Vault/V3/msd/html/.
5. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus. In Sojka, P., Kopecek, I., Pala, K., eds.: Text, Speech and Dialogue. Volume 3206 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2004) 41–47
6. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, Springer (2005) 123–131

7. Vincze, V., Varga, V., Simkó, K.I., Zsibrita, J., Nagy, A., Farkas, R.: Morfológiai újítások a Szeged Korpusz 2.5-ben. In Tanács, A., Viktor, V., Veronika, V., eds.: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2014) 332–338
8. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štepánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics (2009) 1–18
9. Vincze, V., Farkas, R., Simkó, K.I., Szántó, Zs., Varga, V.: Univerzális dependencia és morfológia magyar nyelvre. In Tanács, A., Viktor, V., Veronika, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2016) 322–329
10. Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: Az `e-magyar` digitális nyelvfeldolgozó rendszer. In Vincze, V., ed.: XIII. Magyar Számítógépes Nyelvészeti Konferencia. (2017) 49–60
11. Prószéky, G., Kis, B.: A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. ACL '99, Stroudsburg, PA, USA, Association for Computational Linguistics (1999) 261–268
12. Novák, A.: Milyen a jó Humor? In: I. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2003) 138–144
13. Novák, A.: A Humor új Fo(r)mája. In: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2014) 303–308
14. Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Proceedings of LREC 2014. (2014)
15. Simon, E.: Corpus Building from Old Hungarian Codices. In É. Kiss, K., ed.: The Evolution of Functional Left Perpiheries in Hungarian Syntax. Oxford University Press (2014) 224–236
16. Rebrus, P., Kornai, A., Varga, D.: Egy általános célú morfológiai annotáció. Általános Nyelvészeti Tanulmányok **XXIV.** (2012) 47–80
17. Trón, V., Kornai, A., Gyepesi, Gy., Németh, L., Halácsy, P., Varga, D.: Hunmorph: Open Source Word Analysis. In: Proceedings of the Workshop on Software, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 77–85
18. Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` – Egy formátum mind felett (2019) Jelen kötetben.
19. Kornai, A., Halácsy, P., Nagy, V., Oravecz, Cs., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In: Proceedings of the 2nd International Workshop on Web as Corpus. (2006)

# Named Entity Recognition in the Miskolc Legal Corpus

Üveges István

Szegedi Tudományegyetem, Általános Nyelvészeti Tanszék
uvegesistvan898@gmail.com

**Abstract.** In this paper, a brief study will be presented with regard to the issue of Named Entity Recognition (NER) in legal texts. To get an overall picture, we examined closely the output of two existing analysers: the "*magyarlanc"* linguistic processing toolkit [1] and a Named Entity Recognition system developed by the Natural Language Processing Group at the University of Szeged [2]. Firstly, short references are made to named entity recognition projects in the literature considered important in the current framework. Secondly, quantitative analyses of the data will be presented. At the end of the study, some problematic cases and potential solutions will be discussed which will be followed by the discussion of the future research.

**Keywords**: Named Entity Recognition, Hungarian legal texts, magyarlanc, Szeged NER

## 1 Introduction

The process of finding named entities in a text and classifying them to a semantic type is called *Named Entity Recognition (NER).* The task itself was firstly introduced in the early 1990s in computational linguistics and NER is a cornerstone for tools based on Information Extraction (IE) and key issue in many fields of science nowadays.

Here we focus on NER in the legal domain, where the (semi)automatic anonymization of legal documents and the development of more informative and efficient searching tools get more and more attention. Named Entities (NEs) are not just mentions of persons and organizations in the legal domain, but we also have to take into consideration other categories like names of laws and even concepts. In the international literature, we can see many ongoing projects aimed to develop such systems and applications for Anglo-American legal documents ([3], [4], [5] etc.) and in the Hungarian literature as well ([6], [7]).

With the automatic detection and classification of such elements, legal information extraction can be enhanced for lawyers, courts, governmental organizations, or even non-professionals.

## 2 Data

The examination was carried out on the Miskolc Legal Corpus [8], which was created by a cooperation of lawyers, linguists and IT specialists in order to make the language of law more easily available for NLP studies.

During the creation of the corpus the main goal was to cover the largest segment possible of the Hungarian legal language. It contains six different sources (cf. [8]) of legal texts[1];

- the full text of 5 Hungarian laws (henceforth: Laws)
- randomly selected parts of other legal regulations
- texts of judgements and legal sentences
- explanatory texts (from ministerial arguments and university textbooks)
- legal forums (Forums)
- transcripts (Transcripts).

For our current analysis, the first ~6000 tokens have been chosen from the Laws, Forums and Transcripts sub-corpora.

The Forum part is, as its name may suggest, made of posts, topics and comments of online discussion sites. The transcript part consists of transcripts of courtroom discussions[2] so this section represents the spoken legal language in the corpus. The Laws part has been compiled from full texts of Hungarian laws.

In Table 1, some main properties of the selected texts from the sub-corpora are summarized.

| Sub-corpus | Token number | Word count |
|---|---|---|
| Forums | 6041 | 4718 |
| Transcripts | 6010 | 4594 |
| Laws | 6014 | 4660 |

Table 1: Basic information

When selecting texts from the Miskolc Legal Corpus, the main criterion was that they should represent (intuitively) different aspects of legal language use and text type.

## 3 Methods

Our main goal is to find an explicit evidence that these distinct domains of the legal language may (or may not) require a different treatment from the automatic NE recognizer tools.

To achieve this, a quantitative analysis was carried out on three levels:

[1] From each source, the corpus contains approximately 25.000 sentences, 150.000 sentences in total. The coprus was originally developed in the framework of an OTKA-project: https://sites.google.com/site/otkamiskolc2015/

[2] Recordings and transcripts were made with the consent of all participants of the discussions.

- on the one hand, after a manual annotation (see 3.1) the output of automatic POS-tagging of the *magyarlanc* toolkit, was compared with the output of the Named Entity Recogniser System on the same text,
- on the other hand, in the case of multiword NEs, where *magyarlanc* should tag the affected tokens on the level of dependency grammar with an "NE" label [1], the presence or absence of this specific tag was examined closely.

In the qualitative section of the analysis, the most frequent sources of errors will be examined closely to reveal the domain-specificity of these peculiarities and to provide useful data for increasing the efficiency of future NER-tools in the legal domain.

### 3.1 Manual annotation

To get comparable results, and data, at the first step, all the examined text was checked by a linguist expert, who annotated all the NEs manually. The annotation followed the *tag-for tagging* principle, but apart from this, it was match with the rules defined during the annotation of the HunNER corpus [9].

The used definition for NE categories was based on the ACE 2006 annotation guideline [10]. However, just the name mentions ("Joe Smith")[3], locations and organizations were kept as an annotated category.

The three basic category searched during the annotation process was *person*, *location* and *organization names*. Besides that, the names of legal *regulations* (e.g. Ptk. – *Civil Code*) proved to be important during the annotation process in this specific domain. Table 2 shows the manually annotated NEs in the examined texts.

### 3.2 Automatic NER methods

The selected texts were parsed with *magyarlanc* and the NER-tool, after that we checked whether a label was correctly assigned to a token, or not.

The expected label was PROPN from the *magyarlanc* and an I-TYPE tag from the NER-tool (where "TYPE" stands for one of the above mentioned 4 categories). The NER-tool's classification of tokens into PER, LOC etc. sub-categories is not investigated at this point; here the aim is just to see whether the two systems could find the expected tokens and selected them as a NE, or not.

It is important to mention that *magyarlanc* was originally trained on the Szeged Treebank, which is built up from texts from six different genres, because "the main criteria were that they should be thematically representative of different text types." [11] It contains legal texts from the field of legislation, but only one specific type of it: full texts of laws.

On the other hand, the NER-tool was developed by using the same corpora, but just with another subset of it which contains short business news articles, so the training set of the NER-tool had not contained legal texts at all. The original F-measure calculated from the metrics of different NE type's results (PER, ORG, LOC, MISC) was an overall 94.77% on the Hungarian data [2].

[3] Examples are quoted from the original guideline.

In the next section, the results connected with the actual corpora's NEs will be briefly overviewed.

| Corpora | NEs count (number of annotated tokens) | Number of NEs | Multi-token NEs | Type |
|---|---|---|---|---|
| Transcripts | 56 | 29 | 23 | Person |
| | 41 | 22 | 11 | Location |
| | 69 | 33 | 22 | Organization |
| | 25 | 11 | 7 | Regulation |
| | 191 | 95 | 63 | All in the section |
| Forums | 95 | 51 | 19 | Person |
| | 4 | 4 | 0 | Location |
| | 6 | 5 | 1 | Organization |
| | 6 | 6 | 0 | Regulation |
| | 111 | 66 | 20 | All in the section |
| Laws | 0 | 0 | 0 | Person |
| | 5 | 3 | 2 | Location |
| | 14 | 8 | 4 | Organization |
| | 6 | 1 | 1 | Regulation |
| | 25 | 12 | 7 | All in the section |
| Sum: | 327 | 173 | 90 | |

Table 2: Manually annotated tokens

## 4 Results

In Table 3 the related token-level metrics are represented. The data was calculated from the tokens, which get a PROPN label from the *magyarlanc* and/or which an I-PER, I-ORG, I-LOC or I-MISC label from the NER-tool. The criterion of getting a label[4] from both tool was not expected (so the results of the two systems was handled independently from this aspect).

It can be seen that the NER-tool consequently gets higher scores in all terms of metrics, while there is a remarkable difference in the accuracy between the text types respectively. The Law texts proved to be the less precisely predicted ones, while the best scores were achieved for Transcripts.

In the next sections, the three different genres will be analysed in a more detailed way.

[4] PROPN label form the *magyarlanc* and an I-TYPE from the NER-tool

| | NER-tool | *magyarlanc* | Sub-corpus |
|---|---|---|---|
| Precision | 83.10 | 69.51 | Forums |
| Recall | 51.75 | 50.00 | |
| F-score | 63.78 | 58.16 | |
| Precision | **94.48** | 63.22 | Transcripts |
| Recall | 70.26 | 56.41 | |
| F-score | 80.59 | 59.62 | |
| Precision | 63.33 | **26.67** | Laws |
| Recall | 73.08 | 61.54 | |
| F-score | 67.86 | 37.21 | |

Table 3: Precision, Recall and F-Score

## 5 Discussion

In this section, the detailed results of the analysis will be described from the aspect of the three sub-corpora.

### 5.1 Forums

In internet forums, nicknames may have almost unpredictable forms, capitalization, extent etc. The following examples represent some typical occurrences in the examined text:

(1)

| Token | POS assumed by *magyarlanc* | TYPE labeled by NER- tool |
|---|---|---|
| 55teki55 | PROPN | O |
| heidi1115 | NUM | O |
| ObudaFan | PROPN | I-ORG |

Some “multi-token” nicknames are listed here:

(2)

| Token | POS assumed by *magyarlanc* | TYPE labeled by NER- tool |
|---|---|---|
| Dr. | NOUN | I-PER |
| Attika | NOUN | I-PER |
| Kovács | PROPN | I-PER |
| _ | X | I-PER |
| Béla | PROPN | I-PER |
| _ | X | I-PER |
| Sándor | PROPN | I-PER |

It can be seen that these instances are not always properly identified but we should emphasize that the original training corpora of both tools did not contain instances of NEs like these specific ones.

Handling nicknames as NEs is a more interesting issue from a linguistic point of view. One of the arguments which can support considering nicknames as proper nouns is that they meet with the most fundamental properties of proper nouns mentioned in the literature.

Although we can see that there is no unified definition of proper nouns in the literature, but there are some common points between the definitions.

One of them is usually called as identifying function [12] of proper nouns. Nicknames which are used in websites admittedly fulfil this criterion, because this is the reason why people on websites even use it; to identify themselves with a unique linguistic unit, which only refers to one user. Furthermore, another point worth mentioning is the criterion that a linguistic unit can be called proper noun, if it does not change its referent within a given argumentation (as Kripke says) [13]. Nicknames fit into this expectation as well, since we can say that they usually define more accurately an individual, then a simple first name or last name (or even the two together)[5].

Moreover, from all of the NEs in the Forum sub-corpus, 69.29% (79 out of 114) was a mention of a nickname. All these justify that web nicknames should be seen as NEs.

At the same time, mentions of organisations can rise up questions about what is considered to be a proper noun. There are numerous instances where the same expression (which obviously refers to the same entity or object) occurs twice in the data; one with a capitalized first letter and one in lowercase:

(3) "...ez volt a legfőbb érve a **törvényszéknek**, hogy szabálytalanul lett kézbesítve az idézés."
*"... the main argument of the court of law was that the summon was delivered irregularly."*

But:

(4) "…...a végzés ellen fellebbezést nyújtsak be a várossal egy megyében található **Törvényszéknek** címezve 3 példányban."
*"...against the order, I should submit an appeal in 3 copies to the Court of Law, which is in the same county as the town."*

[5] Let's suppose that there is a class full of students. Although it is not likely, but possible, that there are more than one child in the room whose name is Tamás. It is less likely, but again, it is statistically possible, that there are more than one Kovács Tamás in the room. On the other hand, the list of First Names and Last Names in every language is a well-defined set of linguistic expressions (a definitely finite list). However, the potential combinations of characters (alphanumeric and special ones) are a more extensive set, therefore, the chance of having a unique nickname in a given site can be higher than having a unique name in a class (but indeed, it is not proved statistically yet). Moreover, a unique nickname is necessary in many websites.

In such cases, the two forms of mentioning these organizations are assumed to be distinct in the sense of what they refer to; the capitalized one is assumed to refer to a specific organization (e.g. in (4): Szegedi Törvényszék – Court of Law, Szeged) while the lowercase one is assumed to refer to the "type", or "role" of the organization (e.g. in (3): a type of court which can help you in this problem).

In the statistical data, only the capitalized mentions were included.

### 5.2 Transcripts

In the case of transcripts and in the output of *magyarlanc*, the most typical sources of errors may be related to the beginning of sentence. Within this, two typical problems occur most frequently.

In transcripts the main tool of discourse segmentation is the explicit marking of the speaker in the beginning of every utterance. These marks are abbreviations of the roles which the given person plays in that specific procedure, e.g. "V." stands for "vádlott" (*suspect*), "B." for "Bíró" (*judge*), "Ü." or "Ü / Ügyv." for "ügyvéd" (Lawyer) and so on. (5) is a typical case, where both the abbreviations are parsed incorrectly.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (5) | 1 | Ü | Ü | PROPN | Case=Nom\|Number=Sing | 0 | ROOT |
| | 2 | / | / | PUNCT | _ | 1 | PUNCT |
| | 3 | Ügyv | Ügyv | PROPN | Case=Nom\|Number=Sing | 1 | COORD |
| | 4 | : | : | PUNCT | _ | 1 | PUNCT |
| | 5 | Nem | nem | ADV | PronType=Neg | 1 | NEG |
| | 6 | . | . | PUNCT | _ | 0 | PUNCT |

"*Lawyer: No.*"

The remarkable majority (60.93%, 39 out of 64 instances) of falsely predicted PROPN labels was due to this phenomenon.

The other incorrectly predicted labels have miscellaneous reasons. For instance, it was frequent that the word "Bíró" (judge) at the beginning of the sentence was predicted to be a PROPN (because of the similar capitalization with the Hungarian surname: "Bíró").

Examining the false positive labels of the NER-tool, here we can see some examples for the falsely predicted tags:

| | | | | |
|---|---|---|---|---|
| (6) | a) | . | I-ORG | |
| | b) | Urat | I-PER | (*Sir, ACC*) |
| | c) | Interneten | I-ORG | (*on the Internet*) |

(6) a) is a clear case, while b) and c) are more interesting ones. The word *internet* originally had a capitalized and a lowercase version depending on the referent of the word (Internet as an "organization" or internet as a notion), while the title, "úr" (*sir*) can be attributed as a part of the former proper noun. For instance, if we mention a bare last name, like Kovács, the referent of it can be vague in some cases. If we have two names; Kovács úr (*Sir Kovács*) and a Kovács néni (*Mrs. Kovács*) then, without the title, we cannot decide clearly who the name Kovács actually refers to. In this case, the title can be considered as a part of the NE.

### 5.3 Laws

Both within the laws and transcripts, there are numerous mentions of paragraphs of laws, such as:

- Btk. 236 § (1), *(236§ (1) from the Penal Code)*
- Ptk 6: 494§ (2), *(494§ (2) from the Civil Code)*
- Tht 1§ (2), *(1§ (2) from the Act on Condominium buildings)*

As a convenience, only the name of the acts are considered to be a NE here (for instance; Btk., Ptk., Tht. from the aforementioned ones).

Within the current part of texts, the main reason behind the relatively low scores of *magyarlanc* may be traced back to two distinct sources. Firstly, many of the typographical elements devoted to determine items of lists are predicted to be proper nouns:

(7) "(3a) A (3) bekezdésben foglalt szankciókat..."
"(3a) Sanctions mentioned in the (3) paragraph…"

Example (7) illustrates one of the sentences where this happened: the token "3a" was predicted to be a PROPN. On the other hand, there were a negligible number of cases when real NEs were not predicted as a proper name.

The NER-tool's most conspicuous missed NE was "1952. évi III. Törvény a Polgári perrendtartásról" (*1952. 3rd Act on the Rules of the Court*), because it is fully missed. It is important to mention here that although the structure of the NE is actually very typical in the nomenclature of Laws (for example YYYY, Roman Numeral, Act on *something*), but if the tool did not have access to annotated instances like that, they are very hard to predict

### 5.4 Dependency relations

To get a full picture about the recognition of NEs, the last approach is the analysis of the multi-token NEs in the syntactic level of *magyarlanc*.

Table 4 represents the calculated metrics of the syntax-based NE labelling in each legal domain:

| | | Sub-corpus |
|---|---|---|
| Precision | 80.75 | Forums |
| Recall | 70.00 | |
| F-score | 74.99 | |
| Precision | 76.92 | Transcripts |
| Recall | 66.67 | |
| F-score | 71.43 | |
| Precision | 100.00 | Laws |
| Recall | 57.14 | |
| F-score | 72.73 | |

Table 4: Token-level metrics of syntactic parsing

However, it is important to note that the approach could identify much fewer multiword NEs than expected. In Table 5, the actual count of the NEs represented.

| Corpora | NEs labeled by the *magyarlanc* on the syntactic level | multi-token NEs manually annotated[6] |
|---|---|---|
| Transcripts | 23 | 63 |
| Forums | 15 | 20 |
| Laws | 7 | 7 |

Table5: Syntactic-level data

In the second column, the number of multi-token NEs can be seen, which get a *NE* tag from the *magyarlanc* at the syntactic level, while on the third column, the manually annotated multi-token NEs indicated.

The number of multi-token NEs (both in terms of manually annotated ones and which labelled by the *magyarlanc*) are much lower than expected originally, and not yet suitable for an exhausting statistical analysis.

Therefore, no general conclusions can be determined at this point and further investigations are needed.

## 6. Conclusions

In this paper, we focused on the identification of NEs in legal texts. We analyzed the output of the *magyarlanc* and the Szeged NER system and compared it to manually annotated NEs. The most typical sources of errors of the two negotiated NLP tools were also presented that cause most of the problems in the POS-tagging and in the NE-tagging approaches.

With regard to domain specificity, our investigations supported that all three sub-corpora had some unique peculiarities that need to be handled in order to get a higher rate of correctly recognized and classified NEs.

After a thorough examination, it turned out the multi-token NE category has a much lower presence in all investigated sub-corpora than expected, so a more massive amount of text should be analysed to be able to conclude more precise statements connected to the syntactic labelling efficiency.

## References

1. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In Proceedings of RANLP (2013) 763-771

[6] cf. Table2

2. Szarvas Gy., Farkas R., Kocsor A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: The Ninth International Conference on Discovery Science LNAI 4265 (2006)
3. Quaresma, P., Gonçalves, T.: Using Linguistic Information and Machine Learning Techniques to Identify Entities from Juridical Documents. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.): Number 6036 in Lecture Notes in AI. Springer-Verlag (2010) 44–59
4. Surdeanu, M., Nallapati, R., Manning, C.-D.: Legal claim identification: Information extraction with hierarchically labeled data. In: Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts (SPLeT) (2010)
5. Lenci, A., Montemagni, S., Pirrelli, V., Venturi, G.: Ontology learning from italian legal texts. In: Proceeding of the 2009 Conference on Law, ontologies and the Semantic Web: Channelling the Legal information Flood (2009)
6. Vincze, V., Farkas, R.: Tulajdonnevek a számítógépes nyelvészetben. In: Általános Nyelvészeti Tanulmányok XXIV (2012) 97-119
7. Móra, Gy., Vincze, V., Zsibrita, J.: Szófaji kódok és névelemek együttes osztályozása. In:Tanács, A., Vincze, V. (szerk.): VIII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem (2011) 131-142
8. Vincze V.: A Miskolc Jogi Korpusz nyelvi jellemzői. In: Szabó, M. (szerk.): A törvény szavai. Miskolc (2018)
9. Simon E., Farkas R., Halácsy P., Sass B., Szarvas Gy., Varga D.: A HunNER korpusz. In: Alexin Z., Csendes D. (szerk.): IV. Magyar Számítógépes Nyelvészeti Konferencia (2006) 373-376
10. Linguistic Data Consortium. ACE (automatic content extraction) English annotation guidelines for entities. https://www.ldc.upenn.edu/ collaborations/past-projects/ace, Version 5.6.6 2006.08.01. (2006)
11. 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), Geneva, Switzerland (2004)
12. Farkas, T.: A tulajdonnevek fordíthatóságáról és napjaink fordítási hibáiról, közszók és tulajdonnevek példáján. In: Névtani Értesítő 29 (2007) 167–188.
13. Kripke, S.: Naming and Necessity. Cambridge, Massachusetts: Harvard University Press (1980)
14. Várnai, J.-Sz.: A tulajdonnév a nyelvben és a nyelvészetben – A tulajdonnevek lehetséges megközelítéseiről, PhD értekezés, Debreceni Egyetem (2009)

# End-to-end Convolutional Neural Networks for Intent Detection

**Sevinj Yolchuyeva, Géza Németh, Bálint Gyires-Tóth**
{syolchuyeva, nemeth, toth.b}@tmit.bme.hu
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2., Budapest, 1111 Hungary

## Abstract

Convolutional Neural Networks (CNNs) have been applied to various machine learning tasks, such as computer vision, speech technologies and machine translation. One of the main advantages of CNNs is the representation learning capability from high-dimensional data. End-to-end CNN models have been massively explored in computer vision domain and this approach has also been attempted in other domains as well. In this paper, a novel end-to-end CNN architecture with residual connections is presented for intent detection, which is one of the main goals for building a spoken language understanding (SLU) system. Experiments on two datasets (ATIS and Snips) were carried out. The results demonstrate that the proposed model outperforms previous solutions.

**Keywords**: Spoken Language Understanding (SLU), intent detection, Convolutional Neural Networks, residual connections, deep learning, neural networks.

## 1 Introduction

Spoken dialogue systems are agents that are intended to help users to access information efficiently by speech interactions. Creating such a system has been a challenge for both academic investigations and commercial applications for decades. Spoken language understanding (SLU) is one of the essential components in spoken dialogue systems [1]. SLU is aiming to form a semantic frame that captures the semantics of user utterances or queries. The three major tasks in an SLU system are domain classification, intent detection, and slot filling. Intent detection can be treated as a semantic utterance classification problem [5,10]. Intent detection solutions classify speakers' intent and extract semantic concepts as constraints for natural language. Take a weather-related utterance as an example, "*Weather next year in Canada*", as shown in Figure 1. There are different slot labels for each word in the utterance and a specific intent for the whole utterance.


Fig. 1: Snips corpus sample with the utterance and slot annotation.

Slot filling can be formulated as a sequence labelling task [2,3]. Joint training of intent detection and slot filling models has been investigated [5,6]. The slot-gated SLU model, which incorporates attention and gating mechanism into the language understanding (LU) network was proposed by [5]. Moreover, conditional random field (CRF), introduced in [4], provides a framework for building probabilistic models to segment and label sequences and applies on different natural language processing (NLP) tasks (e.g., part of speech tagging, sentence classification, grapheme-to-phoneme conversion). Jointly modelling intent labels and slot sequences, thus, exploiting their dependencies by the combination of convolutional neural networks (CNN) and the triangular CRF model (TriCRF) can be beneficial [6]. With this approach, the intent error on Airline Travel Information System (ATIS) dataset was 5.91% for intent detection, and the F1-score was 95.42% for slot filling. Bidirectional Gated Recurrent Units (GRUs) could also be used to learn sequence representations shared by intent detection and slot filling tasks [9]. This approach employs max-pooling layer for capturing global features of a sentence for intent detection.

Recently, encoder-decoder neural networks (also referred to as sequence-to-sequence, or seq2seq models) have achieved remarkable success in various tasks, such as speech recognition, text-to-speech synthesis and machine translation [14,15,16]. In this structure, the encoder computes a latent representation of each input sequence, and the decoder generates an output sequence based on the latent representation. This type of network has been extended with attention mechanism [12,13] and applied to grapheme-to-phoneme conversion (G2P) [17]. Applying such models, intent detection and slot filling were also investigated [21,24]. The combination of the attention-based encoder-decoder architecture and alignment-based methods for joint intent detection and slot filling achieved 5.60% intent error on ATIS dataset [21].

In this work, we investigated CNN based residual networks for intent detection. Experiments were carried out on the ATIS and Snips dataset, which is widely used in SLU research. We show the effectiveness of the proposed models in different experimental settings. Using pre-trained Word2vec [25] and Glove [27] embedding also help to get comparable results. The remaining part of the paper is organized as follows. In Section 2, we introduce word embedding methods. In Section 3, the used datasets are described. In Section 4 the proposed method is introduced. Section 5 discusses the experiment setup and results on ATIS and Snips datasets. Section 6 concludes the work.

## 2 Word Embedding

Word embeddings are used for representing words as vectors. Word embedding models generated with tool, such as Word2vec (skip-gram and continuous bag-of-words (CBOW)) [25], and GloVe [27], generate word vectors based on the distributional hypothesis, which assumes that the meaning of each word can be represented by the context of the word. Continuous Bag-of-Words (CBOW) and Continuous Skip-gram models are still powerful techniques for learning word vectors [25]. CBOW computes the conditional probability of a target word given the context words surrounding it across a window with a predefined size. Skip-gram predicts the surrounding context words based on the central target word [25,28]. The context words are assumed to be located symmetrically to the target words within a distance equal to the window size in both directions. GloVe word embedding is a global log-bilinear regression model and is based on co-occurrence and factorization of the matrix in order to produce the word vectors.
Pre-trained word embeddings have proven to be highly useful in neural network models for NLP, e.g., in machine translation and text classification [11,19,26]. In this work, we used 300-dimension Word2vec[1] embeddings trained on Google News and 100-dimension GloVe[2] word embeddings trained on Wikipedia.

## 3 Dataset

We used the Airline Travel Information System (ATIS)[3] dataset, which has been frequently chosen by various researchers [5,11,38]. The dataset contains audio recordings from people making flight reservations. The training set contains 4,478 utterances, the test set contains 893 utterances, and 500 utterances are used for validation (referred to as development set in the paper). Besides ATIS, Natural Language Understanding[4] benchmark dataset was also used. This balanced dataset is collected from the Snips personal voice assistant; the number of samples for each intent is approximately the same. The training set contains 13,084 utterances, the test set contains 700 utterances, and 700 utterances as validation data (development set). All words are labelled with a semantic label in a BIO format, which 'B' means to begin, 'I' means inside, 'O' is outside. Words which don't have semantic labels are tagged with 'O'. For example, 'Weather next year in Canada' contains five words, and these words are labelled according to Figure 1. The sequence 'next year' is labelled as B-timeRange and I-timeRange and 'Canada' is tagged as B-country. The rest of the words in the utterance are labelled as 'O'.

[1] https://code.google.com/archive/p/word2vec/, Accessed: 14th November, 2018
[2] https://github.com/stanfordnlp/GloVe, Accessed: 14th November, 2018
[3] https://github.com/MiuLab/SlotGated-SLU/tree/master/data/atis, Accessed: 14th November, 2018
[4] https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines, Accessed: 14th November, 2018

There are 120 slot labels and 21 intent types in ATIS; there are 72 slot labels and 7 intent types in Snips dataset. Vocabulary size of these datasets is 722 and 11,241 in ATIS and Snips, respectively. Compared to single-domain ATIS dataset, Snips is more complicated, mainly due to the intent diversity and large vocabulary. The intent diversity of ATIS and Snips dataset are shown in Table 1 and Table 2.

| Type of intent | Number |
|---|---|
| PlayMusic | 1914 |
| GetWeather | 1896 |
| BookRestaurant | 1881 |
| RateBook | 1876 |
| SearchScreeningEvent | 1851 |
| SearchCreativeWork | 1847 |
| AddToPlaylist | 1818 |

Table 1: The number of intents in the training data of Snips.

| Type | Number |
|---|---|
| atis_flight | 3309 |
| atis_airfare | 385 |
| atis_ground_service | 230 |
| atis_airline | 139 |
| atis_abbreviation | 130 |
| atis_aircraft | 70 |
| atis_flight_time | 45 |
| atis_quantity | 41 |
| atis_flight#atis_airfare | 19 |
| atis_city | 18 |
| atis_distance | 17 |
| atis_airport | 17 |
| atis_ground_fare | 15 |
| atis_capacity | 15 |
| atis_flight_no | 12 |
| atis_meal | 6 |
| atis_restriction | 5 |
| atis_airline#atis_flight_no | 2 |
| atis_aircraft#atis_flight#atis_flight_no | 1 |
| atis_cheapest | 1 |
| atis_ground_service#atis_ground_fare | 1 |

Table 2: The number of intents in the training data of ATIS.

The intents in Snips are diverse and balanced. The maximal number of utterances are in the PlayMusic domain, the least number of utterances are in AddToPlaylist. The intent types in ATIS are unbalanced. For example, the intent atis_flight equals about 73.8% of training data, while there are intents with one utterance only. Intents with small number of occurrences were excluded from training and evaluation (e.g.

atis_day_name, atis_airfare#atis_flight, atis_flight#atis_airline, atis_flight_no#atis_air line).

# 4 Proposed Work

This section first explains CNN and then introduces the proposed end-to-end CNN approach with residual connections for intent classification.

## 4.1 Convolutional Neural Networks for Intent detection

The architecture of an ordinary CNN is composed of different types of layers (such as the convolutional layers, pooling layers, fully connecting layers, etc.) [34] where each layer realizes a specific function. The convolutional layers are for representation learning, while the fully connected layers on the top of the network are for modelling a classification or regression problem. Convolutional neural networks are jointly performing representation learning and modelling, which makes these models superior to other methods in many cases. Weight sharing in the convolutional layers is essential for the model to become spatially tolerant: similar representations are learned in different regions of the input, and the total number of parameters can also be reduced drastically.
Increasing the number of layers in deep CNNs does not implicitly results in better accuracy, and some issues, such as vanishing gradient and degradation problems may arise as well. Introducing residual connection can improve the performance significantly [29]. These kinds of connections allow the information and gradients to flow more into the deeper layers, increases the convergence speed and decreases the vanishing gradient problem.
Convolutional neural networks were already successfully applied to various NLP tasks [33,38,39]. These results suggest investigating CNN based sequence models for intent classification. We expected that convolutional neural networks enhances the performance of intent detection task.

## 4.2 Model architecture

All utterances and their slots sequences are splatted as Input 1 and Input 2. We use <BOS> and <EOS> tokens as beginning-of-utterances and end-of-utterances tokens in Input 1 and beginning-of-slots and end-of-slots tokens in Input 2, as shown in Table 3.

| **Input 1** | **Input 2** | **Output** |
|---|---|---|
| <BOS> weather next year in Canada <EOS> | <BOS> O B-timeRange I-timeRange O B-country <EOS> | GetWeather |

Table 3: The structure of input and output.

Regarding Input 1, an embedding layer with pretrained word vectors, such as Word2vec or GloVe, was applied. Regarding Input 2, the slots were tokenized, and

embedding was applied, which is intended to map positive integer values in an array to float values. The proposed model was applied on both inputs separately, and then the output of these models (referred to as Model 1 and Model 2) are combined (see Figure 2). Model 1 and Model 2 contains convolutional layers with residual connections. After embedding, a 1D convolutional layer with 16 filters is applied, which is followed by a stack of residual blocks. Through hyperoptimization, the best result was achieved by 3 residual blocks, and the number of filters in each residual block was 32, 64, 128. Each residual block consists of 2 convolutional layers followed by batch normalization layer [32] and ReLU activation. The filter size of all convolutional layers is 5. These blocks are followed by one more batch normalization layer and a ReLU activation. The architecture ends with a fully connected layer coupled with softmax activation function. The model architecture is shown in Figure 3.

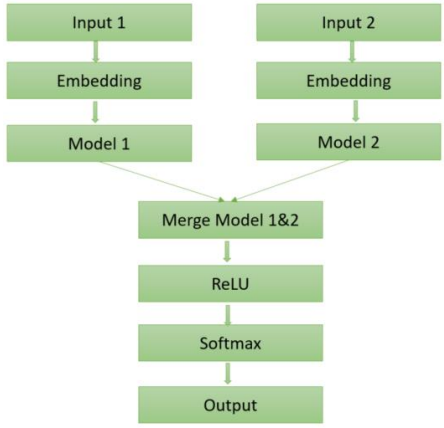



Fig. 2: Proposed model architecture.

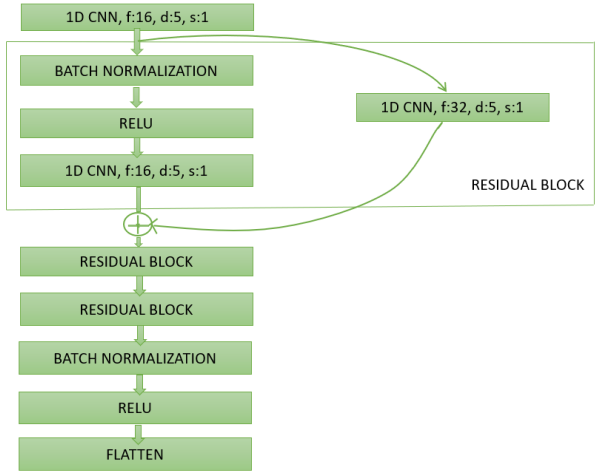



Fig. 3: End-to-end CNN structure for intent detection task. f, d, and s are the number of the filters, length of the filters and stride, respectively.

In general, using CNN for intent detection is similar to a standard classification problem, ATIS dataset is under the flight reservation domain with 17 intents, Snips with 7 intents.

## 5 Evaluation and Results

We used NVidia Titan Xp (12 GB) and NVidia Titan X (12 GB) GPU cards hosted in two i7 workstations with 32GB RAM. For training and inferrrence the Keras deep learning framework with Theano [30] backend was used.
We trained the models both with Word2vec and Glove vector representations.
After training the models predictions were performed on the test dataset and the results were evaluated with confusion matrices and accuracy.
The results of the experiments are shown in Table 4. We compared our solution with state-of-the-art intent detection models, such as Slot-Gated (Intent Attention) [5], Attention-based BiRNN [22], and Recursive Neural Network [36] models. Better results by using different approaches are also published, but in those cases different variations or parts of the ATIS dataset were used [17,23]. In Table 4, the first column shows the applied architecture models; the second and third columns show overall accuracy for each model on ATIS and Snips dataset. For Snips, we are able to get 100% accuracy using pretrained Glove vectors on the test set.
The confusion matrix is an effective method to visualize and to examine the performance of binary and multi-class classifiers [34]. Generally, the confusion matrix shows the detailed number of correctly classified and misclassified intents. The diagonal represents the correct predictions. Each entry outside the diagonal shows how many tokens from each intent (y-axis) were incorrectly assigned to other intents (x-axis).
Figure 4 shows the confusion matrix of the proposed model using GloVe pretrained vectors on ATIS dataset. The intent atis_flight is 73.8 % of training dataset and it is the most part of test dataset too. 629 utterances were classified correctly out of 630. The number of utterances in atis_restriction is zero in test data. Figure 5 and Figure 6 show the confusion matrix of the proposed model using GloVe and Word2vec pretrained vectors on Snips dataset, respectively. In Figure 5, the proposed model correctly classified 629 utterances out of 661 for the atis_flight and 46 out of 51 for the atis_airfare intent. The accuracy of these intents is 95.2 and 90.2%, respectively. More than half of the test utterances of atis_distance and atis_meal were misclassified. In Figure 5, all intents are correctly classified for Snips test dataset by using GloVe pretrained vectors.

| **Model** | **ATIS** | **Snips** |
|---|---|---|
| Slot-Gated (Intent Attention) [5] | 94.1 | 96.8 |
| Attention-based BiRNN [22] | 92.6 | - |
| Recursive Neural Network [36] | 95.40 | - |
| **Word2vec + CNN with residual connections (proposed work)** | 95.46 | 99.7 |
| **Glove + CNN with res. Con. (proposed work)** | 94.40 | 100 |

Table 4: The accuracy (%) of previous works and the proposed models on ATIS and Snips test datasets

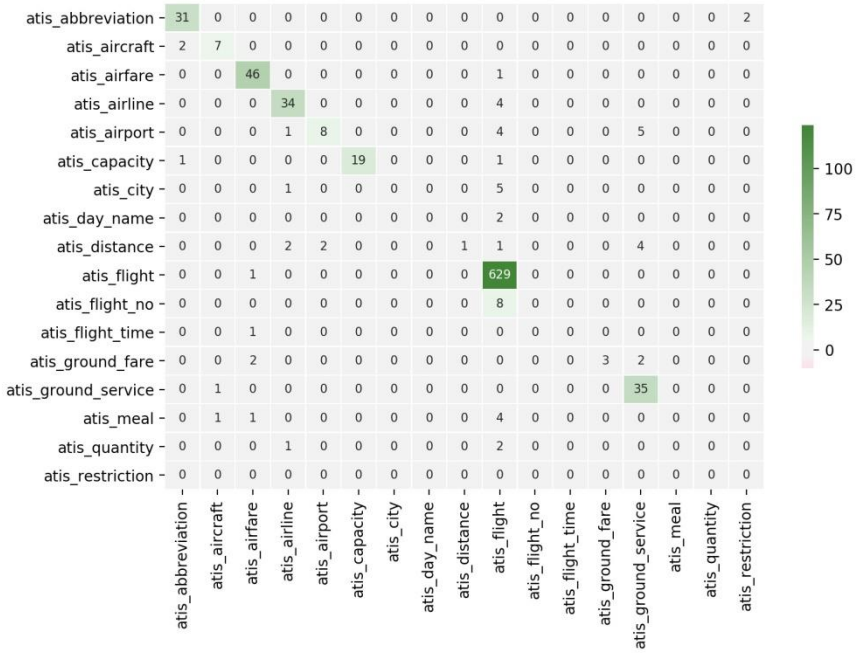



Fig. 4: Confusion matrix of ATIS test dataset by using GloVe pretrained vectors.

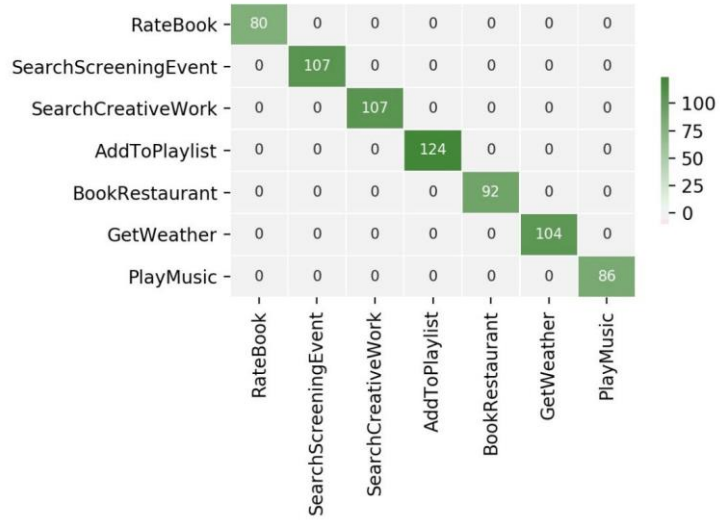



Fig. 5: Confusion matrix of Snips test dataset by using GloVe pretrained vectors.

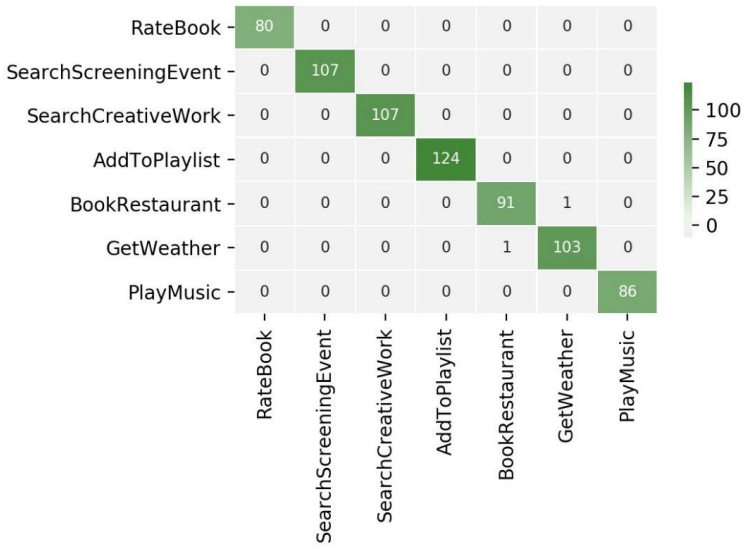



Fig. 6. Confusion matrix of Snips test dataset by using Word2vec pretrained vectors.

## 6 Conclusions and Future Work

In this paper, an end-to-end CNN model with residual connections for intent detection were proposed. 300-dimensional Word2vec embeddings pretrained on Google News and 100-dimension GloVe word embeddings pretrained on Wikipedia were used for word representations. The results were evaluated with the help of confusion matrix and accuracy. The proposed method outperformed previous solutions in terms of accuracy.

## Acknowledgements

The research presented in this paper has been supported by the BME-Artificial Intelligence FIKP grant of Ministry of Human Resources (BME FIKP-MI/SC), by Doctoral Research Scholarship of Ministry of Human Resources (ÚNKP-18-4-BME-394) in the scope of New National Excellence Program, by János Bolyai Research Scholarship of the Hungarian Academy of Sciences, by the VUK project (AAL 2014-183), and the DANSPLAT project (Eureka 9944). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

[1] Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B., and Bengio, Y. (2018). Towards End-to-end Spoken Language Understanding. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5754-5758.

[2] Wang, Y., Shen, Y., and Jin, H. (2018). A Bi-Model Based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling. Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 309-314.

[3] Tur, G., and Mori, R.D. (2011). Spoken language understanding: Systems for extracting semantic information from speech. John Wiley & Sons.

[4] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Machine Learning-International Workshop Then Conference, 282–289.

[5] Goo, C., Gao, G., Hsu, Y., Huo, C., Chen, T., Hsu, K., and Chen, Y. (2018). Slot-Gated Modeling for Joint Slot Filling and Intent Prediction. Proceedings of Annual Conference North American Chapter of the Association for Computational Linguistics, 753-757.

[6] Xu, P., and Sarikaya, R. (2013). Convolutional neural network based triangular CRF for joint intent detection and slot filling. In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 78–83.

[7] Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., and Zweig, G. (2015). Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3):530–539.

[8] Meng, L., and Huang, M. (2018). Dialogue Intent Classification with Long Short-Term Memory Networks. Natural Language Processing and Chinese Computing. Ed. by X. Huang et al. Cham: Springer International Publishing, 42–50.

[9] Zhang, X., and Wang, H. (2016) A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding. International Joint Conferences on Artificial Intelligence, 2993–2999.

[10] Liu, B., and Lane, I. (2016). Joint Online Spoken Language Understanding and Language Modeling with Recurrent Neural Networks. 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), 22-30.

[11] Kim, J., Tür, G., Çelikyilmaz, A., Cao, B., and Wang, Y. (2016). Intent detection using semantically enriched word embeddings. 2016 IEEE Spoken Language Technology Workshop (SLT), 414-419.

[12] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[13] Luong, M.T., Pham, H., and Manning, C.D. (2015). Effective approaches to attention-based neural machine translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1412–1421.

[14] Kalchbrenner, N., and Phil, B. (2013). Recurrent Continuous Translation Models. Proceedings of Conference on Empirical Methods in Natural Language Processing, 1700–1709.

[15] Cho, K., Bart, M., Caglar, G., Dzmitry, B., Fethi, B., Holger, S., and Yoshua, B. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of Conference on Empirical Methods in Natural Language Processing, 1724-1734.

[16] Lu, L., Zhang, X., and Renals, S (2016). On Training the Recurrent Neural Network Encoder-Decoder for Large Vocabulary End-to-End Speech Recognition. IEEE International Conference on Acoustics, Speech and Signal Processing, 5060–5064.

[17] Toshniwal, S., and Livescu, K. (2016). Jointly learning to align and convert graphemes to phonemes with neural attention models. IEEE Spoken Language Technology Workshop (SLT), 76-82.
[18] Hashemi, H.B. (2016). Query Intent Detection using Convolutional Neural Networks. In International Conference on Web Search and Data Mining, Workshop on Query Understanding.
[19] Wang, P., Qian, Y., Frank K. Soong, He, L., and Zhao, H. (2015). Word embedding for recurrent neural network based TTS synthesis. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4879–4883.
[20] Ravuri, S., and Stolcke, A. (2015). A Comparative Study of Neural Network Models for Lexical Intent Classification. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 368–374.
[21] Liu, B., and Lane, I. (2016). Attention-based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. Proceedings of the 17th Annual Meeting of the International Speech Communication Association, 685-689.
[22] Hakkani-Tur, D., Tur, G., Celikyilmaz, A., Chen, Y.N., Gao, J., Deng, L., and Wang, Y.Y. (2016). Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM. In Proceedings of the 17th Annual Meeting of the International Speech Communication Association, 715-719.
[23] Zhang, X., and Wang, H. (2016). A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding. International Joint Conferences on Artificial Intelligence (IJCAI), 2993–2999.
[24] Schumann, R., and Angkititrakul, P. (2018). Incorporating ASR Errors with Attention-based, Jointly Trained RNN for Intent Detection and Slot Filling. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),685-689.
[25] Mikolov, T., Corrado, G., Chen, K., and Dean J., (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR), 1–12.
[26] Qi, Y., Sachan, D.S., Felix, M., Padmanabhan, S.J., and Neubig, G. (2018). When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?. Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics 2018 (NAACL-HLT), 529–535.
[27] Pennington, J., Socher, R., and Manning, C.D. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1532–1543.
[28] Young, T., Devamanyu. H., Soujanya, P., and Cambria, E. (2017). Recent Trends in Deep Learning Based Natural Language Processing. arXiv: preprint arXiv:1708.02709v4.
[29] Kaiming, H., Xiangyu, Z., Shaoqing, R. and Jian, S. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 770-778.
[30] The Theano Development (2016). A Python framework for fast computation of mathematical expressions, arXiv preprint arXiv:1605.02688.
[31] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going Deeper with Convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1–9.
[32] Ioffe, S., and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning. 448-456.
[33] Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text Classification Improved by Integrating Bidirectional LSTM with Two-Dimensional Max Pooling. In Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016), 3485–3495.

[34] Yolchuyeva, S., Németh, G. and Gyires-Tóth, B. (2018) Text normalization with convolutional neural networks. International Journal of Speech Technology, Volume 21, Number 3, 589-600.
[35] Sonmez, C., and Ozgur, A. (2014). A Graph-Based Approach for Contextual Text Normalization. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 313–324.
[36] Guo, D., Tür, G., Yih, W., and Zweig, G. (2014). Joint semantic utterance classification and slot filling with recursive neural networks. 2014 IEEE Spoken Language Technology Workshop (SLT), 554-559.
[37] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y.N. (2017). Convolutional Sequence to Sequence Learning. arXiv preprint arXiv: 1705.03122.
[38] Gehring, J., Auli, M., Grangier, D., and Dauphin, Y.N. (2016). A Convolutional Encoder Model for Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 123-135.
[39] Xiang, Z., Zhao, J., and Yann, L. (2015). Character-Level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015), 1–9.

# An annotation tool for academic literature processing

Molnár Zsolt[1], Polgár Tímea[1], Vincze Veronika[1,2,3]

[1]ScienceBoost Kft.

[2]Szegedi Tudományegyetem, Informatikai Tanszékcsoport

[3]MTA-SZTE Mesterséges Intelligencia Kutatócsoport

zsolt.molnar@hubscience.com; timea.polgar@hubscience.com

vinczev@inf.u-szeged.hu

**Abstract:** In this paper, we present our annotation tool that facilitates research and annotation work by quick, yet efficient literature processing. Our tool helps users create a unique and refined collection of linked information, which can lead to more effective and faster decisions in research. The tool is currently optimized for biomedical domain, but it can adapted to other academic fields with minimal efforts.

## 1 Introduction

Medical institutes usually store considerable amount of valuable information (patient data) as free text. Such information has a great potential in aiding research related to diseases or improving the quality of medical care. The size of document repositories makes automated processing in a cost-efficient and timely manner an increasingly important issue. The intelligent processing of clinical texts is the main goal of Natural Language Processing (NLP) [1] for medical texts.

In order to provide supervised NLP solutions for medical and clinical text mining, there is an intense need for annotated texts. There already exist a number of annotation tools within the community and on the market, which help the researcher collect and annotate relevant data. For instance, [4] and [5] provide a comparison of annotation tools for the biomedical domain, while [3] lists several annotation tools and compares them among parameters such as type of client, content of annotation, applied tags and attributes, annotation format etc.

Some of these annotation tools contain built-in machine learning (ML) methods e.g. for automatically annotating drugs in medical reports [6] or recognizing genes in biomedical articles. These annotators excel on the specific field they were developed for, but would provide poor performance on general texts. These are mostly based on supervised ML methods, in other words, training of the ML model requires domain-specific corpora manually annotated by experts, which can be very expensive. The high costs associated with this approach has led to a shift towards unsupervised or semi-supervised ML methods that, instead of manually labeled data, rely on human expertise encoded in expert-curated knowledge bases [2].

In this paper, we present our annotation tool that facilitates research and annotation work by quick, yet efficient literature processing. With our tool, users can create a unique and refined collection of linked information, which can yield more effective and faster decisions in research.

## 2 The annotation tool

Our work is designed to facilitate research by quick, yet efficient literature processing. This is the driving force behind our work to provide a teamwork-driven, AI-powered literature survey primarily in life sciences but our solutions can be adapted to any academic field.

The tool can be used online from a browser and it helps researchers build up their own knowledge graph over a specific topic while reading scientific publications. In order to reach this goal, an expert needs to do the first steps manually (manual annotation), and based on these annotated data, the algorithm can be automatically trained, hence the rest of the documents are annotated automatically. Documents and the manual annotation task can be shared with other colleagues (see below) to accelerate the training procedure.

In the following subsections, we report the typical annotation process and functionalities of our tool.

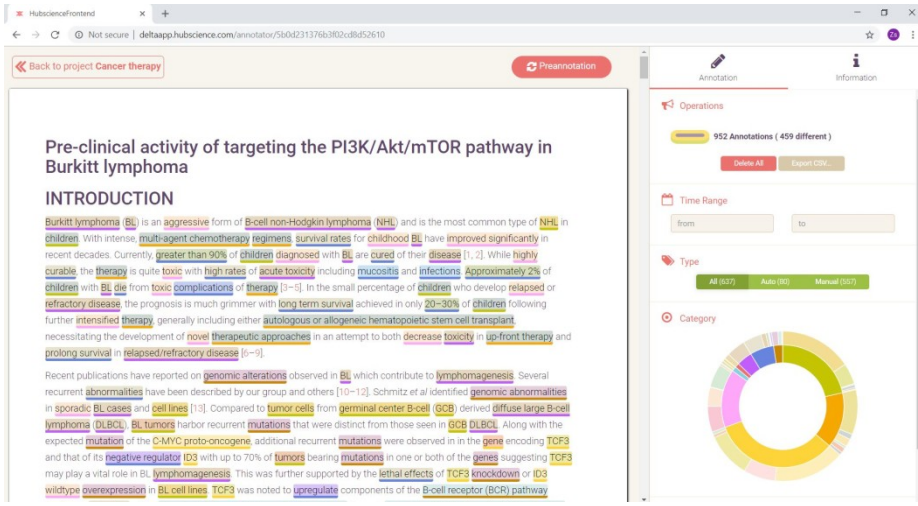



Fig. 1: The annotation interface.

### 2.1 Document Management

In order to be able to annotate, first there is a need for available documents. Documents can be added to the tool in multiple formats:

- Online Journal: The website reference (URL from the browser address bar) can be added and the system will download and insert the full text article for annotation.
- Upload: local PDF or HTML files can also be uploaded from the local computer.
- New Document: new notes or copy-paste fragments from any source can be also added to the user interface for annotation.

### 2.2 Annotation

Documents can be annotated manually and automatically as well. During manual annotation, the annotation categories are visualized above the selected text, and the appropriate one can be chosen by clicking on it.

The preannotation icon will allow users to tag all the text elements, which have been already added to the system. The system can also manage dictionaries containing lists of text elements, these can also be added to the system.

Dictionaries are built up from wordcards and wordpacks (see Fig. 2). The wordcard contains the term, its synonyms, its abbreviations as the most important properties. Additional properties can also be added if needed. Wordcards can be organized and grouped into wordpacks, i.e. lists of word cards that belong together. Typically a wordpack is domain specific. Users can also define their own wordpacks but the tool inherently contains some biomedical wordpacks.

Within the texts, an annotated item can be connected to several other annotations (see Fig. 3). They do not necessarily need to be in the same text, they can be linked to each other in different documents too. The system is able to build up a knowledge graph over a topic (see Fig. 4). The knowledge graph is a bunch of connected information that will help the researcher derive new knowledge on the basis of existing annotations.

### 2.3 Category system

Our tool offers an annotation category system for biomedical text annotation. However, this category system is fully customizable and can be modified according to the needs and the topic that user is interested in. New categories and new subcategories can be added, edited, or removed (see Fig. 5).

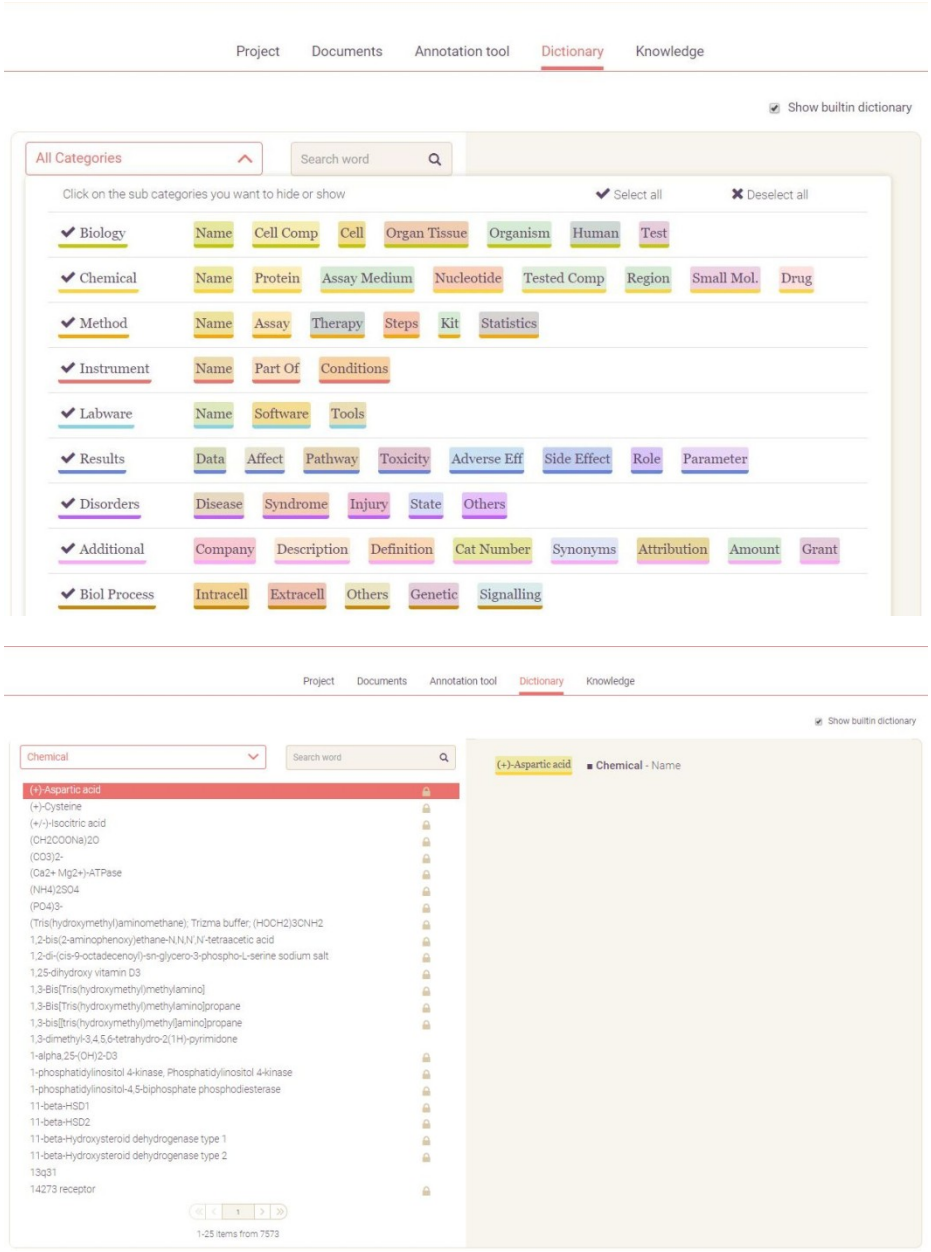



Fig. 2: Dictionary panels.

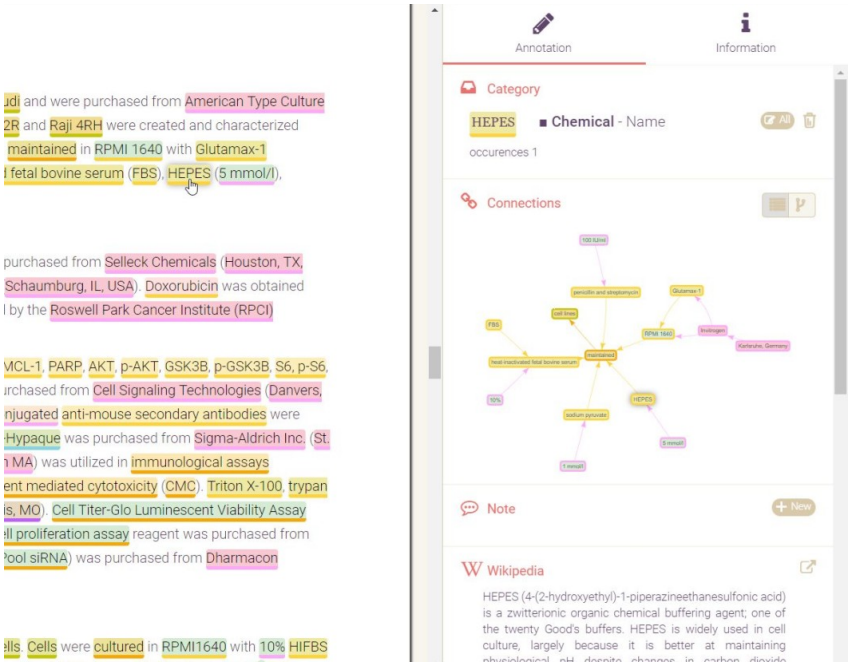



Fig. 3: Connections in an annotated documents.

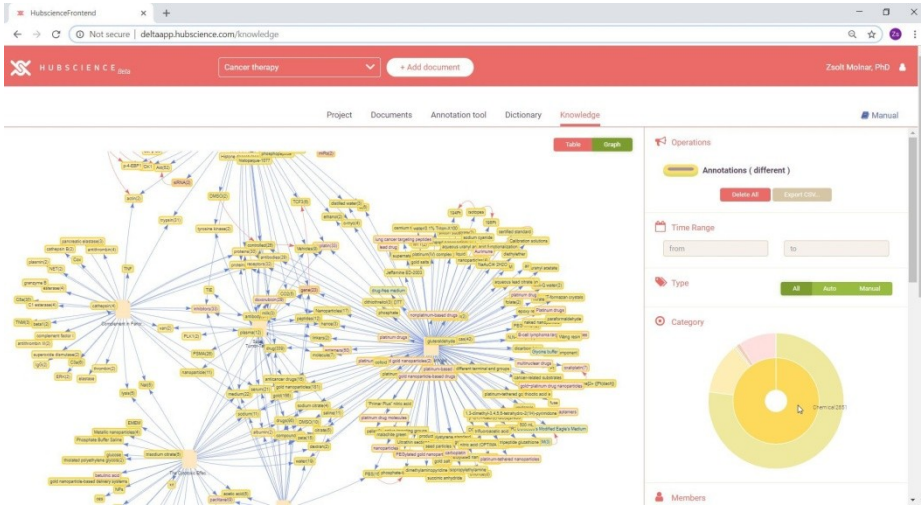



Fig. 4: A knowledge graph.

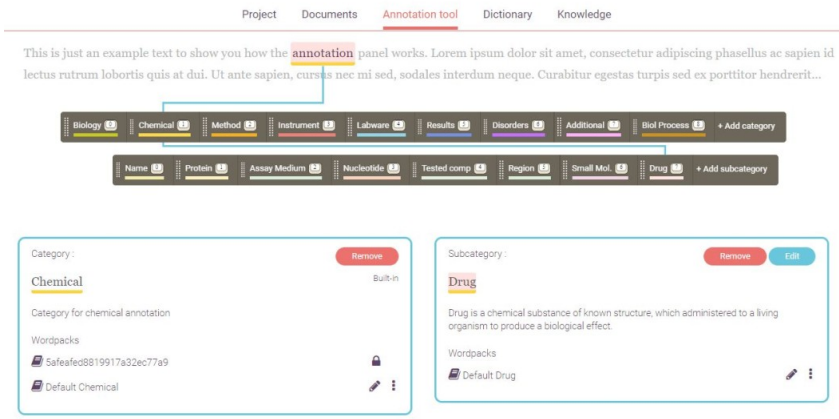



Fig. 5: A category panel.

## 2.4 Working with projects

Documents can be organized into projects. Working with projects allows the users to perform the analysis in an organized way and the machine learning algorithm can be trained directly on topics.

The main advantages of using projects can be summarized as follows:

- Projects allow collaborative work, i.e. several colleagues can work together on the same set of documents
- Documents belonging to the same topic can be organized within one project
- The annotation category system can be customized for each project
- Special dictionaries can be employed for each project
- Annotation statistics/analytics can be aggregated for the whole project, i.e. documents belonging to the same topic

Within projects, members can have multiple roles: owner, admin, or member role. The owner can do everything, admin can invite others and edit the category system, the member can only annotate.

## 2.5 Info Panel

The info panel shows relevant general statistical and meta-information of the document (see Fig. 6). In addition, during annotation the selected word or text element is looked up in Wikipedia or Wikidata providing more information about the specific terms helping students or users inexperienced at a given topic (see Fig. 7).

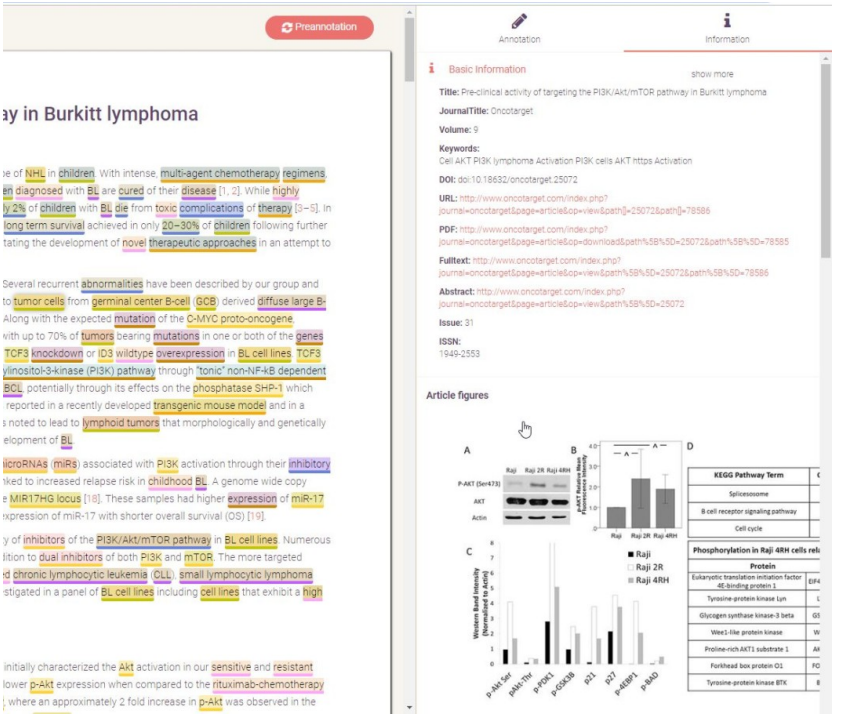



Fig. 6: An info panel with basic information on the article.

### 2.6 Other functionalities

In addition to token-level annotations, users can also apply labels for the whole document. For instance, users can rate documents on a scale of 1 to 10 by relying on their own experience in evaluating scientific information (see Fig. 8). This might help other users to decide whether they need that document or not.

Relevance scores can also be added to the documents (High, Medium or Low relevance) by relying on the user's criteria such as how relevant they can be for them in a given project or topic.

Moreover, any other notes, comments or reminders can be directly added to the annotated information and web contents may also be linked to it.

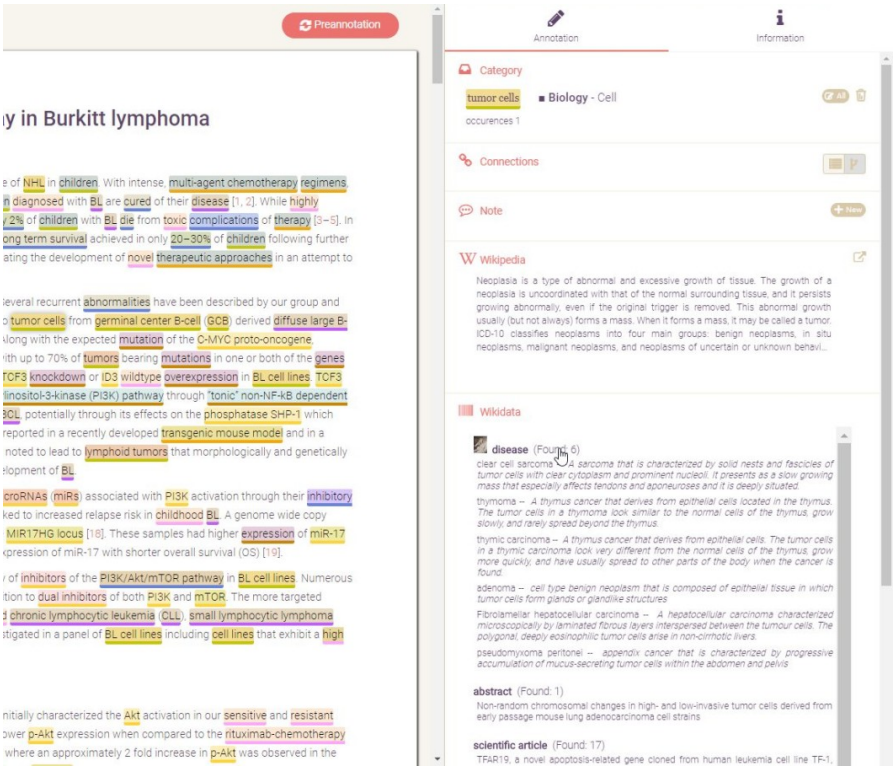


Fig. 7: An info panel with information from Wikipedia.

## 3 Advantages of the tool

We believe that our annotation tool can be fruitfully exploited in several fields of research, and as such, several groups of users can profit from it. For instance, it can be used as supporting material for annotation for researchers and students to stay competitive at universities and research institutions. The tool can also provide an effective way for sharing an annotated literature survey between team members of the same research group, leveraging the power of teamwork.

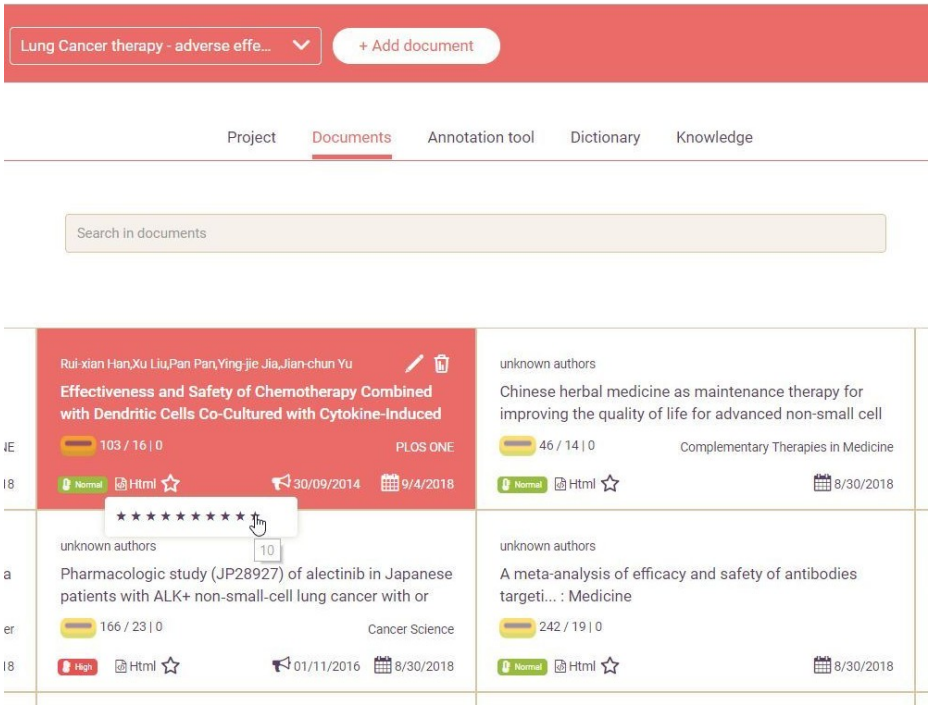



Fig. 8: Rating an article.

The tool facilitates smart ontology building for specific areas. It also offers a way to personalize the annotation labels, hence data curation may become easier and faster. Finally, it offers an easy to use software interface for visualizing annotations and their relations, thus enabling the discovery of novel academic achievements.

## 4 Availability

The basic version of the tool is available for everyone free of charge at our website (www.hubscience.com).

The tool is currently optimized for biomedical domain, but it can be adapted to other academic fields with only minimal efforts.

## References

1. Ananiadou, S., Mcnaught, J.: Text Mining for Biology and Biomedicine. Artech House, Inc., Norwood, MA, USA (2005)

2. Chasin, R., Rumshisky, A., Uzuner, O., Szolovits, P.: Word sense disambiguation in the clinical domain: a comparison of knowledge-rich and knowledge-poor unsupervised methods. Journal of American Medical Informatics Association, Vol. 21 (2014) 842-849
3. http://knot.fit.vutbr.cz/annotations/comparison.html
4. Médigue, C., Moszer, I.: Annotation, comparison and databases for hundreds of bacterial genomes. Research in microbiology, Vol. 158, No. 10 (2007) 724-736
5. Neves, M., Leser, U.: A survey on annotation tools for the biomedical literature. Briefings in bioinformatics, Vol. 15, No. 2 (2012) 327-340
6. Tikk, D., Solt, I.: Improving textual medication extraction using combined conditional random fields and rule-based systems. Journal of American Medical Informatics Association, Vol. 17 (2010) 540-544

# Formális fogalmak a jogi ontológiákban

Syi,[1] Hamp Gábor,[1] Markovich Réka,[2,3,4] Grad-Gyenge Anikó,[3]
Héder Ákos,[5] Nagy Krisztina,[3] Vértesy László[3,6]

[1] BME Szociológia és Kommunikáció Tanszék
i@syi.hu,hampg@eik.bme.hu
[2] Computer Science and Communications Research Unit, University of Luxembourg
markovich.reka@yahoo.com
[3] BME Üzleti Jog Tanszék
{a.gyenge, nagy.krisztina}@eik.bme.hu
[4] ELTE Logika Tanszék
[5] Közbeszerzési Hatóság
akos.heder@gmail.com
[6] BME Pénzügyek Tanszék
vertesy@finance.bme.hu

**Kivonat:** Jelen tanulmányunkban az egyes jogi ontológiákban használt vagy használni érdemes „csúcsfogalmakat” elemezzük. Mint látni fogjuk ezek nem kizárólag a jogi minőségre tisztán és közvetlenül reflektáló fogalmak (pl. kötelezettség, jogosultság), hanem olyan általános ontológiai kategóriák, amelyeknek speciális szerepe vagy minősége van a jogi ontológiák kialakításakor.

## 1 Bevezetés

A jogszabályok gépi értelmezéséhez meg kell tanítanunk a gépet a jogszabályokban használt fogalmak szemantikájának kezelésére. Ennek eszköze formális ontológiák építése. Az ontológiákat érdemes két részre bontani, és elkülöníteni a legáltalánosabb, minden szakterületen alkalmazható, illetve a szakterület-specifikus kategóriákat (csúcs- és domain-ontológiai fogalmakat) egymástól. Jelen tanulmányunkban azokat a formális fogalmakat mutatjuk be, amelyek az általunk használt csúcsontológia körébe tartoznak. A kutatásunk során ezeket a fogalmakat alkalmazzuk konkrét jogszabályok szakontológiájának felépítéséhez. Egy kapcsolódó másik tanulmányban a családtámogatási törvény szakontológiáját bontjuk ki [5], az itt bemutatott csúcsontológiát azonban használni tudjuk más jogszabályok (például a kresz) ontológiájának feltárásakor is.

## 2 Az ontológiaépítéshez szükséges formális fogalmak

Minden fogalmi rendszer, minden ontológia építésekor kezelni kell azt a tényt, hogy a világban létező jelenségeket megragadhatjuk általános és egyedi szinten egyaránt. Ezt a kettősséget a filozófiában bevett **univerzálé-individuum** fogalompár [11] segítségével kezelhetjük (más szakterületen ugyanez a szembeállítás nyilvánul meg az osztály/típus vs. partikuláré/példány/instancia/előfordulás fogalompárban). A jog világá-

ban ez a kettősség abban érhető tetten, hogy amíg a jogszabályszövegek - mivel a jövőre nézve szabályoznak - általános fogalmakat használnak: törvényi tényállásokban rögzítik azokat a tényezőket, amelyek alapján bizonyos eseményeket/cselekvéseket adott jogkövetkezményekkel kapcsolnak össze; addig a jogalkalmazásban, államigazgatási ügyintézésben mindig egyedi eseteket kezelnek, konkrét esetekről, személyekről, jogosultságoktól, kötelezettségekről van szó.

A csúcsontológia tetején az **entitás** fogalma áll. Minden, ami létezik, valamilyen entitás. Az entitások egyik típusa a **dolog** vagy objektum, amellyel önmagában létező entitásokat ragadhatunk meg. Hogy ez mit jelent pontosan, azt a **egzisztenciális dependencia** fogalmával definiálhatunk [23]. A dolgok azok a létezők, amelyek önmagukban, másoktól függetlenül léteznek. A természeti fajták (ember, állat, növény, ház, autó stb.) mind ilyenek. Ezekkel állnak szemben azok a létezők (illetve azok a fogalmak, amelyeket ilyen minőségekre alkalmazunk), amelyek valahogyan függenek a minőség hordozójától (vagy hordozóitól). Az egyik ilyen minőség a **tulajdonság**, amely a dolgokban „egyedileg" van meg: a tulajdonság mindig függ attól a dologtól, amelyben inherens módon benne van (tehát egzisztenciálisan függ más entitástól). Az alma színe az almában nyilvánul meg, nem önmagában, tehát a szín függ az almától, de ez a függés mindig csak egy almára igaz, nem két alma viszonyára. A **reláció** fogalmával viszont olyan függést írhatunk le, amely kettő (vagy több) dolog közti viszonyban érhető tetten. A *házastársa* reláció két ember közti kapcsolatot fejez ki, nem tudjuk a két fél (a férj vagy a feleség) nélkül értelmezni ezt a fogalmat.

A relációban összekapcsolt entitásokat a reláció argumentumainak (paramétereinek) nevezzük, és ezekkel kapcsolatban fontos kérdés, hogy milyen számosságokat engedünk meg a paraméterek helyébe írható előfordulásokra vonatkozóan.

A relációknak is lehet minőségeket tulajdonítani (reflexivitás, szimmetricitás, tranzitivitás stb.), és a relációk között is lehet relációkat értelmezni (inverz, komplementer, azonos stb.). (A matematikában gyakran használnak relációs fogalmakat, de ezeket itt nem mutatjuk be, csak használjuk.)

Egy csúcsontológiában kezelni kell eseményfogalmakat is. Az ontológiai szakirodalomban elkülönítik egymástól a dologszerű és eseményszerű (endurant és perdurant, vagy continuant és occurant) entitásokat [8]. Bár a családtámogatási folyamatban sokféle eseményt lehet találni, az ontológiánkban most nem eseményfogalmakat kezelünk, hanem az események során konstruált államigazgatási nyilvántartások és adatok alapján hozunk létre szerepfogalmakat.

Minden ontológiának központi kategóriája a **generikus alárendeltje** reláció, ami egy másodrendű fogalom, hiszen két univerzálé között teremt kapcsolatot (ugyanerre a fogalomra használt gyakori terminus még az 'is-a', az 'is-a-type-of", az osztály vagy a típus kifejezés).

Egy másik fontos, központi fogalom a **példánya** (instanciája, előfordulása, esete) reláció, amely az univerzálé és individuum közti viszonyt teszi kifejezhetővé. E reláció segítségével tudjuk megadni egy univerzálé terjedelmét.

Gyakran használt relációs fogalom még az **inverze** reláció, ami relációk közti relációként értelmezhető. Egy reláció inverzét úgy képezhetjük, hogy megcseréljük a relátumok sorrendjét. Példa rá a gyermeke reláció, amelynek inverze a szülője reláció, és ha azt állítjuk, hogy x gyereke y-nak, akkor ennek inverze is igaz lesz, vagyis y szülője x-nek. Az inverze reláció másodrendű, hiszen két elsőrendű fogalom között teremt kapcsolatot.

Az ontológiákban használni szokták még a **partitív relációt**, amelyek segítségével az entitások közti rész-egész típusú tartalmazási viszonyokat írják le. Az általunk vizsgált két jogszabály közül a családtámogatási törvény leírásában csak egy-két ponton van szükség erre a fogalomra, a kreszen belül fontosabb szerepe van. A partitív reláció két típusát, az **atomos** és **nem atomos partitív relációt** is meg kell különböztetnünk [24,20]. Előbbire a szervezetek és a bennük dolgozó emberek közti – tagsági – kapcsolat lehet a példa, utóbbira az úttest, illetve az úttest bal és jobb oldali sávja közti átfedési/tartalmazási viszony.

A szakontológiák kibontásakor fontos a reláció fogalmának azon – absztrakt – tipizálása, amely a reláció argumentumainak minőségére tehető állítások alapján különít el típusokat [8,11]. **Formális** relációról akkor beszélhetünk, ha a reláció az argumentumait azok minőségétől teljesen függetlenül kapcsolja össze. Ide tartozik az összes fontos matematikai reláció (rendezések, egyenlőség, azonosság, tolerancia, különbözőség stb.), de a generikus alárendeltje, az inverze, a példánya, a partitív relációk is ilyenek. Ilyenkor semmit sem kell tudnunk a reláció igazságának megállapításához az argumentumokra vonatkozóan. A reláció másik altípusaként definiálhatjuk a **kvalitatív reláció**, azon belül pedig az **intrinzikus** és **extrinzikus reláció** fogalmát. A kvalitatív relációról akkor beszélhetünk, amikor a reláció fennállásának megállapításához szükség van arra, hogy a reláció argumentumainak létezzen valamilyen minősége. Az intrinzikus reláció esetén ez a többletminőség a relátumok valamilyen tulajdonsága lehet: például a *magasabb* vagy az *öregebb* reláció igazságának eldöntéséhez tudnunk kell az összehasonlított dolgok magasságát vagy korát. Ha veszünk két embert, akiknek mindenképpen létezik magassága, illetve kora, akkor a létezésük (és az adott minőségük) alapot ad a köztük levő intrinzikus kvalitatív reláció igazságának eldöntéséhez. Ezzel szemben az extrinzikus reláció esetén nem redukálhatjuk a reláció igazságát a relátumok monadikus tulajdonságaira, szükség van valami másra is: valamilyen harmadik létezőre, ami összekapcsolja az argumentumokat egymással. A *házassága* relációt hozhatjuk fel példaként, amelynek két ember lehet a relátuma, de két ember létezése önmagában még nem elégséges ahhoz, hogy kimondhassuk, hogy a házassága reláció áll fenn köztük. Ezt csak akkor tehetjük meg, ha létezik még valamilyen további minőség a világban: korábban megtörtént a házasságkötés aktusa köztük. A *gyereke* reláció sem értelmezhető csak két ember inherens tulajdonságai alapján, hiszen ez a kapcsolat csak azon emberek között áll fenn, akiket a születés eseménye köt össze. Az ilyen relációkat **materiális relációnak** is nevezik a szakirodalomban [19,8], mi is inkább ezt a terminus fogjuk használni. A materiális relációk azért fontosak az ontológiaépítés során, mert ezek alkalmasak igazán valamely tárgyterület leírására.

A relációfogalmakon belül további fontos megkülönböztetést tehetünk még Geach nyomán [4]. Ő mutatott rá először arra, hogy vannak olyan változások (amiket Cambridge-változásoknak nevezett el), amelyeknek ugyanazok az igazságfeltételeik, de más események, állapotok, folyamatok kapcsolódnak hozzájuk. Mulligan és Smith [19] nyomán a változásokat **Cambridge-relációként** értelmezve a következő példát hozhatjuk: amikor az anya szül, akkor benne/vele nyilvánvaló változások történnek, aminek eredménye a megszületett gyermek lesz. A szülő nő a születés eseményétől válik anyává, de ugyanezen esemény apává teszi a férfit is, miközben benne/vele nem történik semmi „különös” az anyához és a gyermekhez képest (a *nagypapa*, *nagymama* vagy az *özvegy* fogalmak is ilyenek, de van még sok más hasonló kategória).

A jogi szövegek emberi cselekvéseket szabályoznak, így a jogi szakontológia talán legfontosabb kategóriája az *ember* (személy) fogalma, ami a dolog kategóriája alá tartozik. A jogi szövegek azonban ritkán beszélnek az emberről, sokkal inkább – intrinzikus – tulajdonságok vagy – extrinzikus – **szerepek** tulajdonítása mentén szabályozzák az emberi cselekvéseket, amit valahogyan kezelnünk kell.

A személy fogalmát alábonthatjuk bizonyos tulajdonságok segítségével. A *gyermek* vagy a *felnőtt* fogalma az ember életkora, míg a *súlyosan beteg gyermek* fogalma valamilyen testi állapot, a *férfi* vagy a *nő* fogalma a biológiai nem tulajdonság értékei alapján határozható meg.

Ezen tulajdonságok egy része rigid vagyis az adott individuum létezése során nem változik, de vannak nem-rigid – vagyis az individuum létezése során változó – tulajdonságok is [10].

A rigid, illetve nem-rigid tulajdonságok alkalmazhatók a relációs fogalmakra is. A nem-rigid tulajdonságokkal kapcsolható össze a szerep fogalma. Egy jogi ontológia szinte kizárólag szerepekből és az ezekhez rendelt jogosultságok és felelősségek megadásából áll. Az orvos, a plébános, a politikai államtitkár szerepek, amelyeket valaki élete során egy meghatározott ideig valósít meg, hordoz. Filozófia értelemben a szerep a fajtatulajdonságok egy altípusa (phase sortals) [6]. Az, hogy valamely személy egy szerep megvalósítója, hordozója (orvos, plébános vagy államtitkár) lesz, nem valamely inherens tulajdonságából származik, hanem a szerep hordozójának sajátos természeti, társadalmi, intézményi közegében előforduló helyzetek vagy események következménye: kinevezik, felszentelik, megbízzák vagy *úgy tekintenek rá* (1). A jog számára minden szereptulajdonság valamilyen formális aktus eredménye (házasságkötés, munkaképesség 50%-nál nagyobb mértékű elvesztésének deklarása), és azt dokumentáló nyilvántartások és igazolások (anyakönyvi kivonat, szakértői igazolás stb.) tanúsítják a fennállását. Valamely személy individuális személy egyszerre több szerepet is betölthet: egyszerre lehet valaki államtitkár, apa, futballbíró [20].

A jogszabályszövegekben előforduló szerepfogalmakat legtöbbször valamilyen materiális relációfogalom segítségével lehet értelmezni és kifejezni. A családtámogatási törvényben például szerepel a *szülő*, a *házastárs*, a *nagyszülő* fogalma, amelynek segítségével valamilyen emberre lehet rámutatni, de amely fogalmakat a *szülője*, *házastársa*, *nagyszülője* relációkból lehet „levezetni”. Ennek eszköze a **reifikálás** művelete. A reifikáció annyit jelent, hogy a reláció valamelyik argumentumát (vagy a relációt magát) önálló, monadikus fogalomként értelmezzük, vagyis az adott paramétert kiemeljük, projektáljuk a relációból. Mivel a relációnak több argumentuma van, ezért beszélhetünk **jobb-** vagy **baloldali reifikálásról**, sőt, gyakran arra is szükségünk lehet, hogy magát a relációt is reifikáljuk (**közép-reifikálás**). Ennek a műveletnek az az igazi értelme, hogy a relációból mint diadikus (vagy triadikus stb.) fogalomból monadikus fogalmat képzünk, amihez rendelhetünk tulajdonságokat, illetve individuálhatjuk is azt. A *gyermeke(személy,személy)* reláció baloldali reifikálása révén kaphatjuk a *gyermek*, a jobboldali reifikálásból a szülő, a közép-reifikálásból pedig a születés fogalmat, amelyek vagy a személy alá rendelt, monadikus szerepfogalmak, így dolgokként tekinthetünk rájuk, vagy olyan eseményfogalomként, amit példányosítani tudunk. A *házastársa(személy,személy)* relációból kaphatjuk a *házastárs* szerepfogalmat (megint csak a személy fogalma alá rendelve), illetve a *házasság* fogalmat, ami a dolgok (események) másik típusa alá tartozik, de ezek is példányosíthatók.

Az ontológiában szükség van a **gyűjtemény** fogalmára is, amely tetszőleges entitásokat foghat össze (ennyiben különbözik az univerzálé fogalmától, amely alá csak valamilyen tulajdonságban közös fogalmakat rendelhetünk [2]. A gyűjteményre lehet példa egy gépjármű a forgalomban való részvételhez a kreszben előírt forgalmi engedéllyel, egészségügyi dobozzal, elakadásjelzővel együtt.

Eddig azokat az általános ontológiai fogalmakat, kategóriákat mutattuk be, amelyek szükségesek az egyes jogszabályok ontológiájához, de ezekre támaszkodva tartalmilag még semmit sem tudunk mondani a konkrét törvényről. A tartalmi leíráshoz szükségünk van további fogalmakra (relációkra). Ezt a feltárást egy másik cikkben végezzük el. Itt csak annyit említünk meg, hogy a jogontológia számára kiemelten fontos az **ember** fogalma, amivel a legtöbb nyelvhasználati kontextusban szinonim a személy terminusa, de a jogi szaknyelvben utóbbi kifejezést kétféle módon használják: **természetes** és **jogi személy** értelemben. Előbbi az ember jelentésével azonos, utóbbit a jog úgy értelmezi, hogy a személyiségét (jogképességét) a jog teremti.

## 3 Deontikus fogalmak

A jogszabályszöveg normatív állítások rendszere. A normatív állítások különös minőségének megragadására a deontikus logika kínál megfelelő eszközöket. A normákban rejlő, jövőre irányuló elvárás (a kellés) mozzanatát deontikus (modális) operátorok segítségével fejezhetjük ki [21] [16]. A jogi szövegek értelmezéséhez azonban nem csak a deontikus operátorokkal kell foglalkoznunk, de kezelnünk kell a normák leírásához szükséges további fogalmi komponenseket is. Ehhez G.H. von Wright elméletéből indulhatunk ki [25], aki hat komponensre (**tartalom**, **modalitás**, **kibocsátó**, **címzett**, **feltétel**, **körülmény**) bontotta fel a norma fogalmát. Ezt érdemes legalább egy komponenssel kiegészíteni: a jog működésének értelmezéséhez szükséges a **szankció** fogalma is.

A jogszabályokban leggyakrabban megjelenő deontikus modalitások a **megengedett**, **tiltott**, **kötelező** [16] (ezek összefüggéseinek pontos ábrázolásához felvehetők a **mellőzhető**, **preskriptív**, **közömbös** kategóriák is [14,21], de ehhez a jogszabályok elemzésekor gyakorlati érdek nem fűződik, a tényleges jogi gyakorlatban ezek gyakorlatilag nem fordulnak elő). Fontos viszont, hogy egy másik szerző, Hohfeld rendszerét figyelembe vegyük a jogi szövegek elemzésekor [13]. Hohfeld szerint érdemes a jog(osultság) és a kötelezettség fogalmát pontosítani, és elkülöníteni az **igény(jog)**, a **szabadság**, a **felhatalmazottság** és az **mentesség** fogalmait a jogosultságon belül, illetve a **kötelesség**, **joghiány**, **beavatkozásnak kitettség** és **beavatkozásképtelenség** kategóriáit a kötelezettségen belül [17,21]. A hohfeldi jog- és kötelezettségfogalmak értelmét és alkalmazhatóságát a kreszre bemutattuk egy korábbi tanulmányban [22].

## 4 Ténytípusok, fogalomtípusok

A jogi szakontológiákban meg kell különböztetnünk két típust: a természeti tényeket és a társadalmi tényeket megállapító fogalmakat. A kétféle fogalom közti különbség abban áll, hogy emberi akaratoktól független vagy azoktól függő tényekről van-e szó [19]. Egy újonnan felépített szálloda léte természeti tény, míg a szálloda neve társadalmi tény. A természeti tényeket felismerjük, megállapítjuk, a társadalmi tényeket

konstruáljuk, konstituáljuk, deklaráljuk [19]. Az elemzett törvényszövegben dominánsak a társadalmi tényekre hivatkozó fogalmak, de előfordulnak természeti tényekre utaló jogi terminusok is. A gyermek-szülő viszony, a személyek kora, egészségi állapota természeti tényként tekinthető, míg a házastársi, élettársi, nevelőszülői, gyámsági, intézményvezetői vagy jogosultsági szerepek társadalmiaknak. Az, hogy valaki gyermeke valakinek természeti tény, míg az, hogy valaki házastársa valakinek, társadalmi konstrukció eredménye.

A természeti és társadalmi tények kettősségén (és nyilvánvaló különbségén) túl viszont fontos figyelembe venni azt a tényt, hogy a jogszabályok alkalmazása során természeti tényeket is gyakran beemelik a társadalmi tények világába azzal, hogy azokat „elismerik", megállapítják, és a természeti tényeket egy újabb aktussal valamilyen állami **nyilvántartásban** rögzítik. Az államigazgatási nyilvántartásokban így kétféle **adat** keletkezik: egyrészről természeti tények **megállapítása** révén létrejövő adat, másrészről valamilyen konstitutív aktus révén létrejött társadalmi tény deklarálása révén keletkező adat.

A társadalmi tények emberi akaratoktól való függése hatással van a használt fogalmainkra is, mert ha egyszer a társadalmi tények emberi akaratoktól függenek, akkor azok egy későbbi (esetleg más ember által hozott) akarat által meg is változtathatók. A társadalmi tényeket reprezentáló fogalmakat – elméletileg – mindig dinamikusként (időfüggőként) kell értelmeznünk. Ennek a minőségnek komoly jelentősége van minden jogi szakontológia építésekor.

Ahhoz, hogy a társadalmi tények időfüggését kezelni tudjuk, az ontológiánkban alkalmazni kell az **esemény** fogalmát. Nézzünk meg egy konkrét példán keresztül, mit hogyan kell figyelembe venni. Amikor az erre jogosult ágensek konstituálják két ember házasságát, attól kezdve létrejön egy házasság, és a házastársak házas emberek lesznek. Ez az állapot megváltozhat akár egy természeti tény által (meghal az egyik házastárs, aminek eredményeként a másik özvegy lesz), akár egy társadalmi tény által (a felek felbontják a házasságot, és mindketten elvált státusba kerülnek). Előfordulhat az is, hogy egy volt házas (özvegy vagy elvált) ember újabb házasságot köt, és megint házas lesz.

| | | |
|---|---|---|
| egy anyakönyvvezető deklarálja és bejegyzi $t_1$-ben, hogy x házastársa y-nak | házassági anyakönyvi bejegyzés | társadalmi tény |
| y meghal $t_2$-ben | | természeti tény |
| y haláláról bejegyzés készül $t_3$-ban | halotti anyakönyvi bejegyzés | társadalmi tény |
| x özvegy lesz $t_3$-ban | | társadalmi tény |
| egy anyakönyvvezető deklarálja és bejegyzi $t_4$-ben, hogy x házastársa z-nek | házassági anyakönyvi bejegyzés | társadalmi tény |
| egy bíró felbontja x és z házasságát $t_5$-ben | házassági anyakönyvi bejegyzés | társadalmi tény |
| x és z elvált lesz $t_5$-ben | | társadalmi tény |

A gyermek születése természeti tény, az állam ezt – egy konstitutív aktussal – megállapítja, és ezzel – már társadalmi tényként – beemeli a társadalmi cselekvések szimbolikus terébe. Másként kezeli azonban az apaságot és anyaságot. Az anyaság tényét – a megszülés nyilvánvalósága miatt – megállapítja, míg az apaság tényét – annak nem nyilvánvaló volta miatt – vélelmezi. Ha házasságban születik a gyerek, akkor az állam azt a férjet fogadja el apának, aki a szülés időpontjában házasságban élt az anyával, és csak vitatott esetekben lehet ezt a megállapított társadalmi tényt megváltoztatni. Az anyaság és apaság fogalmának eltérő jogi kezelését azzal az ontológiai különbséggel indokolhatjuk, hogy az *apja* fogalma Cambridge-relációként értelmezhető, míg az *anyja* reláció nem.

A cikk utolsó részben bemutatott példák csak a csúcsontológiába felveendő fogalmak szemléltetésére szolgáltak. Azt, hogy a legáltalánosabb fogalmak segítségével hogyan lehet szakterületi ontológiát felépíteni, egy másik tanulmányban mutatjuk be [5].

***

A cikk az EMMI Felsőoktatási Intézményi Kiválósági Program által finanszírozott BME Mesterséges Intelligencia (BME FIKP-MI) kutatás keretében készült.

## Hivatkozások

1. Arp, R., Smith, B.: Function, role, and disposition in basic formal ontology. Nature Precedings. http://hdl.handle.net/10101/npre.2008.1941.1 (2008)
2. Bittner, Thomas, Donnelly, Maureen, Smith, Barry (2004). Individuals, Universals, Collections. On the Foundational Relations of Ontology. In: Varzi, Achille C., Vieu, Laure (eds.): Formal Ontology in Information Systems. Proceedings of the Third International Conference (FOIS 2004*)*, IOS Press, 37–48
3. Dahchour, M., Pirotte, A.: The Semantics of Reifying N-ary Relationships as Classes. In: 4th International Conference on Enterprise Information Systems (ICEIS). (2002) 580–586
4. Geach, P.T.: God's Relation to the World. In: Geach, P.T., Logic Matters. Blackwell (1972) 318–327
5. Grad-Gyenge, A., Hamp, G., Héder, Á., Markovich, R., Nagy, K., Vértesy, L., Syi: A családtámogatási törvény ontológiája. Kutatási jelentés (2018)
6. Grandy, R.E.: Sortals. The Stanford Encyclopedia of Philosophy (Winter 2016 Edition) (2016)
7. Guarino, N., Sales, T., Guizzardi, G.: Reification and Truthmaking patterns. In: Trujillo, J.C., Davis, K.C., Du, X., Li, Z., Ling, T.W., Li, G., Lee, M.L. (eds.): Conceptual Modeling. Proceedings of 37th International Conference. ER 2018. (2018)
8. Guarino, N., Guizzardi, G.: Relationships and Events: Towards a General Theory of Reification and Truthmaking. In: AI*IA 2016 Proceedings of the XV International Conference of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence. (2015) 237–249
9. Guarino, N., Guizzardi, G.: We Need to Discuss the Relationship. Revisiting Relationships as Modeling Constructs. In: Proc. CAiSE 2015, Stockholm. (2015)
10. Guarino, N., Welty, C.: Evaluating Ontological Decisions with Ontoclean. Communications of the ACM 45(2) (2002) 61–65

11. Guizzardi, G., Wagner, G.: What's in a Relationship? An Ontological Analysis. In: Conceptual Modeling (ER 2008). Number 5231 in LNCS (2005) 83–97
12. Heller, B., Herre, H.: Ontological categories in GOL. Axiomathes 14 (2004) 71–90
13. Hohfeld, W.N.: Alapvető jogi fogalmak a bírói érvelésben. In Szabó, M., Varga, C., (eds.): Jog és nyelv. Kiadó nélkül (2000) 59–96
14. Kalinowski, G.. Theorie des propositions normatives. In G. Kalinowski, Etudes de logique deontique. I (1953, 1969), 17-53. Paris: Librairie generale de Librairie générale de droit et de jurisprudence, 1972
15. Moltmann, F.: Events, Tropes and Truthmaking. Philosophical Studies 134(3) (2007) 363–403
16. Markovich, R.: A jogszabályok logikai mélystruktúrája. In: Szabó Miklós, Vinnai Edina (szerk.): A törvény szavai. Miskolc: Bíbor Kiadó (2018)
17. Markovich, R.: Deontic Logic and Formalizing Rights. PhD thesis, ELTE: Eötvös Loránd Tudományegyetem Bölcsészettudományi Kar Filozófiatudományi Doktori Iskolája (2018)
18. Mulligan, K., Smith, B.: A Relational Theory of the Act. In: Topoi 5(2) (1986) 115–130
19. Searle, J.R.: The Construction of Social Reality. The Free Press (1995)
20. Smith, B. et al.: Basic Formal Ontology 2.0. Specification and User's Guide. (2015)
21. Syi: syi.hu/cse. L'Harmattan–Könyvpont (2014)
22. Syi, Markovich, R., Hamp, G.: Jogosultság- és kötelezettségfogalmak a kreszben. In: Vincze, V., ed.: XIV. Magyar Számítógépes Nyelvészeti Konferencia, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2018) 393–404
23. Tahko, Tuomas E. and Lowe, E. Jonathan, "Ontological Dependence", *The Stanford Encyclopedia of Philosophy* (Winter 2016 Edition), Edward N. Zalta (ed.), URL: https://plato.stanford.edu/archives/win2016/entries/dependence-ontological/
24. Varzi, A.: Parts, Wholes, and Part-Whole Relations. The Prospects of Mereotopology. Data and Knowledge Engineering 20 (1996) 177–1982
25. von Wright, G.H.: Norm and Action. Routledge and Kegan Paul, London (1963).

# Kísérletek tudásbázis- és mondatkörnyezet-alapú beágyazásokkal magyar nyelvre

Kardos Péter[1], Berend Gábor[1,2], Farkas Richárd[1]

[1]Szegedi Tudományegyetem, Informatikai Intézet
Szeged, Árpád tér 2.
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
Szeged, Tisza Lajos körút 103.
Kardos.Peter@stud.u-szeged.hu, {berendg,rfarkas}@inf.u-szeged.hu

**Kivonat** Napjainkban a szavak jelentésének folytonos vektortérbeli leírására a mondatkörnyezet-alapú neurális megoldások a legelterjedtebbek. Ebben a cikkben gráfalapú beágyazások használatával kísérletezünk, amelyek a gráf csúcsait (például WordNet synsetek) ágyazzák be vektortérbe, szomszédaik alapján. Javaslunk egy újszerű módszert is, mellyel kombinálhatók ezen beágyazások, majd a beágyazásokat a magyar WordNetből általunk készített szóasszociációs feladaton teszteljük.

## 1. Bevezetés

Hogyan is tudjuk megtanítani a gépeknek, hogy a cipő az egy ruhadarab és a lábunkon hordjuk, nem pedig a fejünkön? Több tudásbázis létezik, melyek megpróbálják a szavak jelentését leírni, szinonimák, hipernímák és más szemantikai kapcsolatok használatával. Ezeknek nagy hátránya, hogy kézzel építettek, nem teljesek, hiányoznak újabb szavak. Másik megoldás lexikai szemantikai ábrázolásra a statisztikai szemantika (distributional semantics), aminek napjainkban legelterjedtebb tagja a szóbeágyazások (*word embedding*) használata. A szóbeágyazó módszerek a szavakat vektortérbe ágyazzák be a szavak mondatkörnyezetének felhasználásával, innen is ered megnevezésük. Ezt úgy teszik, hogy a hasonló fogalmak közel kerüljenek egymáshoz. Ezek rengeteg fontos feladathoz segítséget nyújtanak, mint például gépi fordítás, kérdésmegválaszolás, szófaji kódsorozat meghatározás, ami miatt széles körű felhasználásnak örvendenek. A mondatkörnyezeten alapuló reprezentációk is rendelkeznek természetesen hiányosságokkal, amelyek között megemlíthetjük a reprezentációk egyes dimenzióinak interpretációjának korlátjait [1] vagy a folytonos szóreprezentációk azon tulajdonságát, hogy az ellentétes jelentésű szavakhoz gyakran hasonló reprezentáció társul. A szóreprezentációk tudásbázisokkal való kombinálása alkalmas lehet mindkét előbb említett probléma mérséklésére.

A szóbeágyazások létrehozásának eddig legelterjedtebb formája a szövegekből (*mondatkörnyezet alapú*) való generálás volt, de manapság sorra jelennek meg a gráfokat beágyazó algoritmusok. Ezen gráfalapú algoritmusok a gráf csúcsait ágyazzák be vektortérbe, szomszédaik alapján. A tudásbázisok felfoghatók egy

gráfként, melynek csúcsai a szavak, címkézett élei pedig a szavak közötti kapcsolatok. Munkánkban a magyar WordNetből készítettünk ilyen beágyazásokat a diff2Vec [2] algoritmussal. Majd a kapott beágyazásokat összehasonlítottuk a mondatkörnyezet alapú beágyazásokkal.

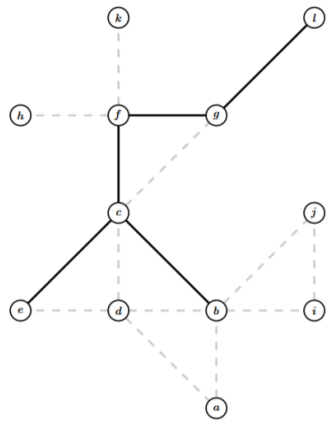
Javaslunk egy újszerű módszert is, mellyel kombinálhatók ezen beágyazások, majd a kombinációkat is összevettük a már előtanított beágyazásokkal. A beágyazásokat a magyar WordNetből általunk készített feladaton teszteltük.

## 2. Kapcsolódó munkák

### 2.1. Gráfbeágyazások

A gráfbeágyazó algoritmusok megjelenését a szóbeágyazó algoritmusokkal elért sikerek ihlették meg. Ezen algoritmusok egy gráf csúcsait ($V$) képzik le $d$ dimenziós lineáris térbe, úgy, hogy az reprezentálja a gráfban lévő távolságokat. Vagyis ezek egy $V \mapsto \mathbb{R}^{|V| \times d}$ leképezést hajtanak végre, ahol $d \ll |V|$. A legtöbb ilyen algoritmus a skip-gram/CBOW beágyazó megoldásra épít, de előtte át kell alakítanunk a gráfot ezzel kompatibilis formába.

A legismertebb gráfbeágyazó algoritmusok a node2Vec [3] és diff2Vec [2], amelyek a csúcsok környezetét adott hosszúságú és számú csúcsból induló bejárással írják le. Ezek a környezet már a mondathoz hasonló szekvenciaként is felfoghatóak. Ezen utakból adott ablak méret mellett, együttes előfordulás mátrixot készítenek a módszerek. Ezek lesznek majd a neurális háló kimenetei, a bemenet pedig a csúcs one-hot vektora. Így már tudjuk tanítani a hálót a skip-gram vagy CBOW módszerrel. A node2vec és diff2vec csak a környezet kinyerésében tér el egymástól. Mivel előzetes kísérleteink azt mutatták, hogy a diff2vec rendre jobb eredményeket ér el és robusztusabb, ezért ebben a munkában csak azt használjuk.




1. ábra. Diffúziós gráf példa. Forrás: [2]

A diff2Vec [2], mint Diffusion to Vector úgynevezett szétterjedés gráfokat hoz létre a következő módon. Először inicializáljuk a $\widetilde{G}$ diffúziós gráfot, melyben az egyetlen csúcs az aktuálisan vizsgált csúcs. Majd minden iterációban a még nem bevett szomszédos élekből véletlenszerűen választunk egyet és a csúccsal együtt bevesszük a $\widetilde{G}$ gráfba. Ezt addig ismételjük amíg a paraméterként kapott $p$ darab csúcs nem lesz a gráfban, vagy már nem tudunk újat bevenni. Most egy útvonalat kell generálnunk ebből a részgráfból. Ezt úgy tesszük, hogy minden élet megduplázunk, így biztos lesz benne Euler séta, amit könnyen meg is tudunk találni. Az Euler sétának megvan az a jó tulajdonsága, hogy megismer minden szomszédsági kapcsolatot a részgráfban. Ezután már tudunk ezekből a csúcs sorozatokból skip-gram/CBOW módszer használatával beágyazásokat tanítani.

A diff2vec a következő paraméterekkel rendelkezik:

- $n$: Generált szétterjedés gráfok száma csúcsonként.
- $p$: Szétterjedés gráfokbeli csúcsok száma.
- $d$: Beágyazás dimenziója.
- $\hat{w}$: Ablak mérete.

A neurális hálók tanításánál a legfontosabb epoch és learning rate paramétereket állítottuk. Az algoritmus implementációja elérhető a `https://github.com/benedekrozemberczki/diff2vec` oldalon.

### 2.2. Magyar nyelvű beágyazási eredmények

Magyar nyelvű szóbeágyazok minőségével kapcsolatban több korábbi munka is napvilágot látott. A [4] cikk magyar nyelvű szóbeágyazások minőségét a [5] által javasolt szóanalógiás feladat magyarra adaptált verzióján, illetve nyelvek közötti fordítási feladatokon értékelte ki. A [6] szerzői csupán elvétve találtak olyan analógiapéldákat, melyekre helyes eredményt kaptak volna, így a szóbeágyazások minőségének ellenőrzésére a szemantikai csoportok hierarchikus klaszterezéssel történő automatikus kinyerését választották. A magyar nyelvű szó– és karakter szintű információt integráló szóreprezentációk hatékonyságát [7] téma– és véleményosztályozási feladatokba építve mutatta meg. A [8] munka különböző szóbeágyazási modellek minőségét hasonlította össze annotátorok segítségével. Az annotátorok feladata az volt, hogy a különböző szóbeágyazási modelleket aszerint rangsorolják, hogy azok mennyire homogén legközelebbi szóhalmazt adnak vissza bizonyos hívószavak tekintetében.

## 3. Mondatkörnyezet- és gráfalapú beágyazások kombinálása

Ebben a munkánkban fő célunk, hogy a szavak mondatkörnyezet- és gráfalapú beágyazásait összehasonlítsuk és kombinálással kiaknázzuk azok előnyeit.

A gráfbeágyazó algoritmusokhoz bemenetként a magyar Wordnetet használtuk fel. Az általunk használt algoritmusok nem veszik figyelembe az élek címkéit,

vagyis a szavak közötti kapcsolatok típusát, viszont ezt nem hagyhatjuk figyelmen kívül, hiszen ezek is információval láthatnak el bennünket. Ezért az adatbázist a kapcsolatok szerint több kisebb részre szedtük, majd ezekre külön-külön futtattuk az algoritmusokat. Az így kapott beágyazásokat a magyar WordNetből generált kiértékelő adatbázison teszteltük.

Ha több különböző beágyazást egyszerre szeretnénk használni, akkor találnunk kell egy módszert azok kombinálására. Alább javaslunk, és tesztelünk egy módszert különböző beágyazások kombinálására.

### 3.1. Beágyazások kiértékelése

A kiértékelő adatbázishoz a magyar WordNetből [9] két gráfot készítettünk, egyet a szinonima, egyet a hiperníma kapcsolatokból. Mondatkörnyezet beágyazásként a Szeged_fasttext-et [7] használjuk. Vettük azon szavak listáját melyek mindkét gráfban és az Szeged_fasttext-ben is szerepelnek. Ezekből a közös szavakból gyűjtöttünk szópárokat melyek 1,2 vagy 3 távolságra (legrövidebb út) helyezkednek el egymástól és mindegyik távolságból véletlenszerűen választottunk 100-100 darab szópárt. A két gráfból külön-külön generáltuk ezeket, így összesen 600 kiértékelő szópárt kaptunk, ahol a távolságok inverzét vettük az asszociáció erősségének. A következőkben ezen a 600 páron hasonlítjuk össze a tudásbázis- és mondatkörnyezet-alapú beágyazásokat úgy, hogy az egyes szópárokra kiszámoljuk a két vektor koszinusztávolságát, majd az egész beágyazás jóságának mértéke a 600 koszinusztávolság és a WordNet szópár távolság inverzének Spearman korrelációja lesz.

*Out-of-vocabulary (OOV)* A modellből hiányzó, de a teszt adatokban előforduló szavak kezelésére három eljárást is megvalósítottunk.

1. **Zero** - Egyszerűen azt mondjuk, hogy ezen példáról semmit nem tudunk mondani, így 0 a hasonlóságértékük.
2. **Skip** - Egyszerűen átugorjuk az adott példát. Itt problémát jelent, hogy a különböző beágyazásokban más-más példák hiányoznak, eltérő mennyiségben, emiatt az ezzel kapott korrelációk nem hasonlíthatók össze. Ellenben ez egy jó módszer annak tesztelésére, hogy a modell által ismert szavak közötti kapcsolatot mennyire jól tudja.
3. **Fallback** - Előfordulhat, hogy csak a szó végén álló toldalék miatt nem ismeri fel a szót. Innen jött az ötlet, hogy akkor levágjuk mindig az utolsó betűt, amíg nem kapunk egy olyan szót, ami szerepel a modellben. Itt kérdéses, hogy mit is kéne tenni, ha elfogynak a betűink. Ebben az esetben átugorjuk a példát.

Miután a 600 szópárt kigyűjtöttük a kiértékelő adatbázisba, a gráfból töröltük az egyes szópárokat összekötő összes legrövidebb út összes élét, hiszen célunk a tudásbázisban nem szereplő asszociációk predikálhatóságának vizsgálata.

### 3.2. Beágyazások kombinációja

A legkézenfekvőbb eljárás beágyazások kombinálásra az, ha egyszerűen konkatenáljuk a szavak vektorait. A beágyazásokban szereplő szavak száma viszont eltérő, így előfordulhat, hogy az egyik beágyazásban szereplő szót nem találjuk a másikban. Ezért a beágyazások vektorait úgy konkatenáljuk, hogy amennyiben valamelyik beágyazásban nem szerepel egy szó, akkor annak helyét nullákkal töltjük fel.

A következő lépésben információtömörítést alkalmazunk, ez lehet SVD vagy PCA. Ennek a motivációja az, hogy redundanciát csökkenthetjük, valamint az erősebb dimenziók jobban fogják meghatározni a beágyazást, mellyel általánosságban javulást érünk el. Ez a művelet nagy mátrix esetén elég költséges mind időigény mind pedig memóriahasználat szempontjából. Az erőforrásigények csökkentése érdekében csak az adatsorok 10%-át vettük a legfontosabb dimenziók meghatározására.

Végül a kapott beágyazás oszlopait L1 normalizáljuk, melyet a [10] cikkben mutatott sikerek miatt teszünk. A módszerek paramétereinek finomhangolását a 3.5 fejezetben mutatjuk be. A következőkben a kombinálást a konkatenálás, majd tömörítést és normalizálást mutatjuk be részletesen.

### 3.3. Diff2Vec beágyazások

Ebben a fejezetben a WordNet szinonima és hiperníma éleiből készítünk beágyazást. Ezt úgy tesszük, hogy előfeldolgozásként a gráfból kivesszük a teszt adatbázisban megtalálható szópárok közötti útvonalat, mely alapján azok létre lettek hozva. Emiatt létrejöttek olyan csúcsok, melyeknek egyetlen élük sincsen és ezeket a beágyazó algoritmus nem tudja beágyazni, vagyis ezek OOV szavak lesznek. Ezeket a fenti lehetőségek közül mindegyik megoldással leteszteltük. A Diff2Vec algoritmushoz a következő paramétereket használtuk.

- dimenziók - 100
- Szétterjedés gráfok száma csúcsonként - 10
- Szétterjedés gráfbeli csúcsok száma - 40
- ablak méret - 10
- learning rate - 0.025
- epoch - 4

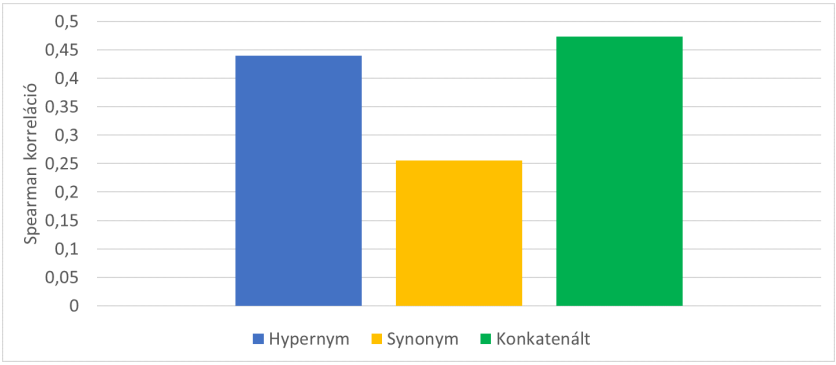
| | **Zero** | | **Skip** | | **Fallback** | |
|---|---|---|---|---|---|---|
| | Hypernym | Synonym | Hypernym | Synonym | Hypernym | Synonym |
| Spearman | 0.439526 | 0.254951 | 0.586998 | 0.300968 | 0.526991 | 0.294253 |
| OOV | 109 | 59 | 109 | 59 | 35 | 25 |

1. táblázat. Hypernym és synonym élek felhasználásával épített gráfbeágyazók eredményei különféle OOV-kezelési stratégiák alkalmazása mellett.

### 3.4. Konkatenálás

Már tudjuk a beágyazások hasznosságát, de kérdés mennyire szerepelnek jól együtt. A konkatenálással kapott beágyazások már többféle kapcsolat típus információját is magában foglalják, így remélhetőleg ez jó irányba befolyásolja a teljesítményt. Ebben a fejezetben ezt teszteljük.

A külön-külön számított gráfbeágyazási vektorok konkatenálásának eredményét a 2. ábra mutatja. A WordNet Hypernym és Synonym éleiből kapott beágyazásokat konkatenálva ez már egy 200 dimenziós beágyazás lesz. A kapott beágyazás 15 darab OOV példát tartalmaz. A kiértékelésnél Zero megoldást használtunk a kezelésükre. A konkatenálás után kapott beágyazás jobban meg tudta oldani a feladatot, mint ezek külön. Ebből következik, hogy a beágyazások konkatenálása hasznos az egyes beágyazások által megtanult információ egyetlen beágyazásban való reprezentálására. A továbbiakban a paraméterek beállítására,illetve tesztelésére ezt a beágyazást fogjuk használni.



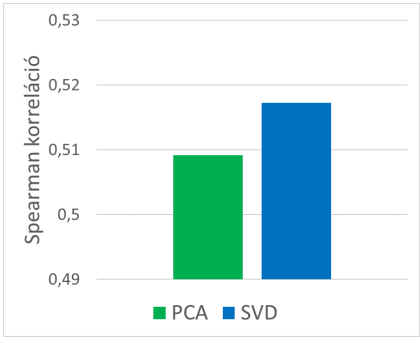

2. ábra. Diff2Vec beágyazások konkatenálásának Spearman korrelációs értékei

### 3.5. Beágyazások utó- és finomhangolása

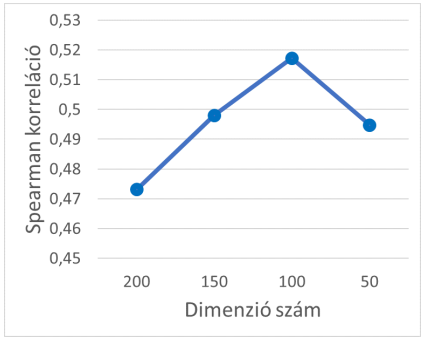
Ahhoz, hogy a kombináló módszerből a lehető legjobbat hozzuk ki minden paramétert tesztelünk, hogy is éri meg beállítani azokat. Ehhez az előző fejezetben kapott konkatenált beágyazást használjuk.

**Információtömörítés** Az információtömörítés segíthet a beágyazásainkon azzal, hogy elhagyjuk az információt nem hordozó dimenziókat. Két lehetőségünk is van információtömörítésre, ezek a PCA és SVD. A teszteléshez 100 dimenziós beágyazást gyártottunk. Ebből a kísérletből az derül ki, hogy a kombináló módszer csak kis mértékben függ attól, hogy PCA-t vagy SVD-t használunk, melyek közül az SVD minimálisan jobb (3. ábra).

Jó kérdés lehet, hogy mennyi dimenzióra érdemes lecsökkenteni a beágyazásokat. A konkatenált beágyazásból kiindulva 50 lépésközzel legyártottunk tömörített beágyazásokat. Ezt a 4. ábra foglalja össze, melyből az derül ki, hogy 100 dimenzióig monoton nőnek az eredmények. Ebből arra következtettünk, hogy a kiindulási dimenzió feléig biztonságos a redukálás a javulás szempontjából.




3. ábra. SVD és PCA összehasonlítása.




4. ábra. Dimenziócsökkentés eredményei 50-es lépésközzel.

**Normalizálás** A különböző beágyazásoknál könnyen meglehet, hogy azok nem ugyanazon az értéktartományon mozognak, emiatt van szükség erre a lépésre. Ennél a kísérletnél az teszteltük javít-e a beágyazás oszlopainak normalizálása. Ehhez L1 normalizációt használtunk, majd összevetettük ezen lépés kihagyásával kapott beágyazásokat (5. ábra). A normalizálás szembetűnően sokat javított, így a továbbiakban ezt mindig elvégezzük.

A dimenziócsökkentés előtti normalizálással kapott eredményeket is megnéztük, mellyel kapcsolatos eredményeinket a 2. táblázatban foglaljuk össze. A normalizálás SVD előtti elvégzése csekély javulást eredményezett a beágyazásokon, így a továbbiakban ezt kihagytuk.

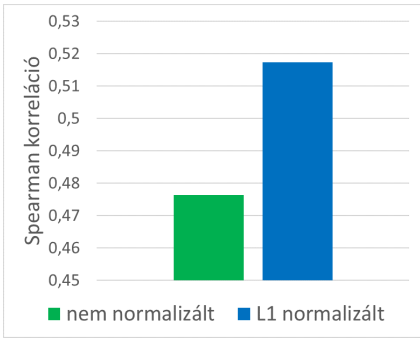
| **Beágyazás** | **Dim** | **WordNet** |
|---|---|---|
| Synonym + Hypernym + Szte_fasttext | 300 | 0.52778 |
| Normalizálás nélkül | 200 | 0.52846 |
| SVD előtti normalizálás | 200 | 0.53007 |
| SVD utáni normalizálás | 200 | **0.59193** |

2. táblázat. Normalizálás összehasonlítása.

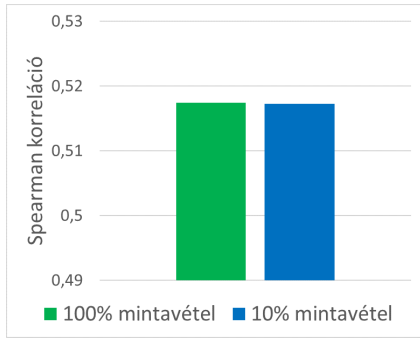
**Mintavételezés** A dimenziócsökkentő algoritmus nagy dimenzió- és szószám esetén erősen számításigényes. Ahhoz, hogy ezen javítsunk megnéztük mi törté-

nik, ha az adott szóhalmaz csak egy paraméterben kapott százalékát használja fel a leghasznosabb dimenziók meghatározására.

Ezt az összes szó felhasználásával kapott beágyazás és a szavak 10%-ának használatával kapott beágyazás összevetésével teszteltük le. A 6. ábrán látható, hogy ez egyáltalán nem befolyásolja a beágyazások minőségét, viszont ezzel csökkenthetjük az erőforrás igényt.




5. ábra. Beágyazás normalizálásának hatása.




6. ábra. Mintavételezés hatása.

**Végső paraméterek** A fent elvégzett tesztek alapján a paraméterek a továbbiakban ennek megfelelően fogjuk beállítani:

- Dimenzió csökkentés: SVD
- Mintavétel: 10%
- Dimenzió szám: kiindulási dimenziók feléhez közelítve
- Normalizáció: L1

### 3.6. Mondatkörnyezet- és gráfalapú beágyazások kapcsolata

A kombináló módszer paramétereinek finomhangolása után, ebben a fejezetben összehasonlítjuk a tudásbázis- és mondatkörnyezet alapú beágyazásokat. A továbbiakban a paraméterek tesztelésére használt él kombinációt fogjuk összevetni más beágyazásokkal.

A Szeged_fasttext [7] egy magyar előtanított szóbeágyazási modell 100 dimenziós vektorokkal. Ezt a beágyazást összehasonlítva az előzőekben tárgyalttal láthatjuk, hogy a gráfalapú modell nem sokkal teljesít rosszabbul. Ha a kettőt konkatenáljuk egy minimális romlás látható, de a kombinálás minden lépésének elvégzése után már sokkal jobban teljesít ezen a feladaton.

Felmerülhet a kérdés mennyire befolyásolja a gráfbeágyazásokat az, hogy voltak OOV szavak. Ezt úgy orvosoltuk, hogy kiszedtük a teszt halmazból azokat a sorokat melyek tartalmaztak OOV-t valamelyik beágyazásban. Az eredmények a 4. táblázatban látottak szerint alakultak:

| **Beágyazás** | **Dim** | **WordNet** |
|---|---|---|
| Synonym + Hypernym | 200 | 0.47308 |
| Szeged_fasttext | 100 | 0.53936 |
| Synonym + Hypernym + Szeged_fasttext | 300 | 0.52778 |
| Előző tömörítve | 200 | **0.59193** |

3. táblázat. Szeged_fasttext és a gráfbeágyazások kombinálása.

| **Beágyazás** | **Dim** | **WordNet** |
|---|---|---|
| Synonym + Hypernym | 200 | 0.51540 |
| Szte_fasttext | 100 | 0.56918 |
| Synonym + Hypernym + Szte_fasttext | 300 | 0.55714 |
| Synonym + Hypernym + Szte_fasttext | 200 | **0.61213** |

4. táblázat. Szeged_fasttext és a gráfbeágyazások kombinálása OOV nélkül.

## 4. Konklúzió

Munkánkban összevetettük a manapság már egyre elterjedtebb gráfokat beágyazó algoritmusokat, a szavakat leíró tudásbázisokból vett kapcsolatokból kiindulva. Általunk vizsgált algoritmus a diff2Vec volt, melyhez a magyar WordNet által leírt szókapcsolatokat vettük kiindulási gráfnak. A beágyazások minőségének kiértékelését a WordNetből generált feladaton végeztük. A WordNet különböző éleit beágyaztuk a diff2Vec algoritmus segítségével, így kiderült mely élek milyen jól tudják ezen feladatokat megoldani. Ezek kombinálására készítettünk egy módszert, mely lehetővé teszi az ezek által megtanult információ egyetlen beágyazásba kombinálását. A módszer a különböző beágyazások konkatenálása után információtömörítést és normalizálást használ.

Mondatkörnyezet alapú előtanított beágyazásokkal (Szeged_fasttext) is összevetettük a csupán tudásbázisokból kapott beágyazások teljesítményét. A gráfalapú beágyazások kombinálása nem sokkal maradt el az előtanított beágyazással szemben. A mondatkörnyezet- és gráfalapú beágyazások kombinálva javítottak egymás eredményein, így kijelenthetjük, hogy a kombináló módszer hasznos különböző beágyazások összeolvasztásában.

A jövőben az olyan úgynevezett heterogén gráfalapú beágyazó algoritmusokkal tervezzük folytatni a kísérletezést, melyek már az élek címkéit is képesek figyelembe venni.

## Köszönetnyilvánítás

Kardos Péter munkáját az "Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein" című, EFOP-3.6.3-VEKOP-16-2017-0002 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg. Berend Gábor munkáját a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal Mester-

séges Intelligencia Nemzeti Kiválósági Programja támogatta a 2018-1.2.1-NKP-2018-00008 azonosítójú projekt keretében.

## Hivatkozások

1. Faruqui, M., Dyer, C.: Non-distributional word vector representations. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics (2015) 464–469
2. Rozemberczki, B., Sarkar, R.: Fast sequence based embedding with diffusion graphs. In: International Conference on Complex Networks. (2018)
3. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2016)
4. Makrai, M.: Comparison of distributed language models on medium-resourced languages. In Tanács, A., Varga, V., Vincze, V., eds.: XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015). (2015)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
6. Siklósi, B., Novák, A.: Beágyázási modellek alkalmazása lexikai kategorizációs feladatokra. In Tanács, A., Varga, V., Vincze, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016), Szeged, Szegedi Tudományegyetem, Szegedi Tudományegyetem (2016) 3–14
7. Szántó, Z., Vincze, V., Farkas, R.: Magyar nyelvű szó-és karakterszintű szóbeágyazások. In Vincze, V., ed.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018), Szeged, Szegedi Tudományegyetem, Szegedi Tudományegyetem (2017) 323–328
8. Novák, A., Novák, B.: Magyar szóbeágyázási modellek kézi kiértékelése. In Vincze, V., ed.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018), Szeged, Szegedi Tudományegyetem, Szegedi Tudományegyetem (2018) 67–77
9. 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. (2008) 311–321
10. Speer, R., Lowry-Duda, J.: Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. CoRR **abs/1704.03560** (2017)

# FrameNet keretek és keretelemek felismerése neurális hálózatok és szódisztribúciós adatok felhasználásával

Tóth Ágoston

Debreceni Egyetem
Angol-Amerikai Intézet
Angol Nyelvészeti Tanszék
toth.agoston@arts.unideb.hu

**Kivonat:** Egyszerű visszacsatolt neurális hálózatok segítségével FrameNet-alapú keretszemantikai elemzést végeztem 9 különböző szóreprezentációs módszer felhasználásával 12 FrameNet keretre és ezek keretelemeire. A kipróbált szóreprezentációs eljárások között szerepeltek a szavak disztribúciós tulajdonságait leíró, nagy méretű korpuszból gyűjtött szóvektorok, melyek lehetővé tették a FrameNet keretek felismerését 91%-os pontossággal 86% fedés mellett (F-mérték: 89%), a keretelemek felismerése pedig 56%-os pontosságú volt 50%-os fedéssel (F-mérték: 53%). A disztribúciós szóábrázolások előnye az eltérő módszerekhez képest jelentős volt. A disztribúciós eszközök közül a környezetszavak leszámlálásán alapuló technikák és a neurális hálózatokban kialakuló prediktív szóbeágyazások egymáshoz hasonló teljesítményt nyújtottak ebben a kísérletben, a prediktív eljárások CBOW és SkipGram osztályai pedig közel azonos eredményt szolgáltattak.

## 1 Bevezetés

A jelentéselmélet három fő irányzata (strukturális, logikai és kognitív szemantika) kijelöli azokat a kereteket, amiben a jelentés gépi feldolgozásának a feladatait a számítógépes nyelvészet és a mesterséges intelligencia kutatásának vonatkozásában is elhelyezzük. Ebben a tanulmányban a kognitív nyelvészet számítógépes nyelvészeti szempontból kiemelt fontosságú eredményének, a FrameNetnek [1] a keretszemantikai kategóriáira támaszkodunk.

A mesterséges neurális hálózatokat gépi tanulási eszközökként egyre jobban megismerjük, napi gyakorisággal megtapasztaljuk széleskörű alkalmazási lehetőségeiket (többek közt az orvosi diagnosztika, arcfelismerés, tőzsdei árfolyamok megjósolása, időjárás-előrejelzés, gépi fordítás területén). A nyelvfeldolgozásban a szerepük túlmutat a más eszközökkel nehezen algoritmizálható részfeladatok végrehajtásán: a természetes nyelvek elsajátításának és feldolgozásának természetes közege az emberi neurális hálózat központja, az agy. A nyelvek kifejlődése, a mai nyelvek elsajátítása és használata is ehhez a közeghez kötődik.

A bemutatott kísérletsorozatban mesterséges neurális hálózatokkal FrameNet-alapú keretszemantikai elemzést végeztem megvizsgálva azt, hogy hogyan befolyásolta különböző szóreprezentációs módszerek használata a feladat megoldásának eredményességét. A kipróbált szóreprezentációs eljárások között szerepeltek a szavak diszt-

ribúciós tulajdonságait közvetlenül leíró tulajdonságvektorok és a neurális hálózatok projekciós rétegében kialakuló prediktív disztribúciós szóbeágyazások is.

## 2 Módszerek

### 2.1 A szemantikai keretek és keretelemek felismerésének feladata

Az 1. táblázatban felsorolt FrameNet keretek felismerését tanítottam be az erre a célra kidolgozott keretspecifikus neurális hálózatoknak.

| Keret-azonosító | Keret (FR) neve | Keret gyakoriság szerinti sorszáma | Keret előfordulási gyakorisága | Keretelemek (FE) száma |
|---:|---|---:|---:|---:|
| 73 | *Leadership* | 5. | 499 | 13 |
| 173 | *Buildings* | 7. | 420 | 12 |
| 408 | *Manufacturing* | 14. | 277 | 13 |
| 191 | *Natural_features* | 16. | 269 | 9 |
| 118 | *Possession* | 17. | 260 | 7 |
| 990 | *Capability* | 18. | 259 | 8 |
| 304 | *People* | 19. | 257 | 8 |
| 34 | *Discussion* | 81. | 87 | 12 |
| 1371 | *Organization* | 89. | 79 | 8 |
| 141 | *Certainty* | 93. | 74 | 7 |
| 172 | *Commerce_sell* | 95. | 73 | 8 |
| 171 | *Commerce_buy* | 145. | 50 | 9 |

1. táblázat. A kísérletben használt FrameNet keretek és keretelemek

A FrameNet 1.7-es változatának full-text kísérőkorpuszában 792 különböző szemantikai kerethez találtam példákat az őket felidéző 28783 szótokent annotáló címke formájában. Ezen annotációk körülbelül 9%-át használtam fel az itt bemutatott kísérletekben. A full-text kísérőkorpusz nem tartalmaz példát minden FrameNet kerethez, továbbá a szemléltetett kereteknek 51%-ához csupán 1-10 példát, további 15%-ához pedig 11-20 példát tartalmaz, ráadásul a hozzájuk tartozó ritkább keretelemek néha egyáltalán nem szerepelnek a példák közt. Mind a betanításhoz, mind a teszteléshez szükségesek voltak ilyen adatok, és a tesztelésnél csak olyan mondatokra támaszkodhattunk, amelyeket a betanítás során nem használt fel a rendszer. Összességében adathiány (data sparsity) miatt a keretek jelentős részét a folyamatból eleve kizártam. A kiválasztott tizenkét keret közül kettő nagyon gyakori volt a korpuszban (420-499 előfordulással), a további keretek közepes- és alacsony frekvenciájúak voltak. A *Commerce_buy* keret 50 előfordulása például (9 keretelem mellett) nagyon kevés

tanító- és tesztadatot eredményezett, azonban későbbi kvalitatív vizsgálatokban (a *Commerce_sell* kerettel együtt) érdekes adatokat szolgáltathat.

A keretelemek felismerése (néhány keretelem ritkasága miatt is) jóval nehezebb feladat volt. A keretelemtípusok keretenkénti száma az 1. táblázatban látható. Néhány példa a keretelemekre:

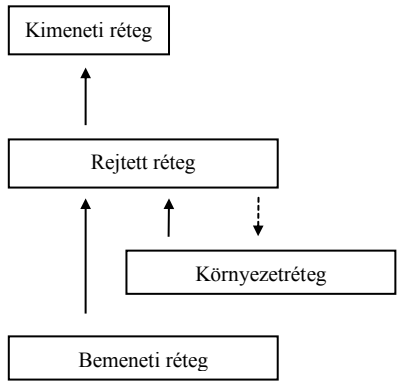
- A *Leadership* keretben: *Leader* („vezető”), *Role* („szerep”), *Governed* („irányított”), stb.
- A *Buildings* keretben: *Name* („név”), *Type* („típus”), *Possessor* („tulajdonos”), stb.
- *Commerce_buy* keretben: *Goods* („áru”), *Buyer* („vevő”), *Seller* („eladó”), stb.

A full-text kísérőkorpuszból vett példamondatok felét betanításra, a másik felét tesztelésre használtam, majd ezek szerepét felcseréltem és az elért eredményeket átlagoltam. Ez a keresztvalidációs eljárás – azzal együtt is, hogy csak kétszeres keresztvalidációhoz állt rendelkezésre elég erőforrás – a tesztelés fontos része volt, mivel a tanító- és a tesztadatok 50-50%-os elosztását kizárólag a keretek vonatkozásában sikerült elérni, a keretelemek esetében nem. Szótövesítést, alaktani vagy mondattani elemzést nem végeztem (ill. ilyen adatot nem használtam fel a korpuszból). Többszavas kifejezéseket, összetevőket nem kezeltem együtt, a szemantikai feladatot végrehajtó hálózattól vártam az ezeket alkotó szavak megfelelő (azonos) címkével történő annotálását. A többszavas kifejezések együttes kezelése és a mondattani összetevők előzetes azonosítása a rendszer teljesítményét minden bizonnyal növelné. Neurolingvisztikai megfigyelések által is motivált az ilyen irányú későbbi továbbfejlesztés, hiszen a nyelvfeldolgozás során az ELAN fázisban jól dokumentált módon lezajlik a „lokális mondattani jelenségek”, pl. bizonyos összetevők elemzése [6]. Egy ezzel analóg feldolgozási mozzanat az egyes keretelemekhez tartozó szócsoportok kiemelését, előfeszítését végezhetné el, egyúttal a többszavas kifejezések kezelését is elősegítené.

Mindegyik szemantikai keretet egy külön neurális hálózat ismert fel, mely Elman-elrendezésben [5] működött. A hálózattípusnak alkalmasnak kellet lennie időbeli mintázatok (szavak szekvenciájának) közvetlen megfigyelésére, ezért visszacsatolt („recurrent”) hálózattípust választottam. Elman a saját hálózati topológiáját „egyszerű visszacsatolt hálózatnak” („simple recurrent network”, SRN) nevezte el, ami a hatékony és gyors betaníthatóságra és alkalmazhatóságra is utal. Az Elman SRN az egyik legelső visszacsatolt hálózati topológia volt, és a nyelvészek számára azért is figyelemre méltó, mert a szerző a hálózattípus eredeti bemutatásakor is kiemelte és demonstrálta a konstrukció felhasználhatóságát nyelvi jelenségek felismerésében is [5]. Amennyiben a rendszer teljesítményének maximalizálása lett volna a cél, akkor összetettebb visszacsatolt hálózattípusok implementálásával (pl. LSTM) valószínűleg további javulást lehetett volna elérni, azonban jelen esetben ez nem volt fontos szempont, hiszen elsősorban a különböző szóreprezentációk összehasonlítását tűztem ki célul.

Az Elman SRN egy rejtett- és egy környezetrétegeket tartalmaz az 1. ábrán szemléltetett módon. A rejtett réteg minden egyes neuronja pontosan 1 környezetneuronhoz van hozzákapcsolva rögzített súllyal. A környezetréteg neuronjai a rejtett réteg idegsejtjeihez kapcsolódnak (jelen implementációban teljes projekcióval, azaz minden neuron a következő réteg összes neuronjához) tanítható súlyokkal. Az itt bemutatott kísérletekben a környezetréteg segítségével mondaton belüli rövid-

távú memóriát alakítottam ki: ennek a rétegnek az idegsejtjeit aktiváció nélküli (0) állapotba hoztam minden mondat utolsó szava után. A bemeneti réteg neuronjaitól a rejtett réteg feldolgozóegységeihez, a rejtett réteg idegsejtjeitől pedig a kimeneti réteg neuronjaihoz az információt teljes projekcióval, tanítható súlyokkal vezettem tovább.




1. ábra: Az Elman SRN topológia áttekintése

A bemeneti rétegen a mondat aktuális szavát jelenítettem meg 9 különböző módon kódolva (ld. 2.2. szakasz) külön kísérletsorokban. A bemeneti réteg mérete a használt szóreprezentációs módszer függvényében 300-10000 neuron volt.

Ahhoz, hogy a hálózatok megfelelő általánosítási képességgel rendelkezzenek, a rejtett rétegnek és az ahhoz közvetlenül kapcsolódó környezetrétegnek a méretét megfelelően alacsonyra kellett beállítani. Ez ebben az esetben 25 idegsejtet jelentett, amit kísérletezéssel választottam ki a 10-300 tartományból.

A kimeneti réteg mesterséges idegsejtjei közül 1 végezte el a szemantikai keretet felidéző lexikális egység („frame-evoking lexical unit”) címkézését, ezáltal a szemantikai keret felismerését. A keretelemeket egy kimeneti mintázat azonosította, melyben egy-egy neuron volt felelős egy-egy keretelem azonosításáért, valamint egy másik neuron jelezte, hogy a hálózat kimenete érvényes mintázatot tartalmaz. A kimeneteken a FrameNet full-text korpusz megfelelő keret- és keretelem kategóriáit reprezentáló mintázatok megjelenését vártam.

A hibát a tanítás során az SRBPTT („simple recurrent backpropagation through time”) algoritmus felhasználásával lépésenként csökkentettem. Egy-egy hálózat betanítását 1200 tanítási menetben végeztem el (a hibadiagramok alapján 800-1200 menet után stabilizálódott a kimeneti hiba a hibaminimum közelében a szemantikai keret és a szóábrázolási módszer függvényében), mindegyik menet egy teljes mondat betanításának felelt meg. A mondatok általagos hossza 21 szó volt (ezek voltak a betanítási menet eseményei; a súlyok nem az események, hanem a menetek végén változtak). A tanítási hibát sikerült a várt módon menedzselni, a feladat a kiválasztott eszközzel megoldható volt, a hálózat hatékonyan megjegyezte a tanítóban lévő FrameNet címkéket. Az új, korábban nem látott mondatokat (és számos először látott szót) tartalmazó tesztadatokra kapott pontossági és fedési értékeket a 3. szakaszban ismertetem.

### 2.2 Szavak ábrázolása a szemantikai feladatot végrehajtó hálózatok bemenetén

A szavakat a neurális hálózat számára numerikus adatokká kell alakítanunk, ennek 9 módszerét próbáltam ki, köztük olyan eljárásokat, amelyek a szavak disztribúciós tulajdonságait (nagy mintán megfigyelt együttes előfordulási adatait) kódolta.

1. 1HOT (one-hot, 1-az-N-ből): A szakirodalomban elterjedt megoldás, melyben a bemeneti vektor elemei közül egyet 1-re, a többit 0-ra állítjuk, és minden szótípushoz más vektort rendelünk. Új szótípus hozzáadása új elem bevezetésével történik, amelyet a neurális hálózatos kísérletekben a következő idegsejt-réteghez megfelelően hozzá kell kapcsolni, majd a hálózatot újratanítani.
2. COUNT-LOGFREQ: A disztribúciós szemantika [11] hagyományos gyakorlatának megfelelően minden szóhoz előállítottam egy olyan tulajdonságvektort, ami megmutatta, hogy az adott célszó más szavakkal milyen gyakorisággal fordult elő együtt egy nagy méretű (de szemantikai annotáció nélküli) korpuszban. Az ilyen eljárás során, például, ha az *iszik* célszót jellemezzük a *vizet*, *teát*, *kólából* és *haza* környezetszavakkal, akkor a *vizet*, *teát* és *kólából* környezetszavaknak megfelelő vektorelemek értéke magasabb lesz, a *haza* szóhoz tartozó vektorelem értéke alacsonyabb. A kapott vektorok egy valós vektortérben úgy kijelölik az adott szó helyét, hogy a hasonlóbb disztribúciójú szavakhoz tartozó vektorok hajlásszöge kisebb lesz (további részletekért ld. [11]). A disztribúció hasonlósága szemantikai, szintaktikai és morfológiai okoknak (együttesen és egymástól elválaszthatatlanul) köszönhető. A disztribúciós adatok összegyűjtéséhez a TC Wikipedia korpusz (http://nlp.cs.nyu.edu/wikipedia-data) véletlenszerűen kiválasztott 100 millió szavas részkorpuszát elemeztem. A többmilliárd szavas korpuszok potenciálisan jobb eredményt adnak, ugyanakkor az emberi tapasztalás korlátait messze túllépik. A TC Wikipedia korpuszból semmilyen annotációt nem használtam fel, és a célszavak 3+3 szavas környezetét vizsgáltam. Környezetszókként az 5000 leggyakoribb angol szót kerestem, ennyi lett a kapott tulajdonságvektorok elemeinek száma. Mivel néhány együttes előfordulásból (különösen a funkciószavak esetében) nagyon sokat találhatunk, a vektor elemeit súlyozni szükséges. A COUNT-LOGFREQ reprezentációban az együttes előfordulási gyakoriság logaritmusával számoltam.
3. COUNT-PPMI: A szóvektorokat a COUNT-LOGFREQ reprezentációnál ismertetett eljárással állítottam össze azzal a különbséggel, hogy a vektor elemeinek súlyozását másképpen végeztem: a célszavak és környezetszavak egyedi kölcsönös információját (EKI [7]; angolul: pointwise mutual information, PMI) használtam, amennyiben az pozitív érték volt, ellenkező esetben nulla lett a vektorelem értéke. Amennyiben a célszó ($c$) és a környezetszó ($k$) előfordulása független egymástól, akkor az együttes előfordulás valószínűsége $P(c) \times P(k)$, ehhez viszonyítjuk $c$ és $k$ megfigyelt együttes előfordulásainak számát ($M(c,k)$); a pozitív egyedi kölcsönös előfordulás értéke $pPMI=max(0,log(M(c,k) / P(c) \times P(k)))$.
4. RND-PPMI: COUNT-PPMI módszerrel készített szóvektorok mindegyikét egy másik, véletlenszám-generátorral kiválasztott szóvektorral felcseréltem, ezáltal a szóvektorokat megfosztottam valós disztribúciós (szövegkörnyezetet kódoló) tartalmuktól. A COUNT-PPMI szóábrázolással összehasonlítva az RND-PPMI reprezentáció alkalmas a disztribúciós információ hatásának megfigyelésére.

5. PRED-CBOW: Visszacsatolás nélküli, 1 rejtett réteget tartalmazó neurális hálózatban Mikolov és mtsai módszerével [8] létrehozott prediktív disztribúciós szóábrázolás. A CBOW („continuous bag of words") eljárás használata esetén a hálózat a bemeneti rétegen egy környezetablak szavait kapja, betanítás után a kimeneten pedig az ablak közepén álló szót jósolja meg. Számunkra nem a hálózat kimenete (a jóslás minősége), hanem a feladat megoldása során az adott célszó előállításához szükséges belső mintázat (a rejtett réteg aktivációs adatsora) az érdekes: ez a beágyazott mintázat lesz az adott célszó PRED-CBOW ábrázolása. A szakirodalomban kialakult gyakorlat szerint az így kapott aktivációs értékeket egy-egy vektor elemeinek tekintjük, a vektorok pedig minden szóhoz kijelölnek egy pontot egy sokdimenziós valós vektortérben úgy, hogy a disztribúciós szempontból hasonlóbb szavakhoz tartozó vektorok hajlásszöge kisebb lesz. A vektorok létrehozásához a TC Wikipedia fent említett részkorpuszát és a word2vec eszközt [9] használtam, 3+3 szavas környezetablak vizsgálatával. A vektorok 300 elemből álltak.
6. PRED-SKIPGRAM: a PRED-CBOW vektorokhoz hasonló eljárással létrehozott prediktív vektorok [8]. A skip-gram szóbeágyazás előállítására használt hálózat a bemenetén a célszót kapja meg, a kimeneten pedig a szó környezetét kell megjósolnia. Itt is igaz, hogy nem a jóslás pontossága, hanem a feladat megoldása során kialakuló aktivációs mintázatok (a rejtett réteg aktivációs szintjei) az érdekesek számunkra, ezeket a célszó disztribúciós tulajdonságait tömörítő szóbeágyazásokként kezeljük. Mérete: 300 valós érték minden szóvektorban.
7. RND-SKIPGRAM: PRED-SKIPGRAM módszerrel készített szóbeágyazások véletlenszám-generátorral kiválasztott párjait felcseréltem, ezt az eljárást minden szóra megismételtem. Az így kapott mintázatokban az adott célszó vonatkozásában valós disztribúciós adat nem maradt. Ez az ábrázolás a PRED-SKIPGRAM szóábrázolással összehasonlítva alkalmas a disztribúciós információ hatásának mérésére. Az RND-SKIPGRAM ábrázolást annak a megfigyelésnek az apropóján vezettem be, hogy a word2vec által generált PRED vektorokban nagyon sok a nullához közeli elem, ami befolyásolhatja a betanítás sikerét, amennyiben ezeket a vektorokat a további feldolgozás során is neurális hálózatokban használjuk fel. (A PRED-CBOW és PRED-SKIPGRAM ábrázolások ebből a szempontból hasonlóak, ezért RND-CBOW ábrázolást nem készítettem).
8. 1HOT + COUNT-PPMI: kombinált 1HOT és COUNT-PPMI reprezentáció minden célszóhoz. Mivel a disztribúciós ábrázolásmód esetén a szó és reprezentációja közt kölcsönösen egyértelmű megfeleltetés nem garantált, ugyanakkor a 1HOT reprezentációt éppen erre vezették be, így a kombinálásukból előny származhat. Az így készült reprezentációk nagy méretűek, esetünkben 5000 vektorelem a COUNT-PPMI reprezentációból és 5000 elem a 1HOT reprezentációból.
9. 1HOT + PRED-SKIPGRAM: kombinált 1HOT és PRED-SKIPGRAM ábrázolás minden szóhoz. Kipróbálásának oka a kölcsönösen egyértelmű megfeleltetés létrehozása prediktív szóbeágyazás mellett. Mérete 5000 + 300 elem vektoronként.

A prediktív szóábrázolások érdekes (bár ritkán tárgyalt) tulajdonsága, hogy úgy hozzuk őket létre, hogy egy nyelvészeti szempontból kevésbé dokumentált feladatot, a szókörnyezet nyelvi kategóriáktól független előrejelzését, illetve a környezet alapján a célszó előrejelzését tanulja meg egy neurális hálózat. A sikeres emberi kommunikáció szempontjából ez egy releváns feladat, hiszen a zajos környezetben az eredeti jel

felismerését nagymértékben segíti a megfelelő szavak előfeszítése, egyúttal hozzájárul ahhoz, hogy a jelek feldolgozása gyorsan és félreértésektől mentesebben történhessen meg.

### 2.3 A szoftverkörnyezet

A kísérletsor szoftveres infrastruktúráját a szerző hozta létre az alábbiak szerint. A kísérlet előkészítő szakaszában a FrameNet keretekről és keretelemekről gyűjtöttem adatokat erre a célra létrehozott programmal. Az előkészületek részeként a korpusz összes szavának minden szóreprezentációs mintázatát elő kellett állítani, amit szintén saját programmal végeztem el a PRED-SKIPGRAM és PRED-CBOW szóábrázolások kivételével, amelyek létrehozásához a word2vec eszközt [9] használtam. Ezután következett az a többlépcsős műveletsor, melyet minden szemantikai kerethez (12 db) és minden szóreprezentációhoz (9 db) külön elvégeztem, a kétszeres keresztvalidáció miatt pedig mindezt kétszer hajtottam végre, azaz összesen 216 kísérletről közlök összesített, átlagolt adatokat. Az eljárás lépései minden esetben ezek voltak:

1. A FrameNethez mellékelt full-text korpusznak az adott szemantikai keretet tartalmazó mondatait a neurális hálózati szimulátor által felismert bemeneti fájlokká alakítottam saját programmal, eközben a szavakat a megfelelő szóvektorokkal helyettesítettem a tanító- és tesztkorpuszban, valamint elhelyeztem a FrameNet kategóriacímkéknek megfelelő elvárt kimeneti mintázatokat.
2. Létrehoztam az adott kerethez és az adott szóreprezentációs módhoz tartozó mesterséges neurális hálózatot a LENS hálózatszimulátorban [10] és betanítottam azt a tanító adathalmazzal.
3. Ugyanezt a hálózatot teszteltem a tesztadatokkal, a tesztszimuláció során lementett kimeneti mintázatokat pedig saját programmal kiértékeltem, összehasonlítva a korpuszban látott és a hálózat kimenetén kapott szemantika keret- és keretelemcímkéket. Kiszámítottam a fedési és pontossági értékeket.
4. Keresztvalidáció céljából a tanító- és tesztadatokat felcseréltem, majd a 2-3. lépéseket megismételtem.

Az Elman hálózat paramétereinek beállításához előzetesen további kísérleteket végeztem (ennek során választottam ki a rejtett réteg méretét), valamint további vizsgálatokat hajtottam végre a leszámlálásos (COUNT) szóreprezentációk paramétereinek beállítása során (pl. a szövegablak méretének meghatározása). A szoftverkörnyezet létrehozása, ellenőrzése során értelemszerűen rengeteg további tesztfuttatást is végeztem. A szoftverfejlesztés és a kísérletek végrehajtása a 2014–2018 időszakban történt.

## 3 Eredmények

A neurális hálózatok a maximális teljesítményüket leszámlálásos disztribúciós szóreprezentációval nyújtották 91% pontosság és 86% fedés mellett a szemantikai keretek (FR) felismerése során. A keret*elemek* (FE) felismerése nehezebb feladat volt a következő okok miatt: *a)* néhány keretelem esetében nagyon kevés tanítóadat állt rendelkezésre, szélsőséges esetként volt olyan keretelem is, amihez csak 1 tanító- és 1

tesztadat volt; *b)* míg a keretfelismerés általában 1 szó (a keretet előhívó lexikai egység, „frame-evoking lexical unit") megfelelő címkézését igényli, a keretelemek általában több szóból állnak, amelyeket a mostani rendszerben egyesével kell felcímkézni, az összetevők előzetes kijelölése nem megoldott.

A 2. táblázat a keretek és a keretelemek felismerésének százalékos pontosságát (*p*), fedési értékét (*r*) és ezek harmonikus átlagát (*F*-mértékét) mutatja a 9 vizsgált szóábrázolás mellett.

| *Szóábrázolás* | *p (FR)* | *r (FR)* | *F (FR)* | *p (FE)* | *r (FE)* | *F (FE)* |
|---|---|---|---|---|---|---|
| 1HOT | 91,1 | 64,8 | *75,7* | 53,8 | 40,4 | *46,1* |
| COUNT-LOGFREQ | 91,2 | **86,4** | ***88,8*** | 55,9 | 42,7 | *48,4* |
| COUNT-PPMI | **92,5** | 84,6 | *88,4* | 56,4 | 46,9 | *51,2* |
| RND-PPMI | 89,8 | 75,9 | *82,3* | 54,2 | 42,6 | *47,7* |
| PRED-CBOW | 88,2 | 84,1 | *86,1* | 56,4 | **49,5** | ***52,7*** |
| PRED-SKIPGRAM | 89,4 | 83,3 | *86,3* | 57,2 | 48,8 | *52,6* |
| RND-SKIPGRAM | 81,1 | 67,8 | *73,8* | 49,6 | 40,1 | *44,4* |
| 1HOT + COUNT-PPMI | 91,9 | 84,7 | *88,2* | **57,7** | 45,7 | *51,0* |
| 1HOT + PRED-SKIPGRAM | 90,0 | 86,3 | *88,1* | 56,4 | 49,2 | *52,6* |

2. táblázat. A szóábrázolás hatása a keret (*FR*) és keretelem (*FE*) felismerésre

### 3.1 Disztribúciós információ nélküli eredmények

A 1HOT (one-hot, 1-az-N-ből) kódolási módszer egyszerűsége és elterjedtsége okán fontos viszonyítási alap a további eredmények értékelése szempontjából. Ezzel a szóábrázolással a keretfelismerés pontossága magas (91%), a fedés viszont alacsony (65%) volt. Ilyen ábrázolás esetén csupán 1 bemeneti egység szolgáltat információt a további feldolgozáshoz a kapcsolatain keresztül (jelen esetben 300 súlyozható kapcsolaton keresztül), a többi bemenetről a rejtett réteghez vezető kapcsolatok aktiváció hiányában nem tudnak a feldolgozáshoz hozzájárulni (ebben a kísérletben kb. 1,5 millió ilyen helyzetben lévő súlyozható kapcsolatról beszélünk). Kismértékű véletlen zaj hozzáadása ezt a problémát valamilyen mértékben orvosolhatja – ezt a lehetőséget ebben a kísérletben nem próbáltam ki, de két másik randomizált ábrázolásmódról közlök adatokat.

A 1HOT ábrázoláshoz hasonlóan a randomizált RND-PPMI és RND-SKIPGRAM szóvektorok szintén csak a szavak azonosítására voltak felhasználhatók a hálózat számára, hiszen az adott célszót jellemző disztribúciós adatokat nem találunk bennük. A 1HOT ábrázolás (melyben csupán 1 aktív elem van), az RND-PPMI (számos aktív neuronnal) és az RND-SKIPGRAM (számos aktív neuronnal, sok nullához közeli elemmel) módszerek egymástól jelentősen eltérően viselkedtek. Az RND-SKIPGRAM reprezentáció eredményei alacsonyak voltak, alulmúlták a 1HOT teljesítményét is, az RND-PPMI vektorok azonban meglepően jó eredményeket hoztak csupán azzal, hogy *nem akadályozták* a neurális hálózat betanítását, működését.

Ezen a ponton azt is meg kell jegyeznünk, hogy ugyan a fenti reprezentációk a szavak általános, nagy korpuszban megfigyelt disztribúciójáról nem tárolnak informá-

ciót, a szemantikai címkézést végző rendszer mégis hozzáfér a szövegkörnyezetre vonatkozó adatokhoz (még ha sokkal kisebb mennyiségben is) a FrameNet tanítómondatokból.

### 3.2 A disztribúciós adatok hatása

A 1HOT módszerhez képest a legjobb disztribúciós ábrázolásmód F-mértékben mért előnye a keretek felismerése közben 13,1 százalékpont, a keretelemek címkézése esetén pedig 6,6 százalékpont volt. A fedés értékét különösen látványos módon (21,6 százalékponttal) növelte a disztribúciós információk megfelelő használata. A COUNT-PPMI módszer használatával a valós szódisztribúciós adatoktól megfosztott RND-PPMI vektorokhoz képest 6,1 százalékpontos F-mérték növekedés következett be.

A 3. szakasz bevezetőjében ismertetett okokból a keretelem-felismerés nehezebb feladat volt, ami a pontossági és fedési értékekben is tükröződött, azonban a disztribúciós adatok jótékony hatása itt is jól látható volt. Az RND-PPMI 47,7%-os eredményéhez (F-mérték) képest a valós disztribúciós adatokat tartalmazó COUNT-PPMI vektorok 51,2%-os (F) eredménye kb. 7 százaléknyi (3,5 százalékpontos) növekedést jelent. A hagyományos 1HOT és az ebben a feladatban legjobb (PRED-CBOW) szóábrázolás közti különbség pedig 14% (6,6 százalékpont) volt.

Baroni, Dinu és Kruszewski munkája [2] bemutatja, hogy a prediktív szóbeágyazások jobban teljesítenek a szokásos benchmark feladatok széles spektrumán, mint a hagyományos leszámlálós eljárással készített szóvektorok. Ez a várakozás ebben a kísérletsorban nem igazolódott, a prediktív (PRED-SKIPGRAM és PRED-CBOW) és leszámlálós (COUNT-LOGFREQ és COUNT-PPMI) módszerek közül ebben az esetben nem tudunk győztest hirdetni. A keretfelismerés során a COUNT módszerek, a keretelem-felismerésben pedig a PRED szóábrázolások voltak valamivel jobbak, kis különbséggel.

Általában is megfigyelhetjük, hogy a különböző disztribúciós eszközök (COUNT-LOGREQ, COUNT-PPMI, PRED-CBOW és PRED-SKIPGRAM) teljesítményének szórása ebben a feladatban alacsony volt. Ennek valószínűsíthető oka az, hogy a szemantikai címkézést végző rendszer a szóreprezentációk adatain kívül is hozzáfér a szövegkörnyezettel kapcsolatos információkhoz a FrameNet tanítómondatokból, hiszen egyrészt a visszacsatolt hálózat a környezetrétegben tárolt információk segítségével emlékszik a mondat korábbi szavaira, másrészt az idegsejtek közti súlyozott kapcsolatok hosszú távú memóriaként működnek az egész hálózatban az összes tanítómondatra vonatkozóan. Ez fontos különbség a disztribúciós szemantikai benchmark kísérletekhez képest, ahol a szavak szövegkörnyezetére vonatkozó adatokat tipikusan csak a szóreprezentációkból nyerhetjük ki, és csupán ezen adatokkal végezhetünk további műveleteket. Amennyiben a szóreprezentációkból olyan adatok hiányoznak, amelyek a feladat végrehajtásához lényegesek lennének, a hiányukat más forrásból nem tudjuk célzottan pótolni, míg ebben a kísérletsorban ez lehetséges volt a szemantikai címkézést végző neurális hálózat számára.

### 3.3 Szóábrázolások kombinálása

Mivel a disztribúciós szóvektorokkal a szavak és ábrázolások kölcsönösen egyértelmű megfeleltetése nem biztosított, kipróbáltam a szóvektorok és a 1HOT ábrázolás együttes alkalmazását is a tanulórendszer bemenetén, ezzel egyszerre megvalósítva a disztribúció kódolását és a szavak egyértelmű azonosításának feladatát. A *keretfelismerés* esetében a 1HOT + PRED-SKIPGRAM mérhető javulást eredményezett a PRED-SKIPGRAM önálló alkalmazásához képest (ld. még a korábbi SKIPGRAM-os megfigyeléseinket a 3.1 szakaszban). A COUNT-PPMI vektorokhoz adott 1HOT minták ugyanakkor nem hoztak további teljesítményjavulást a feladat megoldása szempontjából. A *keretelem-felismerési feladatban* a PRED-SKIPGRAM kiegészítése a 1HOT adatokkal a fedést ugyan enyhén növelte, de a pontosság csökkenése miatt az F-mérték változatlan maradt. A COUNT-PPMI vektorokhoz adott 1HOT minták pedig összességében még csökkentették is a pontosság és a fedés harmonikus átlagát.

## 4 Konklúzió

A FrameNet kiválasztott szemantikai kategóriáit tanuló és felismerő neurális hálózatot 9 különböző szóreprezentáció felhasználásával próbáltam ki egy komplex kísérletsorozatban. A bemutatott kísérletek alátámasztják, hogy a szemantikai keretek felismerésére lehetőség van egyszerű visszacsatolt hálózatokkal, és a feladat végrehajtását elősegíti a disztribúciós adatok megjelenítése a szóreprezentációkban. A továbbfejlesztés legfontosabb területe a keretelemek tekintetében a mondattani összetevők azonosítása és együttes címkézése lehet, ezen kívül további visszacsatolt neurális hálózattípusok kipróbálása és a vizsgált szemantikai keretek körének bővítése jelent közvetlen továbblépési lehetőséget.

## Bibliográfia

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the COLING-ACL, Montreal (1998)
2. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of ACL (2014) 238–247
3. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods 39 (2007) 510–526
4. Bullinaria, J.A. & Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming and SVD. Behavior Research Methods 44 (2012) 890–907
5. Elman, J.L.: Finding structure in time. Cognitive Science 14 (1990) 179–211
6. Friederici, A.D.: The Brain Basis of Language Processing: From Structure to Function. Physiological Reviews 91 (2011) 1357–1392
7. Kálmán L.: Már megint bakot lövünk. https://qubit.hu/2018/07/15/mar-megint-bakot-lovunk (2018)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. https://arxiv.org/abs/1301.3781. (2013)

9. Mikolov, T., Sutskever, I., Chen, K, Corrado, G.S. & Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems (2013) 3111–3119
10. Rohde, D.L.T.: LENS: The light, efficient network simulator. Technical Report CMU-CS-99–164. Carnegie Mellon University, Department of Computer Science, Pittsburgh, PA (1999)
11. Tóth, Á.: The Company that Words Keep: Distributional Semantics. Debrecen University Press, Debrecen (2014)

# Orvosi alkalmazások

# Információkinyerés magyar nyelvű gerinc MR leletekből

Kicsi András[1], Pusztai Péter[1,2], Szabó Ledenyi Klaudia[1], Szabó Endre[3], Berend Gábor[1], Vincze Veronika[2], Vidács László[1,2]

[1]Szegedi Tudományegyetem, Informatikai Intézet, IKK
Szeged, Árpád tér 2.
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
Szeged, Tisza Lajos körút 103.
{akicsi,pusztaip,ledenyik,berendg,vinczev,lac}@inf.u-szeged.hu
[3]Szegedi Tudományegyetem
endrebacsi@gmail.com

**Kivonat** Cikkünkben magyar nyelvű radiológiai leletek automatikus feldolgozásának módszeréről és kezdeti kísérleteink eredményeiről számolunk be. Először bemutatjuk a felhasznált adatbázist és az alkalmazott annotációs elveket, majd ismertetjük kísérleti módszereinket. Bemutatjuk eredményeinket, ezt követően pedig ismertetjük a rendszer jelenlegi erősségeit és gyengébb pontjait, végül szót ejtünk a továbbfejlesztési lehetőségekről is.
**Kulcsszavak:** radiológia, információkinyerés, nlp, annotáció

## 1. Bevezető

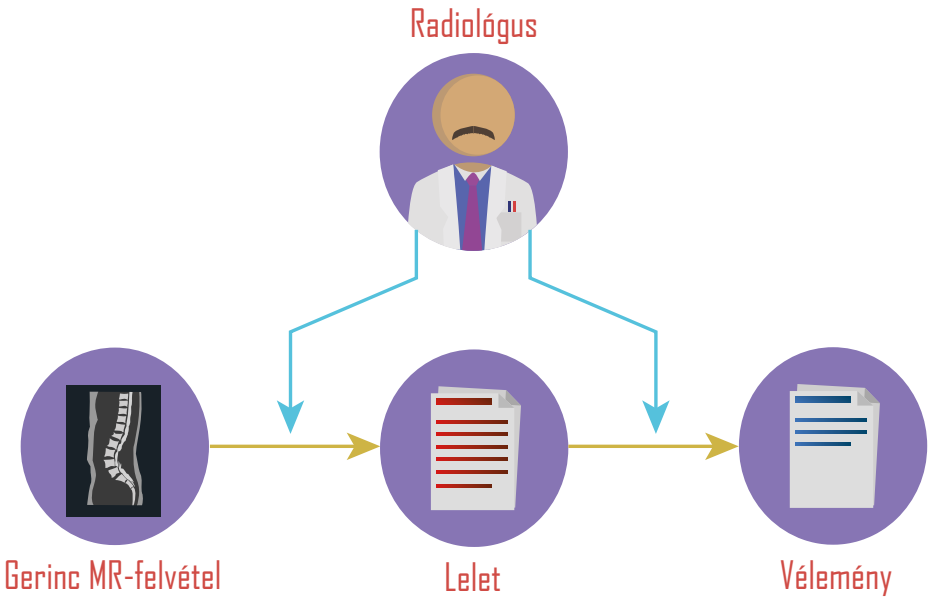
A klinikai gyakorlatban óriási mennyiségű dokumentum keletkezik nap mint nap, melyek között találhatunk leleteket, zárójelentéseket, orvosi lapokban megjelenő publikációkat. Ez a hatalmas szövegmennyiség rengeteg nyers adatot rejt magában, melyek kiaknázásában az informatika egy részterülete, a számítógépes nyelvészet (natural language processing, NLP) nyújthat segítséget. Az információkinyerés feladata, hogy nagy mennyiségű strukturálatlan vagy gyengén strukturált szövegből automatikus eszközökkel összegyűjtse a szövegben meglévő információt. Egy jellemző példa lehet a fehérje–fehérje interakciók kinyerése biológiai szövegekből, ahol a különféle biológiai entitások közti kapcsolatokat kell összegyűjteni. Angol nyelvre már jó ideje léteznek olyan automatizált megoldások, melyek az orvosi szövegekben rejlő információ kiaknázására törekednek, lásd például a páciens dohányzási szokásainak vagy elhízottságának megállapítása orvosi zárójelentések alapján [1,2], a magyar nyelvű orvosi NLP-vel azonban viszonylag kevés munka foglalkozott eddig (lásd pl. [3]).

Ebben a munkában radiológiai leletek automatikus feldolgozásával foglalkozunk. Míg angol nyelvű radiológiai leletek feldolgozására születtek már szép eredmények [4], tudomásunk szerint magyar nyelvű leleteket e szempontból még nem vizsgáltak. Célunk, hogy a leletekből minél több olyan információt nyerjünk

ki automatikus módszerekkel, melyek megkönnyítik a klinikus munkáját. E feladat szakszerű megvalósításához csapatunkban orvos, informatikus és nyelvész kollégák működnek együtt. A továbbiakban részletesen bemutatjuk a felhasznált anyagokat és módszereket, majd ismertetjük elért eredményeinket, végül szót ejtünk a kutatás lehetséges további irányairól is.

## 2. Motiváció

A radiológiai klinikai gyakorlatban a betegről először elkészül egy modern képalkotó eljárással (például CT, MRI, esetleg röntgen) létrehozott felvétel, melynek alapos vizsgálata során a radiológus szakértő megállapítja az esetleges eltéréseket, rendellenességeket, adott esetben diagnosztizálja a betegséget. Szükség esetén a betegről rendelkezésre álló korábbi leleteket, radiológiai felvételeket is tanulmányozza a minél pontosabb véleményalkotás érdekében. Végül mindezt leletbe foglalja, ahol a fentiek szöveges összegzésén kívül diagnosztikai véleményt kell alkotnia, illetve további vizsgálatokra, terápiára is tehet javaslatot. A munkában egy automatikus diktálórendszer is segíti. A fentieket az 1. ábrán láthatjuk összefoglalva.




1. ábra: A radiológus munkája a vizsgálat után

Cikkünk fő célja, hogy megkönnyítsük a radiológus szakember munkáját. Ennek első lépése lehet a leletek automatikus kivonatolása, azaz kezdeti feladatként

fel kell ismernünk a lelet szövegében található legfontosabb kifejezéseket. Ilyenek lehetnek például az egyes testrészek, elváltozások, betegségek nevei stb., illetve a köztük levő kapcsolatok. Ha ezeket az információkat automatikus úton képesek vagyunk kinyerni, a következő lépcsőben automatikus eszközökkel generálhatunk egy diagnosztikai véleményt, melyet használatkor természetesen a radiológus szakember felülvizsgál, szükség esetén felülbírál. A kutatás jelen fázisában a lelet szövegében található fontos kifejezések minél pontosabb kinyerésére törekszünk: cikkünk további részében ennek folyamatát és jelenlegi eredményességét taglaljuk részletesen.

## 3. Annotáció

Jelen munkában gerinc MR-leletekkel dolgozunk, melyeket a Szegedi Tudományegyetem Radiológiai Klinikájának munkatársai bocsátottak rendelkezésünkre. A leletekben található személyes jellegű adatokat természetesen az adatvédelmi előírásoknak megfelelően kezeljük vizsgálataink során. Kísérleteinkben 250 lelettel dolgozunk, azonban a későbbiekben várható a leletállomány kibővülése is.

Az adatok megfelelő annotációja a sikeres gépi tanulás alapfeltétele. Ehhez pedig létfontosságú a megfelelően letisztázott alapelvek lefektetése. Ennek érdekében radiológus, informatikus és nyelvész részvételével több találkozón, iteratívan fejlesztettük ki jelenleg használt annotációs módszerünket. A találkozókra egységesen kijelölt 10 darab leleten végeztek a résztvevők annotációt a korábban megbeszélt elvek szerint. A tényleges annotáció már az így kialakított elvek alapján történt, a munkához a Brat annotációs eszközt [5] használtuk fel, amelyet a kialakított módszer alapján konfiguráltunk.

A következő címkék jelölésének szükségességét állapítottuk meg:

*Testrész:* Az emberi test egy olyan része, amelyet egy átfogó névvel megnevezhetünk. Például szolgálhatnak rá a 2. ábrán látható kifejezések.


2. ábra: Testrész-ként annotált kifejezések

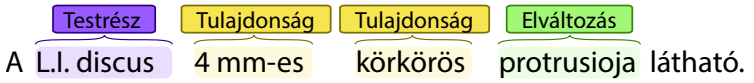
*Hely:* Egy helyet ír le az emberi testen belül. Nagy átfedésben van a Testrész címkével, viszont több különböző testrészt és azok viszonyait is felölelheti, az elváltozás helye gyakran ilyen formában van megadva. Példát láthatunk rá a 3. ábrán.

*Elváltozás:* Az elváltozás, vagy esetleg az elváltozás hiánya, amit a szöveg általában megállapít. Ilyenek például a 4. ábrán látható kifejezések.


3. ábra: Hely-ként annotált kifejezés

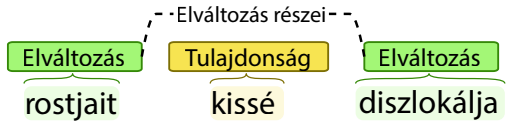

4. ábra: Elváltozás-ként annotált kifejezések

*Tulajdonság:* Olyan módosító kifejezések, amelyek valamilyen tulajdonságot neveznek meg a leleten belül. Módosíthatják vagy pontosíthatják egy Elváltozás típusát, vagy leírhatják annak fokozatát vagy mértékét is. Mindkettőre láthatunk példát az 5. ábrán látható mondatban.




5. ábra: Egy mondat Tulajdonság-ként annotált kifejezésekkel

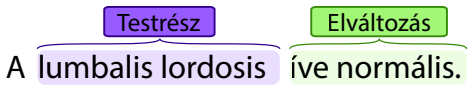
Az entitások közötti relációk közül csupán egyfélét engedtünk meg, ezt *Részei* relációnak hívtuk. Ilyen kapcsolatot azonos címkével ellátott kifejezések között állapítottunk meg, ha úgy találtuk, hogy valójában ugyanazon kifejezés részeit képezik. A reláció fő célja, hogy a kifejezés belsejébe ékelődött egyéb szavak kihagyhatók legyenek a jelölésből. A 6. ábrán látható példa ábrázolja. A feldolgozáskor ezeket a szövegbeli előfordulás sorrendjében kötjük össze és tároljuk egyetlen Elváltozásként.




6. ábra: Részei reláció egy beékelődött Tulajdonság fölött

A találkozók során felmerült kérdések alapján arra jutottunk, hogy a tagadást és bizonytalanságot jelző kifejezéseket nem jelöljük a feladat jelen fázisában, noha a későbbiekben erre mutatkozhat igény. Ezenkívül számos lelet tartalma egy

előző leletre épít, és ehhez viszonyít, mint például „A 2017-es vizsgálathoz képest...” típusú mondatokat tartalmazók, amelyek az elváltozásban bekövetkezett változásokra vonatkoznak legnagyobb részben. Ezen leletek automatizált módszerrel kiszűrésre kerültek, ugyanis az anonimizálás után, és mert egyetlen hónap vizsgálatainak leletei álltak csak rendelkezésünkre, a visszakövetés nem megvalósítható. További kérdéses esetet jelentett például, ha az elváltozás a testrész valamilyen tulajdonságára, például szélességére vonatkozik. Itt azt állapítottuk meg, hogy az ilyen, aspektust leíró szavakat az elváltozás részének jelöljük, mint „szélessége normális”’, és nem tulajdonságként, mivel a szó szervesen az elváltozáshoz kapcsolódik, valamint nem is minden esetben lehetne tulajdonság címkével sem ellátni. Erre mutat be egy például a 7. ábrán látható mondat.




7. ábra: Egy testrész valamely aspektusának elváltozása

A jelen cikkben közölt eredményeink a Helyként annotált adatokat nem tartalmazzák, mivel ezek átfedésben lehetnek a Testrészekkel, illetve nagy mennyiségű általános szót tartalmaznak, ezért a kiértékelést jelentősen komplikálnák. Ezek az annotációk azonban a kutatás későbbi fázisaiban szintén nagyon hasznosnak bizonyulnak, így szükségesnek láttuk keresésüket.

## 4. Kísérletek

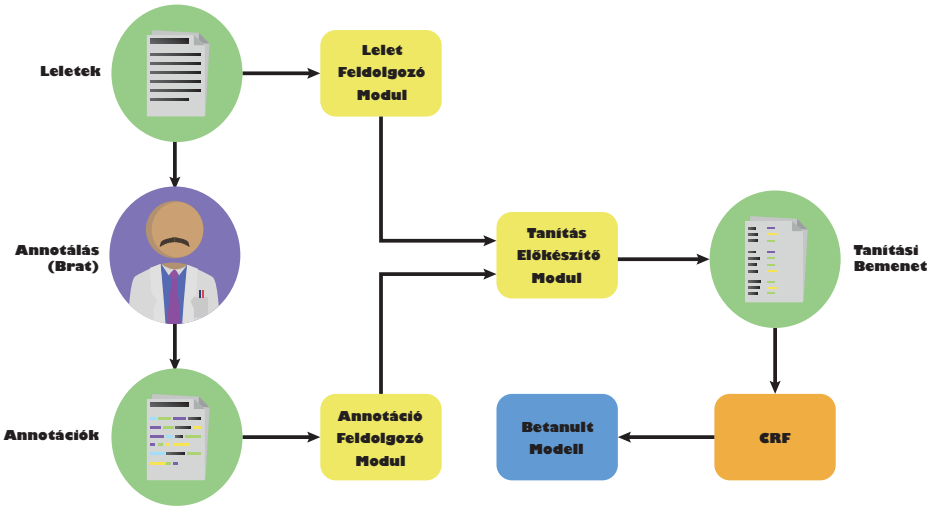
Ebben a fejezetben bemutatjuk a leletekből történő információkinyerést végző szekvenciajelölő osztályozónkat, illetve az ennek segítségével elért eredményeinket. A fejezet végén kitérünk a szekvenciajelölő pontosabbá tételének lehetőségeire.

### 4.1. Gépi tanulás

A teljes folyamat bemutatása a 8. ábrán látható. A leletek feldolgozása után az annotációkat az eredeti leletekhez rendeljük, majd előkészítjük a gépi tanuláshoz az adatokat. Ez lesz az alapja a gépi tanulási szakasznak. A tanuláshoz felhasznált címkék előfordulásainak darabszámát az 1. táblázat szemlélteti.

A testrészek, elváltozások, illetve tulajdonságok beazonosításával kapcsolatban megfogalmazott szekvenciajelölési probléma megoldására egy feltételes valószínűségi modellt (CRF) [6] hoztunk létre. A szekvenciajelölő modell létrehozása, illetve tanítása a CRFsuite [7] csomag segítségével történt.

A szekvenciákban található tokensorozatok leírására az egyes tokenek, illetve a környezetükben – tőlük legfeljebb kettő távolságra – található tokenek felszíni jegyeiből származtattuk a jellemzőkészletet. A jellemzőteret alkotó indikátorváltozók magukat a szóalakokat, illetve a tokenekben előforduló nagybetűket,




8. ábra: A tanulás folyamatának áttekintése

1. táblázat. Az egyes címkék előfordulási gyakorisága a tanító és teszt halmazon. A semelyik érdemi címkével el nem látott tokenek számát az O jelű sor tartalmazza.

| Címke | tanító | teszt |
|---|---|---|
| Testrész | 3912 | 983 |
| Elváltozás | 5109 | 1323 |
| Tulajdonság | 2106 | 514 |
| O | 7547 | 1899 |
| Összesen | 18674 | 4719 |

illetve számjegyeket, kódolták. Az előzőeken túl az egyes tokenek szufixében álló karakterkettesekből-és hármasokból is alkottunk jellemzőket. A karakterprefixekből származtatott jellemzők használatára is tettünk kísérletet, azonban az így kapott modellünk eredményessége elmaradt a kizárólag karakterszuffixeket figyelembe vevőétől. A mondatokra, illetve tokenekre bontást a Spacy [8] nyílt forrású szoftverkönyvtár segítségével végeztük megfelelő konfigurációval.

A jelen kísérletekben a Helyek címkézésére még nem törekedtünk, a Helyként annotált tokeneket az annotálás során érdemi címkével el nem látottakkal azonos módon kezeltük. Az annotációs módszereknél leírt Részei relációk a feldolgozás korai fázisában feloldásra kerülnek.

### 4.2. Eredmények

A kísérleti eredmények a 2. táblázatban láthatóak. A gépi tanuláshoz a 250 leletből 80%-ot választottunk le tanulóhalmaznak, és a kapott modellt a maradék 20% leleten értékeltük ki. A kiértékelést címkénként végeztük el, tehát külön

értékeltük ki a *Testrész*, *Elváltozás* és *Tulajdonság* címkéken mért teljesítményt. Az F1 értékek 90% felett már jó eredményt mutatnak. A modell jó tulajdonsága, hogy kiegyensúlyozott: közel azonos a pontosság (precision) és a fedés (recall) értéke is, egyedül a *Tulajdonság* címke esetén alacsonyabb a fedés. A gerinc egy meglehetősen specifikus területe az emberi testnek, ezért a *Testrész* címke esetén valószínűleg egy viszonylag kis szókészletre tanulunk, így ebben az esetben a 92,3%-os F1 érték még várhatóan növelhető lesz azáltal, ha például az egyes szóalakok anatómiai atlaszokban való előfordulásának tényét további indikátorváltozók formájában beépítjük a modellünk jellemzőterébe.

2. táblázat. Információkinyerés eredmények CRF szekvenciajelölő használatával

| | Pontosság | Fedés | F1-érték | Szupport |
|---|---|---|---|---|
| Testrész | 0,931 | 0,916 | 0,923 | 983 |
| Elváltozás | 0,912 | 0,904 | 0,908 | 1323 |
| Tulajdonság | 0,917 | 0,837 | 0,875 | 514 |
| Összesen | 0,920 | 0,896 | 0,907 | 2820 |

Testrészek esetén tipikusnak mondhatóak azok a mondatok, ahol a mondat elején, topik pozícióban szerepel a csigolya, melyről az orvos megállapításokat tesz. Abban az esetben, ha a csigolya rövid jelölése önmagában szerepel (a *csigolya* szó nélkül) és nem kezdő pozícióban, a szekvenciajelölő már hajlamosabb tévedni. Például a következő mondatokban az S.I. csigolyát a modell jelenleg nem ismeri fel Testrészként: *Az L.V. csigolya 3 mm t hátra csúszott az S.I. fölött Az L.V. discus víztartalma és magassága csökkent enyhe centralis előboltosulása a durazsákot eléri.*

A modell szinte egyik lényeges szót sem ismerte fel a következő mondatban: *A L.IV V discus magasságában az anterior longitudinalis szalag vastagabb.* Ezek közül a *vastagabb* szó elváltozásra utal, amit a modell nem ismert fel annak ellenére hogy nehezebb kifejezésekkel is megbirkózik. Szintén gondot jelentett a két latin kifejezés is. Hasonló mondatok esetén felmerül a tanító minta méretének kérdése, itt úgy gondoljuk, hogy a mintaszám növelésével jelentős előrelépés várható.

### 4.3. Fejlesztési lehetőségek

Bár már az első eredmények is biztatóak, számos továbbfejlesztési lehetőséget látunk a teljes folyamatban. A modell tévesztései alapján és a szakirodalom fényében a következő négy irányvonalat emeljük ki:

*Annotáció ellenőrzése* A módszer több iteráció során alakult ki, azonban a leletek nagy részét egy orvos annotálta, így szükség van független ellenőrzésre orvosi és nyelvészeti szempontból is.

*Nyelvi elemzés* Ebben a kísérletben a nyelvi elemzés eredménye még nem jelenik meg a tanulóadatokban. A szófajok és mondatrészek hozzárendelésétől mindenképp mérhető javulást várunk. A tagadás fontos szerepet tölt be a leletek értelmezésénél, melyet az annotálás figyelmen kívül hagy, ezt az információt is a nyelvi elemzés fogja biztosítani. A magyarlanc elemző [9] tartalmaz dependenciaelemzőt is, melyet szintén használni fogunk a jövőben.

*Mély tanulás* A jelenlegi gépi tanulási módszer kevés adaton is jól működik. Az adatmennyiség növelésével lehetőség nyílik mély neuronhálók használatára, melyet a gépi tanulás sok területén sikerrel alkalmaztak a közelmúltban.

*További leletek* További leletek annotálásával tovább pontosítható a gépi tanulás. Nagyobb mennyiségű lelet további annotálása korlátokba ütközik, viszont a leletek önmagukban is fontos információt jelentenek a mély tanulás számára.

## 5. Kapcsolódó kutatások

A radiológiai leletek jellemzően még mindig szabad megfogalmazású, a radiológus által diktált szövegek, nem pedig jól kategorizált adatgyűjtemények, így azok információtartalmának megfelelő kinyerése kihívás elé állítja a kutatókat. Általánosságban elmondhatjuk, hogy a jelenleg vezető kutatások az adatok kinyerésére valamilyen, gyakran gépi tanulással kiegészített, természetesnyelv-feldolgozó (NLP) módszert használnak. Az alkalmazások köre a kinyert információ típusától függően széles spektrumon változik. Többek között beszélhetünk diagnózissegéd [10], [11], [12], diagnosztikai minőségbiztosítást [13], [14], [15], [16], a leletek automatikus BNO kódolását végző [17], a nem várt elváltozásokra adott válaszlépéseket [18], vagy a további vizsgálatokra vonatkozó ajánlásokat figyelő [19], illetve a páciens egészségi állapotát nyomon követő alkalmazásokról [20]. A közelmúltban több olyan összefoglaló cikk is megjelent, mely jól bemutatja az elmúlt egy évtizedben történt fontosabb előrelépéseket [21], [22], [23], [24], [25], [26].

A terület folyamatos bővülése ellenére a nemzetközi szakirodalom viszonylag szegényesnek mondható a kifejezetten gerincröntgen leleteket, NLP és gépi tanulással feldolgozó tanulmányokat illetően. Tan és munkatársai egy szabályalapú és egy gépi tanuláson alapuló rendszer teljesítményét hasonlították össze 26 alsóháti fájdalomra utaló orvosi megállapítás radiológiai leletekben történő felismerésében [27]. A feladatot a gépi tanulást alkalmazó rendszer 0,98, míg a szabályalapú rendszer 0,90 AUC (vevő működési karakterisztika görbe alatti terület) érték mellett teljesítette. A szerzők egy másik, 2018-as tanulmányukban reguláris kifejezéseket alkalmazó, szabályalapú NLP algoritmust mutattak be, mely a radiológiai leletekben az 1-es típusú Modic véglemez elváltozásokat 0,79 F1-érték mellett ismerte fel [28]. A szerzők a reguláris kifejezéseken alapuló algoritmus hátrányaként említik, hogy a gerincleletek szövegének változatossága miatt nehéz minden esetet lefedő szabályrendszert kialakítani. Wang és szerzőtársai 6 általános, csontritkulásból fakadó töréstípus szabad megfogalmazású radiológiai leletekben történő felismerésére fejlesztettek szabályalapú NLP alkalmazást,

mely elsősorban reguláris kifejezések használatával végzi az osztályozást [29]. A gerinctöréses esetek felismerésében a modell 0,91 F1-értéket mutatott. Hassanpour és munkatársai SVM technológiával kiegészített CRF modellt fejlesztettek az orvosi szempontból releváns elváltozások radiológiai leletekben történő felismerésére. Az alkalmazás ezen túlmenően vizsgálta az elváltozások állapotában bekövetkező változások mértékét és jelentőségét is [30]. A modell a jelentős elváltozások azonosításában 0,75, míg a változás mértékének azonosításában 0,95 F1-értéket ért el. Xu és szerzőtársai gyakori szöveges mintázatok bányászatával (labeled sequential pattern, LSP) támogatott CRF modellt fejlesztettek a radiológiai leletekben található további vizsgálatokat javasló mondatok felismerésére [31]. Az LSP feladata az ajánlást nagy valószínűséggel nem tartalmazó mondatok kiszűrése volt, míg a CRF az ajánlást tartalmazó mondatok azonosítását végezte. A modell az ajánlást tartalmazó mondatok felismerésében 0,88 F1-értéket ért el.

Magyar nyelvű klinikai szövegek feldolgozásában is születtek már eredmények [32]. Például orvosi szövegek automatikus szegmentálására és az orvosi rövidítések automatikus kezelésére gépi tanuláson alapuló rendszert fejlesztettek [33,34]. A leletek szövegeiben gyakran előforduló elírások, elgépelések automatikus javítása szintén elengedhetetlen egy jól működő információkinyerő alkalmazáshoz [35]. Az utóbbi években ezen felül megvalósult egy szemészeti klinikai keresőrendszer [36], illetve egy magyar nyelvű orvosi leletekben, releváns kifejezések azonosítására specializált, nem felügyelt módszereket alkalmazó rendszer is [37].

Tanulmányunk egy további lépés a magyar nyelvű szövegekből történő klinikai információkinyerés felé, mely egyelőre a gerincröntgen leletekben található elváltozás, tulajdonság és testrész típusú szavak detektálásra szorítkozik. Noha összahasonlítható eredmények magyar nyelvre egyelőre nem állnak rendelkezésre, azonban a nemzetközi szakirodalomban található számszerű eredményekkel összevetve rendszerünk versenyképesnek tűnik.

## 6. Összegzés

Cikkünkben magyar nyelvű radiológiai leletek automatikus feldolgozásának módszeréről és az első kapcsolódó kísérletekről számoltunk be. Kidolgoztunk egy módszert a leletek legfontosabb elemeinek annotálására, melyet radiológus, nyelvész és informatikus részvételével iteratívan, több körben finomítottunk. A módszer alapján a radiológus 250 gerinc MR lelet annotálását végezte el. Az információkinyeréshez gépi tanulást alkalmaztunk, szem előtt tartva, hogy a jövőben a gerinc MR-felvételeken kívül tágabb felhasználási területeket is meg szeretnénk nyitni. Egy CRF tanulót alkalmaztunk, mely egy korszerű és gyakorta használt megoldás névelem-felismerésre. Az első kísérletek 87,5%-92,3% közötti F1-értékkel zárultak, melyek összemérhetőek a hasonló célokat kitűző, de angol nyelvű leleteken elvégzett kísérletekkel. A jelenlegi megoldás számos fejlesztési lehetőséget kínál, mint a nyelvi elemzés felhasználása és a mély tanulás alkalmazása.

## Köszönetnyilvánítás

A jelen cikkben közölt kutatást részben az Emberi Erőforrások Minisztériuma támogatta (20391-3/2018/FEKUSTRAT).

## Hivatkozások

1. Uzuner, O., Goldstein, I., Luo, Y., Kohane, I.: Identifying Patient Smoking Status From Medical Discharge Records. Journal of the American Medical Informatics Association: JAMIA **15**(1) (2008) 14–24
2. Uzuner, O.: Recognizing Obesity and Comorbidities in Sparse Data. Journal of the American Medical Informatics Association **16**(4) (2009) 561–570
3. Orosz, Gy., Novák, A., Prószéky, G.: Lessons Learned from Tagging Clinical Hungarian. International Journal of Computational Linguistics and Applications **5** (2014)
4. Friedman, C., Alderson, P.O., Austin, J.H.M., Cimino, J.J., Johnson, S.B.: A General Natural-language Text Processor for Clinical Radiology. Journal of the American Medical Informatics Association **1**(2) (1994) 161–174
5. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: A Web-based Tool for NLP-Assisted Text Annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, Association for Computational Linguistics (2012) 102–107
6. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 282–289
7. Okazaki, N.: CRFsuite: A Fast Implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/ (2007)
8. Honnibal, M.: spaCy: Industrial-Strength Natural Language Processing. https://spacy.io/ (Utoljára látogatva: 2018-11-10)
9. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. http://rgai.inf.u-szeged.hu/index.php?lang=en&page=magyarlanc (2013)
10. Pham, A.D., Névéol, A., Lavergne, T., Yasunaga, D., Clément, O., Meyer, G., Morello, R., Burgun, A.: Natural Language Processing of Radiology Reports for the Detection of Thromboembolic Diseases and Clinically Relevant Incidental Findings. BMC Bioinformatics **15**(1) (2014) 266
11. Rink, B., Roberts, K., Harabagiu, S., Scheuermann, R.H., Toomay, S., Browning, T., Bosler, T., Peshock, R.: Extracting Actionable Findings of Appendicitis from Radiology Reports Using Natural Language Processing. AMIA Joint Summits on Translational Science Proceedings. AMIA Joint Summits on Translational Science **2013** (2013) 221
12. Solti, I., Cooke, C.R., Xia, F., Wurfel, M.M.: Automated Classification of Radiology Reports for Acute Lung Injury: Comparison of Keyword and Machine Learning Based Natural Language Processing Approaches. In: Proceedings - 2009 IEEE International Conference on Bioinformatics and Biomedicine Workshops, BIBMW 2009. Volume 2009., NIH Public Access (2009) 314–319

13. Raja, A.S., Ip, I.K., Prevedello, L.M., Sodickson, A.D., Farkas, C., Zane, R.D., Hanson, R., Goldhaber, S.Z., Gill, R.R., Khorasani, R.: Effect of Computerized Clinical Decision Support on the Use and Yield of CT Pulmonary Angiography in the Emergency Department. Radiology **262**(2) (2012) 468–474
14. Ip, I.K., Mortele, K.J., Prevedello, L.M., Khorasani, R.: Focal Cystic Pancreatic Lesions: Assessing Variation in Radiologists' Management Recommendations. Radiology **259**(1) (2011) 136–41
15. Sistrom, C.L., Dreyer, K.J., Dang, P.P., Weilburg, J.B., Boland, G.W., Rosenthal, D.I., Thrall, J.H.: Recommendations for Additional Imaging in Radiology Reports: Multifactorial Analysis of 5.9 Million Examinations. Radiology **253**(2) (2009) 453–61
16. Dang, P.A., Kalra, M.K., Blake, M.A., Schultz, T.J., Stout, M., Lemay, P.R., Freshman, D.J., Halpern, E.F., Dreyer, K.J.: Natural Language Processing Using Online Analytic Processing for Assessing Recommendations in Radiology Reports. Journal of the American College of Radiology **5**(3) (2008) 197–204
17. Farkas, R., Szarvas, Gy.: Eljárás radiológiai leletek automatikus BNO kódolására. In Tanács, A., Csendes, D., eds.: V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2007), Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2007) 149–157
18. Dutta, S., Long, W.J., Brown, D.F., Reisner, A.T.: Automated Detection Using Natural Language Processing of Radiologists Recommendations for Additional Imaging of Incidental Findings. Annals of Emergency Medicine **62**(2) (2013) 162–169
19. Yetisgen-Yildiz, M., Gunn, M.L., Xia, F., Payne, T.H.: Automatic Identification of Critical Follow-Up Recommendation Sentences in Radiology Reports. AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium **2011** (2011) 1593–602
20. Cheng, L.T.E., Zheng, J., Savova, G.K., Erickson, B.J.: Discerning Tumor Status from Unstructured MRI Reports-Completeness of Information in Existing Reports and Utility of Automated Natural Language Processing. Journal of Digital Imaging **23**(2) (2010) 119–132
21. Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., Liu, H.: Clinical Information Extraction Applications: A Literature Review (2018)
22. Pons, E., Braun, L.M., Hunink, M.G., Kors, J.A.: Natural Language Processing in Radiology: A Systematic Review. Radiology **279**(2) (2016) 329–343
23. Ford, E., Carroll, J.A., Smith, H.E., Scott, D., Cassell, J.A.: Extracting Information from the Text of Electronic Medical Records to Improve Case Detection: A Systematic Review. Journal of the American Medical Informatics Association **23**(5) (2016) 1007–1015
24. Cai, T., Giannopoulos, A.A., Yu, S., Kelil, T., Ripley, B., Kumamaru, K.K., Rybicki, F.J., Mitsouras, D.: Natural Language Processing Technologies in Radiology Research and Clinical Applications. RadioGraphics **36**(1) (2016) 176–191
25. Yim, W.w., Yetisgen, M., Harris, W.P., Kwan, S.W.: Natural Language Processing in Oncology. JAMA Oncology **2**(6) (2016) 797
26. Meystre, S.M., Savova, G.K., Kipper-Schuler, K.C., Hurdle, J.F.: Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. Yearbook of Medical Informatics (2008) 128–44
27. Tan, W.K., Hassanpour, S., Heagerty, P.J., Rundell, S.D., Suri, P., Huhdanpaa, H.T., James, K., Carrell, D.S., Langlotz, C.P., Organ, N.L., Meier, E.N., Sherman, K.J., Kallmes, D.F., Luetmer, P.H., Griffith, B., Nerenz, D.R., Jarvik, J.G.:

Comparison of Natural Language Processing Rules-Based and Machine-Learning Systems to Identify Lumbar Spine Imaging Findings Related to Low Back Pain (2018)
28. Huhdanpaa, H.T., Tan, W.K., Rundell, S.D., Suri, P., Chokshi, F.H., Comstock, B.A., Heagerty, P.J., James, K.T., Avins, A.L., Nedeljkovic, S.S., Nerenz, D.R., Kallmes, D.F., Luetmer, P.H., Sherman, K.J., Organ, N.L., Griffith, B., Langlotz, C.P., Carrell, D., Hassanpour, S., Jarvik, J.G.: Using Natural Language Processing of Free-Text Radiology Reports to Identify Type 1 Modic Endplate Changes. Journal of Digital Imaging **31**(1) (2018) 84–90
29. Wang, Y., Mehrabi, S., Sohn, S., Atkinson, E., Amin, S., Liu, H.: Automatic Extraction of Major Osteoporotic Fractures from Radiology Reports using Natural Language Processing. In: Proceedings - 2018 IEEE International Conference on Healthcare Informatics Workshops, ICHI-W 2018, IEEE (2018) 64–65
30. Hassanpour, S., Bay, G., Langlotz, C.P.: Characterization of Change and Significance for Clinical Findings in Radiology Reports Through Natural Language Processing. Journal of Digital Imaging **30**(3) (2017) 314–322
31. Xu, Y., Tsujii, J., Chang, E.I.C.: Named Entity Recognition of Follow-Up and Time Information in 20 000 Radiology Reports. Journal of the American Medical Informatics Association **19**(5) (2012) 792–799
32. Siklósi, B., Novák, A.: A Magyar Beteg. In Tanács, A., Varga, V., Vincze, V., eds.: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem, Informatikai Tanszékcsoport (2014) 188–198
33. Orosz, Gy., Novák, A., Prószéky, G.: Hybrid Text Segmentation for Hungarian Clinical Records. In Castro, F., Gelbukh, A., González, M., eds.: Advances in Artificial Intelligence and Its Applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part I, Berlin, Heidelberg, Springer Berlin Heidelberg, Springer Berlin Heidelberg (2013) 306–317
34. Siklósi, B., Novák, A.: Rec. et exp. aut. Abbr. mnyelv. KLIN. szöv-ben – Rövidítések Automatikus Felismerése és Feloldása Magyar Nyelvű Klinikai Szövegekben. In Tanács, A., Varga, V., Vincze, V., eds.: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem, Informatikai Tanszékcsoport (2014) 167–176
35. Siklósi, B., Orosz, Gy., Novák, A., Prószéky, G.: Automatic Structuring and Correction Suggestion System for Hungarian Clinical Records. In De Pauw, G., de Schryver, G.M., Forcada, M.L., M. Tyers, F., Waiganjo Wagacha, P., eds.: 8th SaLTMiL Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages, Istanbul (2012) 29–34
36. Siklósi, B., Novák, A.: Digitális Konzílium – Egy Szemészeti Klinikai Keresőrendszer. In Tanács, A., Varga, V., Vincze, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016), Szeged, Szegedi Tudományegyetem, Szegedi Tudományegyetem (2016) 230–240
37. Siklósi, B., Novák, A.: Nem Felügyelt Módszerek Alkalmazása Releváns Kifejezések Azonosítására és Csoportosítására Klinikai Dokumentumokban. In Tanács, A., Varga, V., Vincze, V., eds.: XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015), Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2015) 237–248

# Szkizofrénia azonosítása spontán beszéd temporális paraméterei alapján – egy pilot kutatás eredményei

Bagi Anita[1,5], Gosztolya Gábor[2], Szalóki Szilvia[3,5],
Szendi István[3,5] és Hoffmann Ildikó[1,4,5]

[1] SZTE BTK Magyar Nyelvészeti Tanszék, 6722 Szeged, Egyetem u. 2.
bagianita88@gmail.com
[2] MTA-SZTE Mesterséges Intelligencia Kutatócsoport, 6701 Szeged, Pf. 652.
ggabor@inf.u-szeged.hu
[3] SZTE ÁOK Pszichiátriai Klinika, 6725 Szeged, Kálvária sgt. 57.
szilvi.szaloki@gmail.com, szendi.istvan@med.u-szeged.hu
[4] MTA Nyelvtudományi Intézet, 1394 Budapest Pf. 360.
i.hoffmann@hung.u-szeged.hu
[5] Mentális Betegségek Megelőzése Interdiszciplináris Kutatócsoport

**Kivonat:** A szkizofrénia olyan neurodegeneratív spektrum zavar, melyet különböző alulműködések együttese alkot. A szkizofréniát, számos tünete mellett, jellemzi például a csökkent információfeldolgozási sebesség és a csökkent verbális fluencia teljesítmény is. Jelen tanulmányunkban a beszédtempó folyamatosságát vizsgáljuk szkizofréniával élők és illesztett egészséges kontrollszemélyek irányított spontán beszéd-felvételeiben. Célunk, hogy rámutassunk a különböző beszédbeli temporális paraméterek (úm. artikulációs tempó, beszédtempó és különböző szünettartási mutatók) segítségével arra, hogy a két csoport között specifikus eltéréseket tudunk meghatározni egy korábban korai demencia felismerésre (enyhe kognitív zavarra és Alzheimer-kórra) kifejlesztett és tesztelt eljárás használatával. Munkánk során ezen temporális mutatók alkalmazhatóságát teszteltük gépi tanulással új betegpopuláción. Eredményeink azt mutatják, hogy a két csoport beszélői 70–80 % közti osztályozási pontosságértékekkel meghatározhatók és az F-értékek 81% és 87% közé esnek. Részletes vizsgálatunk feltárta, hogy a két csoport meghatározására a szünettartási temporális paraméterek közül a leghatékonyabbak azok az elemzési utak, melyek estében mind a néma, mind pedig a kitöltött szünetekkel számolunk.

**Kulcsszavak:** spontánbeszéd, temporális paraméterek, szkizofrénia, kitöltött szünetek

## 1 Bevezetés

Bár számos, több szempontból közelítő, széles körű vizsgálat ismert a szkizofrénia hátterének feltérképezéséhez, ezidáig nem tudtak meghatározni egyetlen specifikus genetikai, neurobiológiai vagy környezeti tényezőt sem, mely a betegség kiala-

kulásának hátterében állhat. Crow elmélete szerint [1] a szkizofrénia (fenomenológiai szempontból) olyan univerzális betegségnek tekinthető, mely a Föld valamennyi populációjában megtalálható. Elméletében feltételezi, hogy a szkizofrénia evolúciós szintű fennmaradásának hátterében a lateralizációt eredményező genetikai változások és a kialakuló pszichológiai struktúrák állhatnak. A szkizofrénia diagnózisának felállításához a következő tüneti kritériumok teljesülése szükséges: (1) téveszmék, (2) hallucinációk, (3) inkoherens beszéd, (4) szembeszökően szétesett vagy katatón viselkedés, (5) negatív tünetek, azaz hangulati üresség, alogia vagy akaratnélküliség. A tüneti kritériumok mellett fontos az időtartam aspektusa is, mely szerint legalább 6 hónapig, de az 5 fő tünet egyikének legalább egy hónapig fenn kell állnia ahhoz, hogy a diagnózis felállítható legyen [2].

A szkizofréniát számos kognitív deficit jellemezheti, ezen deficitek közé tartozik a csökkent információ-feldolgozási sebesség és a munkamemória károsodása [3]. Emlékezeti funkciók alulműködését találták szkizofréniával élőknél neuropszichológiai tesztek eredményeiben is, melyek érintették a munkamemóriát, a verbális fluencia teljesítményt és az epizodikus emlékezetet is [4,5,6]. Más kutatások specifikus károsodást mutattak ki szkizofréniában a munkamemória és a tartós figyelmi funkciók tekintetében is [7,8].

A szkizofréniával élők számos, különböző nyelvi szinteket érintő deficittel rendelkezhetnek [9]. Pawełczyk és mtsai [10] azt találták, hogy a szkizofréniával élők egészséges kontrollszemélyek eredményeihez képest szignifikánsan alacsonyabb pontszámot értek el az olyan szubtesztek esetében, mint az implicit információ-feldolgozás, a humorfeldolgozás, a metaforák felfejtése, a nem odaillő vagy helytelen észrevételek és megjegyzések felismerése, az érzelmek megkülönböztetésére irányuló feladatokban, melyek a nyelvben használt intonációk felismerésével operáltak; emellett a különböző diskurzusok feldolgozása és értelmezése esetében is jelentős különbségeket találtak. Eltéréseket találtak továbbá a prozódia területén is, míg más kutatások sze-rint a szkizofrénia negatív tünetei megjelenhetnek a hanghordozás és a hangsúlyozás hiányában is [11,12]. A beszédprodukció felől a spontán beszédet vizsgáló kutatások a kommunikált gondolat összetettségét elemezték, és azt találták, hogy szkizofréniával élőknél ezek a megnyilatkozások kevésbé összetettek, mint az egészséges kontrollszemélyek beszédproduktumai. Ugyanakkor arra is felhívták a figyelmet, hogy azok a páciensek, akik jobb teljesítményt nyújtottak, nagyobb arányban voltak érintettek a depresszió és a szorongás különböző tünetei által [13].

Számos fentebb említett tünetet számítógépes eszközökkel is elemeztek. Rosenstein és mtsai [14] a verbális munkamemóriát vizsgálták a verbális emlékezeti folyamatok mérésére koncentrálva számítógépes nyelvészeti megközelítésekkel és eszközökkel. Corcoran és mtsai [15] azt találták, hogy az automatizált szemantikai és szintaktikai elemzés jól használható kiindulási alapja lehetne egy diagnosztikai eszköznek. További prozódiai eltéréseket és lehetséges karakterisztikákat [16,17], illetve a beszéd folytonosságát, a megakadásjelenségek és szünetek minőségét és arányát is vizsgálták már hasonló eszköztárakkal [18]. Más kutatások azt találták, hogy a formális gondolkodási zavarral (mely szembetűnő tünete lehet a szkizofréniának) rendelkező páciensek markánsan kevesebb kitöltött szünetet produkáltak, mint az egészséges kontrollszemélyek [19].

Jelen tanulmányunkban a spontán beszédben észlelhető emlékezeti folyamatok deficitére koncentrálunk. Munkánk során irányított spontán beszédet vizsgálunk, mely egyben egy emlékezeti feladat is. A feladat pontos instrukciója a következő: *„Kérem, mesélje el a tegnapi napját!”*. Feltételezzük, hogy a spontán beszéd temporális mutatói különbözni fognak az egészséges és a szkizofréniával élő beszélők felvételeiben. A leginkább eltérő különbségeket a hezitációk számában és típusában várjuk. Elemzésünket automatizált elemzési módszerrel végezzük: automatizált beszédfelismerő-szoftver segítségével (ASR) kinyerjük a temporális mutatókat a felvételekből, majd mérlegre tesszük e kinyert paraméterek felhasználhatóságát statisztikai gépi tanulás alkalmazásával a két beszélői csoport elkülönítésére.

## 2 A beszéd temporális paraméterei

A spontán beszéd vizsgálatához a szkizofréniával élők és az egészséges kontrollszemélyek válaszaiból specifikus temporális paramétereket számítottunk ki. Kutatásunkat korábbi munkáinkra építettük [20, 21, 22], melyekben olyan, a hezitációt központba helyező temporális paramétereket mutattunk be, melyek az enyhe kognitív zavar (EKZ) korai detektálására használhatók. Az EKZ-t gyakran tekintik az Alzheimer-kór prodromális állapotának, mely egyben egy olyan mentális zavar is, amit igen nehéz diagnosztizálni. Az EKZ (spontán) beszédre gyakorolt hatása ismert [23]; e hatások közül jelen tanulmányunkban a verbális fluenciára koncentrálunk, mely szintén érintett lehet szkizofréniával diagnosztizált személyek esetében is [4, 5, 6]. Az EKZ-ban szenvedő betegek verbális fluenciájában gyakran mérhető rosszabbodás, mely megkülönböztető akusztikus változásokat eredményez; a két legfontosabb változást említve ezek tetten érhetők a több, ill. hosszabb hezitációkban és az alacsonyabb beszédtempóban is [24, 25]. Ezen eredmények felhasználására kifejlesztettünk egy olyan temporális paramétereket tartalmazó eszköztárat, mely elsősorban az alanyok beszédében mérhető hezitációk mennyiségére fókuszál.

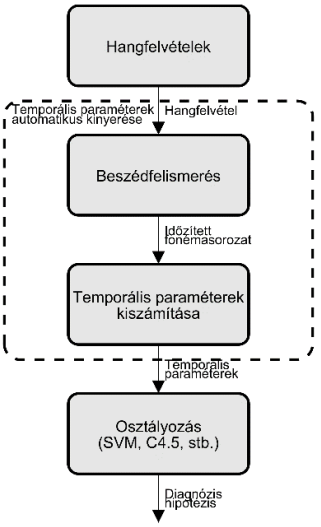
A temporális paramétereket tartalmazó jellemzőkészlet az 1. táblázatban látható. Meg kell jegyeznünk, hogy a paraméterek (4) és (8) között mind az alany spontán beszédében mért hezitáció mértékét írják le, különböző szempontokból fókuszálva a szünetek számára vagy hosszára. Ezen a ponton szükséges definiálnunk a hezitációra vonatkozó meghatározásainkat. A szünet legegyszerűbb formája a *néma* szünet, mely egyenlő a beszéd hiányával. Ugyanakkor a hezitáció megjelenhet *kitöltött* szünetként is, melynek vokalizációi lehetnek például az ’öö’, az ’hmm’ vagy az ’ühm’. Mindkét szünettípus hezitációt jelez a spontán beszédprodukcióban. Ahhoz, hogy mindkét szünettípust elemezni tudjuk, a (4)–(8)-as temporális jellemzőket kiszámítottuk csak néma szünetekkel számolva; csak kitöltött szünetekkel számolva és végül minden szünettartással számolva függetlenül a szünet típusától. Ezen elemzési módszer összesen 18 temporális paraméterhez vezetett.

| | |
|---|---|
| (1) | Artikulációs tempó (hezitációk nélkül számított másodpercenkénti beszédhang-szám) |
| (2) | Beszédtempó (másodpercenkénti beszédhang-szám osztva a megnyilatkozás teljes hosszával) |
| (3) | Megnyilatkozás teljes hossza (ezredmásodpercben mérve) |
| (4) | Szünetek száma (a szünetek előfordulásának száma) |
| (5) | Szünetek hossza (a szünetek összhossza) |
| (6) | Szünetek hosszának aránya (szünetek összhossza osztva a megnyilatkozás hosszával) |
| (7) | Szünetgyakoriság (szünetek előfordulásának száma osztva a megnyilatkozás hosszával) |
| (8) | Átlagos szünethossz (szünetek összhossza osztva a szünetek számával) |

1. Táblázat: A nyolc vizsgált temporális jellemző, Hoffmann és mtsai [21] és Tóth és mtsai [22] nyomán

### 2.1 A beszédfelismerés-alapú temporális paraméterek kiszámítása

A fentebb bemutatott akusztikus-temporális paraméterek manuális feldolgozása meglehetősen hosszadalmas, drága és munkaigényes. Bár korai munkáink során még ezt a kinyerési utat alkalmaztuk [25], jelen tanulmányunkban már automatikusan nyertük ki azokat. Ezt az automatizált utat választva kézenfekvő megoldásnak tűnhet a jelfeldolgozásra támaszkodni [26]. Azonban ez a jelfeldolgozási technikákat alkalmazó megoldás, bár viszonylag egyszerűen és nagy hatékonysággal képes megkülönböztetni a *csend*et az emberi beszéd más hangzó részeitől; más, itt megkövetelt különbségtételeket nem képes megtenni. Így például, pusztán jelfeldolgozási eszközökre támaszkodva képtelenek lennénk megkülönböztetni a kitöltött szüneteket a normál beszédtől, illetve nem tudnánk kiszámítani az artikulációs tempót és a beszédtempót sem.




1. ábra: Automatizált folyamat a temporális paraméterek kiszámítására és elemzésére Tóth és mtsai [20] nyomán

A fenti szempontokat figyelembe véve, az automatikus beszédfelismerési technikák (Automatic Speech Recognition, ASR) mellett döntöttünk, melynek folyamatábrája az 1. ábrán látható. Sajnos egy készen kapott ASR eszköz várhatóan alkalmatlannak bizonyulna erre a feladatra – annál is inkább, mert a szabványos beszédfelismerőket a szószintű átírási hibák minimalizálására tanítják, miközben mi most éppen olyan nem-verbális akusztikus jellemzőket igyekszünk kinyerni, mint a beszédtempó vagy épp a néma és kitöltött szünetek időtartama. Szerencsére azonban az 1. táblázatban bemutatott beszédparaméterek nem követelik meg a hangok azonosítását, csak azok megszámlálását. Továbbá, míg a kitöltött szünetek nem jelennek meg expliciten egy standard beszédfelismerő rendszer kimenetében, a mi jellemzőkészletünk ezek detektálását kimondottan megköveteli. Mindezen megfontolások okán egy standard beszédfelismerő rendszert úgy módosítani, hogy az képes legyen kezelni az ilyen típusú „hibákat", ha nem is lehetetlen, de mindenképpen nagyon munkaigényes lenne.

Mindezen okokból egy olyan beszédfelismerő használata mellett döntöttünk, amely a bemenetként megadott hangfelvételhez kimenetként nem annak szószintű, hanem fonémaszintű átiratát adja meg. (A kitöltött szüneteket, az általánosság megkötése nélkül, kezelhetjük egy speciális „fonémaként".) Természetesen a szószint teljes elhagyása (a szószintű nyelvi modellel és a teljes kiejtési szótárral együtt) várhatóan növelni fogja a fonémaszintű hibák számát is. Azonban, amint arra fentebb rámutattunk, nem minden típusú hangfelismerési hiba „rontja le" a temporális paraméterek kinyerését; jelen esetben csak a fonémák száma és a két szünet típusa (úm. néma és kitöltött) fontos.

## 3 Adatbázis

Tanulmányunkhoz folyó kutatásunk jelenleg is bővülő adatbázisából véletlenszerűen kiválasztottunk tíz szkizofréniával élő személyt, majd hozzájuk korban és nemben illesztettünk nyolc egészséges kontrollszemélyt. A két csoport nemenkénti megoszlása 50-50% volt, tehát a szkizofréniával élők csoportjában (*SZ*) 5 férfi és 5 nő, míg az egészséges kontrollcsoportban (*K*) 4 férfi és 4 nő volt. Jelenlegi kontrollcsoportunk száma meglehetősen alacsony, de folyamatosan dolgozunk az adatbázis bővítésén.

A résztvevőktől származó megnyilatkozásokat 2016 februárja és 2017 márciusa között rögzítettük a Szegedi Tudományegyetem Általános Orvostudományi Karának Pszichiátriai Klinikáján. A kutatást jóváhagyta a Szegedi Tudományegyetem Etikai Bizottsága; a kutatás teljes folyamatát a Helsinki Nyilatkozat szellemében végeztük. A kutatásban résztvevő minden beszélő magyar anyanyelvű volt. Az elemzés során a beszéd temporális jellemzőit mértük. A résztvevőktől irányított spontán beszédet rögzítettünk: arra kértük őket, hogy meséljenek a tegnapi napjukról. A pontos instrukció elhangzása után (*„Kérem, mesélje el a tegnapi napját!"*) a résztvevőknek hozzávetőleg öt perc állt rendelkezésükre, hogy teljesítsék a feladatot – természetesen, ha egy-két perccel rövidebb vagy hosszabb időt vett igénybe a feladat megoldása, akkor sem szakította félbe őket a vizsgálatvezető. A hangfelvételek elkészítéséhez Roland R-05 típusú diktafont használtunk.

A csoportonkénti kormegoszlás az SZ-csoport esetében 39,9 év volt, míg a K-csoport esetében 40,2. Az iskolázottságot években számolva ($t$=-1,82, df=18, $p$=0,09)

és az életkor (t=0,06, df=18, $p$=0,96) tekintetében nem volt szignifikáns különbség a két csoport értékei között. A hangfelvételek mellett minden résztvevővel elvégeztük a Módosított Mini-Mentál Tesztet is (MMSE [27]), melynek eredményeiben a két csoport szignifikáns eltérést mutatott (t=2,55, df=10,55, $p$=0,028). A szkizofréniával élő személyek legtöbbször a *Felidéző emlékezés* altesztben veszítettek pontot: ez azonban nem feltétlenül jelez emlékezeti deficitet – a szórt figyelem eredménye is lehet az altesztben nyújtott csökkent teljesítmény.

# 4 Kísérleti elrendezés

## 4.1 A temporális paraméterek kinyerése

A beszédfelismerő rendszer akusztikus modelljének tanítására a Magyar Beszéltnyelvi Adatbázist (BEA, [28]) használtuk. A BEA adatbázisa tartalmaz spontán beszédet, így jelen kutatásunk szempontjából az egyik leghasznosabbnak tűnt, annál is inkább, mert kitöltött szünetekkel csak spontán beszédben találkozunk. A tanuláshoz közel 7 órányi spontán beszédet használtunk fel. Előzetesen megbizonyosodtunk arról, hogy az átiratokban fonémaszinten következetes módon volt jelölve a kitöltött szünet, a be- és kilégzés, a nevetés, a köhögés és a zihálás.

A beszédfelismerő rendszert arra tanítottuk, hogy felismerje a megnyilatkozásokban lévő beszédhangokat – a fonémakészlet természetesen tartalmazta ezeket a speciális nonverbális címkéket (kitöltött szünet, be- és kilégzés, nevetés stb.) is. Az akusztikai modellezéshez egy standard mély neurális hálót (Deep Neural Network, DNN) alkalmaztunk előrecsatolt (feed-forward) topológiával, melynek három rejtett rétege egyenként ezer ún. ReLU aktivációs függvényt használó neuront tartalmazott. Munkánk során saját implementációnkat használtuk, mellyel korábban kutatócsoportunk érte el a legalacsonyabb publikált szószintű hibaarányt a TIMIT adatbázison [29]. Az alkalmazott nyelvi modell egy egyszerű fonéma bigram volt, mely (még egyszer kihangsúlyozva) tartalmazta a fentebb felsorolt nonverbális hangjelenségeket is. A beszédfelismerő rendszer kimenete egy időzített fonetikus átirat volt; ezekből az átiratokból (melyek a kitöltött szünetet mint speciális fonémát is tartalmazták) az 1. táblázatban felsorolt temporális paraméterek már könnyen kinyerhetők és kiszámíthatók.

## 4.2 Kiértékelési mutatók

A közelmúlt számos orvosbiológiai tanulmánya, mely ASR-alkalmazásokat használt, egyszerű osztályozási pontosságra támaszkodott (vö. [26,30]). Esetünkben azonban a vizsgált csoportok mindkét típusának gyakorisága meglehetősen kiegyensúlyozatlan: a szkizofrénia a populáció 1-1,5%-át érinti. Az ilyen kiegyensúlyozatlan osztályeloszlás miatt a pontosság egyáltalán nem működne megbízható mutatóként. Emiatt jelen kutatásban standard információ-visszakeresési kiértékelési metrikákat használtunk: pontosságot (precision), fedést (recall), és e kettő harmonikus középértékét, az F-

értéket (vagy az F1-értéket; F-measure; F1 score). Ezen felül kiszámítottuk a ROC-görbe alatti terület nagyságát (azaz az AUC mutatót) is az SZ osztályra.

### 4.3 Osztályozási folyamat

Osztályozási folyamatunk alapvetően az orvosbiológiai szokásokat követi, és hasonlít azokhoz a korábbi tanulmányainkhoz, amelyek az EKZ kimutatására koncentráltak (vö. [20,22]). A fentebb bemutatott temporális paraméterekre mint jellemzőkre szupport-vektor gépet (Support Vector Machine, SVM, [31]) tanítottunk, a LibSVM [32] implementációt használva. A nu-SVM metódust használtuk lineáris kernelfüggvénnyel; a $C$ értékét a $10^{\{-5, \ldots, 1\}}$ tartományban teszteltük.

Gépi tanulási szempontból rendkívül kicsi adathalmazon dolgoztunk, hiszen a kontrollcsoportba tartozó résztvevők száma korlátozott volt. Ebből adódóan nem láttuk értelmét külön tanító és teszthalmazok definiálásának, hanem beszélők szerinti keresztvalidációt (cross-validation, CV) alkalmaztunk: az osztályozó modellünket mindig 17 fős korpusz adatain tanítottuk, és mindig a fennmaradó egyre értékeltük ki azokat. Az SVM $C$ meta-paraméterét beágyazott keresztvalidációban határoztuk meg [33]: a 17 beszélő esetében végzett tanításnál a tényleges CV lépésben újabb (beszélő szerinti) keresztvalidációt végeztünk. Azt a $C$ értéket választottuk, amely a legmagasabb AUC pontszámot eredményezte ebben a saját „belső” CV tesztben. Ezt követően az SVM modellt ennek a 17 beszélőnek az adataira tanítottuk, és ezt a modellt értékeltük ki a 18. beszélő adatain. Ez az eljárás garantálja, hogy semmilyen szinten ne használjuk az aktuális tesztadatot az aktuális modell tanítására – ez ugyanis pontszámainkban torzulást eredményezett volna például standard keresztvalidáció használata esetén.

### 4.4 Az adatok előzetes feldolgozása

Kísérleteinkben egy-egy hangfelvételt használhatunk 18 beszélőtől. Adathalmazunk méretének növelése érdekében úgy döntöttünk, hogy kísérleteinkben rövidebb megnyilatkozás-egységeket használunk. Hipotézisünk az volt, hogy temporális beszédparamétereink akkor is értelmezhetőek maradnak, ha viszonylag rövid megnyilatkozásokból számoljuk őket. Ezt szem előtt tartva, a megnyilatkozásokat 30 másodperces szegmensekre osztottuk fel 10 másodperces átfedéseket hagyva (függetlenül a tényleges fonetikai határoktól), és a továbbiakban ezeket a szegmenseket önállóan kezeltük. Ezen lépések után 96 viszonylag rövid, de egyenlő méretű szegmensből álló adathalmazt kaptunk, amely jelentősen növelte SVM tanulókészletünk méretét. Természetesen az osztályozást ezek után is a már bemutatott beszélők szerinti beágyazott keresztvalidációs sémával végeztük; azaz egy-egy fold mindig egy beszélő összes szegmenséből állt.

Bár az eddig használt osztályozási metrikák logikus választásnak tűnnek a 30 másodperces szegmensek esetén is, a pontszámok jobban értelmezhetővé válnak, ha lefordítjuk őket a résztvevőkre. Egyszerű megoldás lehet erre az egyes beszélők kategóriájának (SZ vagy K) meghatározása az egyes szegmensekre adott hipotéziseinkből egyszerű többségi szavazással. Ezt azonban meglehetősen nehéz lenne értelmezni.

Ezért úgy döntöttünk, hogy előrejelzéseinket egy másik megközelítéssel vonjuk össze a beszélő-szintű értékek meghatározásakor: egy beszélőre normalizált tévesztési mátrixot számítottunk ki az egyes beszédszegmensek új súlyozásával: *1/k*, ahol *k* az adott beszélő szegmenseinek száma. Például egy egészséges beszélő 10 beszédszegmenssel (melyek közül 7 lett helyesen azonosítva) 0,7 valódi negatív és 0,3 hamis pozitív esetnek számít. A beszélők szerinti beágyazott keresztvalidálás befejezése után az osztályozási pontosság valamint az információ-visszakeresési metrikák könnyen kiszámíthatóak a beszélők szerint normalizált tévesztési mátrixból. Sajnos az AUC értékeket ebben a megközelítésben nem tudtuk meghatározni, mivel ahhoz az egyes példákra adott poszteriorbecslések is szükségesek lennének, míg most csak egy (normalizált) tévesztési mátrixszal rendelkezünk.

## 5 Eredmények

A 2. táblázat tartalmazza a kiszámított metrikáinkat a **szegmensek szintjén**. Ha mind a 18 temporális beszédparamétert bevesszük a jellemzőkészletbe, a 70,8%-os osztályozási pontosság viszonylag jó teljesítményt mutat. Az $F_1$ 81,3%-os értéke véleményünk szerint meglehetősen magasnak tűnik, különösen, ha figyelembe vesszük a tanítópéldák alacsony számát. A pontossági és fedési mutatókat vizsgálva láthatjuk, hogy a teljesítmény meglehetősen kiegyensúlyozatlan: a szkizofréniával élő betegek által produkált szegmensek mindössze 74%-át találta meg az eljárás, ám ezt megközelítőleg 90%-os pontossággal tette. Ez a probléma a kimeneti poszteriorbecslések küszöbértékelésével kezelhető [34], ugyanakkor úgy véljük, hogy e probléma tárgyalása már szétfeszítené jelen tanulmányunk kereteit.

| | **Osztályozási pontosság (%)** | | | | |
|---|---|---|---|---|---|
| **Jellemzőkészlet** | **Pont.** | **Prec.** | **Fedés** | **$F_1$** | **AUC** |
| Teljes | 70,8 | 89,7 | 74,4 | 81,3 | 0,514 |
| Néma szünetek | 76,0 | 94,1 | 77,1 | 84,8 | 0,599 |
| Kitöltött szünetek | 75,0 | 97,1 | 75,0 | 84,6 | 0,435 |
| Minden hezitáció | 79,2 | 92,6 | 80,8 | 86,3 | 0,726 |
| Tempó + néma szünetek | 80,2 | 97,1 | 79,5 | 87,4 | 0,641 |
| Tempó + kitöltött szünetek | 70,8 | 91,2 | 73,8 | 81,6 | 0,602 |
| Tempó + minden hezitáció | 78,1 | 91,2 | 80,5 | 85,5 | 0,694 |

2. táblázat. A szegmensszintű pontossági értékek a különböző jellemző-részhalmazok használata esetén

A temporális paraméterek egy részhalmazát felhasználó elemzések eredményeit vizsgálva megfigyelhetjük, hogy az osztályozási pontszámok szinte minden esetben javultak. A néma vagy kitöltött szünetekkel kapcsolatos időbeli paraméterek összehasonlításával megállapíthatjuk, hogy a szkizofrénia azonosítására a kitöltött szünetek kevésbé hasznosak, mint a néma szünetek értékei: a 71-75%-os osztályozási pontossági értékek elmaradnak a 76-80%-os értékek mögött, melyek a néma szünetekre koncentrálnak – az F-érték és az AUC pontszám is magasabb az utóbbi két esetben. A

kapott értékek tendenciáit vizsgálva, véleményünk szerint, a vizsgált temporális paraméterek leghasznosabb részhalmazai azok voltak, amelyek a hezitálások alapján számított indikátorokból álltak – függetlenül attól, hogy ezek néma vagy kitöltött szünettel operáltak-e. Bár a néma szünethez tartozó paraméterek az artikulációs tempóval és a beszédtempóval kombinálva valamivel nagyobb pontossághoz és magasabb $F_1$ értékhez vezettek, abban a két esetben, amikor mindkét szünettípust figyelembe vettük, konzisztensen magasabb pontosságértékeket kaptunk, valamint a legmagasabb AUC értékek is ekkor adódtak.

Az osztályozási eredmények értelmezésével a megnyilatkozások számának normalizálásával, az egyes szegmensek külön-külön való számbavétele helyett a mutató értékének enyhe csökkenését láthatjuk (3. táblázat).

| | **Osztályozási pontosság (%)** | | | |
|---|---|---|---|---|
| **Jellemzőkészlet** | **Pont.** | **Prec.** | **Fedés** | **$F_1$** |
| Teljes | 60,8 | 60,0 | 88,0 | 71,4 |
| Néma szünetek | 68,3 | 65,0 | 93,0 | 76,6 |
| Kitöltött szünetek | 65,7 | 62,1 | 98,3 | 76,1 |
| Minden hezitáció | 77,2 | 74,4 | 90,0 | 81,5 |
| Tempó + néma szünetek | 73,4 | 68,7 | 95,6 | 80,0 |
| Tempó + kitöltött szünetek | 61,0 | 59,9 | 89,7 | 71,9 |
| Tempó + minden hezitáció | 76,5 | 74,1 | 88,6 | 80,7 |

3. táblázat. A beszélőszintű pontossági értékek a különböző jellemző-részhalmazok használata esetén

Ami még érdekesebbnek tűnik, hogy a pontossági és visszakeresési eredmények tendenciáját nézve, az eredmények éppen ellenkező irányú tendenciát mutatnak, mint a szegmensek szintjét vizsgálva – itt már alacsonyabb pontosságot (precision), de viszonylag magas fedés értékeket láthatunk. Ez valószínűleg azért van, mert a szkizofréniával élők sokkal részletesebben írták le a tegnapi napjukat, mint az egészséges kontrollok; ebből következően az SZ csoportba tartozók felvételei szignifikánsan hosszabbak voltak, mint az egészséges kontrolloké. Ez azt eredményezte, hogy a megnyilatkozások száma is kiegyensúlyozatlanul alakult: számszerűsítve 68 (SZ) és 28 (K). A felhasznált temporális beszédparaméterek különböző alcsoportjainak vizsgálatát tekintve, minden valószínűség szerint a két beszélői csoportot a leghatékonyabban úgy azonosíthatnánk, ha figyelembe vennénk mindkét szünettípust. Ez következik abból is, hogy a 77,2% és 76,5%-os osztályozási pontossági pontszámok szignifikánsan magasabbak, mint csak a néma szünetekkel (68,3% és 73,4%), vagy csak a kitöltött szünetek használatával (65,7% és 61,0%) kapott értékek. Az így kapott $F_1$-értékek (81,5% és 80,7%) is messze a legmagasabbnak mértek (76,6-80,0% és 76,1%-71,9%, a néma és a kitöltött szüneteket külön-külön vizsgálva).

A különböző temporális paraméterek hasznosíthatóságát illetően tény, hogy a szkizofréniával élő résztvevők felvételei lényegesen hosszabbak voltak, mint az egészséges kontrollok hangfelvételei. A jelenség hátterében felvetődhet lehetséges magyarázatként az olyan pozitív tünetek jelenléte, mint a circumstantialitás, mely a kommunikálni kívánt tartalom túlzott részletességgel való kifejtését jelenti, de hasonlóan e pozitív tünethez, a gondolatrohanások és a szisztematikus önhivatkozások is vezet-

hetnek a hosszabb megnyilatkozásokhoz. A szkizofréniával élők néma szüneteinek magasabb száma további tünetek beszédre gyakorolt hatásával is magyarázható, melyek egyaránt érintik a végrehajtó és emlékezeti funkciókat is, s gyakran eredményeznek zavart gondolkodást, mely a beszédben válik tetten érhetővé. A szkizofréniával élőknek gyakran okoz problémát a gondolatok szervezése, rendszerezése, ami tükröződhet a spontán beszéd temporális paramétereiben is (például a néma vagy kitöltött szünetek számában).

Összegezve az eddigieket, vizsgálatunkban szignifikáns különbséget találtunk a két beszélői csoport (SZ és K) spontán beszédének temporális paramétereiben. A vizsgált temporális paraméterek közül az artikulációs arányra, a beszédtempóra és a hezitációkra koncentrálva, meglehetősen pontosan tudtunk különbséget tenni a két beszélői csoport között. A jövőben további résztvevőket kívánunk bevonni jelenleg is folyó kutatásunkba, hogy megerősíthessük és árnyalhassuk eddigi eredményeinket. Tervezzük továbbá a spontán beszéd fentebb bemutatott elemzését a teljes pszichózis spektrumon is, beteg-kontrollcsoportként együtt vizsgálva a szkizofréniát a bipoláris zavarral és a szkizoaffektív zavarral.

## 6 Összegzés

Jelen tanulmányunkban feltételeztük, hogy különbséget találunk az egészséges kontroll személyek és a szkizofréniával élők spontán beszédének temporális paramétereiben. Automatikus beszédelemzéssel és gépi tanulási technikákkal hatékonyan meg tudtuk különböztetni a két beszélői csoport tagjait. A hezitációs jelenségeket a legfontosabb megkülönböztető jegyeknek feltételeztük, mely feltételezésünket a vizsgálat eredményei igazoltak is: a 77%-os osztályozási pontszámok szignifikánsan magasabbak voltak, mintha csak a néma szüneteket (68-73%) vagy csak a kitöltött szüneteket vizsgáltuk volna (61-66%).

Munkánk pilotkutatás volt: arra kerestük a választ, hogy vajon az automatikus beszédelemzési folyamat használható lenne-e a szkizofréniával élők spontán beszédének temporális elemzésében. Törekedtünk továbbá arra is, hogy kutatásunk hozzájáruljon a neurodegeneratív rendellenességekről alkotott ismereteink bővítéséhez, s ezzel együtt pontosítsa a kapcsolódó szupraszegmentális jegyek leírását is. Természetesen az erősebb kijelentések megtételéhez szükség van a kutatásainkban résztvevők számának növelésére. Jelenleg is folyamatosan vonunk be résztvevőket a pszichózis-spektrum egyéb betegcsoportjaiból is.

## Köszönetnyilvánítás

A kutatást az EFOP-3.6.1-16-2016-00008 a.sz., EU társfinanszírozású projekt támogatta.

## Bibliográfia


1. Crow, T.J: Is schizophrenia the price that Homo sapiens pays for language? Schizophrenia Research **28** (2–3) (1997) 127–141
2. American Psychiatric Association: Diagnostic and statistic manual of mental disorders (DSM-5). American Psychiatric Publishing (2013)
3. Kochunov, P., Coyle, T.R., Rowland, L.M., Jahanshad, N., Thompson, P.M., Kelly, S., Du, X., Sampath, H., Bruce, H., Chiappelli, J., Ryan, M., Fisseha, F., Savransky, A., Adhikari, B., Chen, S., Paciga, S.A., Whelan, C.D., Xie, Z., Hyde, C.L., Chen, X., Schubert, C.R., O'Donnell, P., Hong, E.: Association of White Matter With Core Cognitive Deficits in Patients With Schizophrenia. JAMA Psychiatry **74** (9) (2017) 958–966
4. Heinrichs, R.W., Zakzanis, K.K.: Neurocognitive deficit in schizophrenia: a quantitative review of the evidence. Neuropsychology **12** (3) (1998) 426-445
5. McCleery, A., Ventura, J., Kern, R.S., Subotnik, K.L., Gretchen-Doorly, D., Green, F.M., Hellemann, G.S., Nuechterlein, K.H.: Cognitive functioning in first-episode schizophrenia: MATRICS consensus cognitive battery (MCCB) profile of impairment. Schizophrenia Research **157** (1–3) (2014) 33–39
6. Zhang, T., Li, H., Stone, W.S., Woodberry, K.A., Seidman, L.J., Tang, T., Guo, Q., Zhuo, K., Qian, Z., Cui, H., Zhu, Y., Jiang, L., Chow, A., Tang, Y., Li, C., Jiang, K., Yi, Z., Xiao, Z., Wang, J.: Neuropsychological impairment in prodromal, first-episode, and chronic psychosis: assessing RBANS performance. PLoS One **10** (5) (2015) 33–39
7. Chan, R., Chen, E., Cheung, E., Cheung, H.: Executive dysfunction in schizophrenia: relationships to clinical manifestation. European Archives of Psychiatry and Clinical Neuroscience **254** (4) (2004) 256–262
8. Huang, J., Tan, S.P., Walsh, S.C., Spriggens, K., Neumann, D.L., Shum, D.H., Chan, R.C.: Working memory dysfunctions predict social problem solving skills in schizophrenia. Psychiatry Research **220** (1–2) (2014) 96–101
9. Nagels, A., Kircher, T.: Symptoms and Neurobiological Models of Language in Schizophrenia. In: Hickok, G., Small, S. (eds.) Neurobiology of Language. Academic Press (2016) 887–897
10. Pawełczyk, A.M., Kotlicka-Antczak, M., Lojek, E., Ruszpel, A., Pawelczyk, T.: Schizophrenia patients have higher-order language and extralinguistic impairments. Schizophrenia Research **192** (2017) 274–280
11. Covington, M.A., Congzhou, H., Brown, C., Naci, L., McClain, J.T., Fjordbak, B.S, Semple, J., Brown, J.: Schizophrenia and the structure of language: The linguist's view. Schizophrenia Research **77** (1) (2005) 85–98
12. Rapcan, V., D'Arcy, S., Yeap, S., Afzal, N., Thakore, J.H., Reilly, R.B.: Acoustic and temporal analysis of speech: A potential biomarker for schizophrenia. Medical Engineering & Physics **32** (9) (2010) 1074–1079
13. Moe, A.M., Breitborde, N.J.K., Shakeel, M.K., Gallagher, C.J., Docherty, N.M.: Idea density in the life-stories of people with schizophrenia: Associations with narrative qualities and psychiatric symptoms. Schizophrenia Research **172** (1–3) (2015) 201–205
14. Rosenstein, M., Diaz-Asper, C., Foltz, P.W., Elvevag, B.: A computational language approach to modeling prose recall in schizophrenia. Cortex **55** (2014) 148–166
15. Corcoran, C.M., Carrillo, F., Fernández-Slezak, D., Bedi, G., Klim, C., Javitt, D.C., Bearden, C.E., Cecchi, G.A.: Prediction of psychosis across protocols and risk cohorts using automated language analysis. World Psychiatry **17** (1) (2018) 67–75
16. Bedwell, J.S., Cohen, A.S., Trachik, B.J., Deptula, A.E., Mitchell, J.C.: Speech prosody abnormalities and specific dimensional schizotypy features: Are relationships limited to males. The Journal of Nervous and Mental Disease **202** (10) (2014) 745–751

17. Martínez-Sánchez, F., Muela-Martinez, J.A., Cortés-Soto, P., Meilán, J.J.G., Ferrándiz, J.A.V., Caparrós, A.E., Valverde, I.M.P.: Can the acoustic analysis of expressive prosody discriminate schizophrenia? The Spanish Journal of Psychology **18** (86) (2015) 1–9
18. Alpert, M., Kotsaftis, A., Pouget, E.R.: At Issue: Speech fluency and schizophrenic negative signs. Schizophrenia Bulletin **23** (2) (1997) 171–177
19. Matsumoto, K., Kircher, T.T.J., Paul R.A.: Stokes and Michael J. Brammer and Peter F. Liddle and Philip K. McGuire Frequency and neural correlates of pauses in patients with formal thought disorder. Frontiers in Psychiatry **4** (2013) 67–75
20. Tóth, L., Gosztolya, G., Vincze, V., Hoffmann, I., Szatlóczki, G., Biró, E., Zsura, F., Pákáski, M., Kálmán, J.: Automatic Detection of Mild Cognitive Impairment from Spontaneous Speech using ASR. Proceedings of Interspeech Dresden, Germany (2015) 2694–2698
21. Hoffmann, I., Tóth, L., Gosztolya, G., Szatlóczki, G., Vincze, V., Kárpáti, E., Pákáski, M., Kálmán, J.: Beszédfelismerés alapú eljárás az enyhe kognitív zavar automatikus felismerésére spontán beszéd alapján. Általános Nyelvészeti Tanulmányok **29** (2017) 385–405
22. Tóth, L., Hoffmann, I., Gosztolya, G., Vincze, V., Szatlóczki, G., Bánréti, Z., Pákáski, M., Kálmán, J.: A Speech Recognition-based Solution for the Automatic Detection of Mild Cognitive Impairment from Spontaneous Speech. Current Alzheimer Research **15** (2) (2018) 130–138
23. Laske, Ch., Sohrabi, H.R., Frost, Sh.M, López-de-Ipina, K., Garrard, P., Buscema, M., Dauwels, J., Soekadar, S.R., Mueller, S., Linnemann, Ch., Bridenbaugh, S.A., Kanagasingam, Y., Martins, R.N., O'Bryant, S.E.: Innovative diagnostic tools for early detection of Alzheimer's disease. Alzheimer's & Dementia **11** (2015) 561-578
24. Roark, B., Mitchell, M., Hosom, J.P., Hollingshead, K., Kaye, J.: Spoken language derived measures for detecting mild cognitive impairment. IEEE Transactions on Audio, Speech, and Language Processing **19** (7) (2011) 2081–2090
25. Hoffmann, I., Németh, D., Dye, C.D. and Pákáski, M., Irinyi, T., Kálmán, J.: Temporal parameters of spontaneous speech in Alzheimer's disease. International Journal of Speech-Language Pathology **12** (1) (2010) 29–34
26. López-de-Ipina, K., Alonso, J.B., Solé-Casals, J., Barroso, N., Henriquez, P., Faundez-Zanuy, M., Travieso, C.M., Ecay-Torres, M., Martinez-Lage, P., Eguiraun, H.: On Automatic Diagnosis of Alzheimer's Disease Based on Spontaneous Speech Analysis and Emotional Temperature. Cognitive Computation **7** (1) (2015) 44–55
27. Folstein, M.F., Folstein, S.E., McHugh, P.R.: Mini-mental state: A practical method for grading the cognitive state of patients for the clinician. Journal of Psychiatric Research **12** (3) (1975) 189–198
28. Gósy, M.: BEA A multifunctional Hungarian spoken language database. The Phonetician **105** (106) (2012) 50–61
29. Tóth, L.: Phone Recognition with Hierarchical Convolutional Deep Maxout Networks. EURASIP Journal on Audio, Speech, and Music Processing **25** (2015) 1–13
30. Garrard, P., Rentoumi, V., Gesierich, B., Miller, B., Gorno-Tempini, M.L.: Machine learning approaches to diagnosis and laterality effects in semantic dementia discours. Cortex **55** (2014) 122–129
31. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A:J., Williamson, R.C.: Estimating the Support of a High-Dimensional Distribution. Neural Computation **13** (7) (2001) 1443–1471
32. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (3) (2011) 1–27
33. Cawley, G.C., Talbot, N.L.C.: On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Journal of Machine Learning Research **11** (2010) 2079–2107

34. Waegeman, W., Dembczynski, K., Jachnik, A., Cheng, W., Hüllermeier, E.: On the Bayes-Optimality of F-Measure Maximizers. Journal of Machine Learning Research **1** (15) (2014) 3333–3388.

# Betegségek automatikus szétválasztása időben eltolt akusztikai jellemzők korrelációs struktúrája alapján

Sztahó Dávid, Kiss Gábor, Tulics Miklós Gábriel, Vicsi Klára

Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
{sztaho, kiss, tulics, vicsi}@tmit.bme.hu

**Kivonat:** Egyes betegségtípusok különböző módon befolyásolhatják beszédképzésünk összetett mechanizmusait, patológiás beszédet eredményezve. Biomarkerek kinyerése a beszédből megbízható jelzői lehetnek a különböző betegségtípusoknak. A cikk célja egészséges és különböző betegségtípusokban szenvedő bemondók beszédmintáinak különválasztása. A vizsgált betegségtípusok a következők: depresszió, Parkinson-kór, hangképző szervek morfológiai elváltozása, a funkcionális diszfónia és a rekurrens paresis. Az osztályozó bemenetére formánsfrekvenciák (F1, F2, F3), a mel-szűrő sáv energia értékei, a mel-frekvencia kepsztrális együtthatók (MFCCs), az alapfrekvencia (F0) és az intenzitás időben eltolt értékeinek korrelációs mátrixaiból származtatott értékei kerültek. Szupport vektor gépet, valamint k-legközelebbi szomszéd osztályozási eljárásokat használtunk az eredmények összehasonlítására. Hatosztályos osztályozás esetben a legjobb osztályozási pontosság 54.8%-nak adódott, míg négyosztályos esetben 77.6%. Az elért eredmények alapján kijelenthető, hogy egy beszédalapú rendszer létrehozható, amely segít a klinikai személyzetnek a korai diagnózis felállításában.

## 1 Bevezetés

A biomarkerek alkalmazása egyre népszerűbb, hiszen mérhető információt biztosítanak egy betegség súlyosságára vagy jelenlétére. A beszéd egyike azon biomarkereknek, amelyek számos betegséget jelezhetnek. Ez olcsó, nem invazív és hatékony módszerek fejlesztésére ad lehetőséget, amely segítheti a szakemberek munkáját.

A diszfónia a hangképzés komplex zavarát jelenti. Olyan patológiás állapot, melynek hátterében vagy a hangképző szerv organikus megbetegedése vagy idegrendszeri szabályozási zavar áll. A diszfónia a normálistól (euphonia) eltérő hangszínt, intenzitást, dallamot, hangmagasságot és a hangképző szerv csökkent terhelhetőségét eredményezi. A diszfóniás hang rendszerint rekedt, levegős, fátyolos [1][2]. A diszfóniát rendszerint két csoportra bontják. Az első akkor fordul elő, amikor az orvos hangbeli problémát észlel fiziológiai elváltozás hiányában, amelyet funkcionális diszfónia (FD - functional dysphonia) néven említenek, a második eset, amikor a hangproblémát a beszédképzés egyik alrendszerének fiziológiai torzulása kíséri, amelyet a vokális szervek morfológiai változásaként (MA - morphological alteration) illetnek. Az olyan betegségek, mint a hangszalagcsomók, a polipok, a gastrooesophagealis reflux betegség

(GERD), a ciszta és az egy vagy kétoldali hangszalagbénulás (RP - recurrent paresis, rekurrens paresis) mind a strukturális organikus rendellenességekbe sorolhatók, míg olyan betegségek, mint a stroke, Parkinson-kór (PD - Parkinson's disease) vagy sclerosis multiplex a neurológiai hangrendellenességek csoportjába sorolhatók.

A depresszió egy pszichiátriai betegség. A betegséget elsősorban a stressz vagy a kudarc okozhatja, amelynek érzelmi, kognitív, testi és motivációs tünetei lehetnek. A depresszió felismerési rátája alacsony, a páciensek emiatt nem kapnak megfelelő kezelést vagy félrekezelik őket. Azt jósolják, hogy 2020-ra a mentális fogyatékosság második legszignifikánsabb okozója lesz [6][7]. A beszéd a depresszió észlelésének jó objektív markere lehet, amit számos kutatás is alátámaszt [8][9][10][11][21].

A Parkinson-kór (PD) az egyik leggyakoribb neurológiai rendellenesség. A Parkinson-kórban szenvedő betegek hangjainak jellemzői közé tartozik a pontatlan és koordinálatlan artikuláció, csökkent hangosság, fokozott hangremegés, változó beszédsebesség és lélegzetvesztés, levegős és érdes hangminőség [12][13][14][15][16][19].

Az eddigi tanulmányok többnyire kétosztályos osztályozással foglalkoztak egészséges és patológiás beszéd szétválasztására. Korábbi munkáinkban kétosztályos osztályozási rendszereket fejlesztettünk ki, amely az egészséges beszédet a diszfóniásoktól [3], depresszióban szenvedő betegek hangjaitól [8], valamint Parkinson-kórban szenvedő betegek beszédétől [13] különböztetett meg. A gyakorlatban mindezen betegségek előfordulhatnak a páciensek körében. A jelenlegi kutatásban több (4 vagy 6) különböző betegség típusok szétválasztására fókuszálunk, többosztályos osztályozási módszer alkalmazásával. A vizsgált betegségtípusok a következők: depresszió, Parkinson-kór, vokális szervek morfológiai változása, funkcionális diszfónia és rekurrens paresis. Olyan akusztikai jellemzők, mint a jitter, shimmer, HNR (Harmonics-to-Noise Ratio) hasznosak az egészséges és diszfóniás hangok automatikus osztályozásában, folyamatos beszéd esetén [3][4][5].

Hipotézisünk, hogy ezek a betegségek befolyásolják a formánsfrekvenciákat (F1, F2, F3), a mel-szűrő sáv energia értékei, a mel-frekvencia kepsztrális együtthatók (MFCCs), az alapfrekvencia (F0) és az intenzitás időben eltolt értékeinek korrelációs mátrixait. (Korrelációs struktúra értékeket kétosztályos osztályozásra korábban is használtak [17][18][20][22].)

## 2 Adatbázisok

A kutatásban összesen négy adatbázist használtunk: hármat minden egyes betegségtípusra (a fonációs rendellenességek egy adatbázisban szerepelnek külön kategóriákként), valamint egy egészséges kontroll beszédadatbázist. Minden páciens Aiszóposz meséjét, „Az északi szél és a nap"-ot olvasta fel. Ezen népmese gyakran használt a foniátriai kutatásokban, a szöveganyagát úgy szerkesztették meg, hogy az adott nyelvben előforduló minden beszédhang, valamint a leggyakoribb hangkapcsolatok szerepelnek benne. Számos nyelvre elkészült ez a szöveg, köztük a jelen esetben is használt magyarra. A felvételek átlagosan 41 másodperc hosszúak voltak. Minden bemondó beleegyezett a beszédének rögzítésébe, egy beleegyező nyilatkozatot aláírva. Az adatbá-

zisok felvételeinek számát és leíró statisztikáit az 1. táblázatban foglaltuk össze. A felvételek minden esetben csendes orvosi rendelőben készültek, USB-s hangkártya segítségével.

### 2.1 Fonációs rendellenességek beszédadatbázisa (Phonation disorder Speech Database, PhoDb)

A felvételek az Országos Onkológiai Intézetben, foniáter szakorvos rendelésén lettek rögzítve a páciensek beleegyezésével. A szakrendelésre általában különböző hangpanaszokkal érkeznek a betegek. A beszédadatbázisban lévő betegségek a következők: morfológiai elváltozás (MA - morphological alteration), mint a hangképző szervrendszer különböző pontjain előforduló tumorok, gasztroesophageal reflux (GERD), krónikus gégegyulladás, bulbar paresis (agyideggyulladás), amiotrófiás laterálszklerózis (ALS), leukoplakia, stb.); hangszalagbénulás (RP - recurrens paresis); funkcionális diszfónia (FD). A beszéd minőségét a diagnózist felállító orvos határozta meg az RBH-skála alapján [23]. A négy-fokozatú auditív rekedtségi skálán a 0 a normál hangminőségnek, míg a 3 a súlyos rekedtségnek felel meg. Az R (Rauhingkeit) a hangszalagok rezgési irregularitásából adódó érdességet, a B (Bechauchtkeit) a hangszalagok zárási elégtelenségéből adódó levegő-turbulenciát, a H (Heiserkeit) a rekedtséget általában jellemzik. A felvételek Monacor ECM-100 közel beszélő mikrofonnak készültek.

### 2.2 Depressziós beszédadatbázis (Depressed Speech Database, DSDb)

A depressziós (DE) adatbázis magyar anyanyelvű depresszióban szenvedő hangfelvételek gyűjteménye. A hangfelvételek a Semmelweis Egyetem Pszichiátriai és Pszichoterápiás Klinikával együttműködésben készültek. Az adatbázis az enyhe depressziótól a súlyos depresszióig terjedő páciensek hangfelvételeit tartalmazza, akiket neurológus szakember nem diagnosztizált más neurológiai betegséggel. A depresszió mérésére és a felvételek osztályozására a Beck Depression Inventory II (BDI) skálát alkalmaztuk [24]. A felvételek Audio-Technika ATR3350 csiptetős mikrofonnak készültek.

### 2.3 Parkinson-kór beszédadatbázis (Parkinson's Speech Database, PSDb)

Az adatbázis magyar anyanyelvű, Parkinson-kórban szenvedő páciensek beszédének gyűjteménye. A beszédmintákat két budapesti egészségügyi intézetben gyűjtöttük: a Virányos Klinikán és a Semmelweis Egyetemen. A Parkinson-kór súlyosságát a Hoehn & Yahr skála (H-Y) adja meg [25]. A felvételek Audio-Technika ATR3350 csiptetős mikrofonnak készültek.

### 2.4 Egészséges kontroll csoport (Healthy Control, HC)

Az egészséges kontroll csoport alanyai nem szenvedtek ismert betegségben és semmilyen orvosi kezelés alatt nem álltak. A felvételek ugyanannak a szövegnek a felolva-

sását tartalmazzák, mint a patológiás adatbázisok esetén, valamint a rögzítési körülmények is hasonlók voltak. Az adatbázis 190 személy hangját tartalmazza: 85 férfi és 105 nő bemondóét. A felvételek Audio-Technika ATR3350 csiptetős mikrofonnak készültek.

1. Táblázat: Adatbázisok kor és betegség súlyosság szerinti leíró statisztikája

| **Adatbázis** | ***Súlyossági mérték*** | ***Nem*** | ***Felvételek száma*** | ***Súlyosság*** | ***Kor*** |
|---|---|---|---|---|---|
| PhoDb - MA | RBH (0-3) | férfi | 52 | 2.17(±0.88) | 55.4(±12.8) |
| | | nő | 70 | 1.83(±0.82) | 48.8(±15.3) |
| PhoDb - FD | RBH (0-3) | férfi | 20 | 1.45(±0.69) | 56.2(±14.5) |
| | | nő | 48 | 1.31(±0.59) | 53.1(±17.3) |
| PhoDb - RP | RBH (0-3) | férfi | 22 | 2.50(±0.80) | 50.2(±15.4) |
| | | nő | 51 | 1.86(±0.83) | 58.2(±10.6) |
| DSDb | BDI (0-61) | férfi | 20 | 26.6(±8.9) | 44.1(±14.3) |
| | | nő | 35 | 28.2(±10.2) | 43.4(±13.5) |
| PSDb | H-Y (0-5) | férfi | 40 | 2.74(±1.05) | 64(±9.5) |
| | | nő | 36 | 2.74(±1.10) | 65.4(±9.4) |
| HC | - | férfi | 85 | - | 44.7(±18.7) |
| | | nő | 105 | - | 47.7(±13.8) |

# 3. Módszerek

## 3.1 Akusztikai jellemzők

Számos akusztikai jellemzőt választottunk ki, amelyek patológiás esetekben követik a hangképzés változását. Ezeket az akusztikai jellemzőket alacsony szintű leíróknak neveztük, amelyekből a következő jellemző csoportokat alkottuk: formáns frekvenciák (F1, F2, F3), mel-sávos energia-értékek (27 sáv 60 Hz-től 8 kHz-ig), mel-frekvenciás kepsztrális együtthatók (MFCC-k, amelyek 12 koefficienssel rendelkeznek), valamint az alapfrekvenciát (F0) és intenzitást közösen tartalmazó csoportot. Minden akusztikai jellemzőt Praat [26] szoftverrel számítottunk 10 ms-os időközzel.

## 3.2 Korrelációs struktúra jellemzők

A korreláció és kovariancia struktúrák (mátrixok) számítását és a belőlük származtatott jellemzőket a Williamson és társai eljárása [17][18] alapján végeztük. A korábban említett alacsony szintű leírójellemzőkből képezett idősorokat, mint csatorna használtuk (a [17] és [18] cikkek jelölései szerint) és a következő csoportokat hoztunk létre: „formánsok” (F1, F2, F3), „melsávok” (27 mel-sávos energia érték), „mfcc” (12 mfcc együttható), "enf0" (intenzitás és f0).

A korreláció és kovariancia struktúrákat minden egyes beszédmintára kiszámítottuk. Minden ilyen struktúra egy *(k*n) × (k*n)* mátrixot jelent, ahol a k a csatornák száma (például $k = 3$ a „formánsok" csoport esetében), míg $n$ a késleltetések száma. Ez a mátrix felfogható úgyis, mint $k^2$ darab almátrix, amik elemei $n \times n$ méretű mátrixok. Minden ilyen almátrix adott két csatorna korrelációs vagy kovariancia együtthatóit tartalmazza. Adott két csatorna egyenként $n$ féle különböző késleltetése mellett, ahol a csatorna az $i=1,2,...,k$ és $j=1,2,...,k$ lehetséges értékek között változik, míg az adott almátrixon belül pedig a késleltetés az egyik illetve a másik csatorna esetében $p=0,1,...,n-1$ illetve $q=0,1,...,n-1$ értékek között változik. A struktúra ilyenfajta felépítése következtében az átlón lévő almártixokban az egyes csatornák autokorrelációs együtthatói találhatóak meg különböző késleltetések mellett. A mátrixok az átlóra szimmetrikusak, illetve a sajátértékeik pozitívak. A mátrixokat 4 különböző időskála esetén is kiszámítottuk, ahol az időskála értéke határozta meg, hogy az adott késleltetés mekkora időeltolást jelent a csatorna idősorában. A korreláció és kovariancia struktúra részleteseb leírása megtalálható a [22] irodalomban, ezek korábbi beszédjelre alkalmazott gyakorlati megvalósításai pedig [17][18] irodalmakban.

Az időkésleltetések száma 10 volt ($n = 10$) a „melsávok", „mfcc" és „enf0" csoportok esetében, míg ez az érték 30 volt ($n = 30$) volt a „formánsok" csoport esetében. Ahogy korábban említettük, 4 különböző időskálát használtunk, amik értéke rendre 1,2,4 és 8 voltak minden csoport esetében, ami időben 10 ms, 20 ms, 40 ms és 80 ms időeltolást jelentett $n = 1$ időkésleltetés mellett. Az 1. ábrán példaként láthatóak az átlagos korreláció mátrixok a 4 különböző vizsgált osztály esetében, a „formánsok" csoport, 1-es időskála használata mellett. Összesen 16 korreláció és 16 kovariancia mátrix lett kiszámíttatva minden beszédminta esetében.

A korreláció és kovariancia struktúrákból a következő származtatott jellemzőket számítottuk ki és használtunk fel minden időskála esetében: korreláció mátrix sajátértékei, a korreláció mátrix sajátértékeinek entrópia értéke és a kovariancia mátrix sajátértékeinek négyzetes középben vett átlagát. Ezek a jellemzők voltak a bemenetei az osztályozásnak.

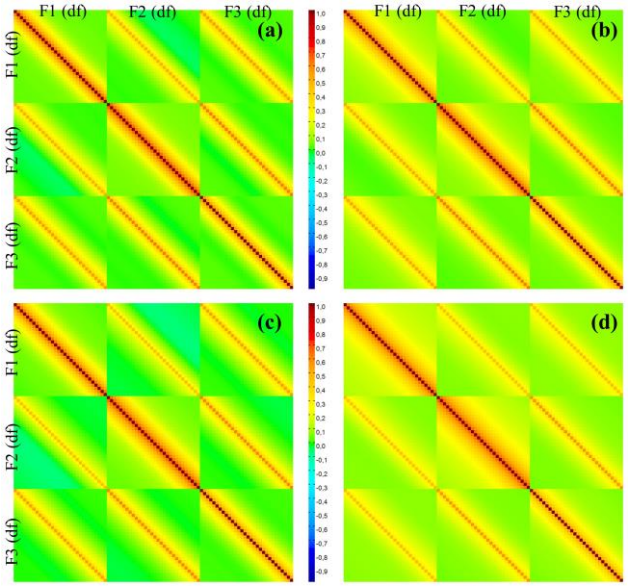
### 3.3 Osztályozás

A RapidMiner Studio 7.5 [27] szoftvert használtuk a gépi tanulási kísérletekhez. Az osztályozási módszerek paraméterei az adott eljárás során bevett alapértelmezett értékei voltak. A kutatás során a k-legközelebbi szomszédok (k-NN, $k$ paramétert 9-re állítva) és szupport vektor gépeket használtunk, c-SVC lineáris ($C = 1$ paraméterrel) és radiális bázis alapú kernelfüggvénnyel (ahol C-nek az akusztikai jellemzők számát választottuk és $\gamma = \frac{1}{akusztikai\ paraméterek\ száma}$). Minden vizsgálatot 10-szeres keresztvalidációval végeztünk, ahol az egyes osztályok eloszlása egyenletes volt.

Először hat csoportot külön kíséreltünk meg osztályozni: HC, DE, PD és az MA, FD és RP osztályokat a Fonációs rendellenességek beszédadatbázisából. A Fonációs rendellenességek beszédadatbázis három csoportját azért is kezeltük külön, mert korábbi munkánkban azt találtuk, hogy az MA és RP csoportok elkülöníthetők lehetnek egymástól [5].

Ezek után a Fonációs rendellenességek beszédadatbázisa három csoportját egybevonva (Fonációs rendellenességek, továbbá FR) négy osztályos osztályozást végeztünk az HC, DE, PD és FR csoportok között.

Optimális akusztikai jellemzők megtalálása érdekében Forward Selection jellemzőkiválasztó eljárást használtunk. Költségfüggvényként pontosságot (accuracy) választottunk, a maximálisan kiválasztott jellemzők számára 20-at választottunk.




**1. ábra.** Formáns frekvencia csoport korrelációs mátrixa, 1-es skálát használva, (a)-egészséges, (b)-depresszió, (c)-morfológiai elváltozás, (d)-Parkinson-kór

## 4. Eredmények

A hat, illetve a négyosztályos osztályozás pontosság ($\frac{\textbf{\textit{helyesen felismert minták száma}}}{\textit{összes minta száma}}$) eredményeit a 2. táblázatban foglaltuk össze. A táblázatban megtalálhatók minden akusztikai jellemző csoporttal külön végzett, valamint együttesen használva kapott eredmények. Különböző akusztikai jellemzők csoportjai különböző elkülönítési teljesítménnyel rendelkeznek. Általánosan az 'enf0' csoport teljesített a legrosszabbul, ebből az következik, hogy az intenzitás és az alaphang auto- és keresztkorrelációs értékei nem rendelkeznek magas elkülönítési képességgel. A további három jellemző csoport mind magasabb osztályozási pontosság értéket ért el, ezek közül is a 'melsávok' csoport teljesített a legjobban.

Hat osztályos osztályozás esetében az MA, FD és RP osztályok esetében sok esetben fordult elő az egymásra tévesztés. A minden akusztikai jellemzőt felhasználó SVM-RBF osztályozás tévesztési mátrixát a 3. táblázat foglalja össze. Az egymásra tévesztés jelensége miatt vontuk össze egy osztállyá az MA, FD és RP osztályokat, így a négy

2. Táblázat: Osztályozási eredmények (pontosság, accuracy) 6, illetve 4 osztályos esetben

| Jellemző csoport | skála | k-nn | svm-linear | svm-rbf |
|---|---|---|---|---|
| **enf0** | 1 | 37,85 / 51,94 | 41,20 / 54,05 | 41,73 / 56,51 |
| | 2 | 38,73 / 54,23 | 42,43 / 54,05 | 41,20 / 54,93 |
| | 4 | 35,92 / 49,65 | 40,32 / 53,00 | 42,08 / 57,75 |
| | 8 | 32,92 / 46,30 | 36,17 / 48,06 | 35,21 / 47,71 |
| | összes | 38,73 / 53,87 | 34,51 / 55,89 | 35,21 / 56,34 |
| **formánsok** | 1 | 38,03 / 55,11 | 46,13 / 64,26 | 44,72 / 65,49 |
| | 2 | 36,27 / 53,87 | 43,31 / 61,80 | 43,31 / 62,15 |
| | 4 | 37,50 / 57,75 | 45,95 / 62,68 | 43,49 / 63,56 |
| | 8 | 38,38 / 57,39 | 47,71 / 63,56 | 45,25 / 65,85 |
| | összes | 38,38 / 58,10 | 42,78 / 64,96 | 42,08 / 64,61 |
| **melsávok** | 1 | 35,21 / 51,58 | 44,54 / 60,56 | 45,95 / 63,91 |
| | 2 | 38,56 / 51,76 | 48,06 / 63,73 | 49,12 / 69,54 |
| | 4 | 39,44 / 55,28 | 49,82 / 65,32 | 47,54 / 66,73 |
| | 8 | 42,43 / 52,28 | 50,53 / 67,08 | 49,47 / 70,25 |
| | összes | 41,55 / 56,34 | 51,06 / 72,36 | 50,35 / 74,12 |
| **mfcc** | 0 | 36,97 / 50,53 | 41,55 / 57,22 | 40,32 / 57,57 |
| | 1 | 36,97 / 53,87 | 42,78 / 60,21 | 42,25 / 63,03 |
| | 2 | 39,44 / 54,93 | 41,78 / 59,15 | 39,61 / 57,75 |
| | 3 | 40,14 / 59,68 | 41,20 / 64,79 | 41,55 / 65,49 |
| | összes | 42,08 / 60,74 | 43,84 / 68,66 | 44,89 / 69,37 |
| **Összes jellemző** | 0 | 39,26 / 57,22 | 45,42 / 72,01 | 45,42 / 71,48 |
| | 1 | 44,54 / 59,68 | 50,00 / 74,30 | 46,30 / 74,47 |
| | 2 | 45,42 / 62,68 | 46,65 / 67,25 | 47,01 / 68,84 |
| | 3 | 46,65 / 63,03 | 48,94 / 75,00 | 47,01 / 73,42 |
| | összes | 47,54 / 63,20 | 48,77 / **76,23** | 48,42 / **77,64** |
| **Összes jellemző jellemző-kiválasztással** | 0 | 43,13 / 65,49 | 46,48 / 72,76 | 53,87 / 72,18 |
| | 1 | 44,89 / 61,27 | 54,93 / 72,40 | 52,64 / 72,36 |
| | 2 | 48,06 / 66,55 | **53,52** / 69,24 | 51,94 / 69,72 |
| | 3 | 48,77 / 68,31 | 52,46 / 72,15 | 52,64 / 71,83 |
| | összes | **51,41 / 71,30** | 53,32 / 76,17 | **54,75** / 77,59 |

osztályos osztályozás eredményeképpen 77,64%-os pontosságot értünk el SVM-RBF-et használva.

Általánosságban elmondható, hogy az összes időskála felhasználása javított az osztályozási eredményeken. A legmagasabb pontosságot akkor értük el, amikor az összes akusztikai jellemző csoport felhasználásra került. A jellemző-kiválasztásos kísérletek során a legnagyon pontosság 54,75%-nak adódott hatosztályos esetben, valamint 77,64% négyosztályos esetben. A jellemző-kiválasztás növelte az osztályozási pontosságot k-NN esetben in. Említésre méltó, hogy jellemző-kiválasztással egy olyan egyszerű algoritmus, mint amilyen a k-NN, összemérhető eredményeket produkált egy sokkal komplexebb osztályozóval, mint amilyen a szupport vektor gép.

3. Táblázat: Tévesztési mátrix minden jellemzőt felhasználva (összes skála) SVM-RBF esetén. A cella értékei százalékok.

| Prediktált\Valós | HC | DE | PD | FD | MA | RP |
|---|---|---|---|---|---|---|
| **HC** | **73.16** | 25.45 | 16.88 | 26.87 | 11.61 | 1.49 |
| **DE** | 8.42 | **56.36** | 9.09 | 1.49 | 1.79 | 0.00 |
| **PD** | 5.79 | 10.91 | **72.73** | 0.00 | 1.79 | 1.49 |
| **FD** | 5.26 | 0.00 | 0.00 | **32.84** | 17.86 | 8.96 |
| **MA** | 6.84 | 5.45 | 1.30 | 28.36 | **19.64** | 80.60 |
| **RP** | 0.53 | 1.82 | 0.00 | 10.45 | 47.32 | **7.46** |

## 5. Következtetések

Ebben a kutatásban kísérletet tettünk különböző típusú patológiás rendellenességek automatikus szétválasztására formánsfrekvenciák (F1, F2, F3), mel-sávos energia értékek, mel-frekvencia kepsztrális együtthatók (MFCC), az alapfrekvencia (F0) és az intenzitás időben eltolt értékeinek korrelációs mátrixai alapján. Többi eltolási skálát és különböző osztályozási eljárást használva a legjobb osztályozási pontosságként 77,64%-ot értünk el négyosztályos osztályozás esetében. Ez ígéretes eredménynek számít, hiszen az adatbázisokban szereplő hangfelvételek száma limitált. Ez az eredmény azt sugallja, hogy valóban vannak korrelációs különbségek a mért időtartománybeli jellemzőkben a négy vizsgált betegségtípus esetén. Az eredmények alapján a korrelációs struktúrák integrálhatók egy automatikus komplex diagnosztikai rendszerbe.

A hatosztályos osztályozási kísérletekben a tévesztési mátrixok alapján elmondható, hogy a fonációs rendellenesség betegségtípusokat (MA, FD és RP csoportokat) a rendszer sok esetben összekeverte. Ahhoz, hogy ezek a betegségtípusok egymástól automatikusan megkülönböztethetőek legyenek, további akusztikai jellemzőket kell bevonni.

## Bibliográfia

1. Tulics, M.G., Kazinczi, F., Vicsi, K., “Statistical analysis of acoustical parameters in the voice of children with juvenile dysphonia,” in: International Conference on Speech and Computer, Springer. 2016, pp. 667–674.
2. Ruotsalainen, J., Sellman, J., Lehto, L., Verbeek, J., “Systematic review of the treatment of functional dysphonia and prevention of voice disorders,” Otolaryngology-Head and Neck Surgery 138, 2008, pp. 557–565.
3. Kazinczi, F., Mészáros, K., Vicsi, K., “Automatic detection of voice disorders,” in: International Conference on Statistical Language and Speech Processing, Springer, 2015, pp. 143–152.
4. Grygiel J. and Strumillo P., “Application of Mel Cepstral Representation of Voice Recordings for Diagnosing Vocal Disorders,” Przeglad Elektrotechzny (Electrical Review), 2012.

5. Tulics, M.G., and Vicsi, K., "Phonetic-class based correlation analysis for severity of dysphonia," in: Cognitive Infocommunications (CogInfoCom), 2017 8th IEEE Conference on, IEEE, 2017, pp. 21-26.
6. Kessler, R.C., Bromet, E.J., "The epidemiology of depression across cultures," Annual review of public health 34, 2013, pp. 119–138.
7. Lépine, J.P., Briley, M., "The increasing burden of depression," Neuropsychiatric disease and treatment 7, 2011, pp 3.
8. Kiss, G., Vicsi, K., "Mono-and multi-lingual depression prediction based on speech processing," International Journal of Speech Technology, 2017, pp. 1–17.
9. Cummins, N., Scherer, S., Krajewski, J., Schnieder, S., Epps, J., Quatieri, T.F., "A review of depression and suicide risk assessment using speech analysis," Speech Communication 71, 2015, pp. 10–49.
10. Asgari, M., Shafran, I., "Improvements to harmonic model for extracting better speech features in clinical applications," Computer Speech & Language 47, 2018, pp. 298–313.
11. Valstar, M., Schuller, B., Smith, K., Eyben, F., Jiang, B., Bilakhia, S., Schnieder, S., Cowie, R., Pantic, M., "Avec 2013: the continuous audio/visual emotion and depression recognition challenge," in: Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge, ACM. 2013., pp. 3–10.
12. Sztahó D, Vicsi, K., "Estimating the severity of Parkinson's disease using voiced ratio and nonlinear parameters," in: Pavel Král, Carlos Martín-Vide, Statistical Language and Speech Processing: 4th International Conference, SLSP 2016, Proceedings. Springer International Publishing, 2016. pp. 96-107.
13. An, G., Brizan, D. G., Ma, M., Morales, M., Syed, A. R., & Rosenberg, A., "Automatic Recognition of Unified Parkinson's Disease Rating from Speech with Acoustic, i-Vector and Phonotactic Features, " Sixteenth Annual Conference of the International Speech Communication Association. 2015.
14. Naranjo, L., Pérez, C.J., Campos-Roca, Y., Martín, J., "Addressing voice recording replications for parkinson's disease detection," Expert Systems with Applications 46, 2016, pp. 286–292.
15. Mekyska, J., Smekal, Z., Galaz, Z., Mzourek, Z., Rektorova, I., Faundez-Zanuy, M., López-de Ipiña, K., "Perceptual features as markers of parkinson's disease: the issue of clinical interpretability," in: Recent Advances in Nonlinear Speech Processing. Springer, 2016, pp. 83–91.
16. 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: International Conference on Text, Speech, and Dialogue, Springer, 2017, pp. 411–419.
17. J. R. Williamson, T. F. Quatieri, B. S. Helfer, G. Ciccarelli, and D. D. Mehta, "Vocal and facial biomarkers of depression based on motor incoordination and timing," in Proceedings of the 4th ACM International Workshop on Audio/Visual Emotion Challenge (AVEC), 2014, pp. 65–72.
18. J. R. Williamson, T. F. Quatieri, B. S. Helfer, R. Horwitz, B. Yu, and D. D. Mehta, "Vocal biomarkers of depression based on motor incoordination," in Proceedings of the 3rd ACM International Workshop on Audio/Visual Emotion Challenge, 2013, pp. 41–48.
19. Williamson, James R., et al. "Segment-dependent dynamics in predicting Parkinson's disease." Sixteenth Annual Conference of the International Speech Communication Association. 2015.
20. B. Yu, T. F. Quatieri, J. W. Williamson, and J. Mundt, "Prediction of cognitive performance in an animal fluency task based on rate and articulatory markers," in 15th Annual Conference of the International Speech Communication Association, September 9–13, Portland, Oregon, Proceedings, 2014.

21. B. S. Helfer, T. F. Quatieri, J. R. Williamson, L. Keyes, B. Evans, W. N. Greene, J. Palmer, and K. Heaton, “Articulatory dynamics and coordination in classifying cognitive change with preclinical mTBI,” in 15th Annual Conference of the International Speech Communication Association, September 9–13, Portland, Oregon, Proceedings, 2014.
22. J. R. Williamson, D. Bliss, D. W. Browne, and J. T. Narayanan, “Seizure prediction using EEG spatiotemporal correlation structure,” Epilepsy and Behavior, vol. 25, no. 2, 2012, pp. 230–238.
23. Wendler, J., Rauhut, A., Kruger, H., “Classification of voice qualities,” Journal of Phonetics 14, 1986, pp. 483–488.
24. Beck, A.T., Steer, R.A., Ball, R., Ranieri, W.F., “Comparison of beck depression inventories-ia and-ii in psychiatric outpatients,” Journal of personality assessment 67, 1996, pp. 588–597.
25. Hoehn, M.M., Yahr, M.D., “Parkinsonism onset, progression, and mortality,” Neurology 17, 1967, pp. 427–427.
26. Boersma, Paul & Weenink, David (2018). Praat: doing phonetics by computer [Computer program]. Version 6.0.39, retrieved 3 April 2018 from http://www.praat.org/
27. Hofmann, M. & Klinkenberg, R. “RapidMiner: Data Mining Use Cases and Business Analytics Applications”. 2013

# MORFOLÓGIA, NYELVI ELEMZÉS

# PoS-tagging and lemmatization with a deep recurrent neural network

Gábor Ugray

memoQ Translation Technologies
gabor.ugray@memoq.com

**Abstract**. Neural networks have been shown to successfully solve many natural language processing tasks previously tackled by rule-based and statistical approaches. We present a deep recurrent network with long short-term memory, identical to engines used in machine translation, to solve the problem of joint PoS-tagging and lemmatization in Hungarian and German. Our model achieves comparable or superior results to a state-of-the-art statistical PoS tagger. We are able to enhance the Hungarian model's performance, as measured on a manually annotated sample unrelated to the initial training corpus, through an additional synthesized dataset.

**Keywords:** PoS-tagging, lemmatization, neural networks, LSTM, Hungarian, German

## 1 Introduction

In recent years we have seen deep neural networks applied to many linguistic modeling tasks that were previously tackled by statistical or rule-based approaches. Németh & Ács [1] achieved promising results for Hungarian hyphenation. Chinese word segmentation is a challenge because of the scripts's lack of spaces. Zheng, Cheng & Xu [2] have shown that a neural model yields results competitive with the state of the art in word segmentation and PoS-tagging.

While many of these approaches formulate the problem as a classification task, Rei, Crichton & Pyysalo [3] have studied sequence labeling and found that an attention model improves performance.

In the problem domain of morphologically rich languages, Yildiz & al [4] have trained neural networks to disambiguate the output of a rule-based morphological analyizer (MA). Zalmout & Habash [5] have successfully used the same approach for Arabic.

In the present paper, we set out to explore a related, but slightly broader, problem: joint PoS-tagging and lemmatization. We define the challenge as a sequence-to-sequence transformation identical to machine translation (MT) between different natural languages. We train an off-the-shelf neural MT engine and achieve outcomes that are competitive or superior to a state-of-the-art PoS tagger. We show that we can boost the neural model's performance on new domains through a training dataset

synthesized via the state-of-the-art statistical PoS tagger trained on a relatively small, manually annotated corpus.

## 2 Experimental setup

### 2.1 Recurrent network with long short-term memory and attention

We formulate the joint task of PoS-tagging and lemmatization as a sequence-to-sequence transformation [6]. The transformation's input is the token to be tagged and lemmatized, surrounded by a chosen number of preceding and following tokens for context. The output is the lemma, followed by one or more tags. For illustration, *Table 1* shows the first few input-output pairs generated from a tokenized sentence, with a context window of 5 surface tokens.

| | |
|---|---|
| [Beg] Néhány [End] pillanat múl■ tán hangot hallott az | néhány [/Num] [Nom] |
| Néhány [Beg] pillanat [End] múl■ tán hangot hallott az ember | pillanat [/N] [Nom] |
| Néhány pillanat [Beg] múl■ tán [End] hangot hallott az ember , | múl■ tán [/Post] |
| Néhány pillanat múl■ tán [Beg] hangot [End] hallott az ember , amely | hang [/N] [Acc] |
| Néhány pillanat múl■ tán hangot [Beg] hallott [End] az ember , amely a | hall [/V] [Pst.NDef.3Sg] |
| Néhány pillanat múl■ tán hangot hallott [Beg] az [End] ember , amely a táb■ lák | az [/Det] |
| pillanat múl■ tán hangot hallott az [Beg] ember [End] , amely a táb■ lák mög■ ül | ember [/N] [Nom] |

**Table 1**: Sample input and output sequences from the neural model's training corpus.

Since we fix the context window's size in surface tokens, a convolutional neural network (CNN) might at first seem like a more natural choice. The experience of neural machine translation, however, is that decomposing the input into subword tokens is a successful way to address the open vocabulary problem. In our model, therefore, we further tokenize both the input tokens and the the target lemma using byte-pair encoding (BPE) [7]. The result is a dataset with random-length sequential input and output.

The models we train for the various experiments are identical, off-the-shelf neural MT models using a bidirectional LSTM and attention [8]. We use OpenNMT's [9] default parameters: 2 hidden layers with 500 hiden units. All models are trained for 13 epochs, with SGD optimization and a learning rate decaying from 1.0 by a factor of 0.7 from epoch 9 onwards. Word embeddings have 500 dimensions.

We use a shared BPE model for the source (surface words) and the target (lemmata), with 12.5 thousand merges. This is a comparatively small vocabulary for neural MT models. Our aim, however, is to model words, not sentences, so we feel even a smaller choice might be warranted. The begin/end delimiters in the source, and the morphological tags in the target, are preserved as distinct vocabulary words; they are exempt from BPE segmentation.

### 2.2 Experiments

We devise a set of experiments to answer various exploratory questions about the neural approach.

**Direct comparison: Hungarian.** How does the accuracy of a neural model compare with a state-of-the-art tagger, when trained and evaluated on a 19:1 split of the same annotated corpus? We train both PurePos [10] [11] and a neural model on 95% of the Szeged corpus [12], and measure accuracy on a 5% evaluation set, after a random split.

This experiment is also a replication study, because for the comparison we remeasure PurePos's reported tagging and lemmatization accuracy. We perform an initial measurement without a morphological analyzer (MA). PurePos's best numbers, however, were reported with an integrated MA. We therefore also reproduce that outcome using the recently open-sourced emMorph analyzer [13], in conjunction with a version of the Szeged corpus converted to the emMorph/HuMor formalism.

**Direct comparison: German.** The publications related to PurePos that we are aware of are all based on Hungarian datasets, but we are curious how well its results generalize to other languages. We therefore perform the same measurements using a comparable German annotated corpus, Tiger [14]. In this case, there is no compatible MA to include. In addition to PurePos, we also measure the tagging accuracy of NLTK's classifier-based tagger.

**Synthesized training data.** Can we improve the neural model by synthesizing additional training data? For our particular supervised learning scenario, the amount of manually annotated text is limited. Meanwhile, PurePos can generalize well to new input in part because of the integrated MA. We first train PurePos on the Szeged corpus, then use it to tag and lemmatize a different, 923-thousand-segment corpus. This automatically annotated dataset, together with the original Szeged corpus's training set, is used to train a neural model.

We compare this neural model's performance with PurePos on the Szeged corpus's validation set, and on a small manually annotated evaluation dataset. The aim is to test whether the neural model can learn a meaningful amount of Hungarian morphology from the examples transmitted through the larger synthesized traininig corpus.

## 3 Data and preparation

| Dataset | Sentences | Tokens | Types | Full tags | Tag vocab |
|---|---|---|---|---|---|
| Szeged | 81,967 | 1,485,306 | 152,057 | 1,246 | 169 |
| Tiger | 50,472 | 888,238 | 89,383 | 694 | 78 |
| Szeged+Synth | 1,005,464 | 10,330,582 | 609,359 | 4,763 | 214 |
| Eval | 491 | 4,959 | 2,331 | 264 | 120 |

**Table 2**: Key statistics about the datasets used for the experiments.

### 3.1 The Szeged corpus and Tiger

We used a version of the Szeged corpus where the annotations have been converted to the formalism of HuMor/emMorph[1]. The numbers related to the Szeged corpus in *Table 2* are from this converted version.

In HuMor's output, a sequence of tags encodes each word's morphological information. E.g., [/N][Pl][Acc] is a noun, in plural form and with an accusative case marker. The *Full tags* column in *Table 2* refers to the number of distinct tag sequences attested in the data; *Tag vocab* is the number of distinct bracketed tags. Referring back to *Table 1*, we can see that in the neural model's training data we chose to treat each bracketed tag as a separate token. This results in a smaller vocabulary and the possibility that the model can output even rare (but correct) sequences not attested in the training data.

The Tiger corpus uses a small set of part-of-speech categories and has additional annotations for each word's inflectional categories. As an example, a particular instance of "größte" is lemmatized as "groß"; the PoS label is "ADJA"; and the inflectional categories are "case=acc|number=sg|gender=fem|degree=sup". For our purposes, we convert this to the following sequence of bracketed tags:

[ADJA][case=acc][number=sg][gender=fem][degree=sup]

### 3.2 Incompatible annotations in the Szeged corpus

As we shall see in the Results section, PurePos's tagging accuracy fell from 97.55% to 79.72%, and its lemmatization accuracy from 96.38% to 90.28%, when we first ran it with an MA, as opposed to relying only on the built-in guesser. This clearly indicated an incompatibility between the converted corpus annotations and emMorph's actual output.

We extracted words where emMorph's analyses did not include the annotation in the corpus. The problem was severe: it affected 32 thousand of the corpus's 152 thousand types, and 312 thousand of its 1.4 million tokens. Because it is not feasible to alter emMorph's rules and lexical database, we chose to adjust the corpus's annotations to make them compatible with emMorph's observed output.

Some problems were trivial, e.g., a difference in the way some punctuation marks were labeled. We also observed that the information following the pipe character ("|") was often incompatible, e.g., emMorph's analysis including a marker about the Latin origin of some words, which is not part of the corpus's annotations. We chose to remove everything from the first pipe onwards in every bracketed tag, both in the corpus and in emMorph's output.

Finally, there was a large number of words where all of emMorph's analyses included at least one derivational suffix, while the corpus annotation was the fully derived form. E.g., "földrajzos" is annotated as "földrajzos[/Adj][Nom]" in the corpus, but analyzed only as "földrajz[/N][_Adjz:s/Adj][Nom]" or

[1] The converted corpus was kindly provided by Veronika Vincze. Unfortunately, we haven't been able to obtain published information about the conversion process.

"földrajz[/N][_Nz:s/N][Nom]" by emMorph. We managed to identify a handful of such patterns and replaced the corpus annotation with the closest, slighly less derived analysis from emMorph.

We did not aim for perfection, as the pattern matching effort soon began to yield diminishing returns. We stopped when we reduced the discrepancy to 7,275 types with 24,152 token instances. With this effort, PurePos's tagging accuracy no longer deteriorated with the MA enabled, and its lemmatization accuracy increased slightly. Details are included in the Results section.

Making PurePos work with morphology was critical for the key experiment, which involves the automatic PoS-tagging and lemmatization of a large dataset with many types and lemmas not attested in the Szeged corpus.

### 3.3 Synthesized dataset

For the synthesized training data we used 923 thousand segments from open sources[2]. The corpus consists of 5% JRC-Acquis, 7% Europarl, 9% modern literature, and 79% movie subtitles. This particular corpus was chosen because it is sufficiently versatile; we had originally created it as a bilingual dataset for training a machine translation engine. For this research's purposes, we took a random subset of the original bilingual dataset's Hungarian sentences.

To prepare for tagging, we tokenized the already sentence-segmented corpus using quntoken, the standalone version of the e-magyar toolchain's [15] emToken component.

We did not find a trivial way to use emMorph as an integrated MA directly invoked by PurePos. We therefore first extracted all surface forms (types), executed HFST from the command line, and fed the analyses via PurePos's morphology table option. For this, we needed to slightly alter PurePos's source code, whose published version ignores lemmata from the morphology table and only returns tags.

Executing HFST itself posed a small challenge. Analyzing the 600 thousand extracted surface forms took over 12 hours, and was only possible in a dozen smaller batches. On larger batches the tool predictably runs out of memory and crashes before completing its job, even with the 1-second timeout option.

### 3.4 Manually annotated evaluation set

After training a neural model on an automatically tagged corpus, there are multiple ways to evaluate it.

First, we can measure to what extent it coincides with PurePos on a smaller, randomly selected validation set. This, however, would not measure how well the neural system learns to model linguistic reality; it would only show how well it learns to replicate PurePos's model. Second, we can check whether the neural model trained on the large corpus makes better predictions on the Szeged corpus's 5% validation set.

[2] http://opus.nlpl.eu/

The most insightful evaluation, however, is on a manually annotated gold standard that is not part of the Szeged corpus. This approach allows us to compare the neural model's performance to PurePos in a new domain.

To create the evaluation set we separated a small random sample of the synthesized corpus and manually corrected its annotations. This 492-sentence evaluation set was excluded from the neural model's training material. For the manual review we relied on the output of PurePos and emMorph's analyses, and frequently cross-checked with the Szeged corpus to mirror its conventions as closely as possible.

We share the manually annotated evaluation dataset, along with the output of the different models, as an Excel file[3].

### 3.5 Limitations

In addition to the remaining inconsistency in the Szeged corpus's annotations, we acknowledge a further limitation of our experimental setup. The 19:1 split of the corpus is different from the standard 9:1 split, and all of our experiments were done with a single random split. For more reliable results, a full roll would be required, retraining models repeatedly and alternating through different subsets of the corpus for evaluation. Due to limited time and resources, this was unfortunately not possible.

## 4 Results

### 4.1 Evaluation on the Szeged corpus

The initial question we set out to answer is whether a neural model can achieve comparable accuracy, or potentially even outperform a state-of-the-art tagger, as measured on a 19:1 split of the annotated Szeged corpus. *Table 3* shows the results we obtained with the converted corpus.

The *Tag-Full* column is tagging accuracy, as measured by the entire tag sequence, and counted by tokens. *Tag-First* is more permissive: it only checks the first bracketed tag (typically, although not always, the part of speech). *Lemma-Strict* is lemmatization accuracy; *Lemma-CI* is a more permissive, case-insensitive measure.

| Model | Tag-Full | Tag-First | Lemma-Strict | Lemma-CI |
|---|---|---|---|---|
| PurePos | 97.55% | 98.58% | 96.38% | 96.99% |
| PurePos+MA | 79.72% | 81.24% | 90.28% | 91.53% |
| Neural | 97.99% | 98.79% | 98.86% | 98.95% |

**Table 3**: Accuracy of the different taggers on the 5% validation set of the converted Szeged corpus.

[3] https://jealousmarkup.xyz/files/MSZNY2019-PoS-EvalSet.xlsx

In this setup, the neural model outperforms PurePos without an MA. As discussed in the previous section, adding an MA produced drastically bad results because of the incompatibility between emMorph's output and the corpus's annotations. Therefore, in *Table 4* we present the results of the same experiment, but this time repeated on the corpus with the adjusted annotations.

| Model | Tag-Full | Tag-First | Lemma-Strict | Lemma-CI |
|---|---|---|---|---|
| PurePos+MA | 97.41% | 98.57% | 97.24% | 97.69% |
| Neural (Szeged) | 97.89% | 98.83% | 98.51% | 98.70% |
| Neural (Szeged+ Synth) | 98.01% | 98.88% | 98.74% | 98.96% |

**Table 4**: Accuracy of the different taggers on the 5% validation set of the converted and adjusted Szeged corpus.

PurePos's tag accuracy with an MA is now effectively identical to its accuracy without an MA from the previous experiment; its lemmatization accuracy has improved. We would expect an improvement across the board if the corpus annotations had been completely brought in line with emMorph.

The neural model, again, slightly outperforms PurePos when trained on the same corpus. The model that was trained on the extended corpus (including Szeged's training set plus the 923-thousand-segment synthesized dataset) yields additional improvements. This is interesting, because the improvements are detected on Szeged's validation set, while the synthesized training data is based on an entirely different corpus.

### 4.2 Evaluation on Tiger

*Table 5* presents the results from Tiger, the 888-thousand-word German annotated corpus, after a 19:1 training/evaluation split. The first row, NLTK-CB, shows the tagging accuracy of the NLTK toolkit's classifier-based tagger. That tagger does not perform lemmatization, and only produces a single tag per token, so the other metrics are not applicable.

| Model | Tag-Full | Tag-First | Lemma-Strict | Lemma-CI |
|---|---|---|---|---|
| NLTK-CB | n/a | 94.07% | n/a | n/a |
| PurePos | 84.82% | 97.19% | 96.57% | 97.10% |
| Neural | 91.85% | 98.01% | 98.43% | 98.58% |

**Table 5**: Accuracy of the different taggers on the 5% validation set of the German Tiger corpus.

PurePos outperforms the classifier-based tagger, and the neural model outperforms PurePos on all metrics. The most drastic difference is in the full tagging accuracy. We conjecture that this may be related to the neural model's 5-word window, which is in a sense larger than PurePos's third-order Hidden Markov Model. We suspect that the

correct value of German inflectional categories (e.g., the gender and number of a form like "größte") might be driven by constituents farther away in the sentence. We did not, however, test this conjecture.

### 4.3 Annotated test set

The key experiment was the evaluation of the different models on a manually annotated dataset. *Table 6* shows the results.

| Model | Tag-Full | Tag-First | Lemma-Strict | Lemma-CI |
|---|---|---|---|---|
| PurePos | 95.72% | 97.90% | 95.62% | 96.51% |
| PurePos+MA | 96.87% | 98.21% | 97.06% | 97.90% |
| Neural (Szeged) | 93.83% | 96.53% | 94.98% | 97.70% |
| Neural (Szeged+ Synth) | 96.55% | 97.98% | 96.85% | 97.70% |

**Table 6**: Accuracy of the different taggers on the small, manual annotated gold standard dataset.

Unsurprisingly, PurePos with an MA outperforms PurePos without morphology. Obviously, both PurePos models were trained on the Szeged corpus's 95% training set, there being no other ground truth. "Neural (Szeged)" is the neural model trained on the same corpus. It significantly underperforms PurePos, particularly on the strict metrics.

"Neural (Szeged+Synth)" is the model that we trained on the extended corpus. On the manual evaluation set it fails to reach PurePos's accuracy with morphology, but it does outperform PurePos without an MA. In particular there is a big improvement in terms of full tagging accuracy and strict lemmatization accuracy.

### 4.4 A qualitative look

The filters of the accompanying Excel file with the results of each model on the evaluation set allow for a lot of exploration. Where "Neural (Szeged)" gets lemmata wrong we frequently see missing morphological insight, which is then corrected in "Neural (Szeged+Synth)". One example would be "odalbben" as the lemma returned for "odalent". Because the neural sequence-to-sequence system's output is generated recursively from the network's activation state, the model always produces *some* output, and that output can easily contain sequences that were never attested in the training data, or which simply don't make much sense.

We also see a few of the sort of "hallucinations" that have been observed in neural MT systems, but which are unimaginable in rule-based tools. One example would be "Robert" as the lemma returned for "4)".

115 tokens in the evaluation set are out-of-vocabulary (OOV), i.e., they were not attested in the training data. For 91 of these, the neural model returns a correct lemma, which we see as evidence that the model has acquired morphological insight.

## 5 Conclusion

We have shown that a deep neural network with a bidirectional LSTM topology can learn to jointly lemmatize and PoS-tag text in dissimilar languages such as Hungarian and German. Neural models achieve comparable or superior results to state-of-the-art statistical PoS taggers such as PurePos, even when these incorporate a morphological analyzer. When trained on the relatively small manually annotated corpora that are available for the PoS-tagging task, the neural model has difficulty generalizing to a new domain. However, if we boost the neural model with a large synthetic dataset automatically annotated by a traditional morphology-aware PoS-tagger, it achieves comparable results on a new domain as well.

We achieved these results using an off-the-shelf neural MT engine without any parameter tuning. We are confident that the results can be improved significantly by exploring different network dimensions and optimization methods, different context windows, and more or less aggressive sub-word segmentation. Much larger automatically annotated datasets are also easy to create, promising to broaden the neural model's morphological coverage even further.

Perhaps most importantly, for supervised learning tasks such as PoS-tagging, the core training data's amount and quality has a tremendous impact on the outcome.

## References

1. Gergely Dániel Németh, Judit Ács: Hyphenation using deep neural networks. *In V. Vincze (szerk.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szegedi Tudományegyetem, Szeged. (2018) pp. 146-158.
2. Xiaoqing Zheng, Hanyang Chen, Tianyu Xu: Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 647–657. (2013) Seattle, WA, USA
3. Marek Rei, Gamal K.O. Crichton, Sampo Pyysalo: Attending to Characters in Neural Sequence Labeling Models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 309–318, Osaka, Japan (2016)
4. Eray Yildiz, Caglar Tirkaz, H. Bahadir Sahin, Mustafa Tolga Eren, Omer Ozan Sonmez: A Morphology-Aware Network for Morphological Disambiguation. In *Proceedings of AAAI. AAAI Press*, pp. 2863–2869. (2016)
5. Nasser Zalmout, Nizar Habash: Don't Throw Those Morphological Analyzers Away Just Yet: Neural Morphological Disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 704-713. ACL, Copenhagen, 2017.
6. Ilya Sutskever, Oriol Vinyals, Quoc V. Le: Sequence to Sequence Learning with Neural Networks. In *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*. Vol. 2., pp. 3104-3112, Montreal, Canada (2014)
7 Sennrich, Rico, Haddow, Barry and Birch, Alexandra: Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Berlin, Germany.
8 Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio: Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations* (2016)

9 Klein, Guillaume; Kim, Yoon; Deng, Yuntian; Crego, Josep; Senellart, Jean; Rush, Alexander M.: OpenNMT: Open-source Toolkit for Neural Machine Translation. In *Proceedings of ACL* 2017.

10 G. Orosz, A. Novák: PurePos 2.0: a hybrid tool for morphological disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pp. 539–545, Hissar, Bulgaria, 2013. INCOMA Ltd. Shoumen, BULGARIA.

11 G. Orosz, A. Novák: PurePos – an open source morphological disambiguator. In *B. Sharp, M. Zock (eds.): Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*, pp. 53–63, Wroclaw, 2012.

12 D. Csendes, J. Csirik, T. Gyimóthy: The Szeged corpus: a POS tagged and syntactically annotated hungarian natural language corpus. In *Sojka, P., Kopecek, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206*, pp. 41–47. Springer, Heidelberg (2004)

13 Attila Novák; Borbála Siklósi; Csaba Oravecz (2016): A New Integrated Open-source Morphological Analyzer for Hungarian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, pp. 1315–1322.

14 Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, Hans Uszkoreit: TIGER: Linguistic Interpretation of a German Corpus. In Journal of Language and Computation, 2004 (2), 597-620.

15 Váradi T., Simon E., Sass B., Gerőcs M., Mittelholcz I., Novák A., Indig B., Prószéky G., Farkas R., Vincze V.: Az e-magyar digitális nyelvfeldolgozó rendszer. Magyar Számítógépes Nyelvészeti Konferencia (2017)

# *Hol ugat a kutya? Örömében.* Helyhatározói esetragos névszók pontosabb annotációja

Ligeti-Nagy Noémi[1,2], Novák Attila[2,3]

[1]Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar
2087 Piliscsaba, Egyetem u. 1.
[2]MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
[3]Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Kar
1083 Budapest, Práter u. 50/A
ligeti-nagy.noemi, novak.attila@itk.ppke.hu

**Kivonat** Tanulmányunkban ismertetjük a helyhatározói esetragos névszók pontosabb annotációját célzó kutatásunkat, melyet egy szövegekkel kapcsolatban releváns kérdéseket megfogalmazni képes elemzőrendszer igényei motiválnak. A *Hol?*, *Honnan?* és *Hová?* kérdésekre felelő három-három-három esetrag egyikét magán viselő névszók kategorizációja, a mondatban betöltött határozói szerepének pontosabb definiálása elkerülhetetlen a határozókra irányuló megfelelő kérdések megfogalmazásához. Cikkünkben a magyar UD-korpusz alapján 30 kategóriát mutatunk be, melyek megfelelőek ahhoz, hogy a velük annotált névszók határozói szerepe felismerhető és kérdezhető legyen.

## 1. Bevezetés

Jelen tanulmányunkban azt vizsgáljuk, hogyan kategorizálhatóak a névszótövek az alapján, hogy a kilenc hagyományos helyhatározói esetraggal való előfordulásuk során milyen szerepet töltenek be a mondatban, elsősorban szabad határozóként, nem az ige kötelező bővítményeként. Az 1. példában szereplő *bAn* esetragos szóalakról egyértelműen tudjuk, hogy nem egy helyszínt jelöl, hanem valamilyen formát; az *írásban* szóalakra nem a *Hol kérte a vezetőség a különvéleményeket?*, hanem a *Milyen formában kérte a vezetőség a különvéleményeket?* kérdéssel tudunk a legjobban rákérdezni.

(1) A különvéleményeket *írásban* kérte a vezetőség.

Amint azt a példánk magyarázatával már részben szemléltettük, vizsgálatunkat egy olyan elemzőrendszer víziója motiválja, amely képes egy szöveggel kapcsolatos releváns kérdések megfogalmazására [1]. Ennek eléréséhez szükséges egy megfelelően annotált korpusz, amely tanítóanyagként szolgálhat. Ebben olyan annotációnak kell szerepelnie, amely a fentihez hasonló, határozói szerepű névszók (névszói csoportok) pontosabb kategorizálását mutatja.

Ha a kérdésfeltevés és válaszolás irányából közelítjük a kérdést, nyilvánvaló, hogy az esetragok önmagukban nem nyújtanak elegendő információt az adott szóalakra irányuló megfelelő kérdés megfogalmazásához, vagy éppen egy adott kérdésre a megfelelő válasz megtaláláshoz. Ha csupán a szótőre és a morfológiai elemzésre (*Case:Ine*) támaszkodunk, nincs elég információnk ahhoz, hogy eldöntsük, a 2a és a 2b példákban a dőlttel szedett főnévi csoport jó válasz-e egy *Hol* kezdetű kérdésre - vagy megfordítva a dolgot: a főnévi csoportra a *Hol...?* kérdéssel kérdezzünk-e.

(2) a. A Péterfy kórház sürgősségi belgyógyászati és klinikai toxikológiai osztálya több szempontból is érintett lehet *a különleges nappal kapcsolatban.*

b. Országszerte „nagyüzem" várható *a detoxikálókban*, illetve a mérgezési osztályokon.

Szeretnénk tehát olyan tudást kódolni a korpuszba, amiből megtanulható, hogy mi a különbség aközött, hogy *Ubul öltönyben ment dolgozni* vagy *Ubul decemberben ment dolgozni.* Illetve hogy a *főiskolán* szóalak minden gond nélkül lehet helyhatározói vonzata egy igének, de a *főiskolában* szóalak kevésbé - holott mindkettő a *főiskola* szótő és egy helyhatározói esetrag kombinációja.

Ehhez a Universal Dependencies korpusz [2] magyar alkorpuszában (mely a Szeged Treebank függőségi elemzett változatának [3] UD verziója[1]) `OBL` függőségi viszonnyal annotált[2], a kilenc helyhatározói esetrag egyikét magukon viselő szóalakokat kategorizáltuk.

Az természetesen minden, a magyar nyelv esetrendszerével foglalkozó irodalomban világosan megfogalmazásra került, hogy az esetragok és a mondatbeli szerepek között nem áll fenn egyértelmű megfeleltetés (ld. például [4], [5]). A szótövek részletes csoportosításával, szótő és esetrag beható vizsgálatával azonban nem foglalkoztak.

## 2. Módszer

Első lépésként a Szeged Dependency Treebank [3] egy részének UD-sémára átírt változatából válogattuk ki a függőségi elemzésben `OBL` éllel az igéhez kapcsolt szóalakokat, amelyeken a *Hol?*, *Hová?* és *Honnan?* kérdésre válaszoló, a belső (inesszívuszi *bAn*, illatívuszi *bA*, elatívuszi *bÓl*), külső (adesszívuszi *nÁl*, allatívuszi *hOz*, ablatívuszi *tÓl*), illetve felületi (szuperesszívuszi *On*, szublatívuszi *rA*, delatívuszi *rÓl*) helyhatározás paradigmáját alkotó 3-3-3 esetrag egyike található.

Ezután vettük a szavaknak ezt a kilenc csoportját, és mindegyik csoportot először „elő-kategorizáltuk" a word2vec modelleken alapuló szóbeágyazási modell [6] segítségével. A modell klaszterezésre hierarchikus klaszterezést alkalmaz.

[1] https://github.com/UniversalDependencies/UD_Hungarian-Szeged

[2] Ezek egy része szabad határozó, más része vonzat.

Ennek bemenete jelen esetben a csoportosítandó szavakhoz tartozó szemantikai vektor. A klaszterezés részleteit Siklósi és Novák [6]-ban és [7]-ben ismerteti. Mindehhez a vizualizációs felület volt a segítségünkre [8].

Végül a klaszterezéssel kapott, általában 3-8 elemű csoportok listáját manuálisan javítottuk, elsősorban azt a célt tartva szem előtt, hogy az azonos kérdőszóval kérdezhető, tehát azonos típusú határozóként előforduló szavak maradjanak egy csoportban. Fontos hangsúlyozni, hogy ez a csoportosítás ebben az első fázisban nem a szótöveket, hanem a ragozott alakokat érintette.

A feladat megfogalmazható úgy is, hogy határozókat csoportosítunk: vannak természetesen helyhatározók, mint a *sarkon*, vagy a *bankban*, vannak időhatározók, mint a *télen*, *decemberben*. De persze találkozunk időtartam-határozókkal is, mint az *Öt hónapra béreltük a lakást.* mondatban a *hónapra*.

Második lépésként már a szótövekre koncentráltunk. Az eddigi kategorizációt igyekeztünk általánosítani. Ha egy szótő egy adott esetraggal valamilyen határozóként funkcionált, akkor az vajon általános tulajdonsága a szótőnek, vagy csak azzal az egy bizonyos esetraggal együtt jellemző rá? Ha általános tulajdonsága, akkor van olyan esetrag, amivel együtt viszont más határozóként viselkedik?

A manuális kategorizálás végére 1100 szótövet soroltunk be valamilyen alapértelmezett kategoriába, illetve jelöltük meg külön, ha egy esetraggal együtt az alapértelmezettől eltérő viselkedés jellemzi.

## 3. Eredmények

### 3.1. A határozóragos névszók főkategóriái

A 2.táblázatban látható az a 30, illetve az alkategóriákkal együtt összesen 50 kategória, amelyekbe a szótöveket soroltuk. A főkategóriák valamilyen szemantikai kategóriának feleltethetők meg; ezen belül az alkategóriák általában az adott kategórián belüli esetrag-preferenciákat jelzik. A kategóriák részletesen a következők:

- *body*: testrészek nevei; egy részük (*bAn-On*, pl. *fej*) a *bAn* és *On* esetragokkal, illetve ezek irányhármasság szerinti paradigmájának egyéb tagjaival jelölnek *testrészhatározót*, más részük (*any*, pl. *derék*) bármely esetraggal állhat.
- *build=inst*: olyan köznevek, melyek egyszerre jelölnek egy fizikai épületet és valamely intézményt, például *bank*
- *cause*: okhatározók
- *circumst*: körülményhatározók; a megfelelő esetraggal együtt a *Milyen körülmények között?* kérdésre válaszolnak, pl. *hátrány*
- *curr*: pénznemek, pl. *forint*
- *date*: időhatározók; ezek között is van olyan, amely a *bAn* esetragot (és annak paradigmáját) preferálja (*percben*), és olyan, amelyik az *On*-t (*héten*)
- *dem*: a mutató névmások, illetve az *-é* birtokjellel ellátott szavak, melyek a *Melyikben?*, *Melyikbe?* stb. kérdésekre felelnek, pl. *előbbi*
- *direct*: irányhatározók; ezek között is megfigyelhető a belső helyhatározás esetragjainak (*oldalirány*) és a felületi helyhatározás esetragjainak kiegészítő eloszlása (*délnyugat*)

- *event*: eseményhatározók; különlegességük, hogy a megfelelő esetraggal egyszerre fejeznek ki helyet és időt. *Találkoztam a barátommal az előadáson. Mikor találkoztál vele?*, vagy *Hol találkoztál vele?* Alkategóriákra osztható a csoport, esetrag-preferencia alapján. Például *háborúban*, *tüntetésen*.
- *form*: formahatározók; a *Milyen formában?* kérdésre válaszolnak. Például *szóban*, *papíron*.
- *group*: embercsoportok határozója, pl. *család*. Bizonyos esetragokkal *Hol?* kérdésre felel, helyhatározói szerepben; más esetragokkal viszont nem helyhatározó (pl. *családon*).
- *loc*: sok elemű csoport; vegyes szemantikai kategóriájú szavak, melyek helyhatározói szerepben állnak a megfelelő esetragokkal. Erős az esetrag-preferencia. Az alkategóriákat az 1. táblázatban részletesen bemutatjuk.
- *loc = who*: olyan, elsősorban földrajzi jellegű nevek, melyek bizonyos esetragokkal az *org = who* kategória elemeihez hasonlóan a *Kitől?* stb. kérdésekre felelnek. Pl. *EU-tagállam*: *Felkérés érkezett négy EU-tagállamtól.* esetében *Kitől érkezett felkérés?* a releváns kérdés. (Természetesen itt nem szabad határozóról, hanem az ige vonzatáról beszélünk.)
- *material*: anyaghatározók; elsősorban *bÓl* esetraggal kapcsolódva, a *Milyen anyagból?* kérdésre válaszolva
- *meas*: mértékegységek nevei; a *Mennyiben?* stb. kérdésekre zömében a saját módosítójukkal együtt válaszolnak.
- *mode*: módhatározók
- *num* és *num2*: számnevek; a különbség a két csoportban a *rA* és *rÓl* esetraggal kapcsolatos: a *négyre*, *3500-ra* típusú *num* kategóriájuk a *Mennyire?*, míg a *negyedére*, *tizenötszörösére* típusú *num2* kategóriájuk inkább a *Mekkorára?* kérdésre felelnek.
- *org*: szervezetek, cégtípusok, vállalatok. Esetrag-preferenciával: *Hol? A Gazpromnál*, de **a Gazpromban*; viszont *a cégben* és *a cégnél*
- *org = who*: olyan szervezetek, hivatalok, melyek bizonyos esetragokkal *Ki?* formájú kérdésre felelnek. Ezekben az esetekben az ige szerepe is jelentős - ha *Jött egy levél a banktól*, akkor *Kitől jött a levél?*; viszont ha *Elindultam a banktól?*, akkor *Honnan indultál el?*. Ebben a kategóriában is megfigyelhető az esetrag-preferencia.
- *part*: részhatározók; bár releváns ezekre irányuló kérdés a *Hol?*, *Honnan?* stb., mindegyik esetraggal kérdezhetőek a *Melyik részében?*, *Melyik részénél?* stb. kérdésekkel is.
- *period*: időtartam-határozók.
- *place*: tárgyak, fizikai helyek nevei. Jellemzőjük, hogy a *nÁl*, *hOz* és *tÓl* esetragok mndegyiknél helyhatározói funkciójuk érvényesül (tehát a *Hol?*, *Hová?* és *Honnan?* kérdésre felelnek ezekkel), a *bAn* és *On* esetragok (és ezek irányhármasság-beli társai) azonban komplementerei egymásnak. *Árokban*, de *autópályán*.
- *posi*: tisztségek, pozíciók, szerepek nevei. Esetrag-preferenciával. A *nÁl* esetrag, és annak irányhármasság-beli társai egyik alkategóriában sem váltják ki a helyhatározói funkciót.

– *pov*: szemponthatározók - a lemmák a megfelelő esetraggal ellátva a *Milyen szempontból?* kérdésre felelnek.
– *state*: állapothatározók; csak a *bAn*, a *bA* és a *bÓl* esetrag váltja ki a szótő állapothatározói funkcióját.
– *thing*: azoknak a szavaknak a kategóriája, amelyek nem töltenek be speciális határozói szerepet a mondatban, hanem a *Miben?*, *Minél?* stb. kérdésekre válaszolnak (legtöbbször az ige vonzataként).
– *way*: úttal, útvonallal kapcsolatos kifejezések. A belső helyhatározás paradigmájának esetragjai kivételével minden esetraggal betöltik ezt a funkciót.
– *who*: *Kiben?*, *Kinél?* stb. kérdésekre felelő szavak.

### 3.2. A *loc* kategória alkategóriái

Az alkategóriák szemléltetésére a valóban helyhatározást szolgáló, *loc* kategóriába sorolt töveket mutatjuk az 1. táblázatban.

| **kategória** | | **példa** | **esetrag** | | |
|---|---|---|---|---|---|
| **fő** | **al** | | **bAn** | **nÁl** | **On** |
| loc | any | *szekrény* | hol | hol | hol |
| loc | bAn | *állam* | hol | minél | min |
| loc | nÁl | *pék* | miben | hol | min |
| loc | On | *címoldal* | miben | minél | hol |
| loc | bAn-On | *könyv* | hol | minél | hol |
| loc | city-bAn | *Párizs* | hol | hol | melyik városon |
| loc | city-On | *Miskolc* | melyik városban | hol | hol |
| loc | country | *Afganisztán* | hol | hol | melyik országon |

1. táblázat. A meghatározott esetragokkal társulva helyhatározói szerepet betöltő szavak alkategóriái. Az alkategória oszlopa a legtöbb esetben valamely esetrag-preferenciát jelöl: a *pék* csak a *nÁl* esetraggal felel a *Hol?* kérdésre, míg az *állam* a *bAn*-nal. A *city-bAn*, *city-On* és *country* példák szemantikai információk alapján is elkülönítik a csoport tagjait. Az országnevek a korpuszban szereplő esetek mindegyikében a belső helyhatározás és a külső, ponthoz kapcsolódó helyhatározás paradigmájának elemeivel töltenek be helyhatározói funkciót. [4]A városnevek két csoportra oszthatók: az inesszívuszi paradigmát (*Esztergomban*) és a szuperesszívuszi paradigmát (*Szegeden*) preferálókra. Az utóbbi csoportba tartozik a történelmi Magyarország területén található települések többsége, az előbbibe minden egyértelműen külföldi település. Mindkét csoport tagjai tölthetnek be helyhatározói funkciót a külső, ponthoz köthető helyhatározás paradigmájának esetragjaival. A táblázatban szereplő *Melyik városban?*, *Melyik városon?* és *Melyik országon?* kérdést nem úgy kell értelmezni, hogy azok egy helyhatározóra kérdeznek rá (pl. *Melyik városban laksz?*); a *Melyik városban bízol? - Budapestben.* esetre utalnak.

[4] Megjegyzendő, hogy bár a korpuszban nem szerepelnek, vannak olyan országnevek, amelyek helyhatározói szerepükhöz az *On* esetragot igénylik: *Fülöp-szigeteken*, *Izlandon*. Ezek mind szigetek, de nem minden szigetország neve viselkedik így (pl. *Kubában, Írországban*).

## 4. Általánosságok és kivételességek

Bár a 3. fejezetben bemutatott fő- és alkategóriák rendszere nagyon merev osztályozásnak tűnhet, részletesen megnézve az adatokat, azaz az 1100 szótő kategorizációját, számtalan példát találunk kivételekre, az alapértelmezettől eltérő viselkedésre.

Nagyon gyakori természetesen az is, amikor egy adott szótő mind a kilenc esetraggal a *default* kategóriájának megfelelő viselkedést produkálja. Ezek a szótövek tulajdonképpen a kategóriáik prototipikus példányai. Néhány példa ezekre:

- body_any: *derék*
- build_inst: *ügyészség*
- date_bAn: *1987*
- group: *család*
- org_bAn: *cég*
- period: *félév*
- place_On: *célállomás*
- posi_On: *hatalom*
- who: *árus*

Szintén nagyon gyakori, hogy a szó az esetek nagy részében (kilencből 6-7-szer) az alapértelmezett kategóriájánál leírt módon funkcionál a mondatban, de 1-2 esetben ettől eltérő viselkedést produkál. Néhány példa ezekre:

- *alkalom*: alapértelmezett kategóriája az *event_ On*; de a *bÓl* esetraggal együtt okhatározó (*cause*).
- *egy*: alapértelmezett kategóriája a *date*; de bizonyos esetragokkal módhatározó (*egyben* vagy *egyből*).
- *eleje*: alapértelmezett kategóriája a *part*; bizonyos esetragokkal időhatározói szerepben állhat (*elején*, *elejétől*)
- *előadás*: ez egy esemény (*event_ On*), de a *bAn* esetraggal *Mikor?* kérdésre felelő időhatározó (*date*) vagy *Milyen formában?* kérdésre felelő formahatározó (*form*).
- *semmi*: alapértelmezetten a *thing* kategóriába soroltuk. A *semmibe* szóalak azonban a mennyiséghatározók *Mennyibe?* kérdésére is felel; a *mennyiből* szóalak pedig a helyhatározók *Honnan?* kérdésére is.

A fenti pár szó csak néhány példa a sok közül, de jól szemléltetik, hogy számos lemmánál a merev kategorizáció helyett rugalmasan kell eljárni, néhol az alapértelmezett kategória funkcióját teljesen felülírva, néhol csak kiegészítve azt más lehetőségekkel.

Sok esetben pedig nem minősül elégnek a korábban bemutatott 50 kategória: néhol az adott tőre és esetragra specifikus kérdéseket kell definiálni a kategóriák helyett. Néhány példa erre:

- A *többség* szótő a *thing* címkét kapta alapértelmezetten, de bizonyos esetragokkal a *Milyen arányban?*, vagy *Az értintettek mekkora részénél?* jellegű releváns kérdéseket lehet feltenni vele kapcsolatban.

– A *méter*, *kilométer* szavak mértékegységek. Kategóriájuk (*meas*) kérdései megfelelnek minden esetragnál. A *rA* esetraggal együtt azonban a *Milyen messze?*, *Milyen messzire?* kérdésekre felelő határozók is lehetnek. (Valamint természetesen *Hol?* kérdésre felelő helyhatározók is.)

A fentieken kívül meg kell említenünk, hogy néhány kategória elemeire kifejezetten jellemző, hogy nem rájuk, hanem velük kérdezünk, az őket módosító elemre.

– Jellemző ez az eljárás a pénznemek kategóriájára: a *rA* esetraggal együtt például mindnek releváns kérdése a *Hány* <pénznem neve>*-rA?*. A *400 forintra emelkedett a benzin ára.* esetében a *Mennyire?* és a *Hány forintra?* is megfelelő kérdések.
– Hasonló a helyzet a mértékegységek neveinél is. *Hány kilométerre?*, *Hány százalékon?*, *Hány tonnától?* stb.: ezek mind megfelelő kérdések, és ez mindegyik esetragnál valid kérdésfeltevési forma.

Végül nem hallgathatjuk el, hogy van egy olyan kategóriáink, mely nem alapértelmezett címkéje egyetlen szótőnek sem. Ez a *cause*, az okhatározók kategóriája. A korpusz alapján azt találtuk, hogy bár bizonyos szótövek bizonyos esetragokkal okhatározói funkciót töltenek be, alapértelmezett kategóriaként egyetlen szóra sem illik az okhatározói címke. Néhány példa:

– *apropó*: *thing*, de *apropóján*, *apropójából*
– *cél*: *loc_ any*, de *célból*
– *megfontolás*: *thing*, de *megfontolásból*
– *nyomás*: *thing*, de *nyomásra*

## 5. Konklúzió

Tanulmányunkban bemutattuk, hogy egy olyan elemzőrendszert szem előtt tartva, amely szövegekről releváns kérdést képes feltenni, a határozók esetében milyen annotációt tartunk megfelelőnek egy tanítókorpusz létrehozásánál. Elemzésünkben a függőségi elemzett korpusznak azokkal az `OBL` viszonnyal annotált elemeivel foglalkoztunk, amelyek az irányhármasság paradigmáinak valamely esetragját viselik magukon. Ezek nagy része szabad határozói szerepet tölt be, más részük vonzat. 30, az alkategóriákkal együtt összesen 50 kategóriát állapítottunk meg, amelyekbe a korpusz fenti kritériumnak megfelelő szavai besorolhatóak. A legtöbb szótőnél nem elégséges egy alapértelmezett kategória jelölése, bizonyos esetragoknál az alapértelmezettől eltérő funkciót tölthetnek be a szavak, melynek címkézése szintén feladat. A bemutatott kategorizáció megfelelő jegyeket biztosít egy tanítókorpuszban egy az előzőekben említett kérdezőrendszer létrehozásához. Maradtak azonban nyitott kérdések: azokban az esetekben, mikor egy szótő és egy adott esetrag kombinációja több lehetséges határozói funkciót is lefedhet, más támpontok kellenek a biztos kérdés megtalálásához. Az *órára* esetében szükséges a névszó módosítóit is ismernünk a döntésünkhöz: ha *az órára*, akkor

*Hova?*, de ha *néhány órára*, akkor *Mennyi időre?*. Néhány kategória elemeinél tehát elkerülhetetlen az elemek módosítóinak ismerete, és a módosító és a névszó együttesének felcímkézése is. Kutatásunkat ebbe az irányba visszük tovább, kiegészítve mindezt azokkal a határozókkal, melyek nem az eddig vizsgált kilenc esetrag egyikét, hanem más esetragot, például az eszközhatározó esetragját viselik magukon. A következő lépés természetesen az itt bemutatott kategorizációnak és annotációs javaslatnak a mérése, elsősorban a kérdezőrendszer működése során tesztelve és kiértékelve.

| kategória | | példa | esetrag | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| fő | al | | bAn | nÁl | On | bA | hOz | rA | bÓl | tÓl | rÓl |
| 1 body | any | *derék* | hol | hol | hol | hova | hova | hova | honnan | honnan | honnan |
| 2 body | bAn-On | *fej* | hol | | hol | hova | | hova | honnan | | honnan |
| 3 build = inst | bAn | *bank* | hol | hol | | hova | hova | | honnan | kitől / honnan | |
| 4 build = inst | On | *akadémia* | hol | hol | | hova | hova | | kitől / honnan | honnan | |
| 5 cause | | | miért | miért | miért | miért | miért | miért | miért | miért | miért |
| 6 circumst | | *hátrány* | milyen körülmények között | milyen körülmények között | milyen körülmények között | milyen helyzetbe | milyen körülményekhez | milyen helyzetre | milyen helyzetből | milyen helyzettől | milyen helyzetről |
| 7 curr | | *forint* | | mennyinél | hány _On | mennyibe | | mennyire | mennyiből | mennyitől | mennyiről |
| 8 date | bAn | *hónap* | mikor | | | | | mikorra | | mikortól | |
| 9 date | On | *kedd* | | | mikor | | | mikorra | | mikortól | |
| 10 dem | | *előbbi* | melyikben | melyiknél | melyiken | melyikbe | melyikhez | melyikre | melyikből | melyiktől | melyikről |
| 11 direct | bAn | *oldalirány* | merre | | | merre | | | honnan | | |
| 12 direct | On | *délnyugat* | | | hol / merre | | | merre | | | honnan |
| 13 event | bAn | *háború* | mikor / hol | mikor / hol | | hova | hova | mire / mikorra | honnan | honnan / mikortól | |
| 14 event | nÁl-On | *tüntetés* | | mikor / hol | mikor / hol | | hova | hova / mikorra | | honnan / mikortól | honnan |
| 15 event | On | *olimpia* | | | mikor / hol | | | hova / mikorra | hova / mikorra | | |
| 16 form | bAn | *szó* | milyen formában | | | milyen formába | | | milyen formából | | |
| 17 form | On | *papír* | | | milyen formában | | | hova | | | honnan |
| 18 group | | *társaság* | hol | hol | kiken | hova | hova | kikre | honnan | honnan | kikről |
| 19 loc | any | *szekrény* | hol | hol | hol | hova | hova | hova | honnan | honnan | honnan |
| 20 loc | bAn | *szféra* | hol | | | hova | | | honnan | | |
| 21 loc | bAn-On | *dokumentum* | hol | | hol | hova | | hova | honnan | | honnan |
| 22 loc | nÁl | *pék* | | hol | | | hova | | | honnan | |
| 23 loc | On | *címoldal* | | | hol | | | hova | | | honnan |
| 24 loc | city-bAN | *Párizs* | hol | hol | melyik városon | hova | hova / kihez | melyik városra | honnan | honnan / kitől | melyik városról |
| 25 loc | city-On | *Somogyszob* | melyik városban | hol | hol | melyik városba | hova / kihez | hova | melyik városból | honnan / kitől | honnan |
| 26 loc | country | *Svájc* | hol | hol | melyik országon | hova | hova / kihez | hol / melyik országra | honnan | honnan / kitől | melyik országról |
| 27 loc = who | | *Hága* | hol | hol / kinél | kin | hova | hova / kihez | kire | honnan | kitől / honnan | kiről |
| 28 material | | *porcelán* | | | | | | | milyen anyagból | | |
| 29 meas | | *tonna* | mennyiben | mennyinél | mennyin | mennyibe | mennyihez | mennyire | mennyiből | mennyitől | mennyiről |
| 30 mode | bAn | *lábasor* | | | hogyan | | | hogyan | hogyan | | |
| 31 mode | On | *hátfok* | hogyan | | | | | hogyan | hogyan | | |
| 32 num | | *sok* | mennyiben | mennyinél | mennyin | mennyibe | mennyihez | mennyire | mennyiből | mennyitől | mennyiről |
| 33 num2 | | *duplája* | mennyiben | mennyinél | mennyin | mennyibe | mennyihez | mekkorára | mennyiből | mennyitől | mekkoráról |
| 34 org | bAn | *közhivatal* | hol | hol | min | hova | hova | | honnan | honnan | |
| 35 org | nÁl | *leányvállalat* | | hol | | | hova | | | honnan | |
| 36 org = who | any | *kormány* | hol | hol | hol | hova | hova | hova | honnan | honnan / kitől | honnan |
| 37 org = who | bAn | *párt* | hol | hol | kin | hova | hova / kihez | kire / mire | honnan | honnan / kitől | kiről / miről |
| 38 org = who | nÁl | *adóhatóság* | kiben | hol | kin | kibe | hova / kihez | kire | kiből | honnan / kitől | kiről |
| 39 org = who | On | *alkotmánybíróság* | kiben | hol | hol | kibe | hova / kihez | hova | kiből | honnan / kitől | honnan |
| 40 part | | *széle* | hol / melyik részében | hol / melyik részénél | hol / melyik részén | hova / melyik részébe | hova / melyik részéhez | hova / melyik részére | honnan / melyik részéből | honnan / melyik részétől | honnan / melyik részéről |
| 41 period | | *félév* | mikor | | | | | mennyi időre | | mikortól | |
| 42 place | bAn | *intézmény* | hol | hol | | hova | hova | | honnan | honnan | |
| 43 place | On | *kép* | | hol | hol | | hova | hova | | honnan | honnan |
| 44 posi | bAn | *munkakör* | hol | | min | hova | | hova /mire | honnan | | |
| 45 posi | On | *hatalom* | | | hol | | | hova | | | honnan |
| 46 pov | | *szempont* | milyen szempontból | | | | | | milyen szempontból | | |
| 47 state | | *egyensúly* | milyen állapotban | mikor | | milyen állapotba | | | milyen állapotból | | |
| 48 thing | | *együttműködés* | | | | | | | | | |
| 49 way | | *útszakasz* | | hol | hol | | hova | hova | | honnan | honnan |
| 50 who | | *ellenfél* | kiben | kinél | kin | kibe | kihez | kire | kiből | kitől | kiről |

2. táblázat. A helyhatározói esetraggal előforduló tövek lehetséges kategóriái (*kategória - fő*) és alkategóriái (*kategória - al*), illetve az adott kategóriába tartozó szavak adott esetraggal való előfordulásának jelentése kérdőszóval szemléltetve. A táblázatot a következőképpen kell értelmezni: ha az adott szótő a *material* kategóriába tartozik és *bÓl* esetragot visel (*bÓl*), akkor a mondatban a *Milyen anyagból?* kérdésre felelő határozóként szerepel. Az üres cellák azt jelzik, hogy az adott szótő az esetraggal kizárólag vonzatként fordul elő, tehát a *Miben?*, *Min?* stb. kérdésekre felel.

## Köszönetnyilvánítás

Jelen kutatás az FK 125217 számú projekt keretében az FK 17 pályázati program finanszírozásában a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással és az Emberi Erőforrások Minisztériuma ÚNKP-18–3-III-PPKE-26 kódszámú Új Nemzeti Kiválóság Programjának támogatásával valósult meg.

## Hivatkozások

1. Novák, A., Laki, L.J., Novák, B., Dömötör, A., Ligeti-Nagy, N., Kalivoda, Á.: Egy magyar nyelvű kérdezőrendszer. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019), Szeged, SZTE (2019)
2. Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: A multilingual treebank collection. In Chair), N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
3. Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., Csirik, J.: Hungarian dependency treebank. In Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (2010)
4. Antal, L.: A magyar esetrendszer. In Bottyán, G., Kis, Á., eds.: A formális nyelvi elemzés. A magyar esetrendszer. SZAK Kiadó Kft. (2005)
5. Kiefer, F.: A ragozás. In Kiefer, F., ed.: Strukturális magyar nyelvtan 3. Morfológia. Akadémiai Kiadó (2000) 569–618
6. Siklósi, B., Novák, A.: Beágyazási modellek alkalmazása lexikai kategorizációs feladatokra. In Tanács, A., Varga, V., Vincze, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). (2016) 3–14
7. Siklósi, B., Novák, A.: Közeli rokonunk, az autó. In Tanács, A., Varga, V., Vincze, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). (2016) 27–36
8. Novák, A., Siklósi, B., Wenszky, N.: Szóbeágyazási modellek vizualizációjára és böngészésére szolgáló webes felület. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia. (2017) 355–362

# emtsv – Egy formátum mind felett

Indig Balázs[1,2], Sass Bálint[1], Simon Eszter[1],
Mittelholcz Iván[1], Kundráth Péter[1], Vadász Noémi[1]

[1]MTA Nyelvtudományi Intézet,
1068 Budapest, Benczúr u. 33.
[2]ELTE Bölcsészettudományi Kar
1088 Budapest, Múzeum krt. 4.
vezeteknev.keresztnev@nytud.mta.hu

**Kivonat** Az `e-magyar` nyelvfeldolgozó rendszer elkészülése óta több ízben felmerült az igény a hatékonyságának növelésére és használhatóságának egyszerűsítésére, melyek figyelembevételével továbbfejlesztettük a meglévő szövegfeldolgozó rendszert. Célunk a modulok közötti hatékony kommunikáció megvalósítása, valamint az egyes modulok láncba építésének és önálló használatának egyenrangú támogatása. Ezt egy nemzetközi szabványokkal összeegyeztethető, egyszerű, egységes és általános be- és kimeneti formátum használatával valósítjuk meg. Ez terveink szerint hosszú időre jövőállóvá teszi a rendszert, valamint még szélesebbre tárja a külső fejlesztők előtt a kaput, hogy saját moduljaikat a rendszerünkhöz tudják illeszteni, megosztva a meglévő kompetenciákat a magyar nyelv számítógépes feldolgozásának területén. A cikkben bemutatjuk az `e-magyar` új verzióját, az `emtsv` elnevezésű rendszert.
**Kulcsszavak:** e-magyar, emtsv, eszközlánc, erőforrás, tsv, modularitás

## 1. Bevezetés

Az `e-magyar` nyelvfeldolgozó rendszer [1] elkészültekor nem kisebb célt tűzött ki maga elé, mint hogy a magyar nyelv feldolgozásához szükséges state-of-the-art eszközöket integrálva egy egységes, könnyen kezelhető, karbantartott és frissített rendszert alkosson, mely elősegíti a magyar nyelv kutatás- és alkalmazásközpontú feldolgozását egyaránt. Fontos cél volt, hogy a rendszer kutatási célra teljesen nyílt legyen – bátorítva ezzel a későbbi bővítést –, ugyanakkor a laikusok számára is könnyen használhatóvá és jó kísérletező tereppé váljon, mely az elérhető legjobb teljesítményt adja úgy a feldolgozás sebessége, mint a kimenet helyessége tekintetében.

A rendszert közzététele óta jónéhányan letöltötték, és használják a mai napig is. Történtek próbálkozások nagyméretű korpuszok (MNSZ2, Webkorpusz) elemzésére is, aminek következtében korábban ismeretlen hibák és gyengeségek kerültek napvilágra. Ezeket a magyar nyelvtechnológiai közösség közreműködésével javítottuk, illetve a további fejlesztéseknél figyelembe vettük. Jelen cikkben két szempont összefonódásának mentén szeretnénk bemutatni az elvégzett munkát.

Az első a modulok közötti egységes kommunikációs formátum kérdése. Ez az `e-magyar` első verziójában, amiatt, hogy a rendszer integrációja a GATE [2] keretrendszerben valósult meg, adott volt, célszerűnek tűnt a GATE által definiált belső formátumot használni. A felmerült igényekből és az üzemeltetési tapasztalatokból az tükröződött, hogy a felhasználók jelentős része nem ismeri vagy nem kívánja használni munkájához a GATE rendszert: a nyelvi érdeklődésű felhasználóknak kényelmetlen volt, a technikai érdeklődésűeknek pedig szükségtelenül nehézkes. Továbbá a GATE sok esetben inkább megnehezíti az eszközök használatát, az eszközökkel kapcsolatos munkát, mivel az általa bevezetett komplexitás (bonyolult telepítés, nehéz hibakeresés, kényelmetlen formátum, túl nagy erőforrásigény az XML-re alapuló standoff annotáció következtében) sok esetben aláássa a stabilitást, mely szolgáltatáskimaradáshoz is vezethet. Ezért úgy döntöttünk, hogy egy GATE-től független új, egységes formátumot hozunk létre, mely könnyen összeegyeztethető a nemzetközi trendeknek megfelelő szabványokkal. Ezzel megnyílik az út a meglévő eszközök külön-külön modulként történő használatára, az egyes modulok kimenete jobban áttekinthetővé (ezáltal manuálisan könnyebben módosíthatóvá) válik, valamint a rendszerbe könnyebben beépíthetők lesznek a mások által készített különféle – akár nyelvfüggetlen – eszközök. Emellett a GATE-hez való kapcsolódás is megmaradhat megfelelő formátumkonverziós eljárások segítségével.

A második fejlesztési szempont magának az architektúrának az átdolgozása volt, mely az `e-magyar` megalkotása előtt rendelkezésre álló korábbi modulok öröksége felől (nem egységes nyelvi kódok és programnyelvek, nem kellően modularizált és átlátható felépítés) a jelenleg és a jövőben elvárt funkcionalitások (egységesség, felcserélhetőség, összehasonlíthatóság, tanulmányozhatóság) kiszolgálása felé tolja a hangsúlyt.

A cikkben bemutatjuk, hogy az első verzióhoz képest milyen módon alakítottuk át a felhasznált eszközöket abból a célból, hogy a Unixból ismert „eszköztár filozófiának” és az „egy modul egy feladatot végezzen el, de azt tegye jól” elvnek megfelelően az átstrukturált modulok akár egymástól maximálisan függetlenül is, de szükség esetén egymással összekapcsolhatóan és egymással teljesen kompatibilis módon működjenek. Ezzel létrejön az a fontos új lehetőség, hogy a szerelőszalag tetszőleges szakasza lefuttatható, azaz bármely ponton be, illetve ki tudunk lépni, ami magával hozza annak a lehetőségét, hogy az egyes modulok között a felhasználó szabadon rendelkezhet az adattal, akár kézzel is módosíthatja azt, amíg betartja a formátum által támasztott elvárásokat.

A fejlesztés során körültekintően jártunk el, hogy lépést tartsunk más, egy adott nyelvből kiinduló, de többnyelvűnek vagy akár univerzálisnak szánt feldolgozóláncokkal, valamint a megváltozott igényekkel, melyek újabban a installálást és karbantartást nem igénylő, skálázható felhőalapú technológiákat részesítik előnyben, mintegy szolgáltatásként tekintve az feldolgozóláncra. A következő fejezetben az `e-magyar`hoz hasonló, jelenleg elérhető nyelvfeldolgozó rendszereket tekintjük át, hogy összehasonlíthassuk őket rendszerünkkel.

## 2. Háttér

A magyart mint elsődleges célnyelvet tekintve, az `e-magyar`-ral egyedül a *Magyarlánc* [3] hasonlítható össze[1], ami jelenleg a 3.0 verziónál tart. Ez egy Java-alapú, zártan integrált (*tightly coupled*) láncot bocsájt a felhasználók rendelkezésére. A rendszer tanulmányozása közben azt látjuk, hogy a rendszer a legfrissebb nemzetközi state-of-the-art modulokat használja, ugyanakkor nem bővíthető kényelmesen új modulokkal. Arra alkalmas, hogy nagy mennyiségű magyar szöveget részletes nyelvi elemzéssel lássunk el megfelelő minőségben, de a rendszer módosítása, esetleges új modulokkal való kiegészítése nem volt elsődleges prioritás a rendszer fejlesztése közben, így az nehézkes. Alkalmazói felhasználásra kiváló, de továbbfejlesztésre kevésbé alkalmas, mely tulajdonságból fakadóan a felhasználót az eszköz létrehozójához nem egyenrangú kapcsolat köti. A Magyarlánchoz hasonló rendszerekre a nemzetközi színtéren is több példa van, melyek általában ugyanezektől a hiányosságokról szenvednek. Ugyanakkor olyan előnyöket is fel tudnak mutatni, mint a nyelvfüggetlenség (vagy legalábbis sok nyelv támogatása), illetve nagy mennyiségű adat gyors feldolgozása. Ezekkel az aspektusokkal nem szándékozunk versenyre kelni, hanem arra koncentrálunk, hogy a magyar nyelvre a legjobb eredményt a leghatékonyabban állítsuk elő, valamint egy a szabványokhoz közel álló, jól átalakítható formátumot hozzunk létre. A teljes irányítást a felhasználó kezébe szeretnénk adni azáltal, hogy egy nyíltan integrált (*loosely coupled*) rendszert hozunk létre.

Fontos jellemzője a több nyelvet támogató eszközöknek, hogy jellemzően a *Universal Dependencies and Morphology*[2] (UD) nevű nemzetközileg elterjedt, univerzálisnak szánt annotációs sémát használják. Az általános célú egységes annotáció nyilvánvaló előnyei mellett érdemes látni, hogy az ilyen annotáció nem feltétlenül képes egy nyelv morfológiai jelenségeinek teljeskörű leírására. Ezért tartottuk fontosnak, hogy a magyar esetében egy jó minőségű, speciálisan magyar nyelvre kialakított morfológiai elemzőt építsünk be a láncba, az `emMorph`-ot [4]. Az általunk ismert nyelvfüggetlen elemzőláncok közül csak két eszközt emelünk ki példaként, mivel a többi meg nem nevezett alternatíva is ugyanazokkal az ismertetett hátrányokkal bír.

A *UDPipe* [5] C++ nyelven íródott nagyjából az `e-magyar` rendszerrel egy időben, és a célja általános szövegek elemzése a UD annotációs sémáját és formátumát követő tanítóanyag felhasználásával. Bár sok nyelvre van interfésze, valamint valóban hatékonynak mondható, nem teszi lehetővé a könnyű kiterjesztést és fejlesztést, annak ellenére, hogy forráskódja elérhető[3]. Többek között nem ad lehetőséget saját modulok, például szabályalapú morfológiai modul bevezetésére. Hasonló programnak indult a Python-alapú *Spacy*[4], mely eredetileg

[1] Habár az összehasonlítás alapját képező láncok moduljai között van átfedés, az összehasonlításban a lánc egészének felépítésére koncentrálunk, mely független az egyes moduloktól.

[2] `http://universaldependencies.org`

[3] `https://github.com/ufal/udpipe`

[4] `https://spacy.io`

zártan integrált modulokból állt, de a 2.0 verzióval a támogatott nyelvek számának növelése céljából architekturálisan egyre nyíltabbá válik a fejlesztés során[5].

Egy másik stratégiát követ a *WebSty*[6], mely a CLARIN-PL, illetve a *Weblicht*[7], mely a CLARIN-D projekt keretében jött lére. Ezekben az eszközökben ugyanis integrálni próbálják a meglévő – akár nyelvfüggő – eszközöket is, hogy az egyes nyelvek jobban támogatva legyenek. Egyedüli kritérium, hogy a felhasznált eszközök támogassák a UD formátumot. A megközelítés lényege, hogy egy nagy számítógépklaszteren a felhőben futó feladatütemező segítségével az egyes modulok szükség szerint skálázhatóak legyenek a feladatok sorbaállításával. Az egész rendszer egy webalapú API-n keresztül érhető el, melyben feladatokat kell megadni az adatfájl kíséretében, és megvárni, amíg a feladat feldolgozásra kerül. Ebben az esetben a szoftver forráskódja nem érhető el saját példány futtatása céljából, valamint a modulok fejlesztése külső fejlesztőként nem lehetséges.

Az `e-magyar` rendszer új verziójában, az `emtsv`-ben az ismertetett rendszerek fent leírt hátrányait szeretnénk kiküszöbölni úgy, hogy ugyanakkor azok előnyös tulajdonságait is át tudjuk venni. A következő fejezetekben ismertetjük az ezzel kapcsolatban végzett munkát.

## 3. Az egységes adatformátum

Az `e-magyar` rendszer klasszikus felépítése [6] nagyban támaszkodott az eredeti eszközök örökölt felépítésére, így függött azok bemeneti és kimeneti formátumaitól. Az eredeti elképzelés szerint a GATE volt az architektúra azon rétege, mely megteremti a közös, egységes adatformátumot, és így átjárhatóságot biztosít a független, egymásról mit sem tudó modulok között. Az elképzelés egészen addig megfelelő volt, amíg a felhasználó a GATE rendszer ökoszisztémáján belül kívánta használni a rendszert.

A közös formátum a GATE által definiált GATE XML formátum volt. Ez nem egy szabványos és bárki által könnyen implementálható megoldás, mivel nem ismert a formátumot leíró DTD vagy Schema fájl. Ezekre elméletben nincs is szükség, hiszen elméletileg a GATE rendszer a formátum egyetlen előállítója és feldolgozója, azaz gyakorlatilag belső formátumnak tekinthető. A felépítését tekintve standoff annotációként teszi hozzá a bemeneti szöveghez az összes elemzési információt. Ez azzal jár, hogy mivel a szöveg és az annotáció egymástól elkülönülten helyezkedik el egyazon XML-fájlban, folyamatosan ugrálni kell a kettő között – például az XML-feldolgozásból ismert DOM stratégiával fát építve –, ehhez pedig végeredményben a teljes szöveget és annotációt a memóriában kell tartani. Ez a követelmény legjobb esetben is lelassítja a nagyméretű XML-fájlok feldolgozását (a GATE nem erre lett tervezve), mivel lehetetlenné teszi az adatfolyamként való feldolgozást. Emellett a GATE rendszer nehézkes telepíthetősége önmagában is nagyban megnövelte a rendszer komplexitását a felhasználók és

[5] Bár a SpaCy és az `e-magyar` fejlesztésének iránya megegyezik, a jelenlegi állapotuk túl távoli ahhoz, hogy összemérhetőek legyenek.

[6] `http://ws.clarin-pl.eu/websty.shtml`

[7] `https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page`

az üzemeltetők számára is, függetlenül attól, hogy valóban szükségük volt-e a biztosított többletfunkciókra.

Ez motivált minket arra, hogy egy nem XML-alapú, GATE-től függetlenül működő, egyszerű, egységes és könnyen kezelhető formátumot tervezzünk, amely könnyen átalakítható más formátumra. Fontosnak tartottuk, hogy ne zárjunk ki egy potenciális felhasználót sem a formátum miatt. A könnyű konvertálhatóság lehetővé teszi, hogy a nemzetközileg elismert szabványos formátumokra, mint a CoNLL-X [7] vagy a CoNLL-U[8], vagy akár GATE XML formátumra is, át lehessen alakítani a meglévő adatot veszteség nélkül. A könnyű átalakíthatóság ugyanakkor azt is megengedi, hogy a saját igényeinknek megfelelően módosítsuk a formátumot, mivel a definíciója rugalmas, főként ajánlásokat tartalmaz (pl. adatként JSON vagy szabad szöveg), és a lehető legkevesebb megkötést.

```
form     lemma  xpostag
# Ez egy mondat eleji komment
A        a      [/Det|Art.Def]
kutyák   kutya  [/N][Pl][Nom]
ugatnak  ugat   [/V][Prs.NDef.3Pl]
.        .      [Punct]

A        a      [/Det|Art.Def]
...
```

1. ábra: Példa a formátumra. Egy három oszlopot tartalmazó fejléces TSV fájl: szóalak, szótő, egyértelműsített morfológiai elemzés.

Az új formátum (1. ábra) valójában egy fejléccel rendelkező TSV (*tab separated values*) fájl, azaz egy (akár táblázatkezelőbe is betölthető) táblázat sorokkal és oszlopokkal. A klasszikus vertikális formátumnak megfelelően egy sor egy tokent ad meg, az oszlopokban (mezőkben) pedig az adott tokenhez tartozó információk, annotációk kapnak helyet. Az egyszerű TSV-hez képest két kiegészítéssel éltünk. Egyrészt a CoNLL-U formátumhoz hasonlóan a mondathatárok üres sorokkal vannak jelölve. Másrészt lehetőség van arra, hogy az egyes mondatok előtt egy kettőskereszt (`#`) karakterrel kezdődő sorban megjegyzéseket toldjunk be, melyek változatlanul átmásolódnak a kimenetre. Bár a CONLL-U formátum miatt a mondateleji megjegyzést megengedtük, használata a kettőskereszt miatt ellenjavallott. Nagy korpuszban bármilyen karakter előfordulhat, ezért bármely (ritkának vélt) karakter speciális használata hosszú távon hibához vezet: a karakter eredeti korpuszbeli előfordulása és speciális használata összeütközésbe kerül. Ezek gyakran csak jóval később felismerhető, nem várt hibákat eredményeznek, valamint lassítják és korlátozzák a rendszer működését és későbbi bővíthetőségét – a speciális karakter ügyes megválasztásakor is.

Kiemelten fontos a fejléc szerepe, ugyanis ez az, ami a teljes rendszer működését meghatározza. A fejlécben szigorúan definiált oszlopnevek segítségével

[8] `http://universaldependencies.org/format.html`

azonosítják az egyes modulok a feldolgozáshoz szükséges bemeneti adatok helyét (függetlenül az oszlopok sorrendjétől!), és ugyanígy kimeneti adataikat szigorúan definiált nevű új oszlopokba helyezik el, az összes többi oszlopot változatlanul hagyva. Ennek a következménye az a kívánalom, hogy egy modul a bemeneti sorok számát ne változtassa meg. (Ha esetleg olyan modul készül a jövőben, mely megváltoztatja a sorok számát, akkor nagyon körültekintően gondoskodnia kell az új sorok mezőinek tartalmáról, az adatok teljeskörű integritásáról, beleértve a szekvenciális címkézés kezelését is.)

Az újonnan létrehozott oszlopok az ajánlásunk szerint, a jelenlegi implementációban egyszerűen mindig a meglévő oszlopok után kapnak helyet, de ez az oszlopok névvel való azonosításának köszönhetően nem kötelező. A szöveg így emberi szemnek is jól olvasható marad, lokálisan van tárolva az annotáció, valamint az opcionálisan elhelyezhető tetszőleges számú extra oszlop teret ad az igény szerinti bővítésnek és a kiegészítő információknak – akár nagyméretű fájlok esetén is. Az oszlopok elnevezése és tartalma az előállító és feldolgozó programok közötti megegyezésen múlik, és elengedhetetlen, hogy az egymásra épülő modulok között szinkronizálva legyen. A mezők tartalma ajánlottan szabad szöveg vagy a szabványos és kiforrott JSON formátum[9], mely lehetővé teszi kötött struktúrák átadását is, valamint használatával elkerülhető a házi formátumok és extremális karakterek használata.

## 4. Az architektúra

A TSV formátum egyszerűsége miatt jól kezelhető, számtalan eszköz támogatja, egyúttal megadja a szükséges szabadságot a későbbi modulok írásához is. Habár – a felhasználói igényeknek eleget téve – Python nyelven implementáltuk az egyes modulokat összekötő interfészeket, ezek a specifikáció alapján más programozási nyelveken is egyszerűen implementálhatók, akár egymástól függetlenül, heterogén összeállításban is. Elsődleges célunk volt, hogy megkönnyítsük a további modulok egyszerű fejlesztését és bekapcsolását a rendszerbe. Továbbá a hagyományos parancssoros (CLI) és a formátumfüggetlen Python könyvtár (library) interfész mellett egy programozási nyelvektől független, úgynevezett REST API-t is létrehoztunk.

A CLI a hagyományos unixos szerelőszalagok segítségével egy olyan jól használható eszközt ad a haladó nem technikai érdeklődésű és technikai érdeklődésű felhasználók kezébe egyaránt, amely akár nagy adatokon is használható a modulok belső működésének ismerete nélkül. A Python könyvtár a nagyobb programrendszerekbe történő könnyebb integrációt segíti az informatikus/nyelvtechnológus felhasználóknak. A REST API viszont sokkal inkább a modern felhős trendeknek megfelelően az akár teljesen laikus (nem nyelvtechnológus) felhasználók, illetve üzleti körök előtt nyitja meg a rendszer igénybevételének lehetőségét: segítségével telepítés nélkül, azonnal, a felhőben futtatva, jól

[9] Bár a JSON formátumnál a strukturáló elemek közötti térköz választható tabulátornak is, a TSV-be történő beillesztés miatt ezt nem ajánlott megtenni.

skálázható módon szolgáltatásként elérhetővé tehető a rendszer széles igényeket kielégítve, bármilyen programozási nyelven keresztül.

Az egyes modulokat az általunk megalkotott, a TSV mint kommunikációs formátum kezelését általánosan megvalósító, `xtsv` keretrendszer fogja össze. Ez teszi lehetővé a 3. fejezetben leírt formátumon keresztüli kommunikációt (a bemeneti oszlopok kiválasztását, a kimeneti oszlopok hozzáillesztését és az egyéb oszlopok megőrzését), a REST API-k létrehozását és a dinamikus formátumellenőrzést (5. fejezet) is, a modulok konkrét tartalmától függetlenül. Az egyes modulokat néhány deklaratív stílusban megadott paraméter révén illeszthetjük a rendszerbe: szükséges a modul funkcióját megvalósító programegység neve, a kimeneti és a bemeneti oszlopok nevei, valamint az esetleges modellek és egyéb paraméterek megadása. Különböző modellek dinamikus használatához egy adott modulból alternatív példányokat is létrehozhatunk ezen a módon. Az `xtsv` a fentiek szerint dinamikusan létrehozza és futtatja a kívánt láncot.

A fenti tulajdonságok (nyílt integráció, új modulok írásának lehetősége, szabványos TSV formátum, a háromféle API, kis erőforrásigény, jó skálázhatóság, szabályalapú és statisztikai rendszerek kombinálásának lehetősége) együttes fennállásával és a nyílt forráskóddal[10] a 2. fejezetben említett versenytársakhoz képest sokkal szélesebb lehetőségeket (csővezeték, programkönyvtár és REST API-n keresztül elérhető szolgáltatás, mely akár saját felhőben is futtatható, saját modulok vagy akár kézi javítás beillesztése a láncba, tanulmányozhatóság, módosíthatóság, újrataníthatóság, összehasonlíthatóság) biztosítunk a rendszer leendő felhasználói számára.

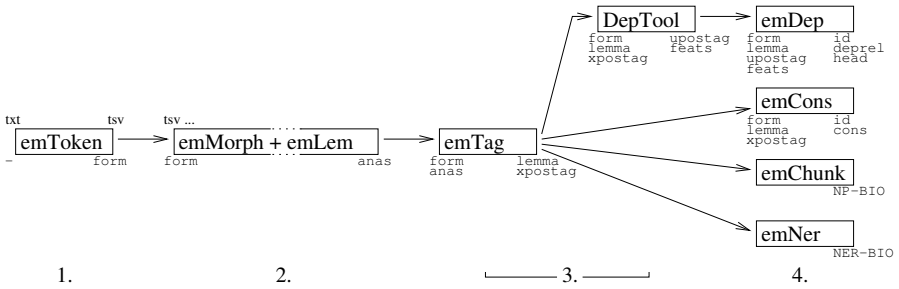
A következő fejezetben ismertetjük az egyes rendelkezésre álló modulok szerepét a láncban, valamint az új modulokkal szemben támasztott minimális elvárásokat, melyek lehetővé teszik a lánc szabad kiterjesztését új modulokkal és a meglévőknek az adott keretek közötti módosításával.

## 5. A modulok

A modulok láncban történő kezeléséhez szükséges, hogy a láncban előrébb levő modulok által felhasznált mezőket gyártó modulok kimenete elérhető legyen a következő modul futtatása előtt. Ennek ellenőrzését a fejléc révén már a szerelőszalag felépítésének idejében meg lehetett oldani, megfelelően korán jelezve az esetleges hibát, akár dinamikusan definiált szerelőszalag esetében is. Ezen megfontolásból úgy alkottuk meg a rendszert, hogy az egyes modulok definíciójának inherens része, hogy milyen oszlopokat igényel, és milyeneket állít elő. Ezenkívül törekedtünk arra, hogy a logikailag különválasztható funkciók külön modulba tartozzanak, akkor is, ha korábban egy modulban egybe voltak építve. Így az egyes modulok feladatköre pontosabban megragadható, és a tesztelésük és fejlesztésük is egyszerűsödik. Az `e-magyar` korábbi verziójából a meglévő modulok felhasználásával jelenleg a 2. ábrán látható láncot definiáljuk. Az `xtsv` általi egységes kezelhetőség miatt (lásd 4. fejezet), a Java-ban írt modulokat egy-egy

[10] A rendszer forráskódja a `https://github.com/dlt-rilmta/emtsv` címen érhető el LGPL 3.0 licenc alatt.

Python nyelvű modulba csomagoltuk, mely minden esetben önálló használatra – Python modulként – is alkalmassá teszi az adott programot. Ezen kiterjesztések nevei egységesen `py` végződést kaptak. A Python interfészekben a Java Pythonon keresztüli meghívásáért egységesen a PyJNIus nevű könyvtár[11] felelős. A kiterjesztett modulok Java natív típusokon keresztül kommunikálnak az eredeti modullal, kiiktatva az eredeti bemeneti és kimeneti formátumok különbségeit, melyeket a láncban az `xtsv` hivatott elfedni. A következőkben az egyes modulokat érintő fentieken túli változásokat ismertetjük.




2. ábra: Az `emtsv` jelenlegi feldolgozó lánca, a bemeneti és kimeneti mezőkkel. A definiált mezők alapján a lánc összeállításának idejében tudható például, hogy a POS taggeléshez kell a `form` és az `anas` oszlop (megfelelő formátumban), vagy hogy a dependenciaelemzést meg kell előznie a POS taggelésnek, a chunkolásnak viszont nem, ahogy ez az ábrán is látszik.

### 5.1. `emToken`

Az új eszközlánchoz – bár maguk a tokenizálási szabályok maradtak a régiek – jelentősen át kellett dolgoznunk a tokenizálót is. Az `emToken`-t [8] alkotó modulok eddig egy bináris fájlra fordultak, ami mindent tartalmazott, ami a futtatásához szükséges. Az új verzióban az egyes modulok külön-külön futtatható binárisokra fordulnak, ezek a szabványos bemenetről olvasnak, a szabványos kimenetre írnak, és egy Python program köti össze őket. Az új struktúrához át kellett írni az `emToken`-hez használt tesztelési rendszert is, ugyanakkor ez lehetővé tette a szerves integrációt az `emtsv` keretrendszerrel.

[11] `https://github.com/kivy/pyjnius`

### 5.2. emMorph és emLem

Az emMorph-ot, valamint az emMorph és emLem együttműködését érintő bizonyos hibákat javítottuk, mások megoldása folyamatban van. A morfológia belső formátumnak tekinthető kimenetét (a kétszintű morfológia felszíni alak–mély alak párjait) nem használjuk fel közvetlenül, hanem elemzéssel ellátott morfémasorozattá alakítjuk, valamint a morfémákból a lemmát is meghatározzuk. E két feladatot végzi az emLem modul, melyhez az eddigi Java implementáció helyett egy új, Pythonban írt változatot[12] készítettünk, amely egyszerűségre és a kód átláthatóságára törekszik.

A modult kiegészítettük egy speciális, saját REST API-val, melynek segítségével a felhasználó egy adott szó elemzéséhez egyszerűen a böngészőből, a szónak egy speciális URL-be történő beírása után férhet hozzá. A https://emmorph.herokuapp.com/dstem/terem címen a felhőben található demó segítségével bárki könnyen meg tudja nézni bármely magyar szó – esetünkben a *terem* – morfológiai elemzéseit az emMorph szerinti kódrendszerben, lemmával együtt.

```
{
  "terem": [
    {
      "lemma": "terem",
      "morphana": "terem[/N]=terem+[Nom]=",
      "readable": "terem[/N] + [Nom]",
      "tag": "[/N][Nom]",
      "twolevel": "t:t e:e r:r e:e m:m :[/N] :[Nom]"
    },
    ...
  ]
}
```

3. ábra: Példa a morfológiai elemző és lemmatizáló JSON kimenetére. A *terem* szó lehetséges elemzései: főnév, ige, valamint a *tér* birtokos személyragos alakja.

A modul kimenete mindkét esetben egy speciálisan formázott JSON (ld. 3. ábra), mely emberi és gépi felhasználásra egyaránt alkalmas. Minden elemzés négy mezőt tartalmaz, rendre: a token szótöve ("lemma"), a morfémákra bontott alak először géppel olvasható ("morphana"), majd ember által is olvasható formátumban ("readable"), a puszta címke morfológiai szegmentumok nélkül ("tag"), végül az emMorph kétszintes kimenete hibakeresés céljából ("twolevel"). A REST API egyszerre több szó elemzését is képes visszaadni, amennyiben HTTP POST metódussal hívjuk a dokumentációban megadott feltételeknek megfelelően. Az új JSON formátum előnye, hogy szabványossága és kiforrottsága révén véd a nagy korpuszokban előforduló, nem várt karakterek okozta hibáktól is. A TSV-be illeszthetőség kedvéért a JSON-ban tilos a sztringen kívüli tabulátor használata.

[12] https://github.com/ppke-nlpg/emmorphpy

### 5.3. emTag

A PurePOS [9] hagyományos, nem szabványos formátumához (ld. a PurePOS dokumentációjában[13]) képest a 3. fejezetben ismertetett új formátum nagy előrelépést jelent. A nagy korpuszokban előforduló, nem várt karakterek okozta hibák így kiküszöbölhetőkké váltak.

A fejlesztésünknek köszönhetően most már natív Java-adatszerkezetként is megadhatók az alternatív elemzések a Java nyelven írt PurePOS számára (akár már Java programból is), így függetlenítve azt az adat mindenkori formátumától. A PurePOS Python interfésze tartalmazza az emtsv-hez szükséges kiegészítéseket. A Python interfész segítségével a PurePOS használható önmagában előelemzett bemenettel, vagy csak a beépített statisztikai morfológiai elemzővel, illetve az emMorph+emLem szabályalapú morfológiát mintegy belső morfológiaként használva.

### 5.4. emChunk és emNER

A modulok alapjául szolgáló HunTag3 [10] konfigurációját átalakítottuk, hogy megfeleljen az e-magyar új formátuma által támasztott követelményeknek: az egyes jellemzőket mostantól nem oszlopsorszám, hanem név alapján éri el a program. Ezenkívül számos belső átalakításon esett át, melynek következtében a be- és kimeneti formátumok kezelése teljesen külön lett választva a program többi részétől, valamint egységesítve lett. Mivel a HunTag3 maga is Python nyelven íródott, ezért a különválasztott és átdolgozott bemenetiformátum-kezelés szolgált elsősorban az xtsv keretrendszer alapjául.

### 5.5. emMorph2UD

A Magyarlánc 3.0-ról leválasztottuk a DepTool modult, mely az emDep függőségi elemző számára konvertálja át az emMorph által kiadott és az emTag által egyértelműsített morfoszintaktikai információkat jegy–érték párok linearizált sorára. Az emDep eddigi modellje is olyan tanítóanyag alapján készült, amelyhez a DepTool konvertálta a szófajcímkéket és a morfoszintaktikai jegyeket. A DepTool-t közelebbről megvizsgálva azonban kiderült, hogy bizonyos morfológiai jegyeket nem kezel, a bemeneti emMorph címkék tartalma sok esetben elvész. A DepTool kimeneteként előálló címkék ráadásul olyan formátummal rendelkeznek, amely csak az e-magyar-on belül, két modul között hasznosítható. Ezzel szemben mi egy teljesebb konverziót szerettünk volna elérni egy olyan formátumra, amely a két modul közötti átjárhatóság mellett önálló kimeneti annotációként is használható.

Mivel az emDep modulhoz rendelkezésre állt egy másik modell, amely a Szeged Treebank UD morfológiai címkéivel ellátott tanítóanyag alapján készült, ezért úgy döntöttünk, hogy lecseréljük az emDep modelljét erre a verzióra. Az új konverter, az emMorph2UD az e-magyar elemzőlánc jelenlegi emtsv változatában egyrészt egy közbülső láncszemként az emMorph kimenetét konvertálja az emDep

[13] https://github.com/ppke-nlpg/purepos

modul számára fogyasztható jegy–érték struktúrájú UD címkékre, másrészt pedig kimeneti formalizmusként lehetővé teszi, hogy az `e-magyar` elemzőláncot használók az eddig elérhető `emMorph` kimenet mellett UD morfológiai címkéket is kaphassanak, amely egy nemzetközileg elterjedt, univerzális annotációs séma szabályait követi [11]. A konverter részletesebb ismertetését és kiértékelését lásd: [12].

### 5.6. `emDep` és `emCons`

A Magyarlánc 3.0-ról leválasztottuk a függőségi elemzést megvalósító Bohnet parsert [13] és az összetevős elemzést megvalósító Berkeley parsert [14], melyek így – megtisztítva azoktól a részektől, amelyeket a Magyarlánc használ, de az `e-magyar`-nak nem szükségesek – kisebb erőforráslábnyommal képesek működni. Az `emDep` modelljét lecseréltük (ld. 5.5. fejezet), így a modul bemenete a UD annotációs sémának megfelelő, az `emMorph` kimenetéből konvertált szófajcímke és morfológiai jegy–érték párok sorozata. A modulok kimenete, vagyis a szintaktikai annotáció nem változott.

## 6. Összefoglalás

A cikkben ismertettük az `e-magyar` nyelvfeldolgozó rendszer megújult és jelentős átalakításon átesett új verzióját, az `emtsv`-t. A rendszer immár nem csak felveszi a versenyt a szabadon elérhető versenytársaival, hanem több ponton meg is haladja azok képességét: szabványos kommunikációs formátumot használ, CLI, Python könyvtár és REST API hozzáféréssel bír, forráskódja elérhető, nyíltan integrált (*loosely coupled*), új modulokra nyitott (legyenek azok szabályalapúak vagy statisztikaiak), kis erőforrásigényű, és jól skálázható. Lehetőséget ad a REST API révén szolgáltatás üzemeltetésére, a CLI által csővezeték készítésére és a Python könyvtár API által nagyobb programrendszerekbe történő beépítésére, a nyílt forráskód miatt pedig mindez akár saját gépen, saját modulok fejlesztésével és beillesztésével is megvalósítható. A rendszer moduláris, továbbépíthető, összevethető, átírható, újrataníthatóó, tanulmányozható, módosítható. Ezzel a magyar nyelvre a funkciókban leggazdagabb, szabadon elérhető elemzőláncot hoztuk létre, mely leváltja a GATE-be integrált eredeti `e-magyar`-t.

Az új rendszerben rejlő valódi potenciál viszont csak akkor lesz teljes egészében kiaknázható, ha a modellek felépítéséhez használt, kézzel annotált korpuszok is szabadon elérhetőek lesznek, hogy az eszközök és az adat változtatásával, cseréjével mindenki szabadon kísérletezhessen új módszerek kifejlesztésével. További hiányzó elem egy olyan szabadon elérhető, kellően nagy méretű, magyar nyelvű korpusz, mely az `e-magyar` eszközkészlettel van leelemezve. Ez jó lehetőséget biztosíthat majd a rendszer részletes tesztelésére, vizsgálatára, a hibák feltárására, más kutatásokban való alkalmazására és végső soron a korpusz nyelvi adatainak elemzése révén új elméletek megszületésére.

## Hivatkozások


1. Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: Az `e-magyar` digitális nyelvfeldolgozó rendszer. In Vincze, V., ed.: XIII. Magyar Számítógépes Nyelvészeti Konferencia. (2017) 49–60
2. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6). GATE (April 15, 2011) (2011)
3. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of RANLP. (2013) 763–771
4. Novák, A., Siklósi, B., Oravecz, Cs.: A New Integrated Open-source Morphological Analyzer for Hungarian. In et al., N.C., ed.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
5. Straka, M., Straková, J.: Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Vancouver, Canada, Association for Computational Linguistics (2017) 88–99
6. Sass, B., Miháltz, M., Kundráth, P.: Az `e-magyar` rendszer GATE környezetbe integrált magyar szövegfeldolgozó eszközlánca. In Vincze, V., ed.: XIII. Magyar Számítógépes Nyelvészeti Konferencia. (2017) 79–90
7. Buchholz, S., Marsi, E.: CoNLL-X Shared Task on Multilingual Dependency Parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), New York City, Association for Computational Linguistics (2006) 149–164
8. Mittelholcz, I.: emToken: Unicode-képes tokenizáló magyar nyelvre. In Vincze, V., ed.: MSZNY 2017. (2017) 61–69
9. Orosz, Gy., Novák, A.: PurePos 2.0: a Hybrid Tool for Morphological Disambiguation. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, Hissar, Bulgaria, INCOMA Ltd. Shoumen, BULGARIA (2013) 539–545
10. Endrédy, I., Indig, B.: HunTag3: a General-purpose, Modular Sequential Tagger – Chunking Phrases in English and Maximal NPs and NER for Hungarian. In: 7th Language & Technology Conference, Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC '15), Poznań, Poland, Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu (2015) 213–218
11. Vincze, V., Farkas, R., Simkó, K.I., Szántó, Zs., Varga, V.: Univerzális dependencia és morfológia magyar nyelvre. In Tanács, A., Viktor, V., Veronika, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2016) 322–329
12. Vadász, N., Simon, E.: Konverterek magyar morfológiai címkekészletek között (2019) Jelen kötetben.
13. Bohnet, B., Nivre, J.: A Transition-based System for Joint Part-of-speech Tagging and Labeled Non-projective Dependency Parsing. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. EMNLP-CoNLL '12, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 1455–1465

14. Durrett, G., Klein, D.: Neural CRF Parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics (2015) 302–312

# The impact of inflection on word vectors

Dániel Lévai, András Kornai

Institute for Computer Science and Control, Hungarian Academy of Sciences
Research Group for Human Language Technologies
{levai, kornai}@ilab.sztaki.hu

**Abstract.** We present a method to evaluate the similarity of word vector clusters, and use it to determine the coherence (self-similarity) and relatedness of morphologically defined clusters
**Keywords:** word vector, cluster similarity, morphology, skip-gram

## 1 Introduction

Word vectors encode not just semantic relations [1], but also morphological ones, as in $\overrightarrow{\text{goes}} - \overrightarrow{\text{go}} + \overrightarrow{\text{see}} = \overrightarrow{\text{sees}}$. In agglutinative languages it is common to treat inflectionally related tokens as separate types (form-based, rather than stem-based modeling). Our main aim is to show that the tokens considered unrelated by the form-based model are indeed related on a morphological level. Furthermore, the more specific case endings (delative, translative, ...) dominate the word vector as opposed to the less specific case endings (nominative, accusative, ...) where the word vectors contain richer semantic relations.

## 2 Methods

The idea of neural networks dates back to the 1940s [2], when McCulloch created a computational network. The main idea is that our brain is composed of neurons (nodes) and synapses (edges). We learn and memorize by creating and strengthening synapses, and an artificial neural network – by analogy – should learn by strengthening and weakening weights on the edges based on the sample it receives. Constructing and training a neural network is a difficult task, because we do not have a strong idea how to interpret the weights of the edges or the nodes themselves – a neural network is a black box, and we do not always know how the architecture of the network should look like, or how we should train a network. The architecture in a skip-gram model [3] [4] consists only of a single hidden layer and an output layer with hierarchical softmax classifier. The task of the model is, for every word in the vocabulary, to learn the probabilities of every other word being in the context of the vocabulary word.

The input is a one-hot vector representing the word, and the output is a probability vector. As we can see in Fig. 1, there are separate weights for each coordinate, and the number of nodes in the hidden layer defines the number of

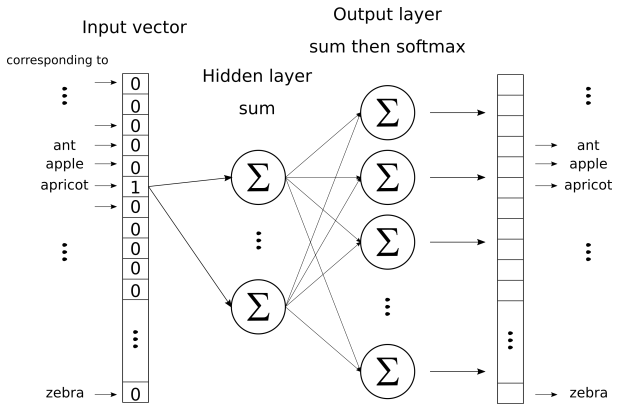



Fig. 1: Network architecture

weights. If the hidden layer counts 200 neurons, then we will have 200 weights for each coordinate, thus for every word. To summarize, we create a model to predict contexts only to learn the input weights to be used as vectors. The same way, the model also learns the output weights, and the classification problem reduces to a matrix dot product and to a softmax classification problem. For adjusting the weights, the model uses backpropagation. The main differences from the traditional neural networks are the subsampling, negative sampling and the use of skip-grams with negative sampling [5].

We use the most noise-reduced tier of the Hungarian Webcorpus[1] [6,7] that has the duplicates, foreign language pages, and script-generated text (such as dates, headlines, tables of content) removed, leaving 710m word and punctuation tokens. For morphological analysis, we used the `emMorph module`[2] [8] of `e-magyar` [9]. To establish the clusters we trimmed the analyses until the last stem, since the derivation does not concern us in this paper, and we used the `<>` sign concatenating the analyses returned by `emMorph`. The following tables show some sample lines before and after the normalization.

| | |
|---|---|
| elméleti | `elmélet[/N]i[_Adjz:i/Adj][Nom]` |
| elméleti | `elméleti[/Adj][Nom]` |
| számítógépes | `számít[/V]ó[_ImpfPtcp/Adj]gép[/N]es[_Nz:s/N][Nom]` |
| számítógépes | `számítógép[/N]es[_Adjz:s/Adj][Nom]` |
| számítógépes | `számítógép[/N]es[_Nz:s/N][Nom]` |

These two words become *elméleti* – `[/Adj][Nom]` 'theoretical', *számítógépes* – `[/Adj][Nom]<>[/N][Nom]` 'computational, computer-related' after the normalization.

We used `gensim` [10] skip-gram with negative sampling with the default hyperparameters in the creation of our models: the dimension of the word embed-

[1] http://mokk.bme.hu/resources/webcorpus/
[2] https://github.com/dlt-rilmta/emMorph

ding is 200, the window used is 5 words in both directions, 5 training epochs. Negative sampling is set to a factor of 5, and minimal sample size is 5. The model was generated using the surface forms only and morphological analyses were assigned to the words subsequently. In the resulting embedding we can observe (Fig. 2) a strong correlation ($\rho = 0.939$) between the log-frequency of the words and the length of their vectors posited in [11]. Knowing this, we can project the vectors to the surface of the unit sphere without great loss of information. Later in this paper we will use a set of 200000 uniformly distributed random vectors as a baseline for comparing to the actual word vectors projected onto the surface of a unit-radius 200-ball.

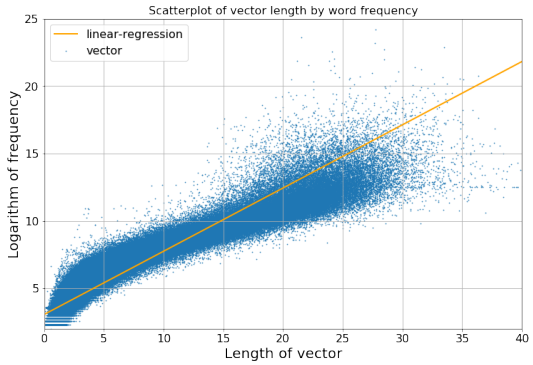



Fig. 2: Length versus $\log_2$(frequency)

Another characteristic of the skip-gram model is that it prefers placing the words in a specific part of $n$-dimensional space [12]. One technique we used to measure the spatial preference of the model is to count the relative frequency of each coordinate being positive, then plotting these numbers in an ascending order. Fig. 3 shows that some coordinates are highly likely to be positive and others negative, whereas for a random set of points the line would be flat since every coordinate would have 0.5 probability to be positive or negative.

## 3 The statistics of grammatically defined clusters

We hypothesize that there is a coherent structure in the embedding and each vector encodes a certain meaning and grammatical structure. The clustering methodology we will be using here is a viable approach to classify word vectors to the extent that we can analyze these clusters in a way that helps understanding them. The model used has no a priori knowledge of these grammatical categories, yet we will see that the clusters are indeed coherent.

Visualizing high-dimensional data is a difficult exercise [13]. We can use principal component analysis to maximize the information retained in the first few

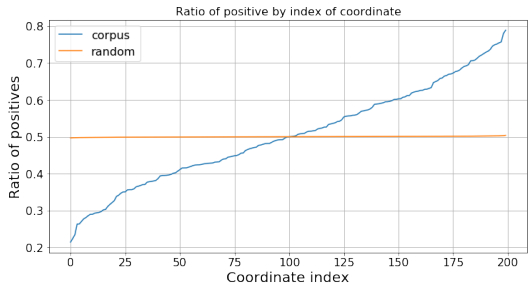



Fig. 3: Probability of a certain coordinate being positive

dimensions. Plotting a sample of 1000 vectors from the spherical projection of the first 3 principal components of some clusters of word vectors yielded Fig. 4, which makes clear we have 3 clusters each restricted to a dominant orthant. What we need to verify is that this phenomenon persists in the whole 200-dimensional space. One way of doing this is by comparing the standard deviations and the entropy of the clusters. If a cluster's standard deviation is high, it indicates low density, the lack of a core, and incoherent structure. If the standard deviation is lower, it indicates a higher density, a more characteristic core. Number of occurrences and entropy ($y$ axis) are plotted against the standard deviation ($x$ axis) in Fig. 5.

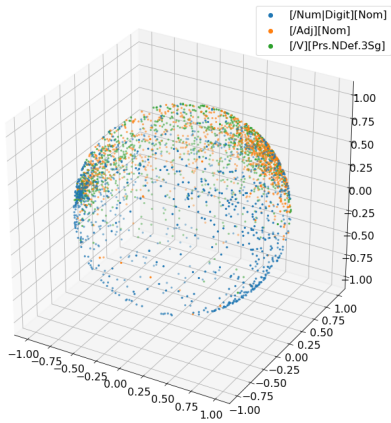



Fig. 4: Clusters on the unit sphere

On the left panel we can see a square-like shape, showing weak correlation between the frequency and the standard deviation. The scatter plot of the entropy-standard deviation shows that higher entropy generally means higher standard

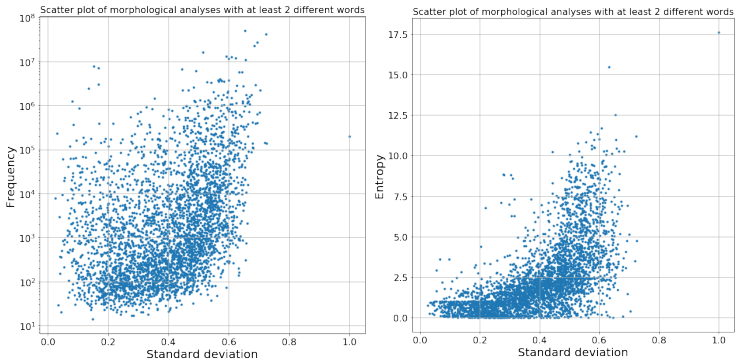



Fig. 5: Scatter plots of clusters

deviation. After filtering out morphological analyses with low number of words, first 5, then 50, Fig. 6 the correlations weaken. The Spearman correlation coefficients for the 4 figures are: 0.512, 0.957, 0.384, 0.057.

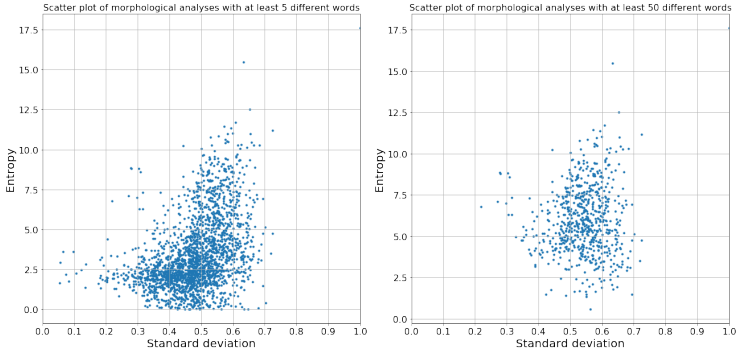



Fig. 6: Scatter plots of clusters

## 4 Quantifying similarity

Here we define and explain the intuition behind a similarity measure between sets of vectors on the $n$-sphere. Since it is hard to have intuition in 200-dimensional space, we begin with the definition of a *cap* (vectors at a small angle to an axis):

**Definition 1.** *$n$-cap*
*Let $\boldsymbol{m} \in S^n$, let $\alpha \in [-\pi, \pi]$. The cap defined by $\boldsymbol{m}$, $\alpha$ is*

$$cap_\alpha(\boldsymbol{m}) = \{\boldsymbol{x} | \boldsymbol{x} \in S^n \wedge \langle \boldsymbol{m}, \boldsymbol{x} \rangle \geq \cos(\alpha)\}$$

*which is equivalent to*

$$cap_\alpha(\boldsymbol{m}) = \{\boldsymbol{x} | \boldsymbol{x} \in S^n \wedge \mathrm{sim}_{cos}(\boldsymbol{m}, \boldsymbol{x}) \geq \cos(\alpha)\}$$

and a theorem about the surface of the $n$-sphere [14]:

**Theorem 1.** *Let $I_x(a, b)$ be the regularized incomplete beta function, and $A_n = 2\pi^{n/2}/\Gamma(\frac{n}{2})$ be the surface area of the $r$-radius $n$-sphere. Then the area of the spherical cap characterized by its $h$ height is:*

$$A = \frac{1}{2} A_n r^{n-1} I_{(2rh-h^2)/r^2}\left(\frac{n-1}{2}, \frac{1}{2}\right) \tag{1}$$

The theorem and the definition together show the ratio of the surface of the cap to the n-sphere. For example, putting $n = 200, h_1 = \cos\left(\frac{11\pi}{24}\right), h_2 = \cos\left(\frac{10\pi}{24}\right)$ into the theorem above we get 0.0327 and $10^{-4}$ respectively. We can verify this upper bound by placing uniformly random points on the surface of the $n$-sphere, counting the points inside $\mathrm{cap}_\alpha$ (we performed simulations with 200k random vectors) and compare the ratio given by theorem 1 to the random sample. To measure the compactness of clusters, we use an increasing cap around the cluster centroid, and plot the ratio of word vectors lying in the cap as a function of the minimal similarity of words to the cluster centroid. As we

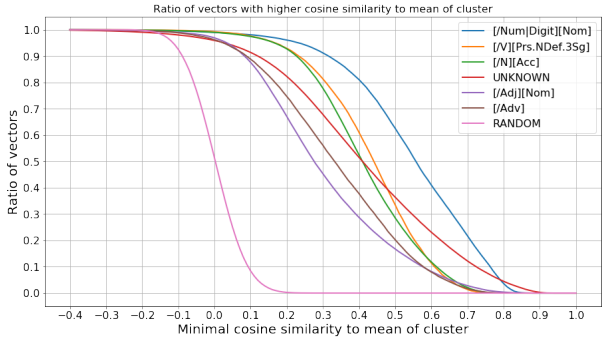



Fig. 7: Ratio of points in a $\mathrm{cap}_\alpha(\mathrm{mean}_{\mathrm{cluster}})$

can see on Fig. 7, the `RANDOM` cap vanishes around $\cos(\alpha) = 0.2$ (for this $\alpha$, theorem 1 limits the relative surface of the cap to 0.0023), while the other clusters, most notably the `[/Num|Digit][Nom]` (digit in nominative case) shows the strongest coherence, which seems intuitive, as the numbers mostly indicate quantity and amount (counterexamples are dates, or symbolical numbers like 7, 3, 24/7). The `UNKNOWN` cluster shows high coherence, as it is dominated by nouns. The `[/V][Prs.NDef.3Sg]` cluster (third person singular verbs) show the same coherence as the `[/N][Acc]` cluster (accusative nouns), while the `[/Adj][Nom]`

cluster (adjectives in noun case) shows lower coherence than any of the clusters other than the `RANDOM` presented on the figure.

Since we want to filter out noise, and our ultimate goal is to measure similarity, we can use the ratio of the words in a $\text{cap}_\alpha$ with fixed $\alpha$ to measure self-similarity, and we can also calculate the ratio of some words in other clusters' cap. That way, we obtain an asymmetrical similarity measure. Obtaining the fixed $\alpha$ is based on filtering out the most noise. We use `RANDOM` as a base of comparison: on Fig. 8, we show the ratios with that corresponding to `RANDOM` subtracted. The plot shows the maximal difference to be around $\cos(\alpha) = 0.13$, so we have chosen $\alpha = \frac{11\pi}{24}$ (82.5°) (for which $\cos(\alpha) \approx 0.1305$). Thus the formal

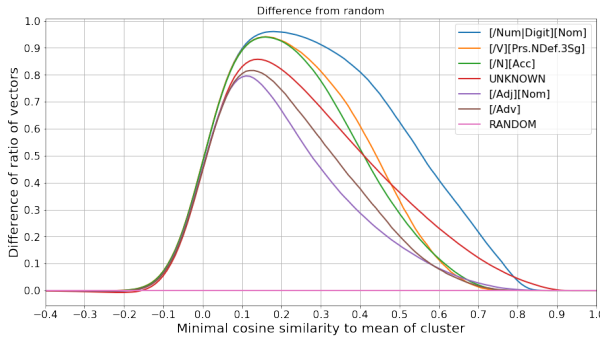



Fig. 8: Difference of the ratios from `RANDOM`

definition of the cluster similarity defined above:

**Definition 2.** *Cluster similarity*
*Let $C_1$, $C_2$ be two sets of points on the n-sphere, $\boldsymbol{m}$ be the mean vector of $C_1$. The (signed) similarity of $C_1$ and $C_2$ is:*

$$\text{sim}_{cl}(C_1, C_2) = \frac{\left|\left\{\boldsymbol{x} \in C_2 | \text{sim}_{\cos}(\boldsymbol{m}, \boldsymbol{x}) \geq \cos(\frac{11\pi}{24})\right\}\right|}{|C_2|}$$

*where $|.|$ is the cardinality of a set.*

## 5 The role of affix frequency

We begin by examining the clusters based on their case endings to see whether some specific case endings contribute significantly more to cluster coherence. Table 1. summarizes the clusters and their respective self-similarities. We can see that the more specific case endings like `[/Adj][Transl]` and `[/Adj][Temp]` (translative and temporal case for adjectives) show higher self-similarity, while the more general ones like `[/Adj][Nom]` and `[/Adj][Supe]` (nominative and superessive) show lower self-similarity. This tendency continues with the cases

affixed to nouns, where [/N][All] and [/N][Transl] (allative and translative) are among the highest self-similarity cases and [/N][Nom] has one of the lowest self-similarity from the paradigm. Let us now consider clusters that are more

| Cluster | Sim | Cluster | Sim | Cluster | Sim |
|---|---|---|---|---|---|
| [/Adj][Nom] | 0.822 | [/N][EssFor:képp] | 0.889 | [/Num][Nom] | 0.908 |
| [/Adj][Supe] | 0.910 | [/N][Nom] | 0.922 | [/Num][Del] | 0.955 |
| [/Adj][Subl] | 0.924 | [/N][Ess] | 0.926 | [/Num][Dat] | 0.957 |
| [/Adj][Ine] | 0.929 | [/N][EssFor:ként] | 0.936 | [/Num][Ter] | 0.960 |
| [/Adj][Ela] | 0.936 | [/N][Ine] | 0.937 | [/Num][Cau] | 0.971 |
| [/Adj][Acc] | 0.941 | [/N][EssFor:képpen] | 0.941 | [/Num][Ill] | 0.977 |
| [/Adj][Ade] | 0.945 | [/N][Cau] | 0.946 | [/Num][All] | 0.978 |
| [/Adj][Ins] | 0.951 | [/N][Ade] | 0.949 | [/Num][Ine] | 0.980 |
| [/Adj][Abl] | 0.959 | [/N][Hyph:Hyph] | 0.957 | [/Num][Acc] | 0.983 |
| [/Adj][Ill] | 0.960 | [/N][Ter] | 0.962 | [/Num][Subl] | 0.984 |
| [/Adj][Cau] | 0.961 | [/N][Supe] | 0.962 | [/Num][Ela] | 0.985 |
| [/Adj][Del] | 0.961 | [/N][Abl] | 0.964 | [/Num][Ade] | 0.988 |
| [/Adj][Ter] | 0.963 | [/N][Acc] | 0.966 | [/Num][Ins] | 0.992 |
| [/Adj][Dat] | 0.967 | [/N][Temp] | 0.966 | [/Num][Abl] | 1.000 |
| [/Adj][All] | 0.978 | [/N][Ela] | 0.968 | [/Num][Transl] | 1.000 |
| [/Adj][Transl] | 0.994 | [/N][Del] | 0.969 | [/Num][Temp] | 1.000 |
| | | [/N][Ill] | 0.969 | [/Num][Supe] | 1.000 |
| | | [/N][Dat] | 0.969 | | |
| | | [/N][Subl] | 0.969 | | |
| | | [/N][Ins] | 0.972 | | |
| | | [/N][Transl] | 0.979 | | |
| | | [/N][All] | 0.979 | | |

Table 1: Clustering by case ending

frequent or have higher entropy. We partitioned the clusters into 20 equal bins, each 0.05 wide, by their respective standard deviation, then calculated the mean and the standard deviation ($\sigma$) of the vectors of each cluster, see Fig. 9.

Most clusters lay in the $1\sigma$ stripe, and the $2\sigma$ stripe is also rather populated. Each stripe is monotonically increasing. The interesting clusters are the ones above the $2\sigma$ stripe, because compared to their high entropy their variance is smaller than expected. Summarizing on Table 2 the biggest non-ambiguous clusters (counting more than 5000 words) outside the $2\sigma$ line on Fig. 9, it shows us that nouns, be they plural or singular, form highly coherent clusters. The presence of infinitive and plural third person verbs among these most coherent clusters is very interesting, because verbs in general did not show strong coherence.

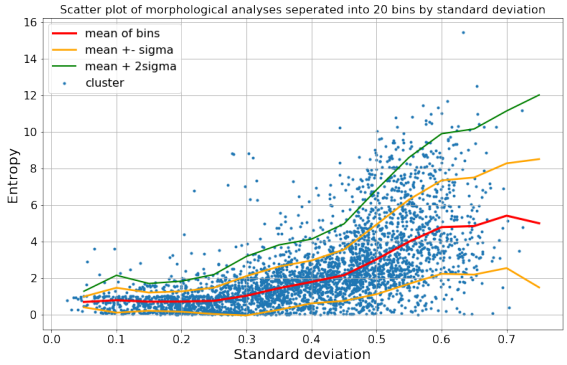



Fig. 9: Binning clusters by standard deviation

## 6 Asymmetrical similarity

In Section 4. we have already given the idea to compare one cluster's mean to another cluster's elements. When comparing not round-shaped clusters, this way of measuring similarity introduces asymmetry. Plotting a histogram of the differences $\mathrm{sim}_{\mathrm{cl}}(C_1, C_2) - \mathrm{sim}_{\mathrm{cl}}(C_2, C_1)$ shows a distribution quite close to normal.

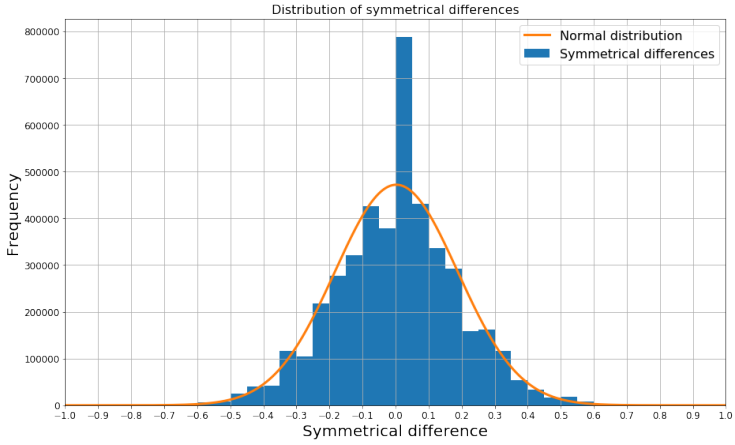



Fig. 10: Distribution of symmetrical differences

Most of these differences are around 0, showing that most of the clusters are round shaped. The two tails of the distribution are the important parts, because they show us pairs of clusters whose pairwise similarity in one direction is 1, while in the other direction this similarity is 0. One example to this phenomenon is the pair of `[/N|Pro][Subl][1Sg]`, `[/N|Pro][3Pl][Dat]<`

| **Bin** | $\sigma_{\text{diff}}$ | **Cluster** | $\textbf{Sim}_{\text{self}}$ | **#Words** | **Frequency** |
|---|---|---|---|---|---|
| 0.45 | 5.75 | `[/V][Inf]` | 0.988 | 14071 | 6677547 |
| 0.50 | 2.88 | `[/Num\|Digit][Nom]` | 0.977 | 26666 | 6000563 |
| 0.55 | 2.25 | `[/V][Prs.Def.3Pl]` | 0.990 | 5230 | 1312497 |
| 0.55 | 2.56 | `[/N][Pl][Subl]` | 0.980 | 6795 | 582972 |
| 0.55 | 2.20 | `[/N][Pl][Supe]` | 0.979 | 5325 | 519220 |
| 0.55 | 2.72 | `[/N][Pl][Ins]` | 0.982 | 10135 | 955157 |
| 0.55 | 2.17 | `[/V][Prs.NDef.3Pl]` | 0.989 | 10486 | 3296388 |
| 0.55 | 2.94 | `[/N][All]` | 0.979 | 13858 | 1073443 |
| 0.60 | 2.60 | `[/N][Ins]` | 0.972 | 32886 | 3868455 |
| 0.60 | 2.42 | `[/N][Subl]` | 0.969 | 25687 | 3469518 |
| 0.60 | 2.21 | `[/N][Pl][Acc]` | 0.974 | 20702 | 3601070 |
| 0.60 | 2.25 | `[/N][Abl]` | 0.964 | 10270 | 649706 |
| 0.60 | 2.22 | `[/N][Ela]` | 0.968 | 12717 | 1028608 |
| 0.65 | 2.04 | `[/N][Ade]` | 0.949 | 7324 | 363706 |
| 0.65 | 3.99 | `UNKNOWN` | 0.892 | 199475 | 5643460 |
| 0.65 | 2.58 | `[/N][Acc]` | 0.966 | 61671 | 12617934 |
| 0.65 | 2.06 | `[/N][Poss.3Sg][Acc]` | 0.962 | 14164 | 2823258 |
| 0.70 | 2.48 | `[/N][Nom]` | 0.922 | 144945 | 50298170 |

Table 2: Clusters with unexpectedly high coherence

`>[/N|Pro][Poss.3Pl][Dat]` clusters. Both of the clusters contain 4 vectors, but the words of the first cluster have 42228 occurrences in the corpus, and the words of the second cluster count 545 occurrences. The words are pronouns in both cases, the first clusters' words are *énrám, reám, rám, énreám,* ($s = 0.1$) meaning 'onto me' with variable spelling, the difference is only stylistic, the words of the second cluster are *némelyiküknek, valamelyiküknek, mindegyiküknek, bármelyiküknek*, ($s = 0.382$) the first one meaning 'to some of them' and 'of some of them', while the rest meaning the same, but changing 'some' to 'specific one', 'all', 'any'. One reason for this strange phenomenon is that the *énrám, reám, rám, énreám* have identical meanings, the standard deviation of their cluster is very low, while the other cluster of 4 words have significant difference in their meanings.

In the following sections, the asymmetry is of less importance. As shown on Fig. 10, most of the pairwise similarities have difference below 0.1, thus we do not lose much by symmetrizing the similarity measure by taking the mean of the similarities, $(\text{sim}_{\text{cl}}(C_1, C_2) + \text{sim}_{\text{cl}}(C_2, C_1))/2$.

### 6.1 Subcategories

E-magyar creates multiple subcategories for adjectives, nouns and numbers, and we can measure the pairwise similarity of their paradigms. If some subcategories show high similarity, we can say that it is not worth preserving as separate categories. Comparison of the subcategories to the `[/Adj]` categories yields interesting results.

| **Cluster$_1$** | **Cluster$_2$** | **similarity** | **cases** |
|---|---|---|---|
| `[/Adj][.]` | `[/Adj][.]` | 0.954 | 22 |
| `[/Adj][.]` | `[/Adj\|col][.]` | 0.921 | 14 |
| `[/Adj][.]` | `[/Adj\|nat][.]` | 0.900 | 16 |
| `[/Adj][.]` | `[/Adj\|Attr][.]` | 0.865 | 7 |
| `[/Adj][.]` | `[/Adj\|Pro][.]` | 0.843 | 18 |
| `[/Adj][.]` | `[/Adj\|Pro\|Rel][.]` | 0.549 | 7 |
| `[/Adj][.]` | `[/Adj][Pl][.]` | 0.884 | 17 |
| `[/Adj][Pl][.]` | `[/Adj][Pl][.]` | 0.956 | 17 |
| `[/Adj][Pl][.]` | `[/Adj\|col][Pl][.]` | 0.949 | 9 |
| `[/Adj][Pl][.]` | `[/Adj\|nat][Pl][.]` | 0.943 | 16 |
| `[/Adj][Pl][.]` | `[/Adj][Poss.1Sg][.]` | 0.855 | 10 |

Table 3: Adjectival subcategories

`[.]` marks the pairwise comparison of single morphemes, so in the first few examples, we compare singular forms to singular form (because singular forms are not marked, thus a single morpheme after the word root must mean singular), and in the cases after, the plural forms. We can see a declining similarity when comparing more and more specific clusters, with the `[/Adj|col][.]` (adjectives describing colors) and `[/Adj|nat][.]` (adjectives describing nationality) being relatively similar to `[/Adj][.]`, while `[/Adj|Pro][.]` (pronominal adjectives) and especially the `[/Adj|Pro|Rel]` (relative pronouns like *amilyen* or *amekkora*, 'such as', 'as large as, as much as') show significantly less similarity. As we noted at the beginning, more specific case endings may dominate the word vectors' similarity clusterwise, which is indeed the case in the last examples. Comparing plural adjectives, the similarities are significantly higher than their singular counterparts' similarities, while comparing singular to plural yields very low similarity.

### 6.2 Paradigm self-similarities

In the previous section, we have already used the `[.]` to indicate the comparison of paradigms. While the nominative forms may have lower similarities, the paradigm comparisons are dominated by the abundance of cases and case endings, producing very high self similarities. `[.]` denotes only a single morpheme, so this table aggregates only the 2-morpheme-long morphological analyses.

## 7 Conclusions and further research

Clustering word vectors by their morphological analysis has proven a good way to examine the impact of inflection on word vectors. Because of the high dimension, naive statistical testing of the distances from the mean does not produce easily interpretable results. In contrast, the 'cap similarity' introduced here,

| $\textbf{Cluster}_1$ | $\textbf{Sim}_{\text{self}}$ | **cases** | $\textbf{Cluster}_1$ | $\textbf{Sim}_{\text{self}}$ | **cases** |
|---|---|---|---|---|---|
| `[/Adj\|Pro\|Rel][.]` | 1.000 | 7 | `[/N\|Unit][.]` | 0.992 | 14 |
| `[/Num\|Pro][.]` | 1.000 | 9 | `[/Adj\|nat][.]` | 0.995 | 16 |
| `[/Num\|Roman][.]` | 1.000 | 6 | `[/V][.]` | 0.989 | 54 |
| `[/N\|Acronx][.]` | 1.000 | 13 | `[/Post][.]` | 0.987 | 8 |
| `[/N\|Pro\|Rel][.]` | 1.000 | 15 | `[/N\|Unit\|Abbr][.]` | 0.984 | 14 |
| `[/Adj\|col][.]` | 1.000 | 14 | `[/Num][.]` | 0.979 | 18 |
| `[/N\|mat][.]` | 0.998 | 17 | `[/Num\|Digit][.]` | 0.975 | 14 |
| `[/N\|Ltr][.]` | 0.997 | 13 | `[/N\|Acron][.]` | 0.974 | 14 |
| `[/N\|Abbr][.]` | 0.996 | 13 | `[/N][.]` | 0.958 | 24 |
| `[/N\|Pro][.]` | 0.996 | 16 | `[/Adj\|Pro][.]` | 0.958 | 20 |
| `[/Adj\|Attr][.]` | 0.995 | 7 | `[/Adj][.]` | 0.955 | 22 |

Table 4: High self-similarity

while asymmetrical, has produced acceptable results, showed high coherence and similarity where expected, and showed lower similarity where difference was expected, thus justifying the selection of clusters for most cases. There are exceptions however, such as treating `[/Adj|Pro|Rel]` as a subcategory of `[/Adj]`, which our method shows to be mistake due to their low similarity. Other future work may also include using disambiguated text corpus to have bigger clusters thus more data to perform the same analysis.

There are supervised methods for creating meaningful ultradense subspaces for polarity, concreteness, frequency and part-of-speech (POS) [15,16], supporting operations like 'give me a neutral word for *greasy*'. We plan on analyzing the POS subspace, comparing the similarities of the clusters projected onto the subspace with the similarities obtained without projection.

## Acknowledgment

This research is partially supported by National Research, Development and Innovation Office NKFIH grant #120145 'Deep Learning of Morphological Structure' and by 2018-1.2.1-NKP-00008 'Exploring the Mathematical Foundations of Artificial Intelligence'.

## References

1. Siklósi Borbála, N.A.: Beágyazási modellek alkalmazása lexikai kategorizációs feladatokra. In Tanács, A., Varga, V., Vincze, V., eds.: Proc. MSZNY 2016. Szegedi Tudományegyetem (2016) 3–14
2. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics **5** (1943) 115–133
3. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013), Atlanta, Georgia, Association for Computational Linguistics (2013) 746–751
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. (2013) International Conference on Learning Representations (ICLR 2013).
5. 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. Curran Associates, Inc. (2013) 3111–3119
6. 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 2004), ELRA (2004) 203–210
7. Kornai, A., Halácsy, P., Nagy, V., Oravecz, C., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In Kilgariff, A., Baroni, M., eds.: Proc. 2nd Web as Corpus Workshop (EACL 2006 WS01). (2006) 1–8
8. Novák, A., Siklósi, B., Oravecz, C.: 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., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, European Language Resources Association (ELRA) (2016)
9. Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: `e-magyar`: digitális nyelvfeldolgozó rendszer. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017), Szeged (2017)
10. 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, Valletta, Malta, ELRA (2010) 45–50
11. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: Rand-walk: A latent variable model approach to word embeddings. Transactions of the Association for Computational Linguistics (TACL) **4** (2016) 385–399
12. Mimno, D., Thompson, L.: The strange geometry of skip-gram with negative sampling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2017) 2873–2878
13. Grinstein, G., Trutschl, M., Cvek, U.: High-dimensional visualizations. In: Proceedings of the Visual Data Mining Workshop, KDD 2, 120. (2002)
14. Li, S.: Concise formulas for the area and volume of a hyperspherical cap. Asian Journal of Mathematics & Statistics **4** (2011) 66–70
15. Rothe, S., Ebert, S., Schütze, H.: Ultradense word embeddings by orthogonal transformation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (2016) 767–777
16. Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L.M., Su, P.H., Vandyke, D., Wen, T.H., Young, S.: Counter-fitting word vectors to linguistic constraints. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (2016) 142–148

# Beszédtechnológia II.

# Érzelmek felismerése magyar nyelvű hangfelvételekből akusztikus szózsák jellemzőreprezentáció alkalmazásával

Vetráb Mercedes[1], Gosztolya Gábor[1,2]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
14vini24@gmail.com, ggabor@inf.u-szeged.hu

**Kivonat** Az érzelmek felismerése a beszédtechnika egy jelenleg is aktívan kutatott területe. A feladaton belül számos probléma fogalmazódott már meg; ezek egyike az egyes hangfelvételek leképezése jellemzőkre. Ennek különlegességét az adja, hogy a hangfelvételek várhatóan eltérő hosszúságúak, míg a következő lépésben alkalmazott osztályozó eljárás fix méretű jellemzővektorokat vár el. Jelen dolgozatban az érzelemfelismerés problémájára egy nemrégiben kifejlesztett jellemzőreprezentációs eljárást, az akusztikus szózsák (bag of audio words, BoAW) módszert alkalmazzuk, mely képes a változó hosszú bemondásokat fix jellemzőtérbe képezni. Kísérleti eredményeink alapján a BoAW eljárás versenyképes osztályozási teljesítményt tesz lehetővé, ugyanakkor a metódus számos paraméterrel rendelkezik, melyeket a megfelelő hatékonyság érdekében pontosan be kell hangolni.
**Kulcsszavak:** érzelemfelismerés, hangfeldolgozás, akusztikus szózsák reprezentáció

## 1. Bevezetés

Az emberi hang nem csupán a szöveg közlésének alapjául szolgál, hanem magában hordoz rejtett, mégis az adott beszélőre nézve fontos, fizikai és lelki jellemzőket is. Ezen indirekt hangi kifejeződések egyike a beszélő emocionális állapota. Napjainkban az automatikus érzelemdetektálás egy aktívan kutatott témakör. A gépek által használt érzelem-felismerő és -monitorozó rendszerek jelenleg is fejlődésben vannak. A technika alkalmazási köre elég széles skálán mozog. Többek közt hasznos az ember-gép interakciók során (az ember kommunikációjának monitorozására) [1], dialógusrendszereknél [2], az egészségi állapot felméréseknél [3,4], valamint a call-centerekben [5]. Az érzelemfelismerés fejlődésével jelenleg is létező munkák könnyíthetők meg, valamint a későbbiekben, mindennapjainkba is beszivárgó robotikai és informatikai rendszerek kiegészítéseképp, vagy akár alapjául is szolgálhat.

Ezen terület kutatásának kezdete óta több módszert is kidolgoztak arra nézve, hogy a hangfelvételekből milyen módon érdemes jellemzőket kivonni, valamint arra, hogy melyek azok a tanulóalgoritmusok, amik a legoptimálisabb és

legeffektívebb eredményeket szolgáltatják egy-egy mintahalmazon. Ezen cikk az akusztikus szózsák (Bag-of-Audio-Words, BoAW) technikát és annak sikerességét vizsgálja, SVM tanulóalgoritmussal ötvözve. Kísérleteinket egy magyar nyelvű hangadatbázison végeztük; eredményeink azt mutatják, hogy a BoAW eljárás hatékony jellemzőreprezentációt tesz lehetővé érzelemfelismerés esetén is, mert a kapott pontosságmetrika-értékek (relatíve) magasaknak adódtak. Ugyanakkor azt a következtetést is levonhatjuk, hogy az eljárás érzékeny a paraméterbeállításokra, így azokra nagy figyelmet kell fordítani, hogy az osztályozás minősége megfelelően magasan alakuljon.

## 2. Az akusztikus szózsák eljárás

Az általunk használt *akusztikus szózsák* technika, azaz a *bag of audio words* (vagy BoAW) hasonló a szövegfeldolgozásban ismert *bag of words* és a képfeldolgozásban alkalmazott *bag of visual words* (BoVW) módszerekhez. Az 1. ábrán látható, hogy a BOAW módszer egyes fázisaiban végrehajtott műveleteket mind a tanító, mind a teszt halmazon elvégezzük. Első lépésben a tanítóhalmaz hangfelvételeiből kinyerjük az előre meghatározott jellemzőket, melyekből minden kerethez egy-egy jellemzővektor áll elő (keretszintű jellemzők). Ezután a jellemzővektorokból klaszterezés segítségével elkészül a kódszavak halmaza (kódhalmaz, *codebook*). A folyamat során megadott számú csoportot hozunk létre, ahol a klaszterek középpontjai lesznek a kódszavak (codewords). A csoportok számát nevezzük a codebook méretének; ez a szózsák eljárás egyik paramétere is.

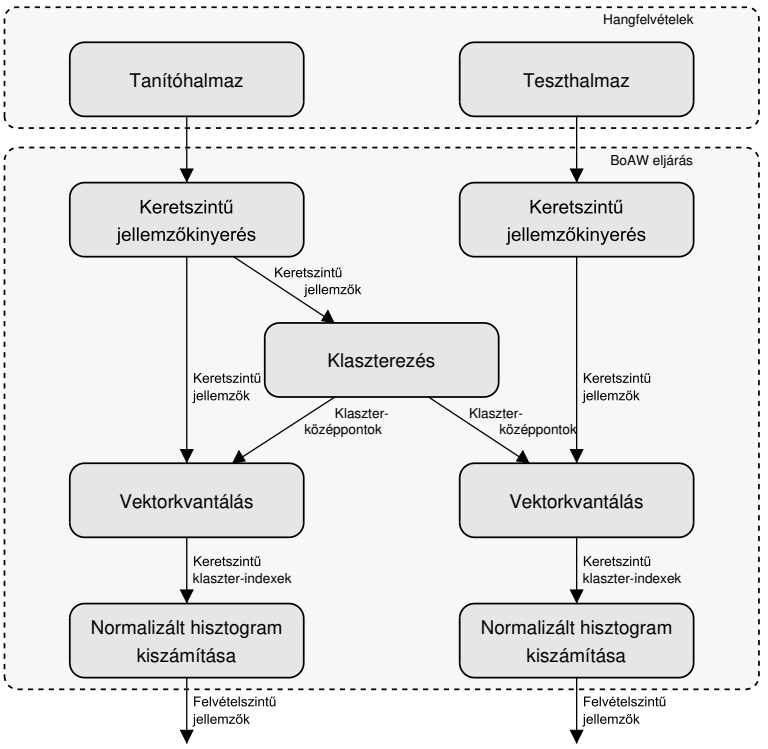
A következő lépés a vektorkvantálás, mely során az egyes felvételekhez tartozó keretszintű jellemzővektorokat kvantáljuk az előző lépésben generált kódszavaktól vett minimális euklideszi távolságuk alapján. Az eredeti jellemzővektorok helyettesítésre kerülnek a hozzájuk legközelebb lévő kódszó indexével. Végül egy hisztogramot készítünk a kódszavak és hozzájuk sorolt vektorok gyakoriságából. Ebből adódóan a hisztogram mérete megegyezik a codebook méretével, és függetlenné válik az adott hangfelvétel hosszától. Az így előállított vektorhalmaz lesz a „*bag of audio words*" reprezentáció, ami majd a tanító algoritmusunk inputjául szolgál.

Látható, hogy a kvantálási lépést a teszthalmaz felvételeire is elvégezhetjük: bár a teszthalmaz felvételeit nem használtuk fel a klaszterezés során, a keretszintű jellemzővektoraikat ettől még besorolhatjuk az egyes klaszterekbe a kódszavaktól vett távolságuk alapján.

### 2.1. A szózsák eljárás paraméterei

A BoAW eljárásnak több olyan tulajdonsága, paramétere is van, mellyel befolyásolhatjuk a szózsák készítésének menetét. A most következő részben ismertetjük, hogy a tanulás sikerességére való befolyás szempontjából mely tényezők hatását vizsgáltuk.

Az egyik befolyásoló tényező a codebook készítése során használt klaszterező eljárás. Pancoast és Akbacak eredeti tanulmányukban k-means-t használtak [6];




1. ábra: Az akusztikus szózsák eljárás működési módja.

ugyanakkor a klaszterezendő keretek nagy száma miatt ennek a megközelítésnek igen magas a futási ideje. Rawat és mtsai. egyszerű véletlen mintavételezést javasoltak [7]; amellett, hogy ennek futási ideje nyilvánvalóan kedvezőbb, mint egy teljes klaszterezésnek, a tapasztalatok szerint (ld. [7,8]) ennek az eljárásnak a használata a teljesítményt is legfeljebb érintőlegesen rontja. Később Schmitt és mtsai. a k-means++ klaszterezési eljárás klaszterközéppont-inicializáló eljárását [9] alkalmazták a teljesen véletlenszerű mintavételezés helyett [10], így a klaszterközéppontok eloszlása kiegyensúlyozottabb lesz. Ezen kutatások alapján az utóbbi metódust választottuk, így kísérleteink során mindvégig azt fogjuk alkalmazni.

Egy másik szabályozható komponens a hisztogram előállításának módja. Pancoast és Akbacak azt javasolták, hogy minden kerethez a legközelebbi klaszter helyett a legközelebbi $a$ db. klasztert rendeljük hozzá, mivel így azonos méretű jellemzővektor mellett pontosabban írhatjuk le az adott felvételt [11]. Ha csupán a legközelebbi komponenst vesszük figyelembe, úgy gondoltuk, hogy túl nagy megszorítást eszközölünk, ezért kipróbáltuk a feltétel lazításának hatását is.

Az eddig taglalt módosításokon túl, a kezdeti keretszintű jellemzőkészleten is hajthatunk végre előfeldolgozást. Előfordulhat, hogy az eredeti adatok túlságosan szétszórva helyezkednek el a térben, valamint vannak köztük olyan minták,

melyek kiugró értékeikkel fals irányba mozdíthatják a tanulást. Ennek kiküszöbölésére a jellemzővektorokat normalizálhatjuk úgy, hogy a minimum és maximum értékekhez igazítva, 0 és 1 közötti skálára hozzuk az adatokat. Egy másik megoldás lehet, ha standardizálást hajtunk végre, tehát a mintákat úgy transzformáljuk, hogy szórásuk 1, átlaguk pedig 0 legyen.

## 3. Kísérletek

A következőkben bemutatjuk az elvégzett kísérletek technikai körülményeit: az alkalmazott adatbázist, az osztályozási eljárást és paramétereit, a kiértékelésre használt metrikát, valamint a keretszintű jellemzőkészletet.

### 3.1. A magyar érzelemadatbázis

A kutatás során használt adatbázis 97 magyar anyanyelvű és magyarul beszélő személy hangját tartalmazza [12]. A beszédek televíziós műsorok során lettek felvéve. A szegmensek túlnyomó része érzelmekben gazdag, folyamatos, spontán beszédből lett kivágva. Kisebb részük improvizációs szórakoztató műsorból jön. Ebből fakadóan az elsőként említett kategóriába tartozó minták a színészi játék miatt, az érzelmek egy feljavított és egyértelműbb változatát tartalmazzák, míg a maradék improvizációs halmazban lévők közelebb állnak a hétköznapi, természetes érzelemkifejezéshez. Az adatbázis összesen 1111 mondatot tartalmaz, melyek egy 831 elemű tanító és 280 elemű teszt halmazra lettek osztva. Az osztályozás során négyféle érzelmet definiálunk a beszédekben: Harag, Öröm, Szomorúság és Semleges hangulat. Korábbi tanulmányok, melyek ugyanezzel az adatbázissal dolgoztak, 66-70%-os osztályozási pontosságot tudtak elérni [13].

### 3.2. Osztályozás

Az osztályozást SVM-ek (Support Vector Machines [14]) használatával végeztük, lineáris kernellel, a libSVM implementációt használva [15]. Az algoritmus komplexitás (complexity, $C$) paraméterét minden minta esetén többféle beállítással teszteltük. A lehetséges konfigurációk az alábbi 10 hatványok voltak: 0.00001; 0.0001; 0.001; 0.01; 0.1; 1 és 10. Az algoritmus tanulását és kiértékelését 10-szeres keresztvalidálással (10-fold cross-validation, CV) végeztük el. Tehát az aktuális mintahalmazt 10 egyenlő részre osztottuk, és minden lehetséges 9 tanító – 1 tesztelő halmaz kombinációra tanítottunk és kiértékeltünk egy SVM modellt. A teszthalmazra adott predikcióinkat a teljes tanítóhalmazon tanított SVM modellek szolgáltatták. Egy adott modell "jóságának" mérésére az UAR metrikát (Unweighted Average Recall: az adott osztályra helyesen osztályozott példák száma osztva az adott osztályba tartozó példák számával) alkalmaztuk. A kereszvalidálás során a tanító halmazra kapott értékek alapján választottuk ki, hogy a kutatás egyes fázisaiban mely paraméterértékekkel dolgozzunk tovább.

| Megközelítés | Maximális UAR | Codebook méret |
|---|---|---|
| Változatlan jellemzők | 44.34% | 32 768 |
| Normalizált jellemzők | 70.77% | 8 192 |
| Standardizált jellemzők | 68.29% | 8 192 |

1. táblázat. A keretszintű jellemzők normalizálásával és standardizálásával elért legjobb pontosságok a keresztvalidáció során.

### 3.3. Keretszintű jellemzőkészlet

Az akusztikus keretszintű jellemzők megválasztásának alapját a 2013-2014-es INTERSPEECH Számítógépes Paralingvisztikai Versenyen kiadott cikk adta. Az ott publikált jellemzőkészlet 65 keretszintű jellemzőt, azaz LLD-t (low level descriptor) tartalmazott (4 darab energián alapuló LLD; 55 spektrális LLD; 6 hangosságon alapuló LLD), valamint ezek első fokú deriváltjait. Kutatásunk során ezen jellemzőket az openSMILE nevű program segítségével számoltuk le. A hangosság alapú leírókat 60 ms-os kerettel (Gaussian window function) és 0.4 értékű szigmával, a másik két csoportot pedig 25 ms-os kerettel (Hamming window function) számítottuk. A Hamming-ablakokat a megszokott módon, átfedéssel, 10 ms-os eltolással helyeztük el.

frekvenciát valósítottak meg.

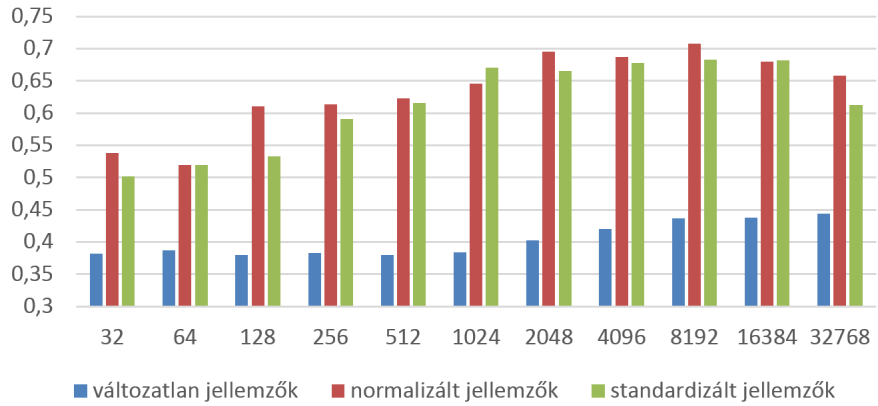
### 3.4. Az akusztikus szózsák eljárás paraméterei

Az egyik olyan paraméter, melyet minden esetben megadott értékekkel vizsgáltunk, az a codebook mérete volt. Az eljárás első lépése során megadhatjuk, hogy hány klasztert hozzunk létre, tehát hány kódszavunk legyen. Az általunk vizsgált értékek minden esetben az alábbi skálára terjedtek ki: 32, 64, 128, 256, 512, 1 024, 2 048, 4 096, 8 192, 16 384, 32 768. Az eljárás végén a számolt hisztogramot minden esetben normalizáltuk, azaz a kapott gyakoriságokat elosztottuk a hangfelvétel kereteinek számával.

## 4. Tesztek és eredmények

A következőkben ismertetésre kerül a kísérletek pontos menete, valamint az egyes fázisokban kapott eredmények kiértékelése.

Az első összehasonlítandó tényező a keretszintű jellemzővektorok kezelése volt a klaszterezés megkezdése előtt. Három esetet vizsgáltunk: 1) a jellemzővektorokat érintetlenül hagytuk, 2) a jellemzővektorokat normalizáltuk, 3) a jellemzővektorokat standardizáltuk. Az eredmények alapján (ld. 2. ábra és 1. táblázat) a normalizálás és a standardizálás közel azonos teljesítményjavulást nyújt ahhoz képest, ha az adatokon semmilyen további módosítást nem hajtunk végre. A bemeneti jellemzők normalizálásának vagy standardizálásának további előnye,



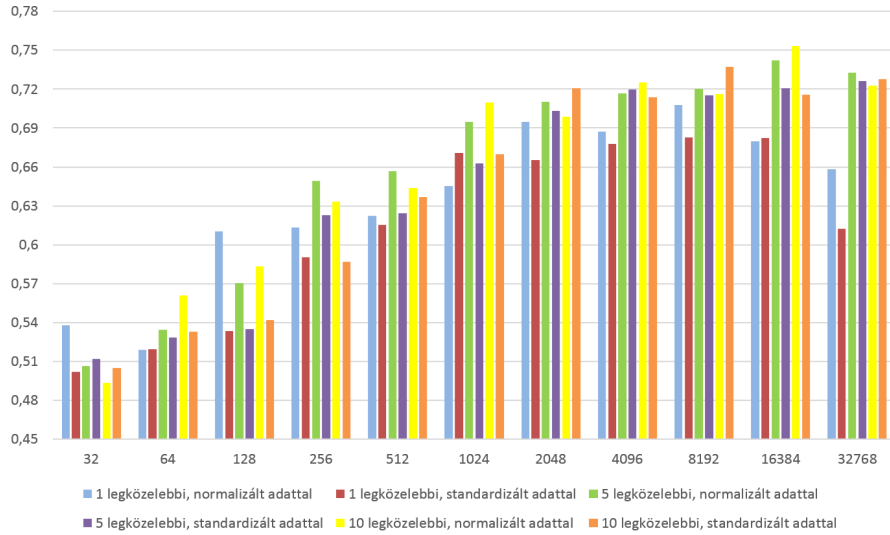

2. ábra: A keretszintű jellemzők normalizálásával és standardizálásával elért eredmények, különböző szózsák méreteknél.

hogy lényegesen kevesebb klaszter (mindkét esetben 8192) szükséges az optimális teljesítményhez, mint ha a jellemzőket változatlanul hagynánk (32768), így a felvételszintű SVM tanítása is kisebb jellemzőtérben történik. Ezen eredmények alapján a további teszteket párhuzamosan, normalizációval és standardizációval is elvégeztük.

A következő összehasonlításban azt vizsgáltuk, hogy a hisztogram létrehozásakor egy-egy keretszintű jellemzővektort hány legközelebbi kódszóhoz érdemes hozzárendelnünk. Most az $a = 1$, $a = 5$ és $a = 10$ eseteket vizsgáltuk; a három lehetőséget mind normalizált, mind standardizált adatokon kiértékeltük. A 3. ábrán szereplő legjobb eredményekből, levonható az a következtetés, hogy az $a = 5$ és $a = 10$ értékek a legtöbb esetben jobb felvételreprezentációt eredményeznek, mivel az osztályozás során kapott UAR értékek magasabbnak adódtak, mint az $a = 1$ beállítás esetén. Ez mind normalizálás, mind standardizálás esetén fennállt. A két hozzárendelés-érték segítségével elért teljesítményértékek között azonban nem tapasztaltunk lényeges különbséget, és az $a$ paraméter az akusztikus szózsák eljárás által eredményül adott jellemzővektor méretét sem befolyásolja, csak annak kiszámítását befolyásolja csekély mértékben.

A 2. táblázatból leolvashatóak a legjobb eredmények a különböző $a$ értékek esetén; ezek jól mutatják, hogy az 5 és 10 legközelebbi kódszót nézve közel azonos mértékű javulást kapunk, az 1 szavas változathoz képest; az $a = 10$ eset mindkét jellemzőnormalizálási eljárás esetén valamivel jobbnak adódott, de a különbség valószínűleg nem szignifikáns.

Az eddig ismertetett döntéseket a tanító halmazon végzett keresztvalidációs technika által adott százalékok alapján hoztuk meg. Ezzel határoztuk meg azt a szűkebb paraméterhalmazt, melyre a teszt mintákon is kiértékeljük az SVM algoritmust. A 2. és 3. ábrán is látható, hogy 1 024-es codebook méretig konzisztensen növekvő tendenciát mutat minden próba. Ezen számok alapján úgy döntöttünk,



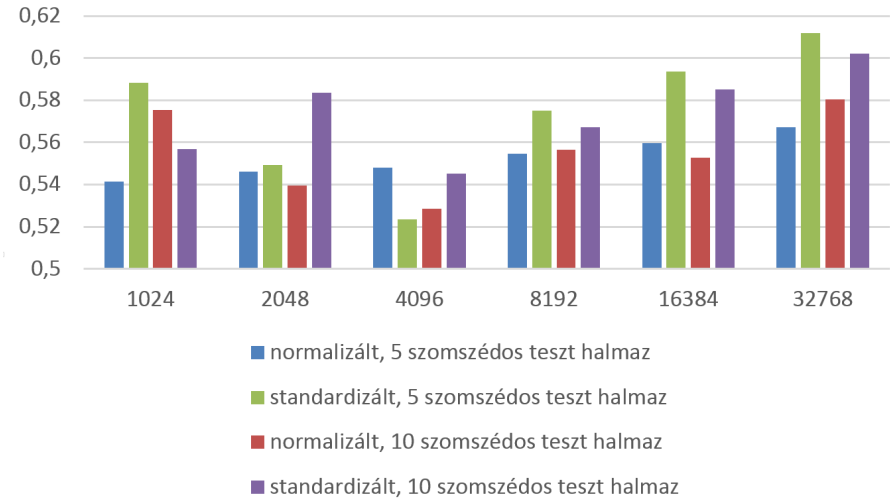

3. ábra: 1/5/10 legközelebbi kódszóhoz való társításnál kapott eredmények az adatok normalizálásának és standardizálásának függvényében.

| **Jellemző-transzformáció** | **a** | **Maximális UAR** | **Codebook méret** |
|---|---|---|---|
| Normalizálás | 1 | 70,77% | 8 192 |
| | 5 | 74,20% | 16 384 |
| | 10 | 75,31% | 16 384 |
| Standardizálás | 1 | 68,29% | 8 192 |
| | 5 | 72,63% | 32 768 |
| | 10 | 73,73% | 8 192 |

2. táblázat. Az 5 és 10 legközelebbi kódszóhoz való igazításnál kapott legjobb eredmények a normalizálás, valamint a standardizálás függvényében a keresztvalidáció során.

hogy a teszthalmazon való kiértékelést elegendő csupán az 1 024 és afeletti méretekre elvégezni, ugyanis ezalatt a tanulás állandó jelleggel rosszabbnak bizonyult. A 4. ábrán láthatók azon eredményeink, melyeket a teszt halmazokon való UAR-ok kiszámítása után kaptunk; a keresztvalidálás során kapott legjobb paraméterekhez tartozó, teszthalmazon mért UAR értékek pedig a 3. táblázatban találhatóak.

Az itt látható értékek lényegesen alacsonyabbak ugyan, mint amiket a keresztvalidáció során kaptunk, de a két halmazon kapott értékek természetesen nem hasonlíthatók össze direktben. Ugyanígy, a korábban erre az adatbázisra elért eredmények sem vethetőek össze közvetlenül az általunk kapott értékekkel, hiszen ott (egyszeres) keresztvalidációt végeztek, és az eredményeket standard osztályozási pontosságban adták meg, míg jelen tanulmányunkban, a kiegyensú-




4. ábra: A teszthalmazon való kiértékelés eredményei.

| **Jellemző-transzformáció** | **a** | **Maximális UAR** | **Codebook méret** |
|---|---|---|---|
| Normalizálás | 5 | 55,97% | 16 384 |
| | 10 | 55,27% | 16 384 |
| Standardizálás | 5 | 61,19% | 32 768 |
| | 10 | 56,83% | 8 192 |

3. táblázat. A teszthalmazon való kiértékelés eredményei.

lyozatlan osztályeloszlást ellensúlyozandó, UAR-t használtunk. A 4. ábrán látható a codebook méretének pozitív hatása, de a tendencia korántsem egyértelmű; például az 1 024 klasztert használó modellek jobbnak adódtak, mint a 4 096 klasztert használók. Elmondható az is, hogy minél nagyobb méretet választunk, annál nagyobb annak is az esélye, hogy túltanulási hibába futunk vele és az osztályozónk elveszíti általánosító képességét.

## 5. Összegzés

Jelen cikkünkben az akusztikus szózsák (Bag-of-Audio-Words, BoAW) jellemzőreprezentációs eljárást alkalmaztuk egy magyar nyelvű érzelemfelismerési feladaton. Az eljárásnak számos paramétere van, így számos gépi tanulási modellt kellett tanítanunk a különböző paraméter-kombinációkra. Mért eredményeink alapján a bemeneti jellemzőket mindenképpen érdemes azonos skálára hoznunk normalizálás vagy standardizálás segítségével, és az alkalmazott kódszavak szá-

mát is érdemes magasnak választanunk (8 192-32 768). Az egyes kereteket is érdemes párhuzamosan több klaszterbe sorolnunk.

Annak kapcsán, hogy az itt elért eredményeink által merre haladhatunk tovább a későbbiekben, több lehetőség is felmerül. Jelenleg a keretszintű jellemzők közül, csupán az első 65-öt vettünk figyelembe; a későbbiekben használhatjuk az LLD-k elsőrendű deriváltjait is. Másrészről a codebook generálás során alkalmazott klaszterező eljárást jelenleg korábbi kutatásokra hivatkozva választottuk ki. Ezen metódusok eredményességét a tanulóadatbázisunkon mi magunk is letesztelhetnénk. Emellett lehetőségünk van más keretszintű jellemzőkészleteket is letesztelni. További érdekes kísérletek végezhetők több adatbázis használatával is; hasonló jellegű korpuszok esetén jogos kérdés a BoAW eljárás paramétereinek stabilitása, de akár a kódszavak átvitele is.

## Hivatkozások

1. James, J., Tian, L., Inez Watson, C.: An open source emotional speech corpus for human robot interaction applications. In: Interspeech, Hyderabad, India (2018) 2768–2772
2. Burkhardt, F., van Ballegooy, M., Engelbrecht, K.P., Polzehl, T., Stegmann, J.: Emotion detection in dialog systems: Applications, strategies and challenges. In: ACII, Amsterdam, Hollandia (2009) 985–989
3. Hossain, M.S., Muhammad, G.: Cloud-assisted speech and face recognition framework for health monitoring. Mobile Networks and Applications **20**(3) (2015) 391–399
4. Norhafizah, D., Pg, B., Muhammad, H., Lim, T.H., Binti, N.S., Arifin, M.: Detection of real-life emotions in call centers. In: ICIEA, Siem Reap, Kambodzsa (2017) 985–989
5. Vidrascu, L., Devillers, L.: Detection of real-life emotions in call centers. In: Interspeech, Lisszabon, Portugália (2005) 1841–1844
6. Pancoast, S., Akbacak, M.: Bag-of-Audio-Words approach for multimedia event classification. In: Interspeech, Portland, USA (2012) 2105–2108
7. Rawat, S., Schulam, P.F., Burger, S., Ding, D., Wang, Y., Metze, F.: Robust audio-codebooks for large-scale event detection in consumer videos. In: Interspeech, Lyon, Franciaország (2013) 2929–2933
8. Schuller, B., Steidl, S., Batliner, A., Hantke, S., Bergelson, E., Krajewski, J., Janott, C., Amatuni, A., Casillas, M., Seidl, A., Soderstrom, M., Warlaumont, A.S., Hidalgo, G., Schnieder, S., Heiser, C., Hohenhorst, W., Herzog, M., Schmitt, M., Qian, K., Zhang, Y., Trigeorgis, G., Tzirakis, P., Zafeiriou, S.: The INTERSPEECH 2017 computational paralinguistics challenge: Addressee, Cold & Snoring. In: Interspeech. (2017) 3442–3446
9. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: SODA, New Orleans, Louisiana, USA (2007) 1027–1035
10. Schmitt, M., Ringeval, F., Schuller, B.: At the border of acoustics and linguistics: Bag-of-Audio-Words for the recognition of emotions in speech. In: Interspeech, San Francisco, USA (2016) 495–499
11. Pancoast, S., Akbacak, M.: Softening quantization in bag-of-audio-words. In: ICASSP, Florence, Olaszország (2014) 1370–1374

12. Sztahó, D., Imre, V., Vicsi, K.: Automatic classification of emotions in spontaneous speech. In: COST 2102, Budapest (2011) 229–239
13. Vicsi, K., Sztahó, D.: Recognition of emotions on the basis of different levels of speech segments. Journal of Advanced Computational Intelligence and Intelligent Informatics **16**(2) (2012) 335–340
14. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation **13**(7) (2001) 1443–1471
15. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 1–27

# Kombinált központozási megoldások magyar nyelvre pehelysúlyú neurális hálózatokkal

Tündik Máté Ákos, Szaszák György

Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
e-mail:{tundik,szaszak}@tmit.bme.hu

**Kivonat** Napjainkban a rekurrens neurális hálókon alapuló szekvenciamodellezés hatékony eszköznek bizonyult több, a természetesnyelv-feldolgozás (NLP) témaköréhez tartozó probléma megoldásában. Ide sorolhatjuk az írásjelek gépi úton történő visszaállítását, vagyis az automatikus központozást is, melynek során a szó- és/vagy akusztikai eseménysorozathoz írásjeleket rendelünk. Ezt a technikát pl. a beszédfelismerő központozatlan kimenetére alkalmazva a szöveg sokkal olvashatóbbá, érthetőbbé válik. Cikkünkben pehelysúlyú kombinált központozási megoldásokat mutatunk be, melyhez karakter- és szószintű beágyazás (embedding) vektorokat, valamint egy 39 dimenziós akusztikai jellemzővektort is felhasználunk. Kísérleteinket két magyar nyelvű, hírműsorokat, illetve felolvasást tartalmazó korpuszon végeztük el. Eredményeinkkel igazoljuk, hogy a kombinált módszerekkel hatékonyabb tud lenni az írásjelek visszaállítása, mintha csak egy-egy szöveges vagy akusztikus komponensre támaszkodnánk.

**Kulcsszavak:** írásjel-visszaállítás, CNN, RNN, LSTM, karakter, szó, akusztika, prozódia, ASR

## 1. Bevezetés

Napjainkban nagy népszerűségnek örvend a kutatók között a gépi beszédfelismerés(ASR) kimenetének minél sokoldalúbb feldolgozása, melynek során a nyers szövegből egy ún. információgazdag átirat (rich transcription) keletkezik. Ehhez segítséget nyújtanak a rekurrens neurális hálózatokon alapuló írásjelező modellek is [1,2,3,4], melyek a korábbi, erre a problémára alkalmazott módszerek teljesítményét is felülmúlják [4]. Tipikusan szöveges [4] vagy prozódiai jellemzők [3] használatosak, de pl. angol nyelvre előfordulnak kombinált módszerek is [5].

Mivel az írásjelek az írott nyelv szerves részét képezik, a szöveges jellemzőkkel történő modellezésük különösebb magyarázatot nem igényel. Az [1,2,4] cikkek szerzői olyan, szóbeágyazásokat használó egy- és kétirányú rekurrens neurális hálózatokat (RNN) hoztak létre, mely széles szövegkontextusokból képes sokféle jellemzőt tanulni, ezáltal az írásjeleket hatékonyan beilleszteni a központozatlan szövegbe. A [4] szerzői az RNN-módszerek magasabb hatékonyságát egy tradicionális, Maximum Entrópia-alapú megoldással szemben is demonstrálták.

A [3] szerzői egy hasonló BiRNN architektúrát hoztak létre, de itt az írásjelek elhelyezése prozódiai jellemzők alapján detektált fonológiai frázisok segítségével történik meg, mivel ezek magas korrelációt mutatnak az írásjelekkel. Ezen módszer esetén szükséges a frázisok előzetes modellezése, mely pusztán akusztikai jellemzőkre támaszkodik (a frázisok detektálása az alapfrekvencia, az energia és ezek deriváltjainak segítségével lehetséges), ezért a központozásra alkalmazott modell nem függ pl. a felvételen elhangzott szavaktól/szósorozattól (ez ASR-hibák esetén előnyös).

A két különböző módszer összehasonlításakor mindegyikben találhatunk előnyös tulajdonságot. A szövegalapú jobb összteljesítményre képes (fedés, pontosság és F-pontszám tekintetében), és hatékonyabb a vesszők visszaállításában. Ezzel szemben a prozódián alapuló modell jóval robusztusabb az ASR-hibákkal szemben (a szóhibák továbbterjedése teljesen blokkolt), a mondatvégi írásjelek (mely legtöbb esetben a pont) predikciója precízebb. Tekintve, hogy a frázisok több szóból is állhatnak, így a modell nem minden szóhatárra jósol írásjelet, kombinálása a szöveges jellemzőket felvonultató rendszerrel ezért nem bizonyult sikeresnek. Így cikkünkben a "nyers" akusztikai jellemzőket használjuk fel, melyek kinyerése minden szó/szóhatár esetén megtörténik. Ennek hátránya, hogy bár a szavaktól maguktól továbbra sem függ a prozódiai modell, a hipotetikus szóhatároktól viszont már igen. Mint látni fogjuk, szerencsére ez nem okoz érdemi pontosságveszteséget, cserébe viszont lehetőségünk nyílik közel végponttól végpontig egyetlen modell tanítására.

Továbbhaladva, fontosnak tartjuk annak az esetnek a megvizsgálását is, amikor az írásjelező modellünk bemeneteként az egyes szavakból származó, karaktersorozatokból adódó információt használjuk fel. A karakteralapú modellek népszerűek a természetesnyelv-feldolgozás (NLP) területén is; segítségükkel lehetséges a szövegek szófaji címkézése [6], a nyelvi modellezés [7] és a névelem-felismerés [8]. Számos példát találunk az irodalomban olyan modellekre is, amelyek a karakterekből és a szavakból származó információt hatékonyan kombinálják, pl. névelem-felismerésre [9], gépi fordításra [10], vagy akár szentimentelemzésre [11].

A [12] cikk szerzői karakteralapú írásjelező modellt hoztak létre, melynek teljesítménye alig maradt el egy szóalapú, feltételes véletlen mező (Conditional Random Field, CRF) technológiát használó megoldással szemben. A karakteralapú modellek segítenek az adatelégtelenségi (data sparsity) probléma áthidalásában. Ez különösen fontos az agglutináló magyar nyelv esetén, ahol rengeteg szóalak használatos, ugyanakkor a valós-idejű gépi megoldásoknál kényszerként csak egy kötött szótárméret engedélyezett. Karaktersorozatokban gondolkodva ilyen kötöttséggel nem kell számolni, így a ritka szavak, mint karaktersorozat-inputok, tovább javíthatják az írásjelező modellek predikciós képességét.

Az alacsony szintű (pl. a karakterekből adódó) jellemzők hatékony kinyerése leginkább a konvolúciós neurális hálózatokhoz (CNN) köthető. A gépi látás mellett a beszédtechnológiai kutatásokban is sikerrel alkalmazták ezt a modellt [13,14], utalva arra, hogy nemcsak az emberek, hanem a mesterséges intelligencia is képes az alacsony szintű információk 'intuitív' észlelésére, mely hozzásegíthet a szöveg vagy beszéd értelmezéséhez [15]. A [12] cikk nyomán azt

gondoljuk, érdemes a karakterszintű automatikus írásjelezést magyar nyelvre is megvizsgálni.

Ezen túlmenően, a szöveges (karakter- és szóbeágyazások) és akusztikai jellemzőket együttesen felhasználva, három darab kétkomponensű, és egy darab, három komponensből álló kombinált központozó rendszert is bemutatunk.

Cikkünk az alábbi struktúra szerint épül fel: a 2. fejezetben bemutatjuk a kísérleteinkhez használt adatbázisokat. Ezt követően a 3. fejezetben ismertetjük a pehelysúlyú szó- és a karakteralapú, valamint az akusztikus jellemzőkön alapuló különálló modelleket, valamint ezek kombinált változatait. Továbbhaladva, a 4. fejezetben ismertetjük kísérleti eredményeinket. Végül az eredményekre vonatkozó tanulságokat levonva, felvázoljuk jövőbeni terveinket.

## 2. Adatbázisok

Az írásjel-visszaállítási kísérleteinkhez két magyar nyelvű adatbázist használtunk fel: a BABEL-t [16] és a Magyar Híranyag-adatbázist [17]. A tanítást és kiértékelést külön-külön végeztük el a két adatbázisra, azok jelentős különbségei miatt. Mindkét adatbázis nagyságrendileg 3-3 óra beszédet tartalmaz. A leggyakoribb és egyben a szöveg érthetősége szempontjából legfontosabb írásjeleket állítjuk vissza: a vesszőt, a mondatvégi pontot, a kérdőjelet és a felkiáltójelet. A kettőspontokat és a pontosvesszőket vesszővel helyettesítettük, minden más írásjeltől eltekintettünk.

Megjegyezzük, hogy mind a szó-, mind a karakteralapú beágyazások tanításához kiegészítő, csak szövegesen elérhető adatbázisokat használtunk fel a [18] irodalomban bemutatottaknak megfelelően.

Az anyagokat 60%-20%-20% arányban osztottuk fel tanítás, validálás és tesztelés céljából; a prozódiai írásjelező modell teljes egészében a BABEL-en illetve a Magyar Híranyag-adatbázison tanult, a [18]-ban ismertetett korpuszon előtanított szó- és karakteralapú módszerek esetén pedig adaptációt hajtottunk végre, tudástranszfert alkalmazva. A BABEL esetében a korpusz szövegrészei részben ismétlődnek; erre gondosan odafigyeltünk a tanító, validáló, és tesztelő halmazok összeállításakor.

A hírkorpuszon 35%-os, a BABEL-en mintegy 50%-os szóhibaarányt mértünk. (Az agglutináló, illetve egybeírandó szóösszetételekben gazdag nyelvekre a WER mindig jóval magasabb az angol nyelven mérthez képest a hasonló felismerési feladatok esetében [19].)

## 3. Írásjel-visszaállító módszerek

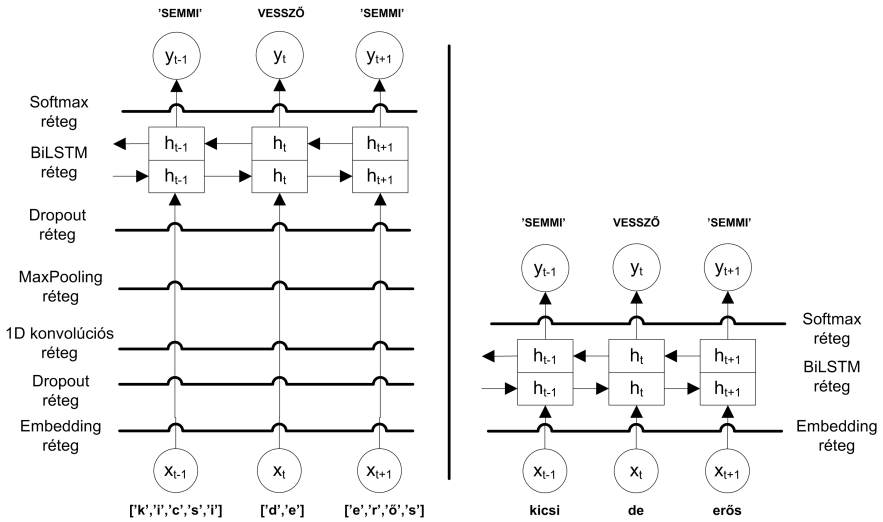
### 3.1. Szóalapú modell

A tanító, a validációs és a teszthalmazt rövid, fix hosszúságú szekvenciákra osztjuk fel, köztes átfedések nélkül. A különböző szóalakok számát limitáljuk, a tanítóhalmaz $k$ leggyakoribb szavából szótárt képezve, a kieső szavakat pedig egy közös *Ismeretlen* címkével látjuk el. A modellhez saját szóbeágyazási mátrixot

képzünk az előre tanított beágyazási modell és a szótárban szereplő szavak segítségével.

Kísérleteinkben egy kétirányú RNN modell teljesítményét vizsgáljuk meg. A modellben az aktuális szót megelőző időpillanatra jósoljuk az írásjelet. A kísérletekhez használt szóalapú RNN-architektúrát az 1. ábrán mutatjuk be.

A szóalapú RNN-modell ("W") a következőképpen épül fel: a szóbeágyazási mátrix alapján a modellnek átadott szószekvenciák a szóbeágyazási térbe ($x_t$ reprezentálja az $x$ szóhoz tartozó n-dimenziós szóbeágyazási vektort $t$ időpillanatban) kerülnek. Ezek a reprezentációk a következő, rejtett rétegbe továbbítódnak, amely BiLSTM rejtett cellákból áll, ezek a kontextus rögzítéséért, az információ kinyeréséért felelősek. A kimenetet egy $softmax$ aktivációs függvény használata után kapjuk meg, mely az $y_t$ kimeneti címkék eloszlását a jelenlegi $x_t$ szó előtti időpillanatra (slot-ra) adja meg.




1. ábra: A karakteralapú "C" RNN modell (bal oldalon) és a szóalapú "W" RNN modell (jobb oldalon) szerkezete

A "W" modellt a tanítókorpusz leggyakoribb 100 ezer szavával tanítottuk, valamint különböző dimenziószámú, előre tanított magyar nyelvű szóbeágyazási modelleket is kipróbáltunk [20]. A tanítás során az RNN modell súlyait a kategorikus keresztentrópia költségfüggvény alapján módosítjuk, valamint minden egyes epoch-ban frissítjük a szóbeágyazásokat is.

### 3.2. Karakteralapú modell

A "C" karakteralapú modellünk az 1. ábrán látható módon épül fel. A modell – hasonlóan a szóalapú megoldáshoz – fix hosszúságú szekvenciákat fogad a bemenetén, melyben a szavak karaktersorozatokként reprezentáltak. Minden egyes

karakter a karakterek által alkotott beágyazási térbe kerül. Fontos eltérés a szóalapú modellhez képest, hogy a karakteralapú modell tanításának kezdetén a beágyazási tér vektorai (előtanítás nélkül) véletlenszerűen inicializáltak. Másfelől, a karakteralapú modell előnye, hogy nem szükséges az OOV-szavakat kezelni, az összes karaktert tartalmazó szótár limitált számossága miatt; így az ezen szavakból képzett karaktersorozatok is befolyásolják/segítik a jellemzőtanulást.

A beágyazási transzformációt követően, az 1D-konvolúció művelete (különböző súlyozású konvolúciós szűrőivel) számos reprezentációt készít a transzformált bemenetből. Ezekből a reprezentációkból a dimenziócsökkentést elvégző MaxPooling réteg segítségével egy új, jóval tömörebb jellemzővektor keletkezik. Végül a BiLSTM réteg ismét a $t$ időpillanat kontextusának rögzítéséért, az információ kinyeréséért felelős. A kimeneti írásjelcímkék posterior valószínűségeit ismét egy $softmax$ aktivációs függvény használata után kapjuk meg. A köztes Dropout rétegek célja, hogy elkerüljük a modell túltanulását.

### 3.3. Akusztikai-prozódiai modell

A [3] szerzői az automatikus központozáshoz fonológiai frázisszegmentálásból származó prozódiai jellemzőket használtak fel. Mivel ezt a szegmentálást egy külön Rejtett Markov-modell végzi el, ehelyett vizsgálatainkhoz csak a frázisszegmentálást segítő akusztikai-prozódiai jellemzőket tartottuk meg a neurális háló alapú írásjelezés esetén. Az alapfrekvencia és átlagos energia kinyerése egy 150 ms-os ablakban történik (mel skálára bontás nélkül), 10 ms-onként mintavételezve, 5-pontos medián szűrővel simítva. Az $x_t$ szóhoz tartozó jellemzővektorba az alapfrekvencia- és energiaértékek első- és a másodrendű deriváltjai is bekerültek ($d_t$), melyeket az alábbi regressziós képlet segítségével számítottunk ki $W = 30$ keret hosszú kontextust figyelembe véve:

$$d_t = \frac{\sum_{i=1}^{W/2} i(x_{t+i} - x_{t-i})}{2\sum_{i=1}^{W/2} i^2} \tag{1}$$

Ahol a beszédfelismerő szóhatárt feltételez, ott újabb, két 6-dimenziós jellemzővektor kinyerése történik meg; egy a szóhatárt megelőző 15 keretben, egy pedig az utána következő 15 keretet befoglalva. Ehhez alap statisztikai értékeket számítunk; a minimum-, maximum- és átlagértékek kerülnek a 6x6 dimenziós vektorokba. A bemeneti vektort végül az aktuális szót megelőző szó időtartamával, és a két szó között eltelt szünetértékkel egészítjük ki. Az akusztikai alapú "P" modellünk is pehelysúlyú; a jellemzővektort egy kétirányú LSTM rétegbe irányítjuk, ezt követően pedig a softmax réteg felel a kimeneti írásjelért. Sajnos kevés hanganyag állt rendelkezésünkre, azonban az akusztikus modellünk tanításához ez is elegendőnek bizonyult; a modellre vonatkozó "legjobb" hiperparamétereket az 1. táblázat mutatja be.

A [3] szerzői által ismertetett módszerrel szemben ugyan szükségünk van az ASR szolgáltatta szóhatárokra, de mint látni fogjuk, ez a modell szóhiba-tűrését nem csökkentette érdemben. Úgy véljük, ez a technika kellően robusztus és mégis

egyszerű, mivel a dinamikus (első- és másodrendű derivált) jellemzők segítségével a legfontosabb prozódiai sajátosságokat, azok kontextusát tudjuk kinyerni a szóhatárokon (lokális hangsúlymintázatok, intonáció és szünet tükrében).

### 3.4. Hiperparaméterek

Szisztematikus, kimerítő keresés (grid search) alapú optimalizációt hajtottunk végre a "C", W" és a "P" modellek hiperparaméterein, a validációs halmaz elemeit értékelve. A szekvenciák hosszát, a rejtett állapotok számát, a minibatch méretét, és az optimizáló típusát mindhárom modell esetén változtattuk, valamint korai leállítást (early stopping, *Patience*) is használtunk, a túltanítás elkerülése érdekében. A szöveges modellek esetén a szótár méretét, valamint a szó- illetve karakterbeágyazási dimenziót is konfiguráltuk. Emellett a "C" modell esetén a konvolúciós szűrők számát és azok hosszát, a MaxPooling-ablak méretét és a bemenet átlapolásánál alkalmazott lépésközt változtattuk. Az 1. táblázat összefoglalja a modelljeinkben használt hiperparaméterek végső értékeit.

1. táblázat. A szöveges és a prozódiai alapú modellek hiperparaméterei

| **Bemenet** | **Modell** | **Szekv. Hossza** | **Szótár Mérete** | **Beágyazási dimenzió** | **Rejtett állapotok** | **Batch mérete** | **Optimalizáló** | **Szűrők hossza** | **#Szűrők** | **Lépésköz** | **MaxPooling ablakméret** | *Patience* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Szavak | "W" | 200 | 100,000 | 300 | 512 | 128 | RMSProp | N/A | N/A | N/A | N/A | 3 |
| Karakterek | "C" | 200 | 100 | 80 | 512 | 128 | RMSProp | 6 | 70 | 2 | 25 | 3 |
| Prozódia | "P" | 200 | N/A | N/A | 512 | 16 | RMSProp | N/A | N/A | N/A | N/A | 3 |

A központozó rendszerek implementálásához a Keras keretrendszert [21] használtuk, a tanítást GPU-n végeztük el.

### 3.5. Hibrid modellek

A különböző inputokat páronként kombinálva ("karakter és szó" ("C+W"), "karakter és prozódia" ("C+P"), "szó és prozódia" ("W+P")) három különböző hibrid modellt vizsgáltunk meg, valamint egy negyediket is, mely mindhárom bemenetet egyszerre dolgozza fel ("C+W+P"). A hibrid modellekhez a különálló előtanított karakter- ("C") és szóalapú ("W") modellek súlyait is felhasználtuk, kombinálva a prozódia ("P") alapú modell bemenetével. Az összekapcsolás a "C" és/vagy "W" modellek softmax kimeneti aktivációs rétegeit megelőző BiLSTM rétegeken történt meg, illetve a softmax rétegekkel történő összeillesztést is kipróbáltuk. Az összeillesztett alsóbb rétegekhez hozzáadtunk még egy új, közös BiLSTM réteget és egy új softmax réteget; így állt össze a teljes hibrid hálózat.

## 4. Kísérleti eredmények

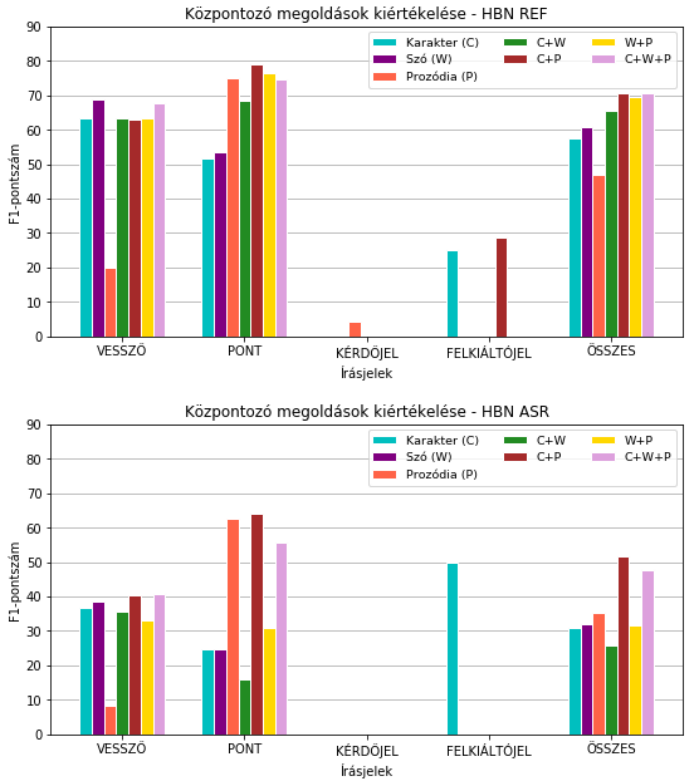
A következő fejezetben bemutatjuk a magyar nyelvű írásjelezési kísérleteink eredményeit. A részletes kiértékelést egy standard információ-visszakeresési mutató, az F1-érték mentén mutatjuk be, melyet az írásjelekre vonatkozó Pontosság (Pr)

és Fedés (Rc) értékekből származtattunk. Ezenkívül a legjobban teljesítő modellekhez megadjuk a Slot Error Rate (SER) [22] értéket is, amely egy metrikában egyszerre tükrözi az írásjel-visszaállításhoz kapcsolódó hibák minden lehetséges típusát - beszúrásokat (Ins), helyettesítéseket (Sub) és törléseket (Del), N helyes találat mellett:

$$SER = \frac{C(Ins) + C(Subs) + C(Del)}{C(slotok\_szama = N + Subs + Del)}, \quad (2)$$

ahol $C(.)$ a számláló operátor, a slot-ok pedig azon szavakat követő helyek a szövegben, amelyekben helyesen szerepel írásjel.

A Magyar Híranyag-adatbázis (HBN) kézi illetve az ASR átiratairaira vonatkozó eredmények a 2. ábrán láthatók.

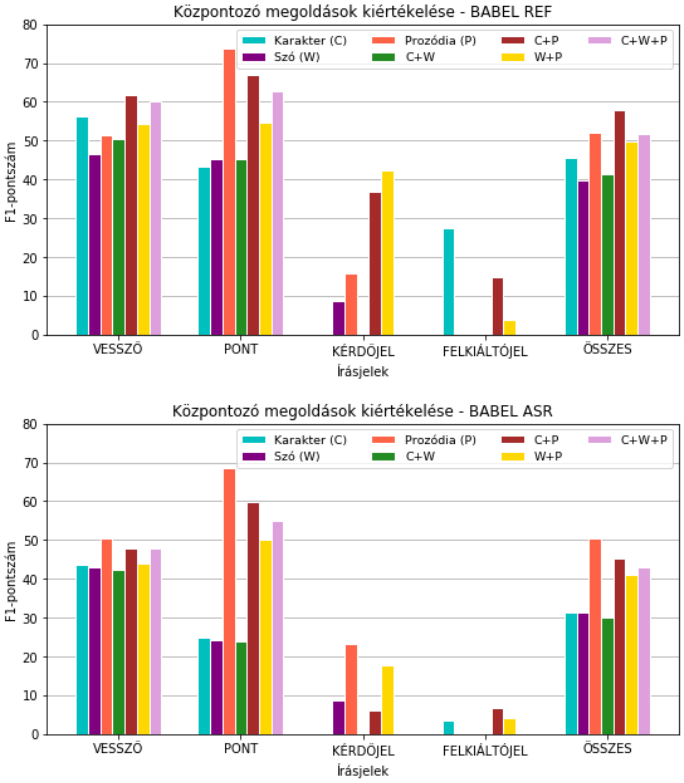



2. ábra: A Magyar Híranyag-adatbázisból származó kézi és ASR feliratok központozása

A szövegalapú ("C" és "W") modellekkel leginkább a vesszők visszaállítása lehetséges, mind a kézi (REF), mind az ASR átiratokon. Ezek a modellek a pont predikciójának tekintetében gyengébb mutatóval rendelkeznek, szemben a

prozódiai "P" modellel, mely ebben a tekintetben jól teljesít, viszont gyenge a vesszők jóslásában. Ígéretes tehát a két jellemzőkészlet kombinálása: a szövegalapú komponenseket a prozódiával kombinálva (akár párban ("C+P", "W+P"), akár hármasban ("C+W+P") további javulás tapasztalható az automatikus írásjelező modellek teljesítményében. A legjobb eredményt a "C+P" inputkombinációval értük el, mind a kézi átiratokon ($F1 = 70,7\%; SER = 45,1\%$), mind az ASR-kimeneten ($F1 = 51,8\%; SER = 78,2\%$).

A BABEL adatbázis kézi illetve az ASR átirataira vonatkozó eredmények a 3. ábrán láthatók.



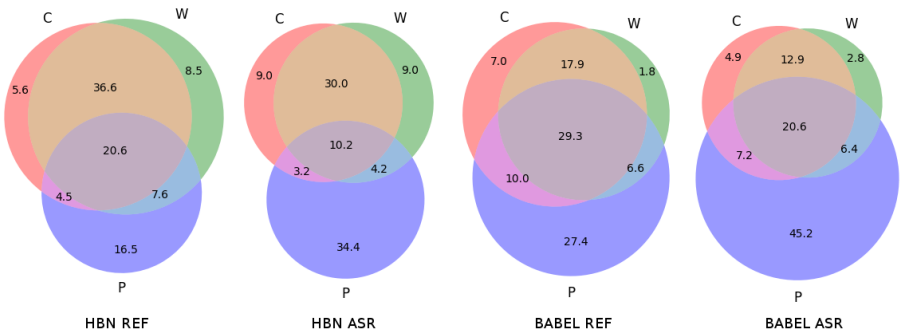

3. ábra: A BABEL adatbázis kézi és ASR átiratainak központozása

A HBN adatbázissal összevetve kiugró a "P" modell szerepe, amit a BABEL kontrollált és gondos artikulációjú felvételeivel magyarázunk. A legjobb eredményt a BABEL-es kézi átiratokon szintén a "C+P" inputkombinációval értük el ($F1 = 58,0\%; SER = 55,6\%$), viszont az ASR-kimeneten a "P" modell önmagában a leghatékonyabb, a magas szóhibaarány következtében a szöveges jellemzőkkel kiegészített hibridek nem tudtak magasabb teljesítményt elérni ($F1 = 50,3\%; SER = 71,7\%$).

A kérdések azonosításában meglepő módon a "P" modell a BABEL-en is gyengén teljesít, ennek okát az adatok kiegyensúlyozatlanságában látjuk, míg a HBN adatbázis esetén arra vezetjük vissza, hogy a kérdések és felkiáltások nem a megfelelő intonációval realizálódnak, hanem a deklaratív irányba tolódnak el jelentősen (de kevés is a minta ezekre a mondatokra az adatbázisban). Ezeket a feltételezéseinket lehallgatással is ellenőriztük, de megfelelnek a [5] irodalom megfigyeléseinek is.

Megjegyezzük, hogy a kísérleteket angol nyelvre is elvégeztük, noha hely hiányában az arra vonatkozó eredményeket nem mutatjuk be részletesen; ott a "P" modell kézi átiraton lényegesen gyengébb, de ASR átiraton szintén felerősödő szerepét tapasztaltuk. Fontos különbség, hogy angol nyelven a mondatvégi írásjelek (pontok) a "W" és a "C" modellekkel is pontosabban detektálhatók voltak kézi átiraton, mint a "P" modellel, illetve az ASR kimeneten a "W+P" hibrid bizonyult a leghatékonyabbnak, igaz „csupán” $p < 0,05$ szignifikanciaszint mellett.

A 4. ábrán Venn-diagramokkal mutatjuk be, hogy a "C","W" és "P" modellek milyen mértékben járulnak hozzá az írásjelek helyes visszaállításához (%-ban megadva). Habár a "P" modell összességében kevesebb írásjelet volt képes megfelelően beszúrni a szövegbe a HBN átiratokon (a gondosabban intonált BABEL-nél nem), de a szerepe látványos; a helyesen beszúrt írásjelek 15-25%-át egyedüliként fedte kézi átiratokon (REF), míg ASR kimeneten az írásjelek harmadát-felét egyedüliként képes detektálni. A kézi átiratokon a szövegalapú modellek (BABEL esetében enyhe) dominanciája figyelhető meg. A "P" modell az ASR átiratok esetén szépen javítja a központozó modell beszédfelismerési hibákkal szembeni robusztusságát.




4. ábra: A szövegalapú és prozódiai modellek kontribúciója a helyesen visszaállított írásjelek halmazát tekintve, a "HBN" és a BABEL korpuszon

Az eredmények alapján az alábbi következtetések rajzolódnak ki: a szakirodalomban is ismert a "P" modell jó szóhibatűrése, amelyet mi is demonstráltunk.

Magyar nyelvre a mondatvégi írásjeleket a szövegalapú modellek pontatlanabbul jelezték előre a vesszőkhöz képest, sőt, angol nyelvre is pontosabb ezen írásjelek predikciója. Ezt a magyar kevésbé kötött szórendjére és a szóalakok relatíve magas számára vezetjük vissza: az agglutináló nyelvek esetén - mint a magyar - a szóbeágyazások alkalmazása kevésbé hatékony; ennek egyfelől az az oka, hogy a több különböző előforduló szóalak miatt az OOV-arány is általában magasabb, mint például az angol nyelvben (esetünkben HBN-re az OOV-arány 8,6%, BABEL-re 11,8% volt), míg nagyobbrészt azt feltételezzük, hogy a kevésbé kötött szórend miatt nagy szókontextusra (akár a beágyazás alapjául szolgáló skip-gram kontextusablakán is kívülre) kiterjedő nagyobb változékonyság miatt a beágyazások szemantikus kapcsolatokat jósló képessége kevésbé robusztus. Karakter N-gramokkal és az ASR-szótár elemeire illesztett szóbeágyazásokkal lehetne javítani az OOV miatti problémán, ezzel a szemantikai pontosságot is növelve, ahogy azt a [23] cikkben több nyelvre be is mutatták. Ezek bevonásától jelen cikkben eltekintettünk, és a hibrid modellekben a karakteralapú modellünkből kinyert jellemzőket használtuk fel, amely kisebb mértékben, de szintén javította a robusztusságot.

Ezzel a hipotézissel összhangban van a szövegalapú modellek vesszőkre vonatkozó magas predikciós képessége is: a magyar nyelvben (is) két esetben gyakori a vesszők használata; egyrészt a kötőszavak előtt (melynek szerepe a különböző tagmondatok elválasztása), másrészt a felsorolásban. Az előbbi esetben a kötőszóhoz tartozó szóbeágyazás általában ismert, az utóbbi esetben pedig tipikusan a szemantikailag hasonló szavakat kapcsoljuk össze. Mindkét esetben megmutatkozik a szóbeágyazások segítő szerepe, melyet másutt a kevésbé kötött szórend miatti „csere-bere” lehetősége itt nem befolyásol.

Összegezve az írásjelezési eredményeket, az agglutináló és kötetlen szórendű magyar nyelven szignifikáns teljesítménynövekedés érhető el a karakterszintű és a prozódiai jellemzők bevonásával, a szóalapú baseline modellünkkel összehasonlítva. Kiemelve az ASR átiratok központozását, a karakter-prozódiai jellemzőpárost használó hibrid modell segítségével közel 40%-os relatív javulás érhető el F1-érték tekintetében, a valós ASR felhasználási körülményeket jól reprezentáló HBN korpuszon, mely $p < 0,01$ érték mellett is szignifikáns.

## 5. Összegzés

Cikkünkben különböző automatikus írásjelező modelleket mutattunk be, szöveges jellemzők (karakterek és szavak) és prozódiai jellemzők egyenkénti, valamint kombinált használatával. Fontos kiemelni a prozódiai modellek teljesítményét, mely a gépi beszédfelismerésből származó hibák ellenére is képes a hatékony írásjelezésre, szemben a szóalapú modellel, mely meglehetősen érzékeny azokra. Kismértékben a karakteralapú modell is emeli a szóhibatűrést. A szóalapú modell teljesítményét befolyásolja az is, hogy a kevésbé kötött szórend és a nagy szótárméret miatt a szóbeágyazások által biztosított szemantikai modellező képesség és koherencia is csak korlátozottabb mértékben tud érvényesülni. A karakter-prozódia jellemzőket együttesen használó hibrid modell bizonyult a leghatéko-

nyabbnak a kézi átiratokon, míg az ASR esetben a karakter-prozódia párossal működő hibrid modellel (a HBN korpuszon), illetve a BABEL-en a prozódiai hálóval értük el a legjobb eredményt, F1 és SER tekintetében. Az írásjelekre kitérve, a prozódiai modell erőssége a pontok, míg a szövegalapúaké a vesszők visszaállítása. Úgy véljük, hogy a karakteralapú és prozódiai modellekhez kapcsolódó eredményeink és megfigyeléseink a többi agglutináló nyelvre is érvényesek lehetnek; ezeket további vizsgálatokkal lenne érdemes alátámasztani.

## Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatalnak, amely az FK-124413 projekt keretében a cikkben ismertetésre került kutatást támogatta.

## Hivatkozások

1. Tilk, O., Alumäe, T.: LSTM for punctuation restoration in speech transcripts. In: Proceedings of Interspeech. (2015) 683–687
2. Tilk, O., Alumäe, T.: Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. In: Proceedings of Interspeech. (2016) 3047–3051
3. Moró, A., Szaszák, G.: A phonological phrase sequence modelling approach for resource efficient and robust real-time punctuation recovery. In: Proceedings of Interspeech. (2017)
4. Tündik, M.Á., Tarján, B., Szaszák, G.: Low Latency MaxEnt-and RNN-Based Word Sequence Models for Punctuation Restoration of Closed Caption Data. In: International Conference on Statistical Language and Speech Processing, Springer (2017) 155–166
5. Klejch, O., Bell, P., Renals, S.: Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE (2017) 5700–5704
6. Hardmeier, C.: A neural model for part-of-speech tagging in historical texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. (2016) 922–931
7. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: AAAI. (2016) 2741–2749
8. Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named entity recognition with character-level models. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, Association for Computational Linguistics (2003) 180–183
9. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. arXiv preprint arXiv:1511.08308 (2015)
10. Chung, J., Cho, K., Bengio, Y.: A character-level decoder without explicit segmentation for neural machine translation. arXiv preprint arXiv:1603.06147 (2016)
11. dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. (2014) 69–78

12. Gale, W., Parthasarathy, S.: Experiments in character-level neural network models for punctuation. Proc. Interspeech 2017 (2017) 2794–2798
13. Abdel-Hamid, O., Deng, L., Yu, D.: Exploring convolutional neural network structures and optimization techniques for speech recognition. In: Interspeech. Volume 2013. (2013) 1173–5
14. Tóth, L.: Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE (2014) 190–194
15. McNamara, D.S., Kintsch, E., Songer, N.B., Kintsch, W.: Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. Cognition and instruction **14**(1) (1996) 1–43
16. Roach, P., Arnfield, S., Barry, W., Baltova, J., Boldea, M., Fourcin, A., Gonet, W., Gubrynowicz, R., Hallum, E., Lamel, L., et al.: BABEL: An Eastern European multi-language database. In: Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On. Volume 3., IEEE (1996) 1892–1893
17. Teleki, C., Szabolcs, V., Levente, T.S., Klára, V.: Development and evaluation of a Hungarian Broadcast News Database. In: Forum Acusticum. (2005)
18. Tündik, M.A., Szaszák, G.: Joint Word-and Character-level Embedding CNN-RNN Models for Punctuation Restoration. In: Proceedings of the 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2018), IEEE (2018) 135–140
19. Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimäki, T., Pylkkönen, J., Alumäe, T., Saraclar, M.: Unlimited vocabulary speech recognition for agglutinative languages. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics -, Morristown, NJ, USA, Association for Computational Linguistics (2006) 487–494
20. Makrai, M.: Filtering Wiktionary triangles by linear mapping between distributed models. In: Proceedings of LREC. (2016) 2776–2770
21. Chollet, F.: Keras: Theano-based deep learning library. Code: https://github.com/fchollet. Documentation: http://keras.io (2015)
22. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of DARPA broadcast news workshop. (1999) 249–252
23. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)

# Mély neuronhálós beszédfelismerők működésének értelmező elemzése

Grósz Tamás, Tóth László

Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék
Szegedi Tudományegyetem, Informatikai Intézet
Szeged, Árpád tér 2.
{groszt, tothl}@inf.u-szeged.hu

**Kivonat** Manapság nyilvánvalóvá vált, hogy beszédfelismerésben a mély neuronhálós modellek teljesítenek a legjobban, azonban fontos kérdés, hogy miért működnek ilyen jól. Az utóbbi pár évben megnövekedett a igény, hogy a mély hálókat ne csupán fekete dobozként kezeljük, hanem azok belső működését próbáljuk megérteni, interpretálni is. Az interpretálásra több eszköz is létezik, jelen cikkben mi két beágyazási technikát alkalmazunk annak vizsgálatára, hogy egy neuronhálós beszédfelismerőn belül pontosan mi történik használat közben. A vizsgált háló egy magyar nyelvű beszédfelismerő része, amelyet egy híradós adatbázison tanítottunk. A háló struktúráját tekintve nem rendelkezik könnyen értelmezhető, keskeny üvegnyak (bottleneck) réteggel, ezért a neuronháló nagy méretű rejtett rétegeinek kimeneteit tanulmányoztuk. Első vizsgálataink során arra a kérdésre kerestük a választ, hogy mennyire jól különíti el az adott réteg a magán- és mássalhangzókat, valamint a csendes részeket. A következő lépésben azt tanulmányoztuk, hogy a magán- és mássalhangzókon belül más csoportok reprezentációja is azonosítható-e. Eredményeink alapján megállapítható, hogy a mély háló számos olyan tulajdonságot is megtanult a beszédhangokról, amelyek felismerésére explicit módon nem tanítottuk a hálót.
**Kulcsszavak:** mély neuronhálók, interpretálhatóság, beszédfelismerés

## 1. Bevezetés

Az elmúlt pár évben egyértelművé vált, hogy a mély neuronhálós beszédfelismerők sokkal jobb eredményeket tudnak elérni, mint más technikák [1]. Megjelenésük óta főleg a technológia finomítására fókuszált a beszédfeldolgozó közösség, minél jobb eredmények elérése céljából és kevésbé törődtek annak a fontos kérdésnek a megválaszolásával, hogy miért is működnek ilyen jól a mély neuronhálók beszédfelismerésben. Ez a trend változni látszik; a közelmúltban több tanulmány is megjelent, amelyek a beszédfelismerőkben található hálók működését elemzik és az interpretálhatóság javítását célozzák [2,3,4,5,6].

Az interpretálhatóság még nem egy teljesen kiforrott tématerület, ám egyre fontosabbá válik, ahogy a mesterséges intelligencia mindennapjaink részévé válik, hiszen az emberek többsége nehezen bízik meg egy olyan rendszerben, amit

nem ért, nem tudja miért működik. Egy betanított modell értelmezésére többféle módszer is létezik; globális vizsgálat esetén magát a modellt próbáljuk értelmezni, míg lokális esetben egy adott bemenethez tartozó kimenetekhez keresünk magyarázatot [7]. Jelen munkában mi ez utóbbira fókuszálunk, azaz azt próbáljuk megmutatni, hogy adott bemenet esetén mi történik a hálózat belsejében. A lokális értelmezés egyik fő eszköze a rejtett rétegek aktivációinak vizualizálása, ehhez viszont át kell transzformálni az általában magas dimenziószámú vektorokat alacsonyabb (általában kettő) dimenziós térbe, hogy emberek számára is átlátható legyen. Ezt a transzformációt dimenzióredukciós módszerekkel tudjuk elvégezni, amelyekből rengeteg létezik. Ezek közül mi két módszert alkalmaztunk vizsgálataink során: a neuronhálókhoz javasolt t-sztochasztikus szomszéd beágyazása (t-Stochastic Neighbor Embedding, t-SNE) [8] és a közelmúltban javasolt egyenletes sokaság becslése és projekciója (Uniform Manifold Approximation and Projection, UMAP) módszert [9].

A korábbi művekben [3,6] speciális neuronháló struktúrát használtak, úgynevezett üvegnyak (bottleneck) réteget alkalmazva. Ez lényegében egy, a háló többi rétegéhez képest kevesebb neuront tartalmazó rejtett réteg, ezen szűk rétegnek a kimeneteit könnyen lehet vizsgálni különböző beágyazási technikákkal. Mi ezzel ellentétben egy már korábban betanított háló működésének elemzését tűztük ki célként, így nem alkalmaztunk szűkített rejtett réteget. Vizsgálataink során két népszerű beágyazási technika segítségével vizsgáltuk meg, hogy egy jól működő magyar nyelvű beszédfelismerő neuronhálója pontosan hogyan is működik. A hálónk egy 5 rejtett réteges háló volt, minden rejtett rétegben 1000 ReLU neuron található (struktúrája és tanítási paraméterei megegyeznek a [10] műben leírtakkal). A neuronháló tanításához egy magyar nyelvű híradós adatbázist [11] használtunk. Az interpretálhatóság céljából kiértékeltük a hálót egy kellően hosszú hangfájlon, amelyet a teszt halmazból választottunk, majd több rejtett réteg kimenetét is beágyaztuk a kettő dimenziós térbe, hogy vizualizálhassuk, milyen belső reprezentációk (fonémakategóriák) alakultak ki a hálóban.

## 2. Beágyazási technikák

Ahogy korábban említettük, több beágyazási technika is létezik. Jelen munkában, hogy biztosan ne vonjunk le téves következtetéseket egyetlen módszer eredményei alapján, két lehetséges technikára fókuszáltunk. Az első módszer, a t-SNE algoritmus [8] eredetileg is mély hálókban található rejtett rétegek kimeneteinek transzformálására lett javasolva, illetve az UMAP beágyazás [9], amely a t-SNE egyik legújabb alternatívája. A továbbiakban röviden bemutatjuk ezen két módszert.

### 2.1. T-SNE

A t-SNE egy felügyelet nélküli módszer, amelynek segítségével mély hálók rejtett rétegeinek kimeneti értékeit ágyazhatjuk be alacsony dimenziós térbe [8]. Ezen

beágyazás segítségével vizualizálhatjuk a háló belső működését annak interpretálása céljából.

A módszer maga tekinthető dimenzióredukciós módszernek, amelynek célja, hogy a lehető legtöbbet megőrizzen a magas dimenziós struktúrából miközben áttranszformálja az adatot egy lényegesen alacsonyabb dimenziós térbe. Esetünkben a rejtett rétegek kimenetei 1000 dimenziós vektorokat generáltak, amelyeket vizualizálás céljából kettő dimenziós síkra redukálunk.

A t-SNE algoritmus két fontos lépésből áll. Az első lépés során a magas dimenziós térben az adatpontok közötti euklideszi távolságot alakítja át feltételes valószínűségekké, amelyek a pontok közötti hasonlóságot fogják reprezentálni. A második szakaszban maga a beágyazás történik, a pontok elhelyezése az alacsonyabb dimenziós térben. Ezt egy optimalizáló algoritmus végzi el, a korábban kiszámolt hasonlóságok alapján.

Tekintsük első körben meg, hogyan pontosan hogyan számolható hasonlóság két pont között magas dimenzióban a t-SNE módszer segítségével. Tegyük fel, hogy $x_i$ és $x_j$ két pont az $N$-dimenziós térben, ekkor a módszer első lépésben egy feltételes valószínűséget ($p_{j|i}$) definiál:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2))}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}. \tag{1}$$

Ez a valószínűség a szerzők szerint úgy értelmezhető, hogy mekkora a valószínűsége annak, hogy $x_i$ pont az $x_j$-t választja szomszédjának, amennyiben a szomszédok kiválasztásának valószínűsége arányos egy $x_i$ középpontú Gauss eloszlással, aminek szórása a $\sigma_i^2$. A szórások beállítását a felező módszerrel tudjuk elvégezni úgy, hogy a feltételes eloszlások perplexitása egy előre megadott értéknek feleljen meg, ezzel tudjuk elérni, hogy a tér sűrűbb részeiben kisebb $\sigma_i^2$ értékek lesznek. A hasonlóságot a pontok között $N$ dimenzióban a $p_{j|i}$ valószínűségek alapján számolhatjuk:

$$d_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{2}$$

és $i = j$ esetén $d_{ij} = 0$.

Maga a transzformáció alacsonyabb ($D$) térbe egy optimalizálási problémának tekinthető, amihez első lépésben definiálnunk kell egy hasonlóságfüggvényt a $D$ dimenziós térben is. Ezen függvénnyel próbáljuk mérni a hasonlóságot a $x_i$ és $x_j$ pontok transzformáltja, az $y_i$ és $y_j$ pontok között:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}, \tag{3}$$

amennyiben $i = j$, akkor $q_{ij} = 0$. A képletből látható, hogy 1 szabadsági fokú Student-féle t-eloszlást (más néven Cauchy eloszlás) használ a módszer, aminek hasznos tulajdonsága, hogy a távoli pontok beágyazása majdnem teljesen invariáns lesz a tér átskálázására, illetve távoli klaszterek pontjai hasonló módon befolyásolják egy pont elhelyezkedését, mint ha különálló pontok lennének. Ez utóbbi tulajdonság az optimalizáló számára lesz hasznos.

Végül az $y_i$ pontok elhelyezéséhez iteratív módon a következő Kullback-Leibler divergenciát minimalizáljuk:

$$KL(P||Q) = \sum_{i \neq j} d_{ij} log \frac{d_{ij}}{q_{ij}}. \quad (4)$$

Ez a módszer az egyik legszélesebb körben elterjedt technika rejtett rétegek aktivációinak vizualizálására és elemzésére, számos területen alkalmazták már pl. képfeldolgozásban [12], természetes nyelvi feldolgozásban [13] és beszédfelismerésben [6]. Hátránya, hogy számos paramétert (perplexitás, optimalizálási iterációk száma, stb.) kell megfelelően beállítanunk ahhoz, hogy jól működjön.

### 2.2. UMAP beágyazás

Az UMAP módszer megértéséhez fontos ismernünk a sokaság (manifold) fogalmát, amit röviden úgy lehet jellemezni, hogy egy olyan topológiai tér, amely lokálisan minden pont környezetében homeomorf a megfelelő dimenziós Euklideszi tér egy-egy nyílt halmazával [14]. A módszer három fontos feltételezésen alapszik:

- az adat egyenletesen oszlik el egy Riemann sokaságon,
- a Riemann metrika lokálisan konstans (vagy becsülhető úgy),
- a sokaság lokálisan összefüggő.

Ezen feltevések alapján az algoritmus első lépésben egy sokaságot keres, amelyen a magas dimenziós adat közel egyenletesen oszlik el, ami természetesen valós adat esetén nem feltétlenül teljesül. A probléma megoldására egy Riemann metrikát kell keresnünk, aminek használata esetén teljesül, hogy a pontok egyenletesen oszlanak el a sokaságon. Ezen Riemann metrika használatával lényegében különböző távolságokat használunk minden pont esetén lokálisan és ezen távolságok nem feltétlenül lesznek kompatibilisek. Következő lépésben a módszer ezeket az inkompatibilis lokális adatokat a sokaságon egyesíti majd átalakítja egy fuzzy topológiai reprezentációvá.

A beágyazást itt is egy optimalizálási problémamegoldásával végezzük el, mégpedig úgy, hogy az alacsonyabb dimenzióban elhelyezett pontokhoz is kinyerjük azoknak a topológiai reprezentációját (hasonló módon mint a magas dimenzió esetén) és a két fuzzy topológiai reprezentáció kereszt-entrópiáját minimalizáljuk a beágyazott pontok átmozgatásával. A módszer részletesebben az eredeti műben [9] kerül bemutatásra a matematikai háttérrel együtt.

Az UMAP módszer 2018-ban jelent meg, így még nem terjedt el olyan széles körben, mint a t-SNE, de használata több szempontból is előnyösebb. Talán a legfontosabb tulajdonsága, hogy lényegesen gyorsabban működik mint a t-SNE nagy méretű és magas dimenziós adatbázisok esetén. A sebességen túl a szerzők szerint az UMAP jobban megőrzi az adatban található globális struktúrát mint a t-SNE módszer [9], ez utóbbi állítást a mi kísérleteink is igazolták.

| Csoport | fonémák |
|---|---|
| magánhangzók | |
| mély hangrendű | a, á, u, ú, o, ó |
| magas hangrendű | e, é, i, í, ö, ő, ü, ű |
| mássalhangzók | |
| zárhangok | p, b, t, d, k, g, ty, gy |
| réshangok | f, v, s, sz, z, zs, h |
| zárréshangok | c, cs, dz, dzs |
| nazális hangok | m, n, ny |
| egyéb | l, ly, r, j |

1. táblázat. A vizsgálataink során használt beszédhang-kategóriák.

## 3. Beszédhang-kategóriák

Az adatokon végzett dimenzióredukció után fontos, hogy megvizsgáljuk, milyen klaszterek alakultak ki. Ehhez első lépésben 3 kategória elkülönülését vizsgáltuk, a magán- és mássalhangzók mellett a csend kategóriába soroltuk azokat a részeket, ahol nem volt beszéd,valamint a zárhangok (closure) szakaszait is. Ezen szinten főleg arra voltunk kíváncsiak, hogy mennyire különülnek el a magán- és mássalhangzók egymástól, hiszen a csendes részeket elég nagy pontossággal felismerte a rendszer, így azt valószínűleg jól elkülönítette a másik két csoporttól. A következő lépésben a magán- és mássalhangzókat osztottuk további kategóriákra, a magánhangzókat hangrend szerint, a mássalhangzókat pedig a képzés módja szerint, remélve, hogy a neuronháló is valami hasonló belső felosztást alakított ki anélkül, hogy erre külön tanítottuk volna. A kialakított csoportokat az 1. táblázat foglalja össze.

## 4. Eredmények

A kísérleteink során a teszthalmazból kiválasztottunk egy hangfájlt, amelyhez a flat-start során használt rendszerünkkel készítettünk kényszerített illesztéssel időben illesztett címkéket. A következő lépésben kiértékeltük a mély hálónkat a hangfájlon és elmentettük a rejtett rétegek kimeneti értékeit. A beágyazás során a t-SNE esetén az első rejtett réteg kimeneteit felhasználva, a beágyazás minőségét vizuálisan értékelve állítottuk be a módszer paramétereit (a perplexitást 50-re, az iterációszámot pedig 5000-re). A továbbiakban is ezeket az értékeket használtuk. UMAP esetén könnyebb volt a helyzetünk, mivel az alapértelmezett paraméterekkel is jól működött az algoritmus, nem volt szükség azok beállítására. Tapasztalataink alapján az UMAP futtatása nagyjából negyed annyi időt igényelt, mint a t-SNE.

Első lépésben megvizsgáltuk, hogy a kimeneti vektoraink mennyire ritkák, hiszen az ismert, hogy ReLU aktivációs függvény használata esetén a neuronok jelentős része inaktív lesz, tehát nullát ad kimenetként. Megfigyelhető, hogy a

| Rejtett réteg sorszáma | Aktivitás |
|---|---|
| 1 | 35.0% |
| 2 | 27.6% |
| 3 | 24.9% |
| 4 | 21.9% |
| 5 | 25.6% |

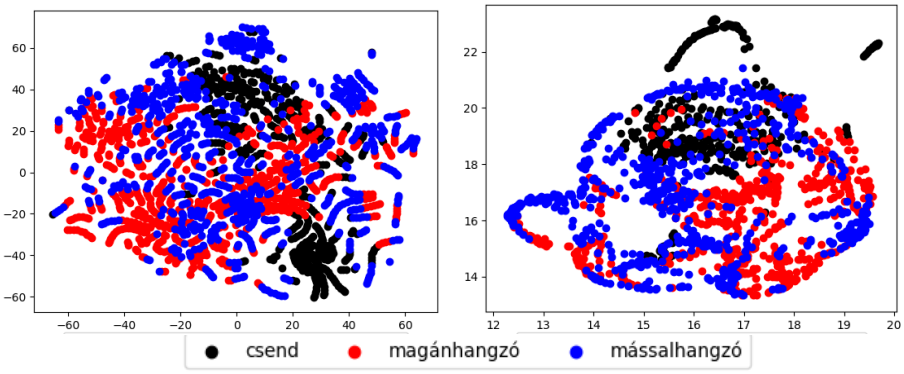
2. táblázat. A rejtett rétegekben az aktív (nem 0 kimenetet adó) neuronok aránya, a rétegek sorszámozása a bemenet felől a kimenet felé növekszik.

legnagyobb aktivitás a bemenetet figyelő rejtett rétegben volt, a neuronok közel 35%-a volt aktív. Érdekesség, hogy a kimenet felé haladva a magasabb rejtett rétegekben az aktív egységek száma csökken, azaz egyre kevesebb neuronnal nyerünk ki hasznos információt, de a kimeneti réteg alatti rétegben hirtelen megnövekszik a nem nulla kimenetek aránya. Véleményünk szerint a magyarázat az lehet erre, hogy a kimeneti réteg ezen réteg kimeneteire támaszkodva hoz döntést, ezért szükséges nagyobb arányú aktivitás. Ezen hipotézisünk igazolásához további vizsgálatok lennének szükségesek, hogy megvizsgáljuk vajon ez a jelenség más rejtett réteg-szám esetén is jelentkezik-e.

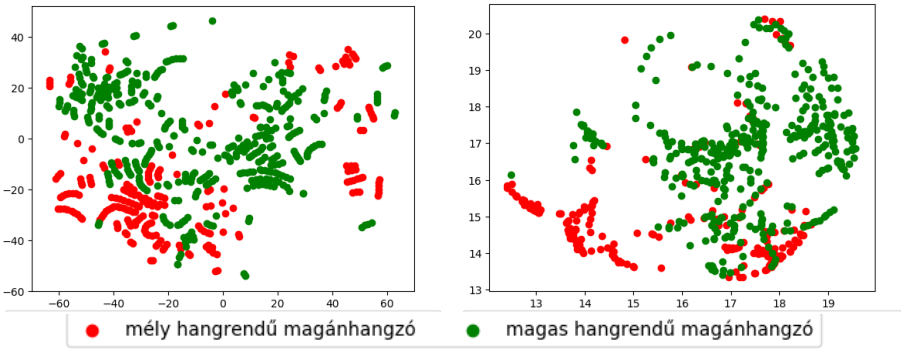
Miután megvizsgáltuk a rétegek aktivitását, figyelmünket a két fontos rétegre fókuszáljuk; a bemeneti réteghez csatolt első rejtett rétegre valamint a kimeneti réteg által figyelt utolsó rejtett rétegre. Tekintsük meg először, hogy egész pontosan milyen kimeneteket generált a legelső rejtett réteg, azaz milyen alacsony szintű jellemzőket nyert ki a bemenetből, azok mennyire jól szeparálják a korábban ismertetett beszédhang-kategóriákat. Első lépésben tekintsük az 1. ábrát, amelyen minden adatkerethez beágyaztuk kettő dimenzióba az első rejtett réteg kimenetét, majd az időben illesztett címkéink alapján minden ponthoz egy kategóriát rendeltünk. Megállapíthatjuk, hogy két csend klaszter alakult ki, az egyik a bemondás elején, végén, illetve a szavak között hallható csendnek felel meg, míg a másik klaszter a szavakban előforduló zár (closure), ez utóbbit a mássalhangzókkal keverve láthatjuk az ábrán. Fontos megemlíteni, hogy az ábrákon láthatunk majd 1-1 kiugró pontot, amely más kategóriák klasztereibe keveredett, ezek általában a fonémahatárok környékére eső kimenetek, ahol a címke bizonytalan, hiszen az időbeli illesztést egy másik háló végezte. Ezt a jelenséget tovább erősítette a tény, hogy három állapotú fonémamodellt használtunk, azaz feltételezzük, hogy minden hang legalább 3 keret hosszú, ami a valóságban nem mindig teljesül.

A magán- és mássalhangzókkal kapcsolatban azt állapíthatjuk meg, hogy ugyan nem teljesen elkülöníthetőek két dimenzióban, de itt is kialakultak csoportok. A továbbiakban ezeket elemezzük alaposabban.

A magánhangzókat tovább vizsgálva a 2. ábrán láthatjuk, hogy már elkezdődött a magas és mély hangrendűek különválasztása, azonban ez még nem tökéletes.



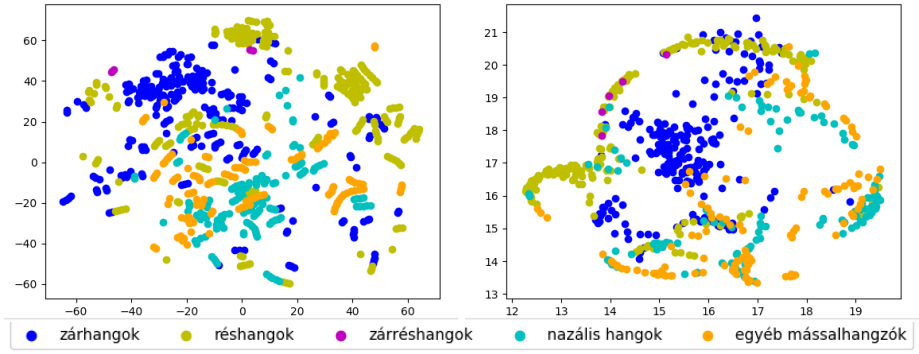

1. ábra: Az első rejtett réteg kimenetének beágyazása, balra a t-sne, jobbra pedig az UMAP módszerrel.



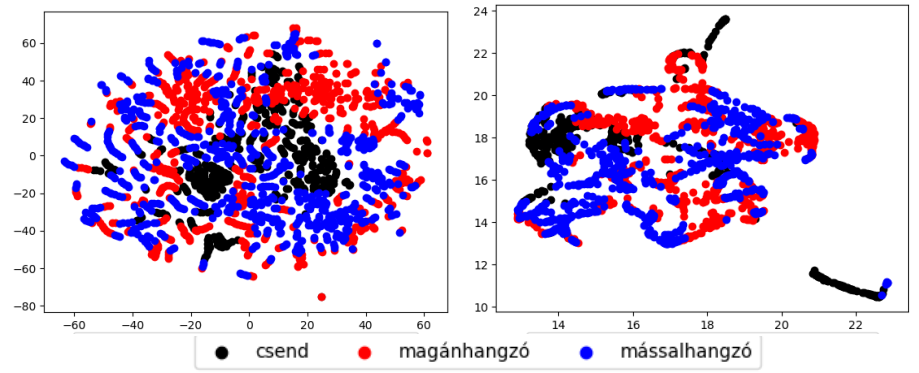

2. ábra: A magánhangzók kategorizálása az első rejtett réteg alapján, balra a t-sne, jobbra pedig az UMAP módszerrel.

Mássalhangzók esetén jól látható a 3. ábrán, hogy a zár- és réshangok elkülönülnek egymástól, azonban a többi kategória nem igazán van megkülönböztetve a háló által. Érdekesség, hogy a réshangok esetén két külön klaszter látszódik kialakulni, t-SNE esetén jól láthatóan, UMAP esetén kevésbé látványosan, de ott is látható egy szakadás a sárga klaszterben a (15,20) pont környékén. Tovább elemezve ezen két csoportot megállapítottuk, hogy az egyikben főleg zöngés, a másikban pedig zöngétlen réshangok találhatóak, tehát a háló erre vonatkozó információt is kinyert.

A legmagasabb szintű jellemzőket kinyerő réteget vizsgálva (4. ábra) látható, hogy az első réteghez hasonló módon itt sem különülnek el markánsan a magán- és mássalhangzók, de a csendes részeket itt három részre bontotta a háló, ismét megkülönböztetve a csendet a zártól. A két elkülönülő csoport közül a t-SNE esetén a nagyobb rész (a (-15,-15) környékén lévő klaszter) a szavak közötti csendnek felelt meg, a (-10,-45) körüli pedig a felvétel elején és végén hallható




3. ábra: A mássalhangzók kategorizálása az első rejtett réteg alapján, balra a t-sne, jobbra pedig az UMAP módszerrel.

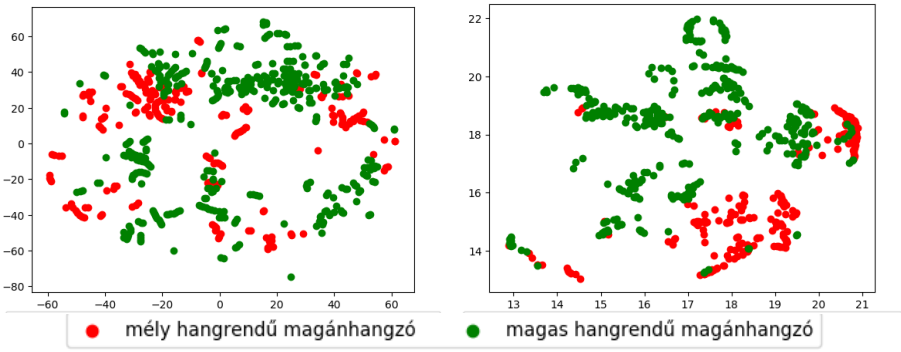



4. ábra: Az legfelső rejtett réteg kimenetének beágyazása, balra a t-SNE, jobbra pedig az UMAP módszerrel.

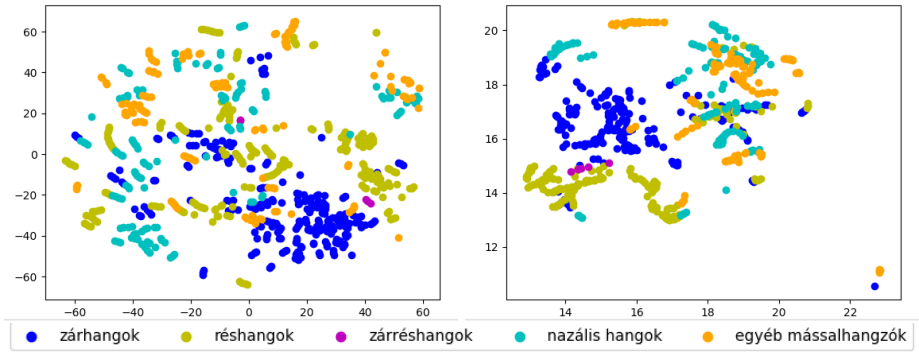
csend. UMAP esetén a két kinyúló rész közül a felső felvétel elején és végén lévő csendes rész, az alsó elkülönülő rész pedig a szavak közötti csend. Az elkülönülés már az első réteg kimeneti esetén is elkezdődött, de nem volt ennyire látványos. Ezek alapján megállapíthatjuk, hogy ez a réteg nem csupán felismeri a csendet, hanem különbséget tesz a hosszabb csend és a szavak közötti rövidebb csend között is.

Magánhangzók esetén azt láthatjuk a 5. ábrán, hogy míg UMAP alapján elég jól elkülönültek a magas és mély hangok, a t-SNE módszer esetén ez kevésbé látható. Ennek egy lehetséges magyarázata, hogy a t-SNE esetén a paramétereket újra be kellett volna állítani a jobb működés érdekében, és lehetséges, hogy nem az optimális értékeket választottuk.

A 6. ábrán a mássalhangzókhoz tartozó kimenetek beágyazása látható, az első rejtett réteghez hasonlóan itt is jól elkülönülnek a rés- és zárhangok, illetve




5. ábra: A magánhangzók kategorizálása a legfelső rejtett réteg alapján, balra a t-SNE, jobbra pedig az UMAP módszerrel.




6. ábra: A mássalhangzók kategorizálása a legfelső rejtett réteg alapján, balra a t-SNE, jobbra pedig az UMAP módszerrel.

a zárréshangok klasztere a kettő közé kerül. Az UMAP módszerrel ismét látható, hogy kialakul a zöngés és zöngétlen zárhangok csoportja, amelyek ezen rétegben már sokkal sűrűbben helyezkednek el. Tekintve, hogy a neuronháló ezen rétege se igazán tesz különbséget a nazális és egyéb magánhangzók között kijelenthetjük, hogy a beszédfelismerő ilyen jellegű információt nem tanult meg kinyerni a tanító adatból.

## 5. Összegzés

Munkánk során egy magyar nyelvű beszédfelismerő mély neuronhálós modulját elemeztük interpretálhatóság céljából. A hálót kiértékeltük egy teszt hangfájlon, majd a kapott rejtett rétegek kimeneteit vizsgáltuk meg alaposabban. A legelső és legfelső rejtett rétegek aktivációs értékeit két beágyazási módszerrel (t-SNE és

UMAP) levetítettük kettő dimenziós térbe, hogy ábrázolhassuk azokat elemzés céljából.

A kapott beágyazások alapján megállapítható, hogy a háló már alacsonyabb rétegeiben is elkezdte különválasztani a csendes részeket a beszédet tartalmazó résztől, illetve megkülönböztette a zárt és a valódi csendet. Magasabb szinten pedig már a szavak közötti csendet is elkülönítette a felvétel elején és végén hallható csendtől. A magánhangzók esetén a legfelső rétegben a magas és mély hangrendű hangok megkülönböztetését is megfigyelhetjük. Mássalhangzókat tekintve két fontos csoportot tanult meg felismerni a háló, mégpedig a zár és a réshangokat, utóbbi esetén még a zöngésséget is figyelembe vette a neuronháló. Az eredményeink alapján megállapítható, hogy a beszédfelismerő számos olyan dolgot is megtanult, amit explicit módon nem vártunk el tőle.

## Köszönetnyilvánítás

Grósz Tamás munkáját a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal Mesterséges Intelligencia Nemzeti Kiválósági Programja támogatta a 2018-1.2.1-NKP-2018-00008 azonosítójú projekt keretében.

Tóth Lászlót az MTA Bolyai János Kutatási Ösztöndíja, valamint az Emberi Erőforrások Minisztériuma ÚNKP-18-4 kódszámú Új Nemzeti Kiválóság Programja támogatta.

A kutatást az Emberi Erőforrások Minisztériuma Emberi Erőforrások Minisztériuma 20391-3/2018/FEKUSTRAT kódjelű pályázata támogatta. A kutatáshoz használt grafikus kártyát az NVIDIA Corporation ajándékozta csoportunknak.

## Hivatkozások

1. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine **29**(6) (2012) 82–97
2. Mohamed, A., Hinton, G., Penn, G.: Understanding how deep belief networks perform acoustic modelling. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (2012) 4273–4276
3. Vu, N.T., Weiner, J., Schultz, T.: Investigating the learning effect of multilingual bottle-neck features for ASR. In: Proc. Interspeech. (2014) Interspeech 2014.
4. Tan, S., Sim, K.C., Gales, M.: Improving the interpretability of deep neural networks with stimulated learning. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). (2015) 617–623
5. Nagamine, T., Seltzer, M.L., Mesgarani, N.: Exploring how deep neural networks form phonemic categories. In: INTERSPEECH. (2015)
6. Bai, L., Weber, P., Jančovič, P., Russell, M.: Exploring how phone classification neural networks learn phonetic information by visualising and interpreting bottleneck features. In: Proc. Interspeech. (2018) 1472–1476

7. Lipton, Z.C.: The mythos of model interpretability. ACM Queue **16**(3) (2018) 30:31–30:57
8. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov) (2008) 2579–2605
9. McInnes, L., Healy, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
10. Grósz, T.: Training Methods for Deep Neural Network-Based Acoustic Models in Speech Recognition. PhD thesis (2018)
11. Tóth, L., Grósz, T.: A comparison of deep neural network training methods for large vocabulary speech recognition. In: Proceedings of TSD. (2013) 36–43
12. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639) (2017) 115
13. Narasimhan, K., Kulkarni, T., Barzilay, R.: Language understanding for text-based games using deep reinforcement learning. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2015) 1–11
14. Lee, J.M.: Riemannian manifolds: an introduction to curvature. Volume 176. Springer Science & Business Media (2006)

# Szintaxis

# Parsing noun phrases with Interpreted Regular Tree Grammars

Evelin Ács[1], Ákos Holló-Szabó[1], Gábor Recski[2]

[1] Department of Automation and Applied Informatics
Budapest University of Technology and Economics
[2] Apollo.AI

**Abstract.** Several common tasks in natural language processing (NLP) involve graph transformation, in particular those that handle syntactic trees, dependency structures such as Universal Dependencies (UD) [1], or semantic graphs such as AMR [2] and 4lang [3]. Interpreted Regular Tree Grammars (IRTGs) [4] encode the correspondence between sets of such structures and have in recent years been used to perform both syntactic and semantic parsing. In this paper we introduce our tool that is capable of automatic IRTG generation. Our IRTG covers 83% of noun phrases (NPs) from the Wall Street Journal section of the Penn Treebank and a pilot experiment had also been made for retrieving surface realizations from UD graphs using independent data. We also describe this generated IRTG which allows for simultaneous generation of structures of various types and can be used for semantic parsing, generation, and semantics-driven translation.

## 1 Introduction

One of the most limiting factors in common tasks in NLP, such as machine translation, question answering and natural language inference, is the absence of high-quality deep semantic parsing. The state-of-the-art tools are mostly based on deep learning, which encode the meaning of words in multidimensional vector spaces, and the understanding of the structures of these representations is very limited. Another approach is using semantic representations based on concept networks. The automatic generation of these representations is also limited, but they facilitate more explicit analysis of tasks close to general artificial intelligence, such as natural language inference or machine comprehension. Syntactic parsers for natural language are key components for most processing pipelines within human language technologies (HLT). A common approach taken by modern HLT systems is dependency parsing, which maps raw text to directed acyclic graphs over words of each input sentence. One of the multiple dependency parsing mechanisms implemented in the Stanford Parser [5] creates dependency graphs by applying rule-based transformations on constituency structures output by a probabilistic context-free grammar (PCFG) parser. The dep_to_4lang component of the 4lang library builds graphs of syntax-independent concepts from the output of any dependency parser that conforms to the Universal Dependencies

format, also using simple template-matching. In this end-to-end semantic parser pipeline all components implement a form of graph transformation so its functionality can be unified in a single graph grammar. We use Interpreted Regular Tree Grammars [4] to simultaneously encode transformations between strings, phrase structure trees and UD and 4lang graphs. Section 2 provides an overview of the used or related tools and technologies, including the dependency parsing component of the Stanford Parser, Universal Dependencies (supported by the Stanford Parser), the 4lang formalism, and the IRTGs. In Section 3 we present a regular tree grammar with four interpretations, corresponding to strings, phrase structure trees, UD graphs, and 4lang graphs. Our grammar is non-deterministic, which means that given an input structure, it can generate more than one derivations, resulting in many different output structure configurations regarding all interpretations. Such a grammar allows converting from any of the above algebraic structures to any or all of the others, e.g. generating English text from dependency graphs. It can also be trained on correspondences between grammatical and semantic structures and surface realizations. The grammar discussed in this paper covers 83% of NPs of the Wall Street Journal section of the Penn Treebank and a pilot experiment was also executed on a small independent (i. e. not used for generating the grammar) test data for generating English text from UD graphs, which resulted in a limited number of successful parses, and also revealed some problems that need further investigation (discussed in Section 3.2).

## 2 Background

### 2.1 Dependency parsing with the Stanford Parser

The Stanford parser includes a component for dependency parsing [5] , which consists of two phases: dependency extraction and dependency typing. After parsing the phrase structure tree of a sentence, semantic heads need to be identified, rather than syntactic ones, to be more useful for semantic dependency analysis (extractions), i.e. choose content words as heads (rather than auxiliaries, complementizers, etc.) and other words as dependents. Ambiguous structures or multi-headed structures (represented as flat structures in the Penn Treebank) also need to be resolved. For dependency labeling, one or more patterns are defined over the phrase structure tree using the `tregex` tree expression syntax [6]. For example, `"ADJP !< CC|CONJP < (JJ|NNP $ JJ|NNP=target)"` describes an ADJP not dominating a CC or CONJP, and dominating a JJ or NP with a sister JJ or NNP. This is one of the patterns which describe the UD relation `amod`. In theory, every node is matched against every pattern and from the matching patterns, the most specific relation decides the type of the dependency.

### 2.2 UD

The Universal Dependencies (UD) project[1] [1] is a cross-linguistically consistent annotation system and treebanks for 60+ languages. It provides a universal inventory of categories and annotation guidelines while allowing language-specific extensions. UD has evolved from Stanford Dependencies [7] by merging it with Google universal tags [8], a revised subset of the Interset feature inventory [9], and a revised version of the CoNLL-X format [10]. The two groups of core dependencies are the clausal relations (which describe syntactic roles concerning the predicate), and the modifier relations (which categorize the ways words modify their heads). For the sake of a uniform analysis, nouns introduced by or having attached prepositions and other case markings are treated as the head of these relations. The formalism follows a lexicalist approach to enable computational use: the syntactic structure consists of lexical elements linked by binary asymmetrical, one-to-one relations as opposed to constituency, which is a one-to-one-or-more correspondence. UD also allows the cross-linguistic evaluation of dependency parsers: more than 30 teams participated in the 2017 CoNLL shared task on multilingual dependency parsing [11].

### 2.3 4lang

4lang [3] is a formalism which builds directed graphs for semantic representation while abstracting away from syntax: in such graphs, nodes stand for language-independent concepts, which do not have any grammatical attributes, and contain shared knowledge of competent speakers. For example, *freeze(N)*, *freeze(V)*, *freezing*, or *frozen* are not differentiated in 4lang representations, resulting in a many-to-one relation between 4lang concepts and between words of a given language.

Nodes are connected via three types of edges: 0-edge represents attribution (*flower* $\xrightarrow{0}$ *beautiful* ), the IS_A relation (*flower* $\xrightarrow{0}$ *plant*) and unary predication (*flower* $\xrightarrow{0}$ *bloom*). 1 and 2-edges connect binary predicates to their arguments (*James* $\xleftarrow{1}$ *like* $\xrightarrow{2}$ *dog*). Binary (transitive) elements, that do not correspond to any word in a given phrase or sentence, are marked by UPPERCASE printnames.

There is another type of edge configuration, $w_1 \underset{0}{\overset{1}{\rightleftharpoons}} w_2$, that may appear in 4lang graphs. This is necessary to consistently represent the relation between the subject and the predicate: considering the example sentences *I'm writing* (*i* $\xrightarrow{0}$ *write*) and *I'm writing a paper* (*i* $\xleftarrow{1}$ *write* $\xrightarrow{2}$ *paper*), these representations would suggest that the relation between *I* and *write* is dependent on whether the object is specified or not. The example graph of Figure 1 represents the meaning of the sentence *John loves Mary's cat.*

The 4lang library contains tools for building directed graphs from raw text (text_to_4lang) and dictionary definitions (dict_to_4lang). The core module of the 4lang library, dep_to_4lang obtains dependency relations from text by

[1] `http://universaldependencies.org/`

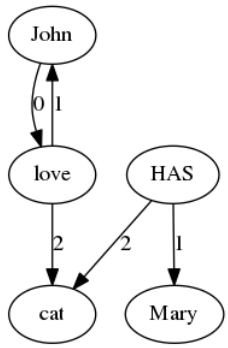



Fig. 1: 4lang graph of the sentence *'John loves Mary's cat.*

processing the output of the Stanford parser [5] and applies a simple mapping from Stanford dependencies to 4lang subgraphs.

4lang is also the name of a manually created concept dictionary [12] which contains more than 2000 definitions of language-independent concepts in four languages (Hungarian, English, Latin and Polish), hence its name.

### 2.4 IRTGs and s-graphs

**IRTG** Interpreted regular tree grammars [4] are context-free grammars in which each rule is mapped over an arbitrary number of algebras. Thus, when one rewrite rule gets applied on one of the algebras, the corresponding operations are executed on objects in each algebra. This means that an IRTG parser can accept inputs in any of the defined interpretations and can convert the input data into any of the other interpretations as output. In an example case of Figure 2, which was implemented using Alto (an open-source parser, will be introduced later in this section) [13], the two interpretations are the `string` and the `tag tree` algebras, so it can either convert a string to a phrase structure tree and vice versa. The `string algebra` has only one binary operation symbol *, which evaluates to `string concatenation`. Operations of the `tag tree algebra` will be discussed later in this section.

The rules are processed as follows: first a derivation tree is built using regular tree grammars, then a function called tree homomorphism maps the derivation tree over a term ($f(t_1, ..., t_n)$ stands for the tree with the root label $f$ and subtrees $t_1, ..., t_n$), then the tree is evaluated over the algebra. For a formal description the reader is referred to [14].

**tag tree algebra** The `tag tree algebra` [15] is a simple tree manipulation language. Trees consist of single edges and nodes. Nodes marked with * are called `holes`. Subtrees can be combined with the elementary tree using `substitution` (leaving a variable in the appropriate place) and `adjunction` (using the @ op-

```
interpretation string: de.up.ling.irtg.algebra.StringAlgebra
interpretation tree: de.up.ling.irtg.algebra.TagTreeAlgebra

S! -> s(NP)
[string] ?1
[tree] ?1

NP -> np(DT, N_BAR)
[string] *(?1,?2)
[tree] @(?2,?1)

N_BAR -> n_bar(JJ, NN)
[string] *(?1,?2)
[tree] NP(*,?1,?2)

// terminals

DT -> a
[string] a
[tree] DT(a)

JJ -> large
[string] large
[tree] JJ(large)

NN -> dog
[string] dog
[tree] NN(dog)
```

Fig. 2: An example IRTG.

erator), as shown in Figure 3. T1 and T3 stand for elementary trees, in which T2 (the subtree, which is allowed to contain additional holes) will be inserted.

```
T1= S(*, VP(V(loves), NP(NN(Mary))))
T2= NP(NN(John))
T3= S(*, VP(V(loves), *))
@(T1,T2)= S(NP(NN(John)), VP(V(loves), NP(NN(Mary))))
@(T3, T2)= S(NP(NN(John)), VP(V(loves), NP(NN(John))))
```

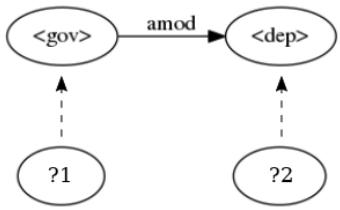
Fig. 3: An example illustrating the operations of the tag tree algebra.

Figure 3 also shows that when there are multiple holes in the tree, the binary operator @ inserts the same subtree to all of them.

**s-graph** The `s-graph algebra` [16] , which has been used for semantic parsing previously [17], is designed for building larger graphs from smaller ones. An s-graph's nodes are marked with `source labels` from a fixed finite set. Source labels identify which nodes should be merged when executing operations. Sources are the nodes which carry source names. An s-graph can consist of a single `root` node. The operations of the s-graph algebra are `merge` (which returns a graph that contains all the nodes and edges of its operands), `rename` (which returns a graph like the original except given source names have been changed) and `forget` (which returns a graph like the original except a given source name is removed from all nodes with that name). When used for semantic parsing [4], source names correspond to the semantic argument positions of the given grammar. The example in Figure 4 demonstrates the usage of these operators by connecting two subgraphs. ?1 and ?2 represent the subgraphs to be merged to the initial graph, which is between the quotation marks. First, the `root` label of ?2 is renamed to `dep`. Then it can be merged with the initial graph's `dep` node. After the merge the `dep` label is removed, because this node will not be used in subsequent rules. In the final step the `root` of ?1 is merged with the `root` of the initial graph.

**Alto** The Algebraic Language Toolkit, or `Alto`[2] [13] is an open-source parser which implements a variety of algebras for use with IRTGs, including the s-graph and the tag tree algebras. It has been used previously for graph transformations and semantic parsing as well [17], [4].

[2] `https://bitbucket.org/tclup/alto`




```
merge(f_dep(merge("(r<root> :amod (d<dep>))", r_dep(?2))),?1)
```

Fig. 4: An example illustrating the operations of the s-graph algebra.

## 3 Parsing NPs with Interpreted Regular Tree Grammars

### 3.1 Rule generation

In this section we present the first steps of creating a framework which encodes transformations between strings, phrase structure trees and UD and 4lang graphs. Our system supports the UD v2.1 format and is capable of automatic rule generation from Penn Treebank lines and those parsed by the Stanford parser. The code can be found at `github.com/evelinacs/semantic\_parsing\_with\_IRTGs/tree/master/code/generate\_grammar/template\_based\_grammar\_generator`.

To generate the IRTG rules, the program compares the phrase structure tree and the dependency graph of each noun phrase. The input for the phrase structure tree data is in Penn Treebank format and the dependency graph data is extracted from the output of the Stanford parser (which is generated by pattern matching). During rule generation, the program compares the relations between two words in the phrase structure tree and in the dependency graph. Given the rigidity and ordered nature of the tag tree algebra, the rule generation is based on the phrase structure of a subtree. This means that the RTG line (its left-hand side, and the arguments on the right-hand side) is derived from the phrase structure tree, and the [tree] interpretation either simply reflects the node type of the head and the number of its children or a merge operation is performed between two subtrees. To generate the [ud] and [fourlang] interpretations, the order of the nodes in the phrase structure tree must be considered, as the direction of some edges in these graphs are reversed with regards to the order of nodes in the phrase structure tree, and this must be reflected in the generated rules. The edge types in the [fourlang] interpretation are derived from the UD edge types using a predefined mapping implemented before Figure 1. This mapping also contains information regarding technical nodes in the 4lang formalism, which must be introduced for some relations in the [fourlang] interpretation to produce the correct 4lang representation of the given structure.

In our experiment we have limited our grammar to trees with an NP as a root node and any node within a tree must have at most three children. This subset makes 84,8% of all NPs of the Penn Treebank.

| **Dependency Edge** | |
|---|---|
| advcl | $w_1 \overset{0}{\rightarrow} w_2$ |
| advmod | |
| amod | |
| nmod | |
| nummod | |
| appos | $w_1 \underset{0}{\overset{0}{\rightleftharpoons}} w_2$ |
| dislocated | |
| csubj | $w_1 \underset{0}{\overset{1}{\rightleftharpoons}} w_2$ |
| nsubj | |
| ccomp | $w_1 \overset{2}{\rightarrow} w_2$ |
| obj | |
| xcomp | |
| **The treatment of `case`** | |
| case + nmod | $w_1 \overset{1}{\leftarrow} w_3 \overset{2}{\rightarrow} w_2$ |
| case + nsubj | |
| case + obl | |
| **English subtypes** | |
| obl:npmod | $w_1 \overset{0}{\rightarrow} w_2$ |
| nmod:tmod | $w_1 \overset{1}{\leftarrow} \texttt{AT} \overset{2}{\rightarrow} w_2$ |
| obl:tmod | |
| nmod:poss | $w_2 \overset{1}{\leftarrow} \texttt{HAS} \overset{2}{\rightarrow} w_1$ |

**Table 1.** UD-conform version of the `dep_to_4lang` mapping.

Figure 5 presents the structure of the phrase *a long way* regarding the tree and graph interpretations.

As the structures of the phrase structure tree and the UD graph are fundamentally different, their respective rules for each interpretation must be implemented to accommodate that. Appendix A presents the rules which are responsible for parsing the aforementioned sentence.

When processing this IRTG, first a derivation tree (Figure 6) is built by parsing the input according to the specified interpretation and the corresponding RTG rules. If a viable derivation tree is found, corresponding outputs are created based on all the other interpretations. For example, if the input is an UD graph, then a phrase structure tree, a string and a 4lang graph can be retrieved as outputs. The outputs are built from the leaf nodes of the derivation tree according to the rules of the interpretations' respective algebra. In the terminal rules, a word, a subtree or a subgraph is produced for the string, the phrase

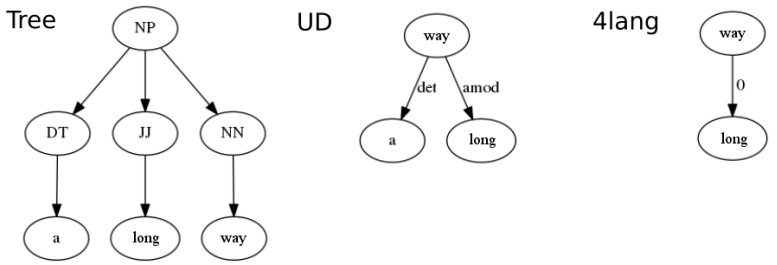



Fig. 5: Configurations of the phrase structure tree, the UD graph and the 4lang graph for the phrase *a long way*.

structure tree, and the UD and 4lang graph interpretations. Then the `N_BAR -> amod_JJ_NN_NBAR(JJ,NN)` rule gets applied. In the string interpretation, a concatenation of the arguments is executed. In the tree interpretation the subtrees are attached to an NP labeled head along with a placeholder node, marked by *. The UD and 4lang interpretations are almost the same in this step: first the root label in the subgraph provided as the second argument is renamed to dep, then merged with the node having the same label in the graph between the quotation marks. Next the dep label is removed, making the node with this label internal (not used in later operations). Then the subgraph provided as the first argument is merged with the root of the graph from the result of the previous operation. In the rule `NP -> det_DT_NBAR_NP(DT, N_BAR)` the same operations are applied in the string and UD interpretations. In the tree interpretation, the subtree provided as the first argument replaces the placeholder node from the previous rule. In the 4lang interpretation only the second argument is passed along, as the determiner doesn't contain much semantic information. Finally the `S! -> s(NP)` rule is simply an identity operation that is, it returns its single argument in all four interpretations.

Given the previous process, transforming the string *a long way* to a UD graph happens as the following. First, each word is mapped to the corresponding terminal rules which in turn create three subgraphs representing each word as a root labeled node. Then following the derivation tree the rule responsible for concatenating the words *long* and *way* is matched (`N_BAR -> amod_JJ_NN_NBAR(JJ,NN)`) which creates an `amod` relation between the nodes representing the words by merging them with the graph provided within the [ud] interpretation. The node label is removed from the node representing the word *long*, and the other node (representing *way*) keeps its root label so it can be used in the following rules. Next, similar to the previous step, the rule concatenating the two substrings *a* and *long way* is matched (`NP -> det_DT_NBAR_NP(DT, N_BAR)`) which creates the `det` relation between the words *a* and *way* by merging the subgraph resulting from the previous step and the node representing the word *a*. Finally the start rule of the derivation tree creation is applied which contains no transformation, so the result is the graph obtained from the previous step.

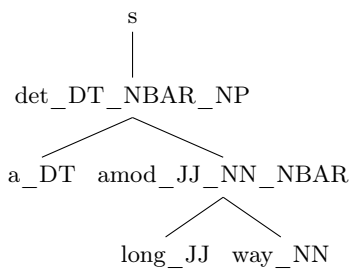



Fig. 6: The derivation tree of the phrase *a long way*.

### 3.2 Evaluation

**Parsing the Penn Treebank** To create a fully functioning IRTG, terminal nodes (e.g. the words) had to be appended to the rule file. We also had to convert from the Penn Treebank format to one that is understood by Alto. These tasks were implemented in Python and the scripts can be found in our repository.

Our grammar have been evaluated on trees with an NP as a root node and any node within a tree must have at most three children. This subset makes 84,8% of all NPs of the Wall Street Journal section of the Penn Treebank. The WSJ section contains 243 914 NPs of which 206 841 meet the aforementioned restrictions. Parsing this subset using the generated grammar resulted in 202 549 successful parses, which makes 97,9% of the test data and 83% of all NPs. We can transform structures in this subset from any interpretation to any other interpretation, including converting UD graphs to strings.

**Generating text from UD graphs** To test our grammar on an independent dataset (i.e. different from the one that was used for generating it), we have used a subset of the test data of the Surface Realization Shared Task [18] as well, which contains UD graphs. We extracted subgraphs corresponding to all NPs of the first 20 sentences, resulting in a small dataset of 38 noun phrases, then manually reviewed the output of parsing this dataset using the previously generated grammar. Our goal was to derive surface realizations from UD graphs. Our grammar successfully parsed 18 NPs, which makes 47% of the test data. For the remaining 20 graphs no output was generated. These either contain dependencies which are not presented in the original input (e.g. flat), or due to some bugs in the grammar (which need to be investigated), these structures cannot be recognized by the parser. Some of the successfully parsed graphs had incorrect word order. As our grammar is non-deterministic, such results are expected due to the possibility of building multiple structures for the same input data. Creating a new grammar using maximum likelihood training might resolve the problem, but the causes of these errors need further investigation.

## 4 Conclusion and ongoing work

In this paper we presented an IRTG which implements a mapping between four formalisms, i.e. strings, the output of the Stanford parser, UD v2.1 and 4lang. Our system covers 83% of the NPs of the Wall Street Journal section of the Penn Treebank and was capable to retrieve surface realizations from a small subset of the test data of the Surface Realization Shared Task with limited success.

The grammar allows converting from any of the above algebraic structures to any or all of the others. A probabilistic version of this grammar (as Alto supports probabilistic IRTGs) can be trained using any parallel data. If a single probabilistic IRTG were to implement the parallel parsing of strings, syntactic constituency structures, dependency graphs and semantic graphs, it could be trained simultaneously on each of these types of gold-standard data, resulting in a single end-to-end system for semantic parsing. This might serve as a basis for semantic generation, paraphrasing or machine translation in the future.

## References

1. De Marneffe, M.C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal Stanford dependencies: A cross-linguistic typology. In: LREC. Volume 14. (2014) 4585–92
2. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Sofia, Bulgaria, Association for Computational Linguistics (2013) 178–186
3. Kornai, A., Ács, J., Makrai, M., Nemeskey, D.M., Pajkossy, K., Recski, G.: Competence in lexical semantics. In: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015), Denver, Colorado, Association for Computational Linguistics (2015) 165–175
4. Koller, A.: Semantic construction with graph grammars. In: Proceedings of the 14th International Conference on Computational Semantics (IWCS), London (2015)
5. DeMarneffe, M.C., MacCartney, W., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC). Volume 6., Genoa, Italy (2006) 449–454
6. Levy, R., Andrew, G.: Tregex and tsurgeon: tools for querying and manipulating tree data structures. In: Proceedings of the fifth international conference on Language Resources and Evaluation, Citeseer (2006) 2231–2234
7. De Marneffe, M.C., Manning, C.D.: Stanford typed dependencies manual. Technical report, Technical report, Stanford University (2008)
8. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. arXiv preprint arXiv:1104.2086 (2011)
9. Zeman, D.: Reusable tagset conversion using tagset drivers. In: LREC. Volume 2008. (2008) 28–30
10. Buchholz, S., Marsi, E.: CoNLL-X Shared Task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, Association for Computational Linguistics (2006) 149–164

11. Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinkova, S., Hajic jr., J., Hlavacova, J., Kettnerová, V., Uresova, 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., dePaiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, c., 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., Mendonca, 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, Vancouver, Canada, Association for Computational Linguistics (2017) 1–19
12. Kornai, A., Makrai, M.: A 4lang fogalmi szótár. In Tanács, A., Vincze, V., eds.: IX. Magyar Számítógépes Nyelvészeti Konferencia. (2013) 62–70
13. Gontrum, J., Groschwitz, J., Koller, A., Teichmann, C.: Alto: Rapid prototyping for parsing and translation. In: Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics. (2017) 29–32
14. Koller, A., Kuhlmann, M.: A generalized view on parsing and translation. In: Proceedings of the 12th International Conference on Parsing Technologies (IWPT), Dublin (2011)
15. Koller, A., Kuhlmann, M.: Decomposing tag algorithms using simple algebraizations. In: Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 11). (2012) 135–143
16. Courcelle, B.: Graph grammars, monadic second-order logic and the theory of graph minors. Contemporary Mathematics **147** (1993) 565–565
17. Groschwitz, J., Koller, A., Teichmann, C.: Graph parsing with s-graph grammars. In: Proceedings of the 53rd ACL and 7th IJCNLP, Beijing (2015)
18. Mille, S., Belz, A., Bohnet, B., Graham, Y., Pitler, E., Wanner, L.: The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In: Proceedings of the First Workshop on Multilingual Surface Realisation. (2018) 1–12

## A An example IRTG

The following is an example IRTG for the phrase *a long way*.

```
interpretation string: de.up.ling.irtg.algebra.StringAlgebra
interpretation tree: de.up.ling.irtg.algebra.TagTreeAlgebra
interpretation ud: de.up.ling.irtg.algebra.graph.GraphAlgebra
interpretation fourlang: de.up.ling.irtg.algebra.graph.GraphAlgebra

// IRTG for the phrase 'a long way'

S! -> s(NP)
[string] ?1
[tree] ?1
[ud] ?1
[fourlang] ?1

NP -> det_DT_NBAR_NP(DT, N_BAR)
[string] *(?1,?2)
[tree] @(?2,?1)
[ud] merge(f_dep(merge("(r<root> :det (d<dep>))", r_dep(?1))),?2)
[fourlang] ?2

N_BAR -> amod_JJ_NN_NBAR(JJ,NN)
[string] *(?1,?2)
[tree] NP(*,?1,?2)
[ud] merge(f_dep(merge("(r<root> :amod (d<dep>))", r_dep(?1))),?2)
[fourlang] merge(f_dep(merge("(r<root> :0 (d<dep>))", r_dep(?1))),?2)

// terminals:

DT -> a_DT
[string] a
[tree] DT(a)
[ud] "(a<root> / a)"
[fourlang] "(a<root> / a)"

JJ -> long_JJ
[string] long
[tree] JJ(long)
[ud] "(long<root> / long)"
[fourlang] "(long<root> / long)"

NN -> way_NN
[string] way
[tree] NN(way)
[ud] "(way<root> / way)"
[fourlang] "(way<root> / way)"
```

# Argumentumszerkezet-variánsok korpusz alapú meghatározása

Szécsényi Tibor

Szegedi Tudományegyetem, Általános Nyelvészeti Tanszék,
6722, Szeged, Egyetem u. 2.
szecsenyi@hung.u-szeged.hu

**Kivonat:** A tanulmány a lexikai egységek, tipikusan igék argumentumszerkezetének a leírására javasol egy új reprezentációs formát, ami nem a klasszikus kötelező vonzat – szabad bővítmény bináris oppozíciós lehetőségeket ragadja meg. Ehelyett az egyes bővítménytípusoknak a korpuszban való megjelenési gyakoriságai alapján a típusokhoz egy-egy valószínűségi értéket rendel, így az argumentumszerkezeti variánsok egy argumentumszerkezeti valószínűségi vektorral jellemezhetőek. A javasolt reprezentáció kizárólag a korpuszbeli adatok morfológiai és szintaktikai tulajdonságaira támaszkodik. Az argumentumszerkezeti variánsok argumentumszerkezeti vektorként való értelmezése új elméleti modellként a grammatikaelméletekben hozhat új eredményeket, másrészt a természetesnyelv-feldolgozásban is használható.

## 1 Bevezetés, célok

A teljes szintaktikai elemzés elengedhetetlen feltétele a szövegben található igék és más régensek argumentumszerkezetének valamilyen szintű ismerete, ez teszi lehetővé, hogy a mondatban a régens mellett kötelezően megjelenítendő kifejezések számát és azok tulajdonságait leírhassuk. Az argumentumszerkezet az igék/régensek egyedi, lexikai tulajdonsága, ami tulajdonságokat a nyelv nyelvelméleti igényű grammatikái explicit módon használnak fel a mondatszerkezet kialakítása során: a transzformációs nyelvtanokban például a projekciós elv [1], a HPSG-ben az alkategorizációs elv [2, 3] biztosítja, hogy az egyes igék/régensek mellett csak a megfelelő összetevők jelenhessenek meg vonzatpozícióban. Az ezen elméleteket alkalmazó számítógépes szintaktikai elemzők is hatékonyan használják a lexikai elemek argumentumszerkezeti információit, például a [4] tanulmányban bemutatott szabály alapú elemző az igék argumentumszerkezetének lexikai leírására támaszkodva csupán négy újraíró szabállyal képes elemezni a magyar mondatokat.

A leíró nyelvészeti munkák az argumentumszerkezetet az argumentumok, argumentumtípusok felsorolásával adja meg. Jelen dolgozatban egy olyan argumentumszerkezet-reprezentációra teszek javaslatot, amely nem ilyen bináris, argumentum – nem argumentum oppozícióként kezeli az argumentumszerkezetet, hanem a régensek lehetséges bővítményeihez egy-egy [0–1] intervallumon található értéket rendel, jelezve ezzel, hogy az adott bővítmény mekkora valószínűséggel jelenik meg a régens mellett egy mondatban. Ekkor a régens argumentumszerkezete *n* lehetséges bővítménytípus esetén

egy *n*-dimenziós egységkocka belsejébe mutató vektorral jellemezhető. Ez az argumentumszerkezeti vektor korpusz alapján automatikusan is meghatározható, illetve a bővítmények megjelenését befolyásoló tényezők feltérképezése után azok figyelembevételével a klasszikus argumentumszerkezeti lista is visszanyerhető.

A javasolt reprezentáció nem a szóbeágyazási modellek [5] egy változata, hanem a többdimenziós értelmezés miatt inkább Sass Bálint duplakocka-modelljével [6] rokonítható. Korábban Kálmán László is javasolta a vonzatság bináris felfogásának az elvetését [7, 8], de nála ez egyrészt vagy csak az ige és bővítmény közötti többféle lehetséges viszonyt jelentette, vagy ha a kapcsolatuk erősségének a gradualitását is megemlítette, ennek a gradualitásnak az értékét nem a bővítmények megjelenési valószínűségéhez kötötte. További különbség, hogy Kálmán az egyes ige-bővítmény kapcsolatokat egyenként elemezte, nem az ige teljes argumentumszerkezetét próbálta meg így leírni.

A dolgozat fő újdonságaként az első részben először bevezetem az argumentumszerkezeti vektorok fogalmát (2. szakasz), majd bemutatom, hogyan lehet egy ige argumentumszerkezeti variánsait korpusz alapján meghatározni (3. szakasz). Ezután az argumentumszerkezeti vektorokat befolyásoló néhány tényezőt mutatok be (4. szakasz). A dolgozat végén néhány lehetséges felhasználási területet is ismertetek.

## 2 Az argumentumszerkezetek megfigyelése természetes környezetükben

Az igék és más régensek argumentumszerkezetét hagyományosan egy táblázatban adhatjuk meg, ahol minden egyes ige minden vonzatszerkezeti variánsához egy sor tartozik, ahol jelöljük, hogy milyen tulajdonságú vonzatokkal kell együtt szerepelnie egy teljes mondatban. A Szeged Korpuszban [9, 10] a *bíz/bízik* ige 6 variánsával található meg, ezek az 1. táblázatban láthatóak (a főnévi vonzatokat az esetükkel jellemzem). A hat argumentumszerkezeti variánsra példák: *Péter bízik*$_1$ *Mariban*; *Péter megbízik*$_2$ *Mariban*; *Péter Marira bízta*$_3$ *a könyvet*; *Péter rábízta*$_4$ *a könyvet Marira*; *Péter megbízta*$_5$ *Marit a feladattal*; *Péter elbízta*$_6$ *magát*, bár itt nagyon speciális tárgyról beszélünk, csakis visszaható névmási tárgy lehet.

| | igekötő | NOM | ACC | BAN | RA | VAL |
|---|---|---|---|---|---|---|
| bíz/bízik$_1$ | - | + | - | + | - | - |
| bíz/bízik$_2$ | meg | + | - | + | - | - |
| bíz/bízik$_3$ | - | + | + | - | + | - |
| bíz/bízik$_4$ | rá | + | + | - | + | - |
| bíz/bízik$_5$ | meg | + | + | - | - | + |
| bíz/bízik$_6$ | el | + | + | - | - | - |

1. táblázat. a *bízik* ige argumentumszerkezeti variánsai

A Szeged Korpuszban ez a hat argumentumszerkezeti variáns nem mindig a vonzataival együtt jelenik meg, továbbá nem csak a vonzatai találhatóak mellette, hanem más szabad bővítmények is. A dependenciakorpuszból [10] saját korpusztranszformációval és kézi annotálással (MMAX2 [11]) a 2. táblázatban látható vonzat-előfordulási adatokat kapjuk.

| | | | Argumentumtípus (X) | | | | |
|---|---|---|---|---|---|---|---|
| | | | NOM | ACC | BAN | RA | VAL |
| bíz/bízik | $n_V$=157 | $n_X$ | 64 | 66 | 91 | 44 | 23 |
| | | $f_X$ | 0,41 | 0,42 | 0,58 | 0,28 | 0,15 |
| bíz/bízik$_1$ | $n_{V1}$=66 | $n_{1X}$ | 29 | 0 | 65 | 2 | 0 |
| | $f_{V1}$=0,42 | $f_{1X}$ | 0,44 | 0 | 0,98 | 0,03 | 0 |
| bíz/bízik$_2$ | $n_{V2}$=17 | $n_{2X}$ | 9 | 0 | 17 | 0 | 0 |
| | $f_{V2}$=0,11 | $f_{2X}$ | 0,53 | 0 | 1,00 | 0 | 0 |
| bíz/bízik$_3$ | $n_{V3}$=37 | $n_{3X}$ | 9 | 33 | 4 | 37 | 0 |
| | $f_{V3}$=0,24 | $f_{3X}$ | 0,24 | 0,89 | 0,11 | 1,00 | 0 |
| bíz/bízik$_4$ | $n_{V4}$=6 | $n_{4X}$ | 2 | 4 | 1 | 3 | 0 |
| | $f_{V4}$=0,04 | $f_{4X}$ | 0,33 | 0,67 | 0,17 | 0,5 | 0 |
| bíz/bízik$_5$ | $n_{V5}$=28 | $n_{5X}$ | 14 | 26 | 4 | 2 | 23 |
| | $f_{V5}$=0,18 | $f_{5X}$ | 0,5 | 0,93 | 0,14 | 0,07 | 0,82 |
| bíz/bízik$_6$ | $n_{V6}$=3 | $n_{6X}$ | 1 | 3 | 0 | 0 | 0 |
| | $f_{V6}$=0,02 | $f_{6X}$ | 0,33 | 1,00 | 0 | 0 | 0 |

2. táblázat. a *bízik* ige bővítményeinek előfordulási száma és megjelenési gyakorisága a Szeged Korpuszban

A korpuszban összesen 157-szer szerepel az ige ($n_V$), ebből 64-szer szerepel vele egy tagmondatban alanyesetű maximális főnévi csoport ($n_{NOM}$), ami 0,41 relatív gyakoriságot jelent ($f_{NOM} = \frac{n_{NOM}}{n_V}$) stb. Ezek a korpuszból automatikusan, kézi annotálás nélkül kigyűjthető adatok.

Kézi annotálással meghatározható, hogy a hat argumentumszerkezet-variáns egyenként 66-szor, 17-szer stb. fordul elő ($n_{Vi}$), ami 0,42, 0,11 stb. relatív gyakoriságot jelent ($f_{Vi} = \frac{n_{Vi}}{n_V}$). A táblázat többi részében az egyes argumentumszerkezet-variánsok mellett megjelenő egyes bővítmények megjelenési száma ($n_{iX}$) és megjelenési gyakorisága ($f_{iX} = \frac{n_{iX}}{n_{Vi}}$) található. Láthatjuk, hogy a kötelező vonzatok nem jelennek meg minden esetben az ige mellett, az alany például, amely mindegyik variánsnak vonzata, csak 0,33–0,53 gyakorisággal. Ennek egyrészt az az oka, az alanyi és tárgyi vonzat sokszor elhagyható (pro-drop), máskor egyenesen tilos kitenni (főnévi igenév alanya), az ellipszis (pl. *Péter keringőzött ~~Marival~~, Lajos pedig foxtrottozott Marival*) is látszólagos vonzathiányt okoz, illetve vannak egyszerűen hiányos mondatok (rövid válasz, pl. – *Találkoztál Marival? – Találkoztam ~~Marival~~*). Az alanyon kívüli vonzatok azonban igen nagy gyakorisággal megjelennek (>0,6). Azonban az is megfigyelhető, hogy a vizsgált bővítmények akkor is megjelenhetnek az igék mellett, ha annak nem vonzatai. A BAN esetű bővítmény például helyhatározóként a 3., 4. és 5. variánst is módosíthatják, ezekben az esetekben azonban aránylag kicsi a megjelenési gyakoriságuk (<0,2).

Hogy az argumentumszerkezetet közvetlenül a korpuszban ténylegesen megfigyelhető adatok alapján értelmezhessük, ezáltal számot tudjunk adni az esetleges vonzatelmaradásokról is, továbbá hogy egységes keretben tudjuk kezelni a kötelező vonzatokat és a szabad bővítményeket úgy, hogy közben a két csoport tagjainak a megkülönböztethetősége megmaradjon, a továbbiakban

- az igék vonzatszerkezetét illetve a vonzatszerkezeti variánsait nem a vonzatok felsorolásával, bináris listaként jellemezzük (1. táblázat), hanem a skaláris argumentumgyakorisági értékek listájával (2. táblázat), vagyis egy-egy argumentumgyakorisági vektorral.

Tegyük fel, hogy a magyar nyelvben a lehetséges bővítménytípusok a *bíz/bízik* ige kapcsán tárgyalt [NOM; ACC; BAN; RA; VAL] listával adhatók meg. Ekkor az ige első argumentumszerkezeti variánsát a [0,44; 0; 0,98; 0,03; 0] ötdimenziós vektorral jellemezzük, a második variánsát a [0,53; 0; 1; 0; 0] vektorral stb. Jelöljük ezeket a vektorokat $\bar{v}_1$-gyel, $\bar{v}_2$-vel, … $\bar{v}_6$-tal, a 2. táblázatban látható összesített [0,41; 0,42; 0,58; 0,28; 0,15] előfordulási gyakorisági vektort pedig $\bar{v}$-vel. $\bar{v}$ a *bíz/bízik* ige Szeged Korpuszban való előfordulásaiból közvetlenül meghatározható, $\bar{v}_i$ pedig kézi annotáció utáni számlálással. Ekkor a következő összefüggés áll fenn:

$$\bar{v} = \sum_{i=1}^{6} f_{Vi} \cdot \bar{v}_i \qquad (1)$$

vagyis az ige korpuszban megfigyelhető bővítménygyakorisági vektora az ige argumentumszerkezet-variánsainak a variáns előfordulási gyakoriságának a súlyozásával vett vektori összegével egyenlő.

Az igei argumentumszerkezetek-variánsok vektorainak birtokában és a variánsok korpuszbeli előfordulási gyakoriságának ismeretében tehát megkaphatjuk az ige bővítményeinek a korpuszbeli előfordulási gyakoriságvektorát.

## 3 Argumentumszerkezeti vektor meghatározása korpuszból

Egy V ige bővítményeinek egy adott korpuszban megfigyelhető előfordulási gyakoriságát tehát a $\bar{v}$ vektorral jellemeztük. Ez a vektor a korpuszból közvetlenül kinyerhető, amennyiben a korpusz szavai megfelelő morfoszintaktikai annotálással vannak ellátva, illetve automatikusan meghatározhatók a korpuszban a mondat és tagmondathatárok és a maximális főnévi kifejezések. Ha az ige $V_i$ argumentumszerkezeti variánsokkal rendelkezik, ezeket az alternánsokat a $\bar{v}_i$ vektorokkal kívánjuk jellemezni, illetve az egyes variánsok korpuszbeli előfordulási gyakoriságát $f_{Vi}$-vel. Ezek, vagyis a variánsok argumentumszerkezet-vektorai és a variánsok gyakorisági együtthatói a korpuszból közvetlenül nem meghatározhatóak, viszont tudjuk, hogy teljesül rájuk az (1) egyenlőség.

Automatikusan meghatározható viszont az az információ, hogy az ige a korpuszban ugyanabban a tagmondatban ténylegesen milyen bővítményekkel fordul elő. Ezeket a megfigyelhető ige–bővítmény előfordulásokat kombinációstípusonként összegezhetjük is, illetve az ige összes előfordulásához viszonyítva a gyakoriságukat is megadhatjuk. Jelöljük a megkülönböztetett bővítménytípusok halmázát *ArgType*-pal (vagy AT-val). (Az előbbi *bíz/bízik* példában *ArgType* = {NOM, ACC, BAN, RA, VAL} volt.)

Az *ArgType* = {A, B, C stb.} *k* elemű bővítménytípus-halmaz esetén azoknak a mondatoknak a számát, ahol a *V* ige bővítmény nélkül jelenik meg, jelöljük $n_{V+0}$-val, relatív gyakoriságát $f_{V+0}$-val. Annak a számát, amikor csak A bővítménnyel fordul elő, jelöljük

$n_{V+A}$-val, relatív gyakoriságát $f_{V+A}$-val, amikor A-val és B-vel fordul elő, $n_{V+A+B}$-vel és $f_{V+A+B}$-vel és így tovább: $n_{V+B}$ és $f_{V+B}$, $n_{V+A+C}$ és $f_{V+A+C}$ stb. Ha *k* darab különböző bővítménytípus veszünk figyelembe, akkor $2^k$ különböző kombinációban jelenhetnek meg ezek a bővítmények az ige mellett, vagyis ennyi előfordulási adatot és gyakorisági adatot kaphatunk a korpuszból, bár ezek nagy része valószínűleg egyszer sem fordul elő: például szinte nulla a valószínűsége annak, hogy egy ige az összes lehetséges bővítménnyel együtt jelenjen meg egy mondatban.

Azon mondatok számát, ahol az ige az A bővítménnyel jelenik meg, függetlenül más bővítménytípusok jelenlététől, jelöljük $n_{V+A+*}$-gal, relatív gyakoriságát $f_{V+A+*}$-gal, azon mondatok számát, ahol az ige A-val és B-vel jelenik meg, függetlenül más bővítménytípusok megjelenésétől, jelöljük $n_{V+A+B+*}$-gal stb. Azt várnánk, hogy $n_{V+A+*} = n_A$, vagyis az igét és az A bővítményt egyaránt tartalmazó mondatok száma megegyezik az A bővítmények számával, de ez nem szükségszerűen igaz: vannak eseteket, amikor a kettő eltérhet, pl. *A múlt évben még bíztam Mariban* esetében az ige egyszer fordul elő BAN bővítmény mellett, de a BAN bővítmény kétszer fordul elő az ige környezetében.

A *bíz/bízik* ige esetében nem csak a korábban bemutatott 5 lehetséges vonzattípussal kel számolni, hanem több szabad bővítménnyel is. Ezeket a további bővítményeket főnévi kifejezés esetében szintén az esetükkel lehet jellemezni, más esetekben pedig a megjelenő névutóval, az igenévi típussal (pl. főnévi igenév – NI) vagy a mondattípussal (pl. HKM). A Szeged Korpuszban az ige az említett bővítményeken kívül szerepelt még *hogy* kötőszavas mellékmondattal (HKM), szuperesszívusz esetű bővítménnyel (ON), terminatívuszi bővítménnyel (IG), ablatívuszi bővítménnyel (TÓL), különböző határozószókkal (ADV) és néhány névutós kifejezéssel (PP). Ez utóbbi öt típus csak néhányszor fordult elő, ezért most a határozószókat, illetve a névutós kifejezéseket összevontan kezelem. Az így kapott 11 bővítménytípussal összesen $2^{11} = 2048$ különféle bővítménykombinációt lehetne létrehozni, de a korpuszban – már csak azért is, mert összesen csak 157-szer szerepel a kérdéses ige – nem található meg mindegyik, hanem csak 40. Ebből a 40-ből is csak 10 olyan van, ami kettőnél többször fordul elő, ezekben pedig a HKM-en kívül csak a korábban ismertetett 5 bővítménytípus van jelen, ezek lefedik a *bíz/bízik* ige előfordulásának több mint a kétharmadát, összesen 116-ot a 157-ből.

| Típus | előfordulási szám ($n_{V+X}$) | gyakorisági szám ($f_{V+X}$) |
|---|---|---|
| V+BAN | 26 | 0,17 |
| V+ACC+RA | 21 | 0,13 |
| V+NOM+BAN | 21 | 0,13 |
| V+BAN+HKM | 12 | 0,08 |
| V+NOM+BAN+HKM | 11 | 0,07 |
| V+NOM+ACC+VAL | 7 | 0,04 |
| V+ACC+VAL | 6 | 0,04 |
| V+NOM+ACC+RA | 5 | 0,03 |
| V+RA | 4 | 0,03 |
| V+ACC | 3 | 0,02 |

3. táblázat. a *bízik* ige 10 leggyakoribb megjelenő bővítménykombinációja

A táblázatban szereplő adatok esetében nem tettem különbséget az igekötős és az igekötő nélküli igék között, csak az igével előforduló bővítménykombináció alapján összegeztem az adatokat. Az igekötők szerepére a 4.2.7 szakaszban térek vissza.

A 2. táblázat adatai az új bővítménytípusokkal kiegészítve a következő (a táblázat alsó sora $\vec{v}$):

| | biz/bízik | NOM | ACC | BAN | RA | VAL | HKM | ON | IG | TÓL | ADV | PP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | 157 | 64 | 66 | 91 | 44 | 23 | 31 | 2 | 2 | 1 | 11 | 6 |
| f | 1,00 | 0,41 | 0,42 | 0,58 | 0,28 | 0,15 | 0,20 | 0,01 | 0,01 | 0,005 | 0,07 | 0,04 |

4. táblázat. a *bízik* ige összes bővítményének az előfordulási adatai

Korábban kézi annotációval, azaz a mondat értelmezésével és a mondatban szereplő bővítmények tulajdonságainak figyelembevételével határoztuk meg, hogy a korpusz egyes mondataiban melyik argumentumszerkezeti variáns található, ami alapján a 2. táblázatot összeállítottuk, vagyis az argumentumszerkezeti vektorokat és a variánsok előfordulási gyakoriságát meghatároztuk. A kérdés az, hogy meghatározhatjuk-e ezeket a vektorokat és gyakoriságokat automatikusan a korpuszból, kizárólag a hozzáférhető morfológiai és szintaktikai információkra hagyatkozva, a mondatok értelmezése nélkül, vagyis meghatározhatóak-e az 2. táblázatban látható adatok kizárólag a 3. és 4. táblázatban található információk alapján? Mivel itt már nem a megfigyelhető szerkezetek előfordulásait számoljuk, vagyis a relatív gyakoriságukat (f), hanem becsüljük azokat, ezért ezeket a meghatározandó értékeket előfordulási valószínűségnek (p) tekintjük.

### 3.1 Két triviális argumentumszerkezet-variáns

Az első probléma az argumentumszerkezeti vektorok automatikus meghatározásánál az, hogy nem tudjuk, hogy hány vektort keresünk, azaz hány variánsa van az igének. Erre a kérdésre két triviális válasz is lehetséges. A két triviális megoldás legtöbbször nem megfelelő leírása az adatoknak, de két fontos általánosítás megfogalmazására teremtenek lehetőséget.

#### 3.1.1 Maximális variánsszámú ige

Tekinthetjük a 3. táblázatban felsorolt és a felsorolásból kihagyott, összesen 40 megfigyelhető bővítménykombinációt mind különálló argumentumszerkezet-variánsnak, ahol a megadott (megjelent) bővítmények előfordulási valószínűsége mind 1,00, a meg nem jelent bővítményeké pedig egyre 0,00, a variánsok előfordulási gyakorisága pedig megegyezik a megfigyelhető kombinációk előfordulási gyakoriságával, vagyis minden megjelenő bővítmény kötelező vonzat is egyben. Az argumentumszerkezeti variánsoknak ez a triviális listája így megfelel a (1) azonosságnak is. Azonban ekkor nem tudunk számot adni arról a jelenségről, hogy a természetes nyelvekre úgy tekintünk, hogy azokban vonzatok sem jelennek meg mindig, bizonyos esetekben a vonzatot is elhagyhatjuk. Továbbá szeretnénk olyan nyelvi leírást adni, ami a lehető leggazdaságosabb reprezentációt igényli, azaz

- Az ige argumentumszerkezeti variánsainak a számának minimalizálására törekszünk.

### 3.1.2 Egyvariánsos ige

Feltételezhetjük, hogy az igének csak egyetlen variánsa van. Ekkor mondhatjuk azt, hogy az ige egyetlen argumentumszerkezet-variánsa a 4. táblázatban látható argumentumszerkezeti vektorral jellemezhető, és a variáns gyakorisági együtthatója 1,00.

Ebben az esetben nem tudjuk megmagyarázni azt a tényt, hogy bár a korpuszban a vizsgált ige környezetében a BAN bővítmény ($f_{BAN}$ = 0,58) és a tárgyi bővítmény ($f_{ACC}$ = 0,42) a két leggyakrabban előforduló, együtt mégis csak nyolc mondatban találjuk meg mindkettőt (kb. 5%). A kisebb előfordulási gyakoriságú RA bővítmény ($f_{RA}$ = 0,28) tárggyal együtt viszont sokkal többször, 37-szer szerepel (kb. 24%).

Feltételezzük ugyanis, hogy egy egyvariánsos ige különböző bővítményeinek a megjelenési valószínűsége független egymástól, az egyik megjelenése nem befolyásolja a másik megjelenési valószínűségét. Ez igaz a többvariánsos igék egyes variánsa esetében is:

- Egy ige egy argumentumszerkezeti variánsa esetében a variáns különböző bővítményeinek a megjelenési valószínűségei függetlenek egymástól.

Vegyük a V igének egy $V_i$ variánsát (vagy egy egyvariánsos igét), ami mellett az A, B, C és D bővítmények jelenhetnek meg. Annak a valószínűsége, hogy a variáns mellett megjelenik az A bővítmény, $p_{iA}$ (illetve $p_{iB}$, $p_{iC}$, $p_{iD}$), annak a valószínűsége pedig hogy az A bővítmény nem jelenik meg mellette, $1\text{-}p_{iA}$ (illetve $1\text{-}p_{iB}$, $1\text{-}p_{iC}$, $1\text{-}p_{iD}$). Ekkor a V+A+C bővítménykombináció előfordulási valószínűsége a $V_i$ variáns mellett $p_{iV+A+C} = p_{iA}\cdot(1\text{-}p_{iB})\cdot p_{iC}\cdot(1\text{-}p_{iD})$, az A és a C bővítmény együttes előfordulásának a valószínűsége (függetlenül attól, hogy a B és a D megjelenik-e) $p_{iV+A+C+*} = p_{iA}\cdot p_{iC}$.

A *bíz/bízik* ige mellett a tárgy és a BAN bővítmény együttes előfordulásának a valószínűsége egyvariánsos igének feltételezve így $p_{V+ACC+BAN+*} = p_{ACC}\cdot p_{BAN} = 0{,}42\cdot 0{,}58 = 0{,}24$, a tárgy és a RA bővítményé pedig $p_{V+ACC+RA+*} = p_{ACC}\cdot p_{RA} = 0{,}42\cdot 0{,}28 = 0{,}12$ kellene hogy legyen, a megfigyelt 0,05 és 0,28 helyett.

### 3.2 Az argumentumszerkezeti vektor és a korpuszban megfigyelhető gyakoriságok közötti összefüggések

Az előző részben használt számolás mögötti összefüggések általánosítva a következők:

Legyen *ArgType* (vagy AT) a lehetséges bővítménytípusok halmaza, C pedig ennek egy részhalmaza. Jelöljük V+C-vel a azokat a bővítménykombinációkat, amikor az ige a C-ben levő bővítményekkel együtt jelenik meg (pl. ha $C = \{c_1, c_2, c_3\}$, akkor $V+C = V+c_1+c_2+c_3$). Ekkor

- a V ige $V_i$ argumentumszerkezet-variánsa melletti V+C bővítménykombináció megjelenési valószínűsége

$$p_{iV+C} = \prod_{c \in C} p_{ic} \cdot \prod_{c \in AT \setminus C} (1 - p_{ic}) \qquad (2)$$

- ha az igének *k* különböző argumentumszerkezeti variánsa van, akkor az ige melletti V+C bővítménykombináció megjelenési valószínűsége

$$p_{V+C} = \sum_{i=1}^{k} \left( p_{Vi} \cdot \prod_{c \in C} p_{ic} \cdot \prod_{c \in AT \setminus C} (1 - p_{ic}) \right) \quad (3)$$

### 3.3 Argumentumszerkezeti vektor meghatározása – egyszerű példa

Vegyünk egy egyszerűsített példát, a *bíz/bízik* ige első (*vki bízik vmiben*) és harmadik (*vki bíz vmit vkire*) variánsát, és csak az ACC, BAN és RA bővítményeket vegyük figyelembe. A két variáns a korpuszban összesen 103-szor fordul elő, ebből 66 az első variáns, 37 a harmadik variáns előfordulási száma, vagyis $p_{V1} = 0{,}64$ és $p_{V3} = 0{,}36$. Tárgyi bővítmény 33-szor jelenik meg az ige mellett, mind a harmadik variánsnál, BAN bővítmény 69-szer, 4 kivételével az első variánsnál, RA bővítmény pedig 39-szer, kettő kivételével a harmadik variánsnál.

A korpuszból automatikusan kigyűjthető adatokat az 5. táblázat tartalmazza, kiemelve az adatok száma, illetve ezekből kiszámolhatóak a bővítménykombinációk gyakorisági értékei és az összesített $\bar{v}$ argumentumszerkezeti vektor. Megjegyzem, hogy ebben a példában az egyes igei bővítménykombinációk egyes korpuszbeli megjelenései minden esetben ugyanahhoz az argumentumszerkezeti variánshoz tartoztak, nevezetesen az első három sor a $bíz_1$, a második három pedig a $bíz_3$ variánshoz, de ez nem szükségszerű. Az ige csak ACC, vagy csak ACC és BAN bővítményekkel egyszer sem fordul elő.

| kombinációk | ACC | BAN | RA | n | |
|---|---|---|---|---|---|
| V+BAN | | + | | 63 | $f_{V+BAN} = 0{,}611650$ |
| V+BAN+RA | | + | + | 2 | $f_{V+BAN+RA} = 0{,}019417$ |
| V+0 | | | | 1 | $f_{V+0} = 0{,}009709$ |
| V+ACC+RA | + | | + | 29 | $f_{V+ACC+RA} = 0{,}281553$ |
| V+ACC+BAN+RA | + | + | + | 4 | $f_{V+ACC+BAN+RA} = 0{,}038835$ |
| V+RA | | | | 4 | $f_{V+RA} = 0{,}038835$ |
| V+ACC | | | | 0 | $f_{V+ACC} = 0{,}0$ |
| V+ACC+BAN | | | + | 0 | $f_{V+ACC+BAN} = 0{,}0$ |
| össz. | 33 | 69 | 39 | 103 | |
| $\bar{v}$ | 0,320388 | 0,669903 | 0,378641 | | |

5. táblázat. a *bízik* ige megfigyelhető előfordulási adatai három bővítménytípussal kombinálva

Ezen adatok ismeretében az a feladatunk, hogy meghatározzuk azokat a $\bar{v}_1 = [p_{1ACC}; p_{1BAN}; p_{1RA}]$ és $\bar{v}_3 = [p_{3ACC}; p_{3BAN}; p_{3RA}]$ vektorokat és a $p_{V1}$ és $p_{V3}$ valószínűségi együtthatókat ($p_{V1} + p_{V3} = 1$), amelyekkel a két argumentumvariánst jellemezhetjük. A kézi annotálás segítségével megszámolt értékek a 6. táblázatban találhatóak, nekünk ezt most azonban becsülnünk kell.

| | | ACC | BAN | RA |
|---|---|---|---|---|
| bíz$_1$ ($n_1$=66) | $n_{1X}$ | 0 | 65 | 2 |
| $p_{V1}$=0,640777 | $\bar{v}_1$ | 0 | 0,984848 | 0,030303 |
| bíz$_3$ ($n_3$=37) | $n_{3X}$ | 33 | 4 | 37 |
| $p_{V3}$=0,359223 | $\bar{v}_3$ | 0,891892 | 0,108108 | 1,00 |

6. táblázat. a *bízik* ige két argumentumszerkezeti vektora kézi annotálással

Ha feltételezzük, hogy 2 argumentumszerkezeti variáns van, akkor a megbecsülendő adatokból az (1) és a (3) képletek szerint a következő számolt valószínűségi értékek határozhatóak meg:

$$p_{ACC} = p_{V1} \cdot p_{1ACC} + p_{V3} \cdot p_{3ACC} \quad (4)$$
$$p_{BAN} = p_{V1} \cdot p_{1BAN} + p_{V3} \cdot p_{3BAN}$$
$$p_{RA} = p_{V1} \cdot p_{1RA} + p_{V3} \cdot p_{3RA}$$
$$p_{V\text{-}0} = p_{V1} \cdot (1\text{-}p_{1ACC})\,(1\text{-}p_{1BAN})\,(1\text{-}p_{1RA}) + p_{V3} \cdot (1\text{-}p_{3ACC})\,(1\text{-}p_{3BAN})\,(1\text{-}p_{3RA})$$
$$p_{V+ACC} = p_{V1} \cdot p_{1ACC}\,(1\text{-}p_{1BAN})\,(1\text{-}p_{1RA}) + p_{V3} \cdot p_{3ACC}\,(1\text{-}p_{3BAN})\,(1\text{-}p_{3RA})$$
$$p_{V+BAN} = p_{V1} \cdot (1\text{-}p_{1ACC})\,p_{1BAN}\,(1\text{-}p_{1RA}) + p_{V3} \cdot (1\text{-}p_{3ACC})\,p_{3BAN}\,(1\text{-}p_{3RA})$$
$$p_{V+RA} = p_{V1} \cdot (1\text{-}p_{1ACC})\,(1\text{-}p_{1BAN})\,p_{1RA} + p_{V3} \cdot (1\text{-}p_{3ACC})\,(1\text{-}p_{3BAN})\,p_{3RA}$$
$$p_{V+ACC+BAN} = p_{V1} \cdot p_{1ACC}\,p_{1BAN}\,(1\text{-}p_{1RA}) + p_{V3} \cdot p_{3ACC}\,p_{3BAN}\,(1\text{-}p_{3RA})$$
$$p_{V+ACC+RA} = p_{V1} \cdot p_{1ACC}\,(1\text{-}p_{1BAN})\,p_{1RA} + p_{V3} \cdot p_{3ACC}\,(1\text{-}p_{3BAN})\,p_{3RA}$$
$$p_{V+BAN+RA} = p_{V1} \cdot (1\text{-}p_{1ACC})\,p_{1BAN}\,p_{1RA} + p_{V3} \cdot (1\text{-}p_{3ACC})\,p_{3BAN}\,p_{3RA}$$
$$p_{V+ACC+BAN+RA} = p_{V1} \cdot p_{1ACC}\,p_{1BAN}\,p_{1RA} + p_{V3} \cdot p_{3ACC}\,p_{3BAN}\,p_{3RA}$$

A célunk tehát az, hogy a $\bar{v}_1 = [p_{1ACC}; p_{1BAN}; p_{1RA}]$ és $\bar{v}_3 = [p_{3ACC}; p_{3BAN}; p_{3RA}]$ vektorokra és a $p_{V1}$ és $p_{V3}$ valószínűségi együtthatókra olyan becslést adjunk meg, amelyek alapján a (4)-ben számolt valószínűségi tényezők a ténylegesen megfigyelt $f_{ACC}$, $f_{BAN}$, $f_{RA}$, $f_{V\text{-}0}$, $f_{V\text{-}ACC}$, $f_{V\text{-}BAN}$, $f_{V\text{-}RA}$, $f_{V\text{-}ACC\text{-}BAN}$, $f_{V\text{-}ACC\text{-}RA}$, $f_{V\text{-}BAN\text{-}RA}$, $f_{V\text{-}ACC\text{-}BAN\text{-}RA}$ gyakorisági tényezőket legjobban megközelítik, vagyis az azokhoz viszonyított különbségeik négyzeteinek összege minimális:

$$(f_{ACC}\text{-}p_{ACC})^2 + (f_{BAN}\text{-}p_{BAN})^2 + (f_{RA}\text{-}p_{RA})^2 + (f_{V+0}\text{-}p_{V+0})^2 + (f_{V+ACC}\text{-}p_{V+ACC})^2 + (f_{V+BAN}\text{-}p_{V+BAN})^2 + (f_{V+RA}\text{-}p_{V+RA})^2 + (f_{V+ACC+BAN}\text{-}p_{V+ACC+BAN})^2 + (f_{V+ACC+RA}\text{-}p_{V+ACC+RA})^2 + (f_{V+BAN3RA}\text{-}p_{V+BAN+RA})^2 + (f_{V+ACC+BAN+RA}\text{-}p_{V+ACC+BAN+RA})^2 \quad (5)$$

Mivel most 3 bővítménytípus és 2 variáns van, ez egy $2\cdot(3+1)$ dimenziós térben való minimumkeresés. $k$ bővítménytípus és $n$ variáns esetében ez a keresés $n\cdot(k+1)$ dimenziós térben történik.

Természetesen elvégezhetjük a számítást több argumentumszerkezeti variánst feltételezve is. A helyes variánsszám meghatározásánál figyelembe kell venni azt, hogy egyrészt törekednünk kell a minél kisebb variánsszámra (3.1.1 szakasz), de azért az adatokat minél jobban megmagyarázni képes modellt szeretnénk kialakítani (3.1.2 szakasz).

## 4 Az argumentumszerkezeti vektort befolyásoló tényezők

Az argumentumszerkezeti vektor értékét több tényező is befolyásolja, például a korpusz egyedi tulajdonságai, ami alapján meghatározzuk a vektort, de vannak grammatikai befolyásoló tényezők is. Ezen tényezők számbavétele és a hatásuk leírása egyrészt a hatás kiküszöbölésével pontosíthatja az argumentumszerkezeti vektor meghatározását, másrészt feltárásukkal hasznos összefüggésekre lelhetünk a nyelv és a nyelvtan működését illetően.

### 4.1 Korpuszhatások

Ha az argumentumszerkezeti vektort korpusz alapján határozzuk meg, akkor a vektor a korpusz adatait fogja visszatükrözni, más korpuszt választva más értékeket kaphatnánk. A korpusz mérete is befolyásolja ezt a folyamatot, nagyobb korpusz esetén csökken az adatok esetlegességének a mértéke.

Az argumentumszerkezeti variánsok egymáshoz viszonyított előfordulási valószínűsége például erősen korpuszfüggő. A különböző variánsok ugyanis különböző jelentést hordozhatnak, ezért a korpuszban szereplő szövegek típusa, témája meghatározza, hogy mely variánsok lesznek a gyakoribbak benne. A hivatalos, jogi vagy gazdasági szövegekben várhatóan kevesebbszer fordul elő a *bíz/bízik* ige 6. variánsa: *vki elbízza magát*, az iskolások fogalmazásaiban vagy a szépirodalmi szövegekben, a *vki megbíz vkit vmivel* viszont gyakoribb lesz a gazdasági hírekben.

A korpuszban szereplő szövegek típusa az egyes vektorokban megjelenő argumentumok előfordulási valószínűségét is befolyásolja. Az iskolai fogalmazásokban sokkal többször jelenik meg az első és második személyű névmás, ugyanígy a szépirodalmi művekben is, mint a formálisabb szövegekben, a névmások viszont hajlamosabbak a meg nem jelenésre, mint a kifejtett főnévi kifejezések. Ezért ezekben fogalmazásokban várhatóan kisebb lesz az alanyi és tárgyi bővítmények megjelenési valószínűsége, mint a jogi szövegekben (ha csak ezt a különbséget vesszük figyelembe). De a fogalmazások és az irodalmi művek között is találhatunk különbséget, például a nem kötelező bővítmények megjelenési valószínűségét illetően.

Az előző bekezdésben ismertetett hatások azonban nem közvetlenül szövegtípusok és az argumentumszerkezeti vektorok között érvényesülnek, hanem a következő szakaszban ismertetett grammatikai hatásokon keresztül. Az egyes szövegtípusokra jellemző ugyanis azok névszó- és bővítményhasználata, és ha ezeknek tényezőknek az argumentumszerkezeti vektorokra való befolyását elkülönítve tudjuk jellemezni, akkor már csak azt kell megállapítani, hogy ezek a tényezők mennyire jellemzők a korpuszokra.

### 4.2 Grammatikai hatások

#### 4.2.1 Pro drop

A magyarban a hangsúlytalan alany és tárgy esetű névmások elhagyhatóak. Ha korpuszvizsgálattal meghatározzuk, hogy az alanyi, illetve a tárgyi vonzattal rendelkező igék alanya, illetve tárgya mekkora valószínűséggel lesz (megjelenő vagy elhagyott) személyes névmás ($p_{pron\text{-}NOM}$, $p_{pron\text{-}ACC}$), továbbá meghatározzuk, hogy névmási alany és tárgy mekkora valószínűséggel kerül elhagyásra ($p_{prodrop\text{-}NOM}$, $p_{prodrop\text{-}ACC}$), akkor a névmáselhagyás hatása kiküszöbölhető. Ha ugyanis az ilyen alanyok és tárgyak nem lennének elhagyva, akkor a ténylegesen megfigyelhető adatokból számolt $p_{iNOM}$, illetve $p_{iACC}$ alanyi és tárgyi valószínűség helyett a $p'_{iNOM} = p_{iNOM} + p_{pron\text{-}NOM} \cdot p_{prodrop\text{-}NOM}$ stb. korrigált alanyi előfordulási valószínűséggel dolgozhatunk.

A $p_{pron\text{-}NOM}$ és $p_{prodrop\text{-}NOM}$ valószínűségek nem csak egy igére vagy igevariánsra jellemző értékek, hanem az összes igére és variánsra: $p_{pron\text{-}NOM}$ korpuszfüggő valószínűség, $p_{prodrop\text{-}NOM}$ viszont korpuszfüggetlen.

### 4.2.2 Ellipszis

Nem csak az alanyi és a tárgyi névmás hagyható el a magyarban, hanem más vonzatelhagyási jelenségek is megfigyelhetőek. Az összetett mondatokra, különösen a mellérendelésekre jellemző, hogy ha ugyanaz a kifejezés több tagmondatban is jelen van, akkor csak az egyik tagmondatban jelenik meg: *Péter csak találkozott ~~Marival~~, de Pál beszélgetett is Marival.* A különböző típusú vonzatok elliptálhatósága a névmáselhagyási jelenséghez hasonlóan egy valószínűségi értékkel jellemezhetők, bár ebben az esetben a korpuszhatás nehezebben elhatárolható a teljes valószínűségi értéktől, és a különböző igék is különböző mértékben hajlamosak az ellipszisben való részvételre.

### 4.2.3 Szabad bővítmények igefüggetlen megjelenése

A 3. szakaszban bevezetett argumentumszerkezeti vektor nem tesz különbséget vonzat és szabad bővítmény között, azonban a vonzat és a szabad bővítmény ebben az értelmezésben is jól elkülöníthető egymástól: a hagyományosan vonzatnak tekintett bővítmények nagy valószínűséggel megtalálhatóak az ige mellet ($p>0,6$), míg a szabad bővítmények előfordulási gyakorisága kicsi ($p<0,4$). Ez alól csak az alanyi és tárgyi bővítmények jelenthetnek kivételt, de azok meg mindig vonzatok.

Míg a vonzatok esetében a vektor megfelelő értékének értelmezésekor azt kell megindokolni, hogy mikor, mekkora valószínűséggel nem jelenik meg mégsem az ige mellett, a szabad bővítményeknél a megjelenést kell alátámasztani: mivel a szabad bővítmény nem kötelező, mikor jelenik meg mégis, mekkora ennek a valószínűsége. A szabad bővítményeket nem az igék szelektálják, ezért egy adott szabad bővítménytípus megjelenés valószínűsége csak kis mértékben igefüggő, a különböző igék és argumentumszerkezeti variánsok melletti megjelenési gyakorisága állandónak tekinthető. Az igék különböző szabadbővítmény-felvevő hajlandósága csak közvetetten köthető az igéhez: a szabad bővítmények jellemzője az, hogy milyen típusú, milyen jelentéskategóriájú igéhez tudnak kapcsolódni, ezáltal az igék osztályozása áttételesen ad magyarázatot a varianciára. Mindazonáltal egy szabad bővítménytípus megjelenési valószínűségét több ige vizsgálatával korpusz alapján egységesen lehet megállapítani, az egyes alternánsok esetében pedig ezt lehet irányadónak venni.

### 4.2.4 Szabadbővítmény-csoportok

A korábbiakban az egyes bővítménytípusokat a bővítmény esetével vagy névutójával jellemeztük. Azonban vannak olyan esetcsoportok, amelyeket érdemesebb együtt kezelni, ugyanannak a bővítménytípusnak a különböző megnyilvánulásainak tekinteni őket. Például a helyhatározói funkciójú bővítmények hasonlóan működnek, ugyanolyan predikátumtípusokhoz illeszthetőek, egymással helyettesíthetőek, bár a morfológiai esetük többféle is lehet: BAN, ON, NÁL vagy MELLETT stb. Az ugyanolyan funkciójú, de különböző morfológiai esetű szabad bővítményeket ezért kívánatos egy bővítménytípusnak tekinteni és egységesen meghatározni a megjelenési valószínűségét, gyakoriságát: $f_{\text{HELY}}$. Ugyanakkor az ugyanolyan funkciójú, de különböző esetű bővítmények egyenként is jellemezhetőek aszerint, hogy az adott funkciójú megjelenő szabad bővítmény mekkora valószínűséggel realizálódik egy bizonyos esetű kifejezésként. Ez esetenként változó nagyságú lehet, a realizálódási értékek nagysága független az igétől, ami mellett megjelennek. Ha egy argumentumvariáns mellett a kérdéses bővítménytípusok (esetek) a funkcióra jellemző valószínűségekkel jelennek meg egymáshoz képest, akkor az adott funkciót betöltő szabadbővítmény-csoport tagjainak tekinthetőek.

### 4.2.5 Argumentumszerkezet-típusok, argumentumszerkezet-változtató műveletek

Az argumentumszerkezeti vektorok segítségével az egyes igei lexikai egységek is összevethetőek: megvizsgálhatjuk, hogy melyek azok a lexikai egységek, variánsok, amelyek azonos vagy nagyon hasonló argumentumszerkezeti vektorral jellemezhetőek. Ezek – az igék jelentésének az előzetes vizsgálata és ismerete nélkül – utalhatnak arra, hogy a talált hasonló lexikai egységek valamilyen szintaktikai vagy szemantikai tulajdonságukban megegyeznek, ugyanabba a szintaktikai vagy szemantikai csoportba tartoznak.

Továbbá megvizsgálható, hogy vannak-e olyan igealakok, amelyek hasonló argumentumszerkezeti variánsokkal rendelkeznek, van-e értelmezhető grammatikai kapcsolat a több argumentumszerkezeti variánssal rendelkező kifejezések variánsai között.

Érdekes grammatikai általánosítások megfogalmazásához vezethet annak vizsgálata, hogy a nyilvánvaló morfológiai kapcsolatot mutató tövek különböző argumentumszerkezeti variánsai között van-e valamilyen kapcsolat. A *készül-készít, hárul-hárít, gurul-gurít* unakkuzatívuszi-akkuzatívuszi párok argumentumszerkezet-variánsai például egyértelműen párba állíthatóak, de a párhuzamokon túl érdekesek az egyes argumentummegjelenési valószínűségek változásai is, illetve az egyediségek is: melyek azok a variánsok, amik csak az egyik párnál jelennek meg, a többinél nem. Ezek az egyedi variánsok idiomatikus variánsai.

Például a *készül-készít* ( és a *gurul-gurít* stb.) párok esetében megfigyelhető, hogy a *készül* ige alanya a *készít* ige tárgyának feleltethető meg (pl. *elkészült a leves – Péter elkészítette a levest* vagy *a cipő bőrből készült – a cipész bőrből készítette a cipőt*), vagyis az igék alanyi és tárgyi bővítményeinek a megjelenése korrelál. Vannak azonban olyan bővítményi környezetek, ahol ez a korreláció nem figyelhető meg (pl. *Péter Debrecenbe készül* – ?*Mari Debrecenbe készíti Pétert*), így ezek a variánsok nem célpontjai részt az argumentumszerkezet-változtató műveletnek: idiomatikusabbak.

Hasonlóan lehet jellemezni az egyes igeképzők argumentumszerkezet-változtató képességét is.

### 4.2.6 Örökölt vonzatok

Nem csak az ige argumentumai, vagyis a kötelező és szabad bővítményei jelenhetnek meg az ige mellett ugyanabban a tagmondatban, hanem más szintaktikailag önálló összetevők is. Ilyenek például az alany jelenlétében, de nem vele egy összetevőt alkotó VAL típusú bővítmények (pl. *Péter elment Marival a moziba*), vagy a szétváló birtokos kifejezések esetében a DAT birtokos (pl. *Péternek elment a barátja a moziba*). Ezek a bővítmények nem argumentumai az igének, nincsenek azzal szemantikai kapcsolatban, de az olyan argumentumszerkezeti modellekben, ahol csak a morfológiai és szintaktikai tényezőket vesszük figyelembe a bővítménység megállapításánál, ezek nem különböztethetőek meg egyszerűen a szabad bővítményektől. (Az ilyen jellegű problémák egyedi kezelésére lásd pl. Sass Bálint disszertációjának 2.2. szakaszát: [12].)

Azonban arról, hogy ezek milyen argumentumok mellett jelenhetnek meg (annak típusával vagy szótövével azonosítva), lehet feltételeket meghatározni, mint ahogy ahhoz is lehet valószínűségi értéket rendelni, hogy a megadott feltételek teljesülése esetén mekkora valószínűséggel jelenik meg az ilyen örökölt bővítmény.

Külön említést érdemelnek azok az igék, amelyeknek főnévi igeneves vonzatuk is van: ezeknek a főnévi igeneveknek a vonzatai, argumentumai megszorítás nélkül kerülhetnek a mátrix igével azonos tagmondatba is a mátrix ige bővítményeként (pl. *A házi feladatot tegnap elfelejtettem megcsinálni*, ahol a tárgy nem az *elfelejt* ige saját tárgya).

### 4.2.7 Igekötők

A 2. szakaszban az igekötős igéket és az igekötő nélkülieket megkülönböztettük, külön argumentumszerkezeti variánsnak tekintettük őket, ezáltal az igekötőket az igekötős ige részeként elemeztük, nem az ige önálló argumentumaként. Az igekötők azonban időnként átveszik valamely kötelező argumentum szerepét, a *rá* igekötő jelenlétében például nem jelenhet meg a *néz* ige mellett az egyébként kötelező névmási *rá* vonzat: *Péter ránézett *rá*. Az első személyű *rám* névmási vonzat esetében viszont az igekötő az, amit nem tehetjük ki: *Péter* (**rá*)*nézett rám*. Egyébként pedig, főnévi fejű RA bővítmény mellett, az igekötőt szabadon megjelenhet vagy elhagyható: *Péter Marira nézett/Péter ránézett Marira*. Ezekben az esetekben nem egyértelmű, hogy két argumentumszerkezeti variánst látunk-e, egy igekötőset és egy igekötő nélkülit, vagy pedig hármat, ahol a harmadik egy olyan igekötős *néz* variáns van, aminek nincs RA vonzata. És bármelyik megoldást is választjuk, a *Péter rám nézett* mondat igéjének besorolása elméleti szempontból is kérdéses.

Vagy választhatjuk azt a leírási módot is, hogy az igekötőket mint önálló mondatbeli összetevőket bővítménynek tekintjük, és megjegyezzük, hogy a *rá* igekötői bővítmény és a RA esetű bővítmény hajlamosak együtt megjelenni, mintegy szétváló, de szemantikailag összetartozó bővítményt alkotva. Ekkor hasonló leírást kívánnak meg, mint az elváló birtokos és a birtok: az egyik a bővítmény, a másik pedig annak a vonzata, ami esetlegesen az ige bővítményeként jelenik meg, példánkban a RA esetű bővítmény vezeti be a klitikumszerű *rá* igekötői bővítményt.

Máskor viszont az igekötős ige argumentumszerkezetében olyan vonzat jelenik meg, ami az igekötő nélküli esetben nem engedélyezett: **Péter megy az ajtón* de *Péter átmegy az ajtón*. Ebben az esetben az igekötő megjelenése az, ami engedélyezi az ON vonzat megjelenését.

Az igekötős igék argumentumszerkezeti vektorainak a vizsgálatával megállapíthatjuk, hogy egy adott igekötő milyen bővítménytípusokkal szokott együtt megjelenni: ezeket az igekötő-bővítménytípus párokat így összetartozókként kezelhetjük. Ugyanezen igék igekötő nélküli változatainak a vizsgálatával leírhatjuk, hogy az igekötő megjelenése milyen argumentumszerkezeti változást okoz, mint ahogyan a képzők argumentumszerkezet-változtató képességét is leírjuk. Megállapíthatjuk, hogy milyen feltételek mellett, vagyis milyen argumentumszerkezeti variáns esetében lehet egy bizonyos igekötővel ellátni egy igét, és hogy az igekötő megjelenése hogyan változtatja meg az argumentumszerkezeti variáns argumentumvektorát. Ha a feltárt feltételeknek megfelelő variáns hiányában is megjelenhet egy igekötő, vagy nem az elvárt módon változtatja meg az ige argumentumszerkezetét, akkor idiomatikus igekötő-ige párt találtunk. Az *el* igekötőtől például azt várnánk, hogy ha van valamilyen argumentumszerkezet-változtató képessége, akkor valamilyen BÓL/BA jellegű bővítmény megjelenését erősíti, nem pedig mondjuk BAN bővítményt (*Péter elindult az iskoláBÓL*/*iskoláBA*/?*iskolában*). A korábban vizsgált *bízik* ige esetében azonban ACC bővítmény megjelenését tapasztalhatjuk: *vki elbízza magát*. Ez nem magyarázható az igekötő szokásos viselkedésével, vagyis az *elbízik* igekötős ige idiomatikus szerkezetű.

## 5 Összefoglalás, alkalmazási lehetőségek

A tanulmány a lexikai egységek, tipikusan igék argumentumszerkezetének a leírására javasol egy új reprezentációs formát, ami nem a klasszikus kötelező vonzat – szabad bővítmény bináris oppozíciós lehetőségeket ragadja meg. Ehelyett az egyes bővítménytípusoknak a korpuszban való megjelenési gyakoriságai alapján a típusokhoz egy-egy valószínűségi értéket rendel, így az argumentumszerkezeti variánsok egy argumentumszerkezeti valószínűségi vektorral jellemezhetőek. A javasolt módszer kizárólag a korpuszbeli adatok morfológiai és szintaktikai tulajdonságaira támaszkodik, nem is célja a lexikai elemek szemantikai jellemzése, továbbá nem a vizsgált lexikai elemek környezetében levő kifejezések alakját vagy szótövét veszi figyelembe, hanem csak néhány absztraktabb, általánosabb tulajdonságát, ezért nem tekinthető a szóbeágyazási modellek egy változatának [5]. Az argumentumszerkezet többdimenziós értelmezése miatt inkább Sass Bálint duplakocka-modelljével [6] rokonítható.

Az argumentumszerkezeti variánsok argumentumszerkezeti vektorként való értelmezése új elméleti modellként a grammatikaelméletekben hozhat új eredményeket: a 4.2. szakaszban bemutatott, az argumentumszerkezeti vektorokat befolyásoló grammatikai tényezők feltárásával korpuszra, vagyis valós nyelvi adatokra támaszkodó grammatikai összefüggéseket lehet megfogalmazni. Az elméleti eredményeken túl azonban az argumentumszerkezeti vektorok a nyelvfeldolgozás során is több helyen alkalmazhatóak:

- Az argumentumszerkezeti vektorok a bővítmények valószínűségi értékeinek felhasználásával közvetlenül átalakíthatóak valószínűségi frázisstruktúra nyelvtanná ([13] 494.o.).
- A régensek környezetét vizsgálva valószínűsíthetjük, hogy az adott mondatban melyik argumentumszerkezeti variánsát találjuk. Abban az esetben, amikor a különböző argumentumszerkezeti variánsok más jelentést hordoznak, ez a jelentésegyértelműsítést is magával hozza.
- Az alanyi és tárgyi névmások elhagyásának a valószínűségét ismerve egy vizsgált szövegben az is megállapítható lehetne a legvalószínűbb argumentumszerkezeti variáns megtalálásával, hogy mellette szerepel-e zéró névmás, ami az anaforafeloldás során fontos információ.
- Elég nagy korpusz segítségével a szövegtípusok argumentumszerkezetivektor-változtató képességét is megadhatjuk, aminek a segítségével egy ismeretlen szöveg típusára adhatunk becsléseket.
- A lexikai elemek szokásos argumentumvektorainak ismeretében egy nyelvhasználónál az azoktól eltérő vektorok meglétéből következtethetünk a beszélő nyelvhasználati tulajdonságaira, így például a beszélő korára, társadalmi helyzetére vagy a mentális képességeire, nyelvi zavaraira is.

Mindezek fényében a lexikai elemek argumentumszerkezeti variánsainak vektoros reprezentációja mind elméleti, mind gyakorlati szempontból átgondolni érdemesnek látszik.

## Bibliográfia


1. Carnie, A.: Syntax: a generative introduction. Wiley-Blackwell, Hoboken, New Jersey (2013).
2. Pollard, C., Sag, I.A.: Head-driven phrase structure grammar. CSLI, University of Chicago Press, Stanford, Chicago (1994).
3. Szécsényi T.: Magyar mondatszerkezeti jelenségek elemzése HPSG-ben. In: Bartos H. (ed.) Új irányok és eredmények a mondattani kutatásban. pp. 99–138. Akadémiai Kiadó, Budapest (2011).
4. Kovács V., Simkó K., Szécsényi T.: Szabályalapú szintaktikai elemző szintaktikai szabályok nélkül. In: Tanács A., Varga V., and Vincze V. (eds.) XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). pp. 251–259. Szegedi Tudományegyetem, Szeged (2016).
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. ArXiv13104546 Cs Stat. (2013).
6. Sass B.: Az igei szerkezetek algebrai struktúrája, avagy a duplakocka modell. Argumentum. 14, 12–44 (2018).
7. Kálmán L.: Miért nem vonzanak a régensek? In: Kálmán L. (ed.) KB 120. A titkos kötet. Nyelvészeti tanulmányok Bánréti Zoltán és Komlósi András tiszteletére. pp. 229–246. MTA Nyelvtudományi Intézet, Tinta Könyvkiadó, Budapest (2006).
8. Kálmán L.: Bővítménykeretek mint konstrukciók. In: Kas B. (ed.) "Szavad ne feledd" Tanulmányok Bánréti Zoltán tiszteletére. pp. 61–72. MTA Nyelvtudományi Intézet, Budapest (2016).
9. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Matoušek, V., Mautner, P., and Pavelka, T. (eds.) Text, Speech and Dialogue. pp. 123–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2005).
10. Vincze, V., Szauter, D., Almási, A., Móra, G., 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).
11. Müller, C., Strube, M.: Multi-level annotation of linguistic data with MMAX2. In: Braun, S., Kohn, K., and Mukherjee, J. (eds.) Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods. pp. 197–214. Peter Lang, Frankfurt a.M., Germany (2006).
12. Sass B.: Igei szerkezetek gyakorisági szótára - Egy automatikus lexikai kinyerő eljárás és alkalmazása, http://real-phd.mtak.hu/342/, (2011).
13. Jurafsky, D., Martin, J.H.: Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, Pearson Education Internat, Upper Saddle River (2009).

# Véges erőforrás végtelen sok igekötős igére

Kalivoda Ágnes

MTA Nyelvtudományi Intézet
MTA–PPKE Magyar Nyelvtechnológiai Kutatócsoport
Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar
kalivoda.agnes@nytud.mta.hu

**Kivonat** A PrevLex egy szabadon elérhető, manuálisan ellenőrzött erőforrás, amely 54 955 igekötős igét tartalmaz gyakorisági adatokkal együtt. Bár lefedi az MNSZ 2.0.4 korpusz összes igekötős finit igéjét, soha nem lehet teljes: bizonyos igekötők rendkívül produktívak, és tetszőleges számú új szó képezhető velük. A cikk központi témája az, hogyan mérhető az igekötőknek ez a tulajdonsága, és hogyan használhatók fel a kvantitatív eredmények a lexikai erőforrások teljesebbé tételére. Az ismeretlen szavak mintázatainak számítógépes vizsgálata rámutat azokra a szabályokra, amelyekkel az ilyen szavak nagy része automatikusan felvehető a lexikonba. Nyelvészeti szempontból szintén lényegesek ezek a szabályok, mivel az anyanyelvi beszélő is ezek mentén képes korábban ismeretlen szavakat alkotni és érteni.
**Kulcsszavak:** igekötős igék, produktivitás, korpusznyelvészet

## 1. Bevezetés

A morfológiai produktivitás szerves része a természetes nyelvek működésének. Ennek segítségével folyamatosan, tudatos erőfeszítés nélkül hozunk létre új szavakat [1]. Nyelvtechnológiai szempontból ez felvet egy fontos kérdést: Hogyan dolgozzunk fel olyan szavakat, amelyek nem szerepelnek a lexikonban? A neurális hálón alapuló módszerek számára ez kevésbé problémás, a lexikalista megközelítésben viszont nehézséget jelent. A cikk az utóbbi szellemében járja körül a problémát, az igekötős igék morfológiai produktivitását vizsgálva.

A cikk első felében bemutatom a Magyar Nemzeti Szövegtár 2.0.4 [2] felhasználásával készült PrevLex-et[1], amely a magyar igekötős igék jelenleg legbővebb, manuálisan ellenőrzött táblázata. Szerepelnek benne a korpuszban `UNKNOWN`-nak címkézett szavak és a hapaxok (egyszer előforduló szavak) is. Az utóbbiak alkalmassá teszik a PrevLex-et arra, hogy meg lehessen vele határozni az egyes igekötők produktivitásának mértékét. Erre teszek kísérletet a cikk második felében. A mérések alapján sorra veszem a legproduktívabb igeképzési szabályokat, valamint a produktivitás kapcsolatát a stílusregiszterrel és a gyakorisággal. Végül szó lesz arról, hogy a cikkben ismertetett módszert lehet-e használni az igekötő-állomány meghatározására.

[1] https://github.com/kagnes/prevlex

## 2. A PrevLex

### 2.1. Az adatfeldolgozás menete

A PrevLex előállításához közvetlenül az MNSZ 2.0.4 forrásfájlt használtam. Három szűrést végeztem az eredeti korpuszon annak érdekében, hogy a lehető legjobb minőségű szöveganyagot kapjam. Egyrészt kiszűrtem a verseket, mivel sokuk nem természetes nyelvhasználatot tükröz. Másrészt – amennyire csak lehetett – eltávolítottam az idegen nyelvű, valamint a magyar, de ékezet nélkül írt mondatokat, mert torzíthatták volna a keresések eredményét. Például az ékezetet eleve nem tartalmazó igekötős igék sokkal gyakoribbnak tűntek volna, mint az ékezetet tartalmazók. Ehhez azt a heurisztikát alkalmaztam, hogy töröltem minden olyan mondatot, amelyben a tokenek 80%-a `UNKNOWN` vagy `SKIP` elemzést kapott. Ez a módszer inkább a pontosságnak, mintsem a fedésnek kedvezett. Végül igyekeztem kiszűrni a korpuszban található duplumokat. Itt is a pontosságot tartottam szem előtt. Csak a nyolc tokennél hosszabb mondatokat vettem figyelembe a szűrésnél, feltételezve, hogy ennél rövidebb mondatoknál (pl. köszöneséknél) természetes lehet a többszörös jelenlét. Még ezzel az óvatos módszerrel is rendkívül magasnak bizonyult a duplumok aránya (20,12%), a személyes alkorpuszon belül akadt olyan – meglehetősen hosszú – mondat, amely száznál többször ismétlődött. A szűrések eredményét az 1. táblázat foglalja össze.

| **korpusz** | **token** | **százalék** |
|---|---:|---:|
| **eredeti MNSZ2** | **1 348 000 000** | **100** |
| versek | 5 661 000 | 0,42 |
| `UNKNOWN`/`SKIP` | 26 825 200 | 1,99 |
| duplumok | 271 217 600 | 20,12 |
| **módosított MNSZ2** | **1 044 296 200** | **77,47** |

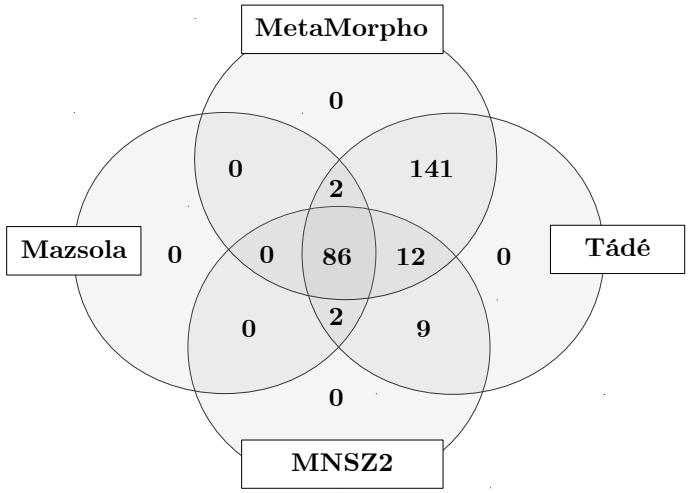
1. táblázat: Az MNSZ 2.0.4 mérete a versek, értelmes elemzés nélküli mondatok, valamint a duplumok szűrése előtt és után. A tokenszám írásjelekkel együtt értendő.

A lehetséges igekötők listája a Manócska[2] integrált igei vonzatkeret adatbázisból származik [3]. Olyan szavak szerepelnek benne, amelyeket a magyar igei vonzatkerettárak közül legalább egy igekötőnek jelöl. Az adatbázis készítői átnézték ezeket a szavakat, és javították az egyértelműnek tűnő hibákat (pl. a *vissz*, *nyug* igekötőnek jelölését a *visszhangoz*, *nyugdíjaz* szavak esetében). Ennek ellenére a végső lista hosszú – összesen 252 tagot számlál –, és több mint kétharmada esetében (pl. *szénné, pofon, zsebre*) az igekötői státusz erősen vitatható. Ez egyúttal jól tükrözi azt is, mennyire nincs egyetértés abban, hogy mely szavakat tekintjük igekötőnek (erről áttekintést ad Komlósy [4]). A 1. ábra azt

[2] http://github.com/ppke-nlpg/manocska

szemlélteti, hogy melyik erőforrás hány igekötőt nevez meg. A MetaMorpho [5] és a Tádé [6] kezelik a legtágabban ezt a kategóriát. Az MNSZ2 alapján készült listák [7] feleannyi jelöltet sem tartalmaznak, a legszigorúbb pedig a Mazsola ([8] és [9]), 90 igekötővel.

A MetaMorpho adatbázis esetében elsősorban a szubjektív annotátori osztályozás határozza meg, hogy mi igekötő és mi nem. A többi erőforrásnál az tűnik vízválasztónak, hogy a kérdéses szó elég gyakori-e, és elég gyakran van-e egybeírva az igével. Például az *utol* és a *zokon* is csak egy-egy igével állnak, de az *utolérte* lényegesen gyakoribb, mint a *zokonvette*. Csak az előbbi annotált igekötős igeként. Megjegyzendő, hogy az egybeírás nem szükséges feltétele az igekötős igévé válásnak. Az egybeírásra való hajlandóságban számíthat a szavak hossza és az is, hogy a főnév ragos vagy ragtalan-e.




1. ábra: A Manócskában szereplő erőforrások összesen 252 szót minősítenek igekötőnek. A halmazok azt mutatják, hogy az egyes erőforrások hány másikkal és hány darab szót illetően értenek egyet.

A kiinduló, 252 szavas listában 13 hibát találtam (ilyen pl. a *vízi*, amely egyszerűen bizonyos szóösszetételek első tagja). Így végül 239 szó maradt, amely az ismertetett források valamelyike szerint igekötő, és én is fenntartom ennek a lehetőségét – hangsúlyozva, hogy az igekötő-állomány összetétele bizonytalan.

A következő lépésben lekértem a módosított MNSZ2-ből minden olyan finit igeként vagy `UNKNOWN`-ként annotált szót, amely egy adott igekötővel kezdődik. Ennek a döntésemnek két része is magyarázatra szorul. Először is az, hogy miért csak a finit igéket vettem figyelembe, amikor az igekötők például igenevekhez is kapcsolódhatnak. A korpuszvizsgálat során azt tapasztaltam, hogy az igenevek esetében erős a tendencia az igemódosító és az igenév egybeírására (pl. *jóltáplált vendég, földreszállt angyal*), míg ugyanezeket az igemódosítókat a fi-

nit igével már kevésbé írják egybe. Valószínű, hogy az igenevek figyelembevétele nem változtatott volna jelentősen a PrevLex összetételén, viszont sokkal több ellenőrizendő adathoz vezetett volna. Másodszor, az `UNKNOWN` szavakra azért volt szükség, mert sok jó találat csak így jelenik meg (pl. *visszacuccol, felstócol, benyammog*). Ugyanakkor az `UNKNOWN` szavak legnagyobb része hibás találat (elírt vagy idegen nyelvű szó), és a finit igék között is akadnak álpozitív találatok (pl. a *túlélősködik* mint igekötős ige). Emiatt az eredményül kapott, közel 178 000 szavas listát át kellett nézni.

Ez a munka körülbelül huszonegy órát vett igénybe. Először eltávolítottam a lehetséges igekötőket a szavak elejéről, és a megmaradó szórészeket néztem át aszerint, hogy egyáltalán igék-e vagy sem. Ezután a már jóval rövidebb listát átnéztem úgy, hogy az ige az adott igekötővel is megfelelő-e (ezen a szinten szűrtem ki pl. a *túlélősködik* és *feltűnősködik* igéket). Néhány olyan esetben, ahol az igekötő+ige kombináció nem volt értelmetlen, viszont nagyon valószínűtlennek tűnt, csak a konkrét szövegbeli előfordulások segítségével tudtam dönteni (pl. a *túltejesít*-ről így derült ki, hogy mindig a *túlteljesít* hibásan írt változata). Ezután lokálisan újraelemeztem a forrásfájlt a javított adatokkal (pl. a korábban `UNKNOWN` *hype-olok, hype-ol* szavakat összevontam egy lemmává). A javított korpuszból állt elő a PrevLex végső változata.

### 2.2. Nehézségek

Az adatok átnézése során többször felmerült a kérdés, hogy bizonyos szóalakokat nem kellene-e valahogyan normalizálni. Három esetben az igető okozott bizonytalanságot, mert (1) teljesen azonos jelentésű igék történetileg eltérő tőváltozattal rendelkeznek (pl. *verekedik – verekszik*), (2) két igének minimálisan eltérő töve van (pl. *gyömöszköl – gyömöcköl*), (3) egy-egy neologizmus többféle írásváltozatban létezik (pl. *dizájnol – design-ol – designol*). Egyedül az utóbbi csoport kapcsán voltam biztos abban, hogy a különbség csak ortográfiai jellegű. Ezeket a szavakat normalizáltam – rendszerint a magyar kiejtés szerint írt változatra –, mindenhol megőrizve az eredeti szóalakot is.

Elkülöníthető továbbá három olyan probléma, amely a képzőt érinti: amikor (1) két vagy több ige képzőjében csak a kötőhang tér el (pl. *feccel – feccöl – feccol*, (2) opcionálisan -ikes végződésű az ige (pl. *szörföz – szörfözik*), (3) ugyanaz az ige -(O)z és -(O)l képzővel is előfordul (pl. *offtopicol – offtopicoz*). Bár itt is szólhatnak érvek a normalizálás mellett, annyi biztos, hogy nem egyszerű ortográfiai különbségekről van szó. A (3)-asban látható példák egyelőre még ugyanazt jelentik, de elképzelhető, hogy idővel kis jelentésbeli eltérés kapcsolódik hozzájuk (ahogy azt pl. a *házal – házaz* párnál látjuk). A normalizálást ezekben az esetekben önkényesnek találtam, és nem vállalkoztam rá.

### 2.3. A PrevLex felépítése

Az erőforrás egy `TSV` fájlként érhető el, amely öt oszlopból áll. Az első oszlopban szerepel az ige (igekötő+igelemma formában). Ezt követi az MNSZ2-ben

mért tokengyakoriság. A harmadik oszlopban kétféle érték szerepelhet attól függően, hogy az ige kapott-e megfelelő annotációt az MNSZ2-ben (`FIN`, ha igen és `UNKNOWN`, ha nem). A negyedik oszlop azt jelzi, hogy az ige hány dokumentumban fordult elő. Ez fontos információ lehet akkor, ha a tokengyakoriság és a tartalmazó dokumentumok száma nincs arányban (pl. az ige százszor fordul elő, de mindössze egy dokumentumban). Utolsóként szerepel a normalizált alak, amely csak a neologizmusoknál térhet el az első oszlop tartalmától.

Bár az igekötős igék listája manuálisan ellenőrzött, a gyakorisági adatok fenntartással kezelendők. Néhány igealak ugyanis több lehetséges elemzéssel rendelkezik (pl. a *leszel* egyik lehetséges elemzése a *lenni* E/2. alakja, a másik a *leszel* igekötős ige). Ezek az elemzések sokszor eleve rosszak a forrásfájlban, így kissé torzíthatják a gyűjtött statisztikát.

| **kategória** | **típus** | **token** |
|---|---:|---:|
| összes igekötős ige | 54 955 | 11 959 379 |
| hapaxok | 22 043 | 22 043 |
| `UNKNOWN` szavak | 5 156 | 26 542 |
| `UNKNOWN` hapaxok | 3 335 | 3 335 |

2. táblázat: A PrevLex számokban. Az értékek az eredeti igealakokra vonatkoznak, nem a normalizáltakra.

A 2. táblázat számszerű áttekintést ad a PrevLex-ről. A várakozásnak megfelelően az igekötős igék Zipf-eloszlást mutatnak: néhány ige rendkívül nagy tokengyakorisággal bír, míg a hapaxok ritkák, de sokfélék. Az utóbbi tulajdonságuk miatt bizonyulnak hasznosnak a morfológiai produktivitás kvantitatív meghatározásában.

## 3. Az igekötők produktivitásának vizsgálata

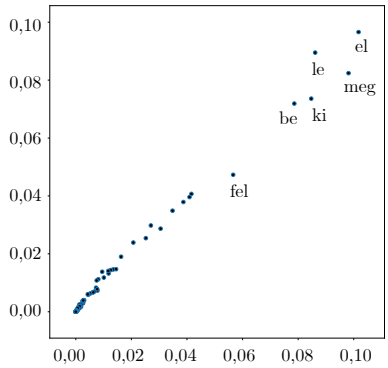
Kiefer és Ladányi (2000) [10] szerint egy szóalkotási mintát akkor tekinthetünk produktívnak, ha a minta alapján tetszőleges számú, szemantikailag transzparens szó hozható létre egy adott szemantikai tartományban. A morfológiai produktivitás esetében is – mint minden nyelvi jelenségnél – célszerű rávilágítani a kompetencia és a performancia közti különbségre. A nyelvi rendszer szintjén a produktivitás egy lehetőség, a minden további nélkül létrehozható szóalakok nem biztos, hogy ténylegesen létrejönnek, és még kevésbé valószínű, hogy benne lesznek egy korpuszban. Emiatt több elméleti morfológus (pl. Dressler [11]) nem tartja helyesnek a kompetenciaszintű lehetőség performanciaszintű valószínűségen alapuló vizsgálatát. Ez a cikk mégis az utóbbit célozza meg, mivel általában véve ezt a módszert tartom a leginkább objektívnek és reprodukálhatónak, az eredmény pedig tendenciák szintjén érdekes lehet a kompetenciát kutatóknak is.

A morfológiai produktivitás kvantitatív meghatározása Baayen nevéhez köthető ([12] és [13]). Három típust különböztet meg: a megvalósult (*realized*), a terjeszkedő (*expanding*) és a lehetséges (*potential*) produktivitást. A következőkben arról lesz szó, hogy pontosan mik ezek a típusok, és hogyan jellemeznek egy-egy igekötőt.[3]

### 3.1. Megvalósult és terjeszkedő produktivitás

A megvalósult produktivitás annak a mértéke, hogy egy adott affixum a mérés időpontjáig mennyire vett részt a szóalkotásban, tehát a múltbeli és a jelenlegi szerepe jellemezhető ezáltal. Úgy kapjuk meg, hogy az affixumot (itt: igekötőt) tartalmazó lemmák darabszámát elosztjuk a korpuszban (itt: a PrevLex-ben) található összes lemma darabszámával.

A terjeszkedő produktivitás arról ad jóslatot, hogy az affixumnak várhatóan mekkora szerepe lesz a szóalkotásban a közeljövőben. Ehhez az affixumot tartalmazó hapaxok darabszámát osztjuk el a korpuszban található összes hapax darabszámával. Ez azért is jó mérték, mert a hapaxok jelentése szinte minden esetben kompozicionális, ezért kevesebb „hamis produktív" találat adódik hozzá az eredményhez, mint a megvalósult produktivitás esetében. A 2. ábra áttekintést ad a PrevLex igekötőinek kétféle produktivitásáról.[4]




| igekötő | $P_m$ | $P_t$ |
|---|---|---|
| el | 0,1018 | 0,0970 |
| meg | 0,0984 | 0,0825 |
| le | 0,0863 | 0,0899 |
| ki | 0,0848 | 0,0737 |
| be | 0,0785 | 0,0718 |
| fel | 0,0563 | 0,0471 |
| át | 0,0418 | 0,0408 |
| bele | 0,0411 | 0,0398 |
| vissza | 0,0389 | 0,0380 |
| össze | 0,0348 | 0,0349 |

2. ábra: Az igekötők megvalósult ($P_m$) és terjeszkedő ($P_t$) produktivitása. Bal oldalt a két mérték összefüggése síkban ábrázolva látható, az X-tengelyen a $P_m$, az Y-tengelyen a $P_t$ értékeivel. Jobb oldalt a tíz legmagasabb értéket kapott igekötő szerepel.

[3] A következő igekötőket alakvariánsokként kezeltem, és összevontam minden mérés előtt (a párok első tagját meghagyva): *bele – belé, be – bé, fel – föl, odább – odébb, rá – reá, tele – teli.*

[4] Az ábrát Makrai Márton készítette. Kiegészítő információkkal együtt elérhető az alábbi címen is: https://github.com/makrai/misc/blob/master/tade/kalivoda19-mszny.ipynb

Az ősi igekötők (*meg, el, le, ki, be, fel*) mindkét mérték szerint kiugróan produktívak, bár a *fel* elmarad a többitől. Látható az is, hogy a kétféle produktivitás nagyjából egyenesen arányos. A tendenciától csak a *le* és a *meg* térnek el. A *le* azért is figyelemre méltó, mert a terjeszkedő produktivitása nagyobb, mint a *meg*-é. Ez azt jelenti, hogy várhatóan az igekötős igék alkotásában is egyre nagyobb szerepe lesz.

Mielőtt áttérnénk a harmadik produktivitás-típusra, érdemes alaposabban megvizsgálni az eddig látott eredmények okait azok fontossági sorrendjében. Három tényezőről lesz szó, amelyek közül leglényegesebbek a produktív szóképzési szabályok.

### 3.2. Produktív szóképzési szabályok

Minden olyan igekötőnél, amelynek a $P_m$-je és $P_t$-je nagyobb 0-nál, megfigyelhető az igéből igét képző produktív szabályok megléte (ilyen például a *-gat/get* gyakorító képző használata, ha ennek nincs szemantikai korlátja). Lényegesen kevesebb igekötőre igaz viszont az, hogy névszóból képzett igéhez kapcsolódhat. Épp ezért a névszóból igét képző szabályok azok, amelyek produktivitás szempontjából látványosan kiemelnek bizonyos igekötőket a többi közül.

A korábban nem hallott, de „mintaszerűen" képzett szavaknak azért tudunk jelentést tulajdonítani, mert egy ismert, alapjelentéssel bíró sémába illeszkednek (ld. 3. táblázat). Az igekötők ezt teszik specifikusabbá, illetve gyakran további jelentéssel gazdagítják a létrejövő szót.

| **alapjelentés** | **sematikus szerkezet** | **néhány példa** |
|---|---|---|
| N-nel kapcsolatosat csinál | N+(O)z(ik)<br>N+(O)l | *kisfiamozik, testékszerez*<br>*rajzszögel, bokroscsomagol* |
| N-né változik | N+Ul/sUl<br>N+Odik/sOdik | *mémesül, szinglisül*<br>*vékonyodik, tahósodik* |
| N-né változtat | N+ít/(O)sít | *részegít, szálkásít* |
| N-ként viselkedik | N+kOdik/skOdik | *vandálkodik, jópofáskodik* |

3. táblázat: A hat legproduktívabb szabály, amellyel névszóból ige képezhető. A táblázatban az N tetszőleges névszót jelöl, az /O/ archifonéma az /o/, /ö/ és /e/, az /U/ pedig az /u/ és /ü/ fonémák helyett áll.

Példaként vizsgáljuk meg – a teljesség igénye nélkül – azokat a többletjelentéseket, amelyeket a *le* igekötő ad a névszóból képzett igének. Kifejezhet (1) lefelé történő mozgást (pl. *leszánkózik, leteherautózik*), (2) egy felület lefedését valamivel (pl. *leszőnyegpadlóz, leszemfedelez*), (3) támadást vagy rombolást valamilyen eszközzel (pl. *lemacsetéz, levízágyúz*), és azt, hogy (4) valakit vagy valamit nevezünk valahogy (pl. *lelófogúzik, lebölcsészlányoz, legyíkarcoz*). Ez a

sokféle többletjelentés az oka annak, hogy terjeszkedő produktivitás szempontjából a *le* felülmúlja a *meg* igekötőt.[5]

Említést érdemelnek még azok a produktív szabályok is, amelyek egy szótagszerkezeti séma alapján tetszőleges számú hangutánzó igét hoznak létre. Ezek aztán tovább kombinálódhatnak bizonyos – főként irányjelölő – igekötőkkel. A leggyakoribbak a `CVC:+Og` (pl. *cimmog, nyammog, kaffog, hümmög*) és a `CVC:+An`, `CVC:+En` (pl. *nyekken, csisszen, toccsan, suppan*) sémákra illeszkedő igék.

### 3.3. Produktivitás és stílusregiszter

Az MNSZ 2.0.4 összesen 2952 dokumentumból áll, amelyek mindegyike egy alkorpuszhoz van rendelve. Az alkorpuszok a következők: személyes, beszéltnyelvi, szépirodalom, sajtó, tudományos, hivatalos. Ezeknek a metaadatoknak a segítségével könnyen ki lehetett mérni, hogy van-e összefüggés az igekötők produktivitása és a vizsgált szövegek stílusregisztere között. A 4. táblázatban látható, hogy tíz igekötő hapaxai milyen arányban szerepelnek az egyes alkorpuszokban.

| | **személyes** | **beszélt** | **szépirod.** | **sajtó** | **tud.** | **hivatalos** |
|---|---|---|---|---|---|---|
| **MNSZ2** | **28,96** | **7,33** | **7,75** | **35,09** | **11,34** | **9,52** |
| **el** | ↑ 34,62 | 6,10 | ↑ 28,56 | ↓ 18,21 | 9,57 | ↓ 2,93 |
| **meg** | 31,02 | 7,44 | ↑ 31,25 | ↓ 14,94 | 12,59 | ↓ 2,75 |
| **le** | ↑ 45,18 | 6,90 | ↑ 17,64 | ↓ 20,33 | 7,00 | ↓ 2,95 |
| **ki** | 33,06 | 6,53 | ↑ 29,66 | ↓ 17,36 | 10,38 | ↓ 3,01 |
| **be** | ↑ 41,70 | 8,17 | ↑ 22,09 | ↓ 16,20 | 8,77 | ↓ 3,08 |
| **fel** | ↑ 34,55 | 6,59 | ↑ 24,54 | ↓ 19,05 | 13,68 | ↓ 1,59 |
| **át** | 27,46 | 6,84 | ↑ 28,16 | ↓ 23,06 | 11,24 | ↓ 3,24 |
| **bele** | ↑ 33,99 | 6,23 | ↑ 33,13 | ↓ 17,97 | ↓ 5,99 | ↓ 2,69 |
| **vissza** | 27,74 | 7,21 | ↑ 33,21 | ↓ 20,52 | 7,21 | ↓ 4,10 |
| **össze** | 32,70 | 6,89 | ↑ 31,62 | ↓ 17,03 | 9,86 | ↓ 1,89 |

4. táblázat: A tíz legmagasabb $P_m$ és $P_t$ értékű igekötő hapaxainak százalékos eloszlása az MNSZ 2.0.4 alkorpuszaiban. A második sorban vastagon kiemelve az látható, hogy az adott alkorpusz tokenjei az MNSZ2-nek mekkora részét képezik. A ↑ azt jelzi, ha az adott igekötőnél egy alkorpusz legalább 5%-kal nagyobb arányban van jelen az eredetinél, a ↓ azt jelzi, ha legalább 5%-kal kisebben.

Ahogy várható volt, a formális regiszter (a hivatalos, tudományos és sajtószövegek nyelvezete) kevés teret enged a produktivitásnak. Érdekes viszont, hogy a sajtónyelvre a tudományoshoz képest kevésbé jellemző a produktivitás, pedig

[5] A többletjelentések is eltérő produktivitással bírnak, például a *le* esetében a (4)-es jelentés jóval produktívabbnak tűnik, mint a (3)-as. A jelentéscsoportok automatikus elkülönítése nem lehetetlen ugyan – például szóbeágyazást alkalmazó módszerekkel –, de komoly utómunkálatot igényel, ezért ebben a cikkben nem vállalkozok rá.

a beszélt nyelv jobban hat rá. Az új szóalakok alkotása az informális regiszterhez (főképpen a személyes szövegekhez) és a szépirodalomhoz köthető. Néhány igekötő (pl. *tova, által*) produktivitása szinte csak a szépirodalmi alkorpuszban mutatkozik meg.

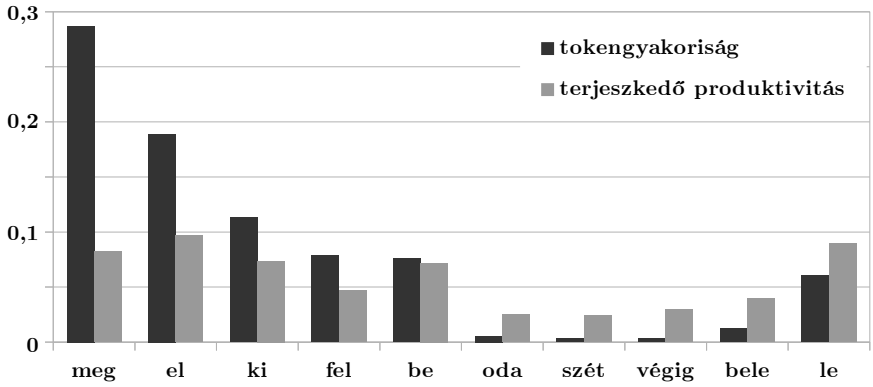
### 3.4. Produktivitás és gyakoriság

A kapott eredmények alapján feltételezhetnénk, hogy a produktivitás szorosan összefügg a gyakorisággal. Ez nem így van: jó ellenpélda az *agyon*, amely a tokengyakoriság szerinti rangsorban a 46., a $P_t$ szerintiben a 20. helyet foglalja el. A produktív affixumok nem feltétlenül gyakoriak. A gyakoriság és a produktivitás kapcsolata egy 2x2-es mátrixszal írható le, amelyet az 5. táblázat szemléltet.

| | **produktív** | **nem produktív** |
|---|---|---|
| **gyakori** | • sok típus, sok token<br>• pl. *el, be* | • kevés típus, sok token<br>• pl. *létre, egyet* |
| **nem gyakori** | • sok típus, kevés token<br>• pl. *pofon, szénné* | • kevés típus, kevés token<br>• pl. *hajba, síkra* |

5. táblázat: A produktivitás és a gyakoriság kapcsolata, Pakerys [14] alapján.

A szoros összefüggés hiányát igazolja a 3. ábra is. Az egyik szélsőséges csoportot a *meg*, *el*, *ki*, *fel* és *be* igekötők alkotják, amelyek sokkal kevésbé produktívak, mint ahogy a gyakoriságuk alapján várnánk. A másik szélsőséges csoport az *oda*, *szét*, *végig*, *bele* és *le*, amelyek esetében épp az ellenkező tendencia látható.




3. ábra: Az a tíz igekötő, amelynél a legnagyobb az eltérés a tokengyakoriság és a terjeszkedő produktivitás mértéke között.

### 3.5. Lehetséges produktivitás

A Baayen által definiált produktivitás-típusok harmadik tagja a lehetséges produktivitás. Ez az egészen távoli jövőről ad jóslatot: mik azok a most még viszonylag ritkán előforduló affixumok, amelyeknek jó esélye van arra, hogy később sok szó képzésében vegyenek részt? A terjeszkedő produktivitáshoz hasonlóan ez is hapax-alapú mérték, de az eddig látottaktól élesen eltérő eredményt hoz.

Úgy kapjuk meg, hogy egy adott affixumhoz tartozó hapaxok tokengyakoriságát elosztjuk az affixumhoz tartozó összes szó tokengyakoriságával. A méréshez célszerű gyakorisági küszöböt választani. Minél kisebb tokengyakorisággal osztunk, annál magasabb – és annál kevésbé informatív – lesz a lehetséges produktivitás. A mérést 5-ös és 5000-es küszöbbel (ld. 6. táblázat) végeztem el.

| igekötő | token | hapax | $P_1$ |
|---|---|---|---|
| mennybe | 6 | 4 | 0,6667 |
| oldalba | 18 | 9 | 0,5000 |
| égbe | 10 | 5 | 0,5000 |
| szarrá | 18 | 8 | 0,4444 |
| szénné | 7 | 3 | 0,4286 |
| fejen | 16 | 6 | 0,3750 |
| torkon | 11 | 4 | 0,3636 |
| seggre | 9 | 3 | 0,3333 |
| tűzbe | 13 | 4 | 0,3077 |
| porba | 10 | 3 | 0,3000 |

| igekötő | token | hapax | $P_1$ |
|---|---|---|---|
| tele | 5 824 | 225 | 0,0386 |
| agyon | 7 852 | 293 | 0,0373 |
| körbe | 11 304 | 298 | 0,0264 |
| ide | 15 499 | 305 | 0,0197 |
| körül | 12 940 | 238 | 0,0184 |
| hátra | 8 381 | 142 | 0,0169 |
| végig | 41 772 | 629 | 0,0151 |
| utána | 8 407 | 125 | 0,0149 |
| előre | 12 633 | 166 | 0,0131 |
| egybe | 10 163 | 132 | 0,0130 |

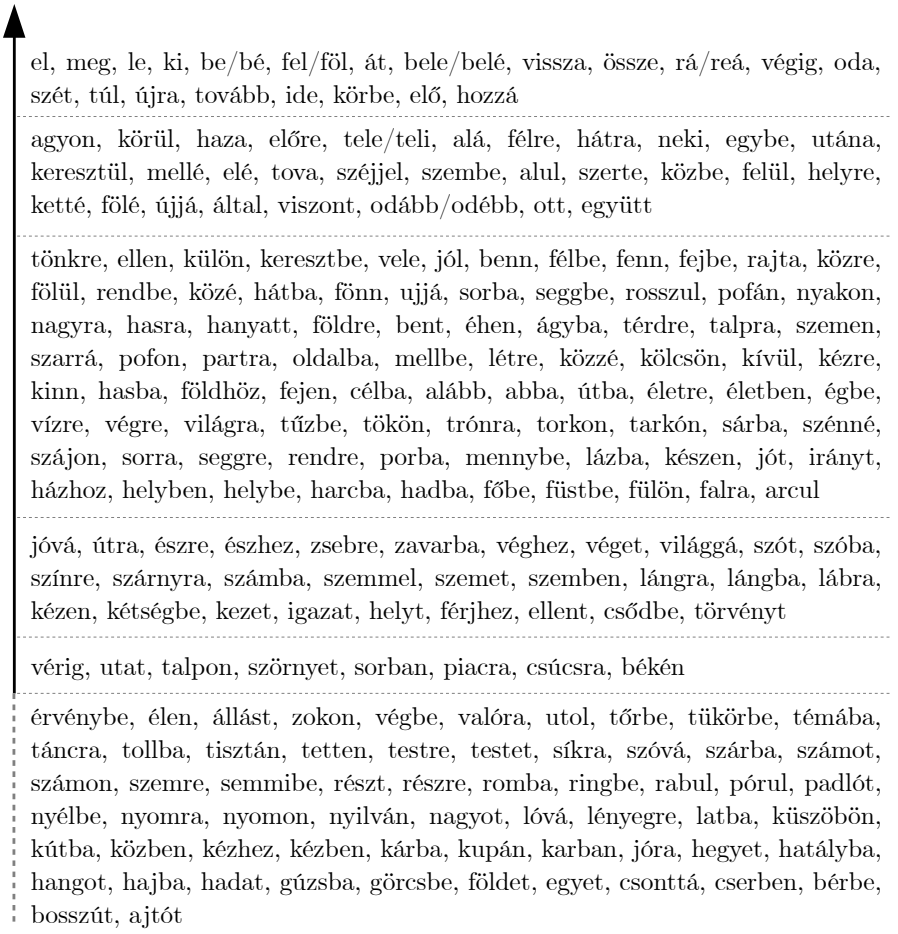
6. táblázat: Lehetséges produktivitás ($P_1$), a 10 legmagasabb értéket kapott szó 5-ös küszöb (bal oldalon) és 5000-es küszöb (jobb oldalon) mellett.

A 6. táblázat bal oldalán szereplő szavakat egyetlen nyelvészeti szakirodalom sem sorolná az igekötők közé. Tény viszont, hogy „igekötőszerűbben" viselkednek sok más igemódosítónál, és emiatt nem ritka, hogy egybe is íródnak az igével. Az előfordulásaik több dokumentumra oszlanak el, tehát nem csak egy ember szóhasználatát látjuk. Többségük egy jól meghatározható szemantikai tartományban mutatja a produktivitás jeleit, például a *mennybe* és az *égbe* mozgásigékkel (*megy, száll*), az *oldalba* támadást kifejező igékkel (*szúr, rúg*) áll. A *szarrá* (*ázik, fagy, bombáz*) és a *szénné* (*ég, vakuz, tetovál*) már absztraktan értendők – a *szét* stilisztikailag jelölt változatai.

A 6. táblázat jobb oldalán rangsorolt szavak státusza kevésbé megosztó: a szakirodalomban is felbukkannak igekötőként, bár az egyetértés ezeket illetően sem általános. Könnyen elképzelhető, hogy idővel lényegesen több új szó alkotásában vesznek majd részt. Az itt látható igekötők mindegyike kapcsolódhat névszóból képzett igéhez (pl. *végigszambáz, telekommentel, agyonpárnáz, körbekordonoz, idekontárkodik*), ami táptalajt nyújt a kreatív szóalkotásnak.

### 3.6. Következtetések

A kapott eredmények amellett szólnak, hogy az igekötők állományát ne intuitív módon megszabott szükséges és elégséges feltételek mentén határozzuk meg. Célravezetőbb lehet az a felfogás, amely szerint az igekötőségnek különböző fokozatai vannak (ld. még [15] és [16]). Az általam vizsgált 239 szó a háromféle produktivitása alapján jól elhelyezhető egy kontinuumban, és ezen belül hat nagyobb csoportra osztható (ld. 4. ábra).




4. ábra: Kontinuum, amely a vizsgált 239 szó háromféle produktivitása alapján rajzolódik ki. A nyíl mentén felfelé haladva a produktivitás mértéke egyre nő. A vízszintes, szaggatott vonalak az egyes csoportok közötti, semmiképp sem éles határokat jelzik.

A 7. táblázat azokat a szempontokat mutatja be, amelyek szerint az igekötőjelölteket csoportosítottam. Ezek sorrendje lényeges: ha egy szó a produktivitásértékei alapján az 1. csoportba tartozás feltételét nem teljesítette, akkor vizsgáltam a 2. csoportba tartozást, és így tovább. A csoportra bontás nem áll ellentétben az igekötő-kategória kontinuum-jellegével. Egyszerűen az eredmények áttekintését és tárgyalását hivatott segíteni.

| **csoport** | **feltétel** |
|---|---|
| **1.** | $P_m > 0{,}01$ és $P_t > 0{,}01$ |
| **2.** | $P_m > 0{,}001$ és $P_t > 0{,}001$ |
| **3.** | $P_m > 0{,}0001$ és $P_t > 0{,}0001$ |
| **4.** | $P_m > 0$ és $P_t > 0$ |
| **5.** | $P_m = 0$ és $P_t = 0$ és $P_l > 0$ |
| **6.** | $P_m = 0$ és $P_t = 0$ és $P_l = 0$ |

7. táblázat: Feltételek, amelyek mentén a hat csoport kialakult. A $P_m$ a megvalósult, a $P_t$ a terjeszkedő, a $P_l$ a lehetséges produktivitás mértékét jelöli.

Az 1. csoportot, egyúttal a kontinuum egyik végpontját a kimagaslóan produktív igekötők alkotják (pl. *be, össze*). A 2. csoport igekötői (pl. *agyon, félre*) közepesen produktívak. A legnépesebb, 84 tagú 3. csoport most még nem túl produktív, de várhatóan azzá váló szavakat foglal magába (pl. *tönkre, szénné*). Ezek között már nagy számban találunk olyan igemódosítókat, amelyeket a nyelvészeti szakirodalmak többsége nem sorolna az igekötők közé.

A 4. csoport tagjai (pl. *jóvá, világgá*) jellemzően csak egy-két igével állnak (pl. *zsebrevág*, ritkábban *-dug, -tesz, -rak*). Az 5. csoportról az mondható el, hogy a tagjai (pl. *csúcsra, békén*) nem produktívak, de minimális esély van arra, hogy idővel produktívabbak lesznek. A kontinuum másik végpontját alkotó 6. csoport 61 olyan igemódosítót tartalmaz, amely kizárólag egy igével áll (pl. *pórul* → *póruljár*, *lóvá* → *lóvátesz*).

## 4. Összefoglalás

A cikkben bemutattam a nyilvánosan elérhető, 54 955 igekötős igét tartalmazó PrevLex táblázatot, amelyet arra használtam, hogy kvantitatív módon meghatározzam az egyes igekötők morfológiai produktivitását. Az így kapott eredmények azt az elképzelést támasztják alá, miszerint az igekötő-kategória kontinuumként értelmezendő.

A PrevLex anyagával bővíthetők a morfológiai elemzők (például az `emMorph` [17]) lexikonjai, ezáltal csökkenthető az `UNKNOWN`-ként elemzett szavak száma. A produktív szóképzési szabályoknak a 3.2. alfejezetben felvázolt, ám ennél szisztematikusabb és teljesebb leírásával a lexikonírás kevesebb humán erőforrást igényel. Mindez jobb lexikont – ezáltal pontosabb nyelvmodelleket – eredményezhet.

## Köszönetnyilvánítás

Köszönöm Olsvay Csabának, Indig Balázsnak és Makrai Mártonnak a cikk többszöri átnézését és a hozzá fűzött értékes megjegyzéseket. Köszönet illeti Prószéky Gábort és mindkét névtelen bírálót a hasznos tanácsokért. Sass Bálintnak köszönöm az MNSZ 2.0.4-hez adott közvetlen hozzáférést.

Jelen kutatás az FK 125217 és a PD 125216 számú projekt keretében az FK17 és a PD17 pályázati program finanszírozásában a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással valósult meg.

## Hivatkozások

1. Ladányi, M.: Produktivitás és analógia a szóképzésben: elvek és esetek. Tinta Könyvkiadó, Budapest (2007)
2. Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In Calzolari, N., et al., eds.: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Izland, European Language Resources Association (ELRA) (2014) 1719–1723
3. Kalivoda, Á., Vadász, N., Indig, B.: Manócska: A Unified Verb Frame Database for Hungarian. In Sojka, P., Horák, A., Kopeček, I., Pala, K., eds.: Proceedings of the 21st International Conference on Text, Speech and Dialogue (TSD), szeptember 11–14, 2018, Brno, Csehország, Springer-Verlag (2018) 135–143
4. Komlósy, A.: Régensek és vonzatok. In Kiefer, F., ed.: Strukturális magyar nyelvtan 1., Mondattan, Budapest, Akadémiai Kiadó (1992) 299–527
5. Prószéky, G., Tihanyi, L., Ugray, G.: Moose: a robust high-performance parser and generator. In Hutchins, J., ed.: Proceedings of the 9th EAMT Conference, La Valletta, Málta, Foundation for International Studies (2004) 138–142
6. Kornai, A., Nemeskey, D.M., Recski, G.: Detecting Optional Arguments of Verbs. In Calzolari, N., et al., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Szlovénia, European Language Resources Association (ELRA) (2016) 2815–2818
7. Kalivoda, Á.: A magyar igei komplexumok vizsgálata. (2016) Mesterszakos szakdolgozat. PPKE-BTK. https://github.com/kagnes/hungarian_verbal_complex.
8. Sass, B., Váradi, T., Pajzs, J., Kiss, M.: Magyar igei szerkezetek – A leggyakoribb vonzatok és szókapcsolatok szótára. Tinta Könyvkiadó, Budapest (2010)
9. Sass, B.: 28 millió szintaktikailag elemzett mondat és 500 000 igei szerkezet. In Tanács, A., Varga, V., Vincze, V., eds.: XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015), Szeged, Szegedi Tudományegyetem Informatikai Intézet, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2015) 399–403
10. Kiefer, F., Ladányi, M.: A szóképzés. In Kiefer, F., ed.: Strukturális magyar nyelvtan 3., Morfológia, Budapest, Akadémiai Kiadó (2000) 137–164
11. Dressler, W.U.: Degrees of grammatical productivity in inflectional morphology. Rivista di Linguistica (Italian Journal of Linguistics) **15**(1) (2003) 31–62
12. Baayen, H.: A Corpus-Based Approach to Morphological Productivity (Statistical Analysis and Psycholinguistic Interpretation). (1989) Doktori értekezés. Centrum voor Wiskunde en Informatica, Amszterdam, Hollandia.
13. Baayen, H.: Corpus linguistics in morphology: morphological productivity. In Lüdeling, A., Kytö, M., eds.: Corpus Linguistics. An international handbook, Berlin, Mouton De Gruyter (2009) 900–919

14. Pakerys, J.: Measuring morphological productivity (2017) Graduate School of Linguistics, Philosophy and Semiotics (GSLPS), Tartu, Észtország, március 20, 2017. Handout. http://web.vu.lt/flf/j.pakerys/wp-content/uploads/pakerys-measuring-morphological-productivity-tartu-2017-handout.pdf.
15. Kerekes, J.: Az igekötők meghatározásának problémái. In Gécseg, Zs., ed.: LingDok 10. Nyelvészdoktoranduszok dolgozatai, Szeged, JATEPress (2011) 109–130
16. Forgács, T.: Grammatikalizálódás az igekötők körében. In Oszkó, B., Sipos, M., eds.: Uráli grammatizáló, Budapest, MTA Nyelvtudományi Intézet (2005) 88–116
17. Novák, A., Siklósi, B., Oravecz, Cs.: A New Integrated Open-source Morphological Analyzer for Hungarian. In Calzolari, N., et al., eds.: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Szlovénia, European Language Resources Association (ELRA) (2016) 1315–1322

# Különböző függőségi elemzők teljesítményének vizsgálata magyar nyelven

Tálas Dalma[1], Novák Attila[1,2]

[1]Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Kar
[2]MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
Budapest, Práter u. 50/a.
talasdalmaalexandra@hotmail.com, novak.attila@itk.ppke.hu

**Kivonat** Cikkünkben összehasonlítjuk néhány különböző elven működő függőségi elemző magyar szövegkorpuszon nyújtott teljesítményét. Emellett bemutatjuk, hogy a szövegkorpusz annotációjának akár teljesen automatizált javításával teljesítménybeli javulás érthető el az annotációban használt címkekészlet felbontásának növelése mellett is.

## 1. Bevezetés

A függőségi elemzés a mondatelemzés egy fajtája, amely során azt vizsgáljuk, hogy a mondatban lévő szavak milyen kapcsolatban állnak egymással. Ezeket a kapcsolatokat irányított élekkel írjuk le úgy, hogy egy szóba csak egy él mutathat, de kifelé bármennyi él mehet. A függőségi elemzésre különböző gépi tanulási algoritmusok léteznek, amelyek alapulhatnak valamilyen neurális hálózaton, vagy használhatnak egyéb lineáris vagy nemlineáris módszereket.

Az elemzéshez szükség van nagyméretű, annotált szövegkorpuszra. A legnagyobb, manuálisan ellenőrzött függőségi elemzést is tartalmazó magyar szövegkorpusz a Szeged Dependency Treebank [1]. Probléma azonban, hogy az ebben alkalmazott annotációs sémában számos egymástól meglehetősen különböző szintaktikai szerkezet annotációja nem különbözik a használt függőségi relációk szintjén (pl. jelzők, birtokosok és mellékmondatok), vagy valamilyen egyéb az annotáció megtervezésénél tett megfontolás nehezíti a szerkezetek értelmezését, illetve olyan manuálisan beszúrt elemeket tartalmaz (például a névszói állítmányok mellett feltételezett zérus létigéket), amelyek az eredeti korpuszban nem szerepelnek, és nem is áll rendelkezésre megfelelő gépi modell az ilyen elemek beszúrására a függőségi elemzés folyamán.

A Universal Dependencies (UD) projekt[1] célja, hogy kiküszöbölje vagy lehetőleg minimalizálja a különböző nyelvek függőségi elemzésére használt annotációs sémák közötti idioszinkratikus eltérésekből adódó azon hatást, hogy nem vagy nagyon nehezen összehasonlíthatóak a különböző – akár nagyon közeli rokoni kapcsolatban álló – nyelvekhez készült függőségi elemzést tartalmazó treebankek,

[1] `http://universaldependencies.org`

és így a különböző nyelvekben előforduló szintaktikai szerkezetek is. Célja, hogy egy olyan függőségi annotációs szabályrendszert alkosson meg, amely minél nagyobb mértékű nyelvészeti konszenzuson alapul, minél könnyebben értelmezhető az emberek számára, és minél helytállóbb módon és egységesen írja le a különböző nyelvek sokszor nagyon eltérő szerkezeteit is. Az ideális tanítókorpuszban egyesülne a minőség és a mennyiség, azaz a Szeged Dependency Treebank mérete és a Universal Dependencies korpusz logikus és átlátható elemzési módszere. Az itt bemutatott munkában nem valósítottuk meg a Szeged Dependency Treebank UD 2.0 formátumra hozását. Célunk pusztán egy olyan annotáció automatikus létrehozása volt, amely – miközben összehasonlítható marad az eredetivel – annál pontosabban azonosít bizonyos szerkezeteket, a gép számára mégis jól tanulható. Ehhez az UD specifikációjából merítettünk ihletet. Ugyan az UD-nek része a Szeged Treebank egy konvertált kis részlete, az ebben alkalmazott annotáció sem felel meg pontosan a kurrens UD 2.0 specifikációnak.

A függőségi elemzés kiértékelését három metrika alapján végeztük. A *Label Accuracy* (LA) csak a címke, az *Unlabeled Attachment Score* (UAS) csak az él, és a *Labeled Attachment Score* (LAS) az él és a címke együttes egyezését vizsgálja.

## 2. Kapcsolódó munkák

Három különböző elemző teljesítményét vizsgáltuk meg. A MateParser [2] egy gráf-alapú, support vector machine módszert használó elemző, amelyet 2010-ben fejlesztettek ki. A SyntaxNet [3] egy átmenet-alapú, neurális hálózatot használó elemző, amelyet a Google fejlesztett ki és tett publikussá 2016-ban. Az Parser v2.0 (eredeti nevén Unstable Parser) [4] egy neurális hálózaton alapuló, gráf-alapú algoritmus, amelyet a Stanford Egyetemen fejlesztettek ki, és amely megnyerte a 2017-es CONLL függőségi elemzési versenyfeladatot az összes, feladatkiírásban szereplő nyelvre, és 2018-ban is egy ugyanezen az elemzőn alapuló rendszer lett a nyertes. Az utóbbi két algoritmus a függőségi elemzőn kívül magában foglal egy szófaji egyértelműsítőt és egy morfoszintaktikai elemzőt is.

A kézi függőségi annotációt tartalmazó Szeged Dependency Treebank [1] 82 ezer mondatból áll, ami 1,2 millió szónak felel meg. Megtalálhatók benne különböző újságcikkek, informatikai és jogi szövegek, 14-16 éves tanulók írásai, üzleti és pénzügyi szövegek és fiktív történetek; tehát sokféle doménből származó szövegek. A treebankben használt függőségi címkék teljes listája megtalálható az [1]-es cikkben.

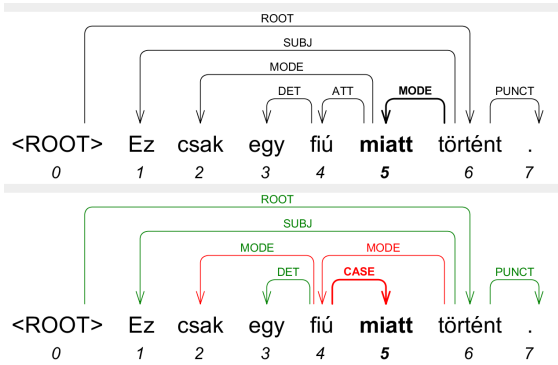
A modellek által készített elemzések kiértékeléséhez, az átalakítások ellenőrzéséhez, és az összehasonlító ábrák készítéséhez a MaltEval nevű programot használtuk [5]. A MaltEval alkalmas mind számszerű kiértékelésre (fedés, pontosság, F-mérték mutatására), mind vizuális megjelenítésre.

## 3. Az annotáció minőségének javítása

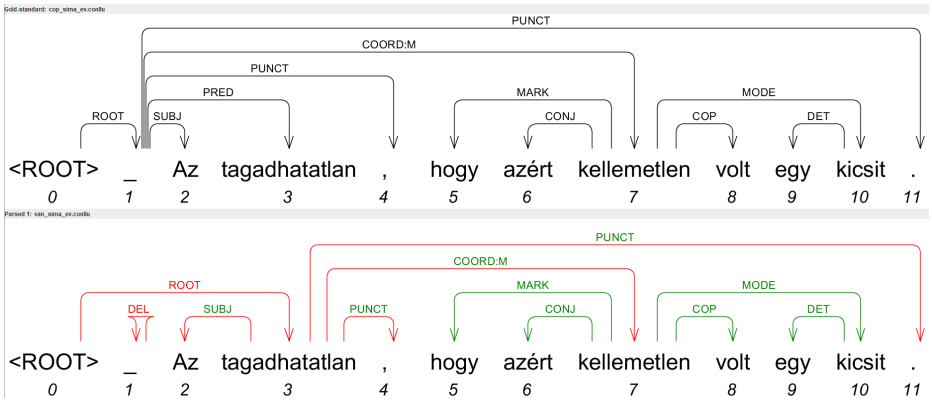
A Szeged Dependency Treebank annotációjában előfordulnak különböző típusú hibák. Egyrészt szerepelnek véletlen hibák, elgépelések, másrészt előfordul, hogy

következetlen egy-egy elemzés, amely nem követi a korpuszban egyébként alkalmazott annotációs sémát, harmadjára pedig vannak szerkezetek, ahol az adott szerkezet elemzési stratégiája nem tűnik a legintuitívebb megoldásnak. Ezen hibák automatikus módszerekkel javítható eseteit próbáltuk meg megtalálni és kijavítani. Emellett az automatikusan szűrhető, de nem javítható hibák esetében megvizsgáltuk, hogy milyen teljesítményt kapunk egy automatikus minőségi szűrés alkalmazásával létrehozott részkorpuszon.

Az annotáció átírása során a Universal Dependencies projektben használt annotációs elvekből merítettünk ihletet, bár az egyszerűség és az eredetivel való könnyebb összehasonlíthatóság érdekében nem tértünk át az ott alkalmazott függőségi címkékre. Hogy az *az a kutya* típusú predeterminánsos szerkezetekben a mutató névmási determinánst is az NP fejéhez kapcsolhassuk, de mégse kapjon azonos `DET` címkét a két determináns, a mutató névmás számára külön címkét definiáltunk ezekben a szerkezetekben. A névutós szerkezetek fejének a névszót tettük meg (a névutó helyett) és külön címkét definiáltunk a névutóra (1. ábra). A többszörös `ROOT` címkéket megszüntettük. Az `ATT` címkét, amit korábban sok egymástól eléggé különböző viszony jelölésére használtak, alcímkékre bontottuk. Bevezettünk minőség-, mennyiség- és birtokos jelzői címkéket, megszüntettük az `ATT` címke névutós szerkezetekben való használatát, és a tagmondatokat összekötő `ATT` címkét helyettesítettük a tagmondatok közötti viszonyt leíró címkékkel (2. ábra). A fráziskoordinációk elemzését átalakítottuk úgy, hogy a frázisok egymással legyenek összekötve, és ne a kötőszón keresztül, továbbá a szerkezet fejének az utolsó frázist tettük meg az első helyett (ebben eltértünk az UD specifikációtól is). A névszói-igei állítmányok esetében a szerkezet fejének a névszót választottuk, amihez kapcsolódik az ige az újonnan bevezetett `COP` címkén keresztül. Az annotációban korábban használt, a testetlen létige jelölését szolgáló tokeneket kivettük és a megmaradó szavak viszonyait megfelelően újrageneráltuk (2. ábra). Ez utóbbi javítás volt talán a legfontosabb ahhoz, hogy a kapott modell nyers szövegre alkalmazva is működőképes maradjon.




1. ábra. Példa a névutós szerkezetek annotációjának átalakítására




2. ábra. Példa a testetlen létige és az alárendelő mondat annotációjának átalakítására (a törlendő token az illusztráció érdekében szerepel `DEL` címkével)

## 4. A tanítás és a tesztelés folyamata

A neurális hálózaton alapuló elemzők tanításához (SyntaxNet és Parser v2.0) három szöveghalmazra volt szükségünk: tanító-, validációs és teszthalmazra, ezért ezekhez a korpuszt 8:1:1 arányban osztottuk fel. A MateParser esetében nem volt szükség validációs halmazra, ezért itt kétfelé osztottuk a korpuszt 9:1 arányban úgy, hogy a teszthalmaz teljes mértékben egyezzen a többi elemző tesztelésére használt halmazzal, a tanítókorpusz pedig magában foglalja a validációs halmazt is. A korpusz felosztásánál figyeltünk rá, hogy mindegyik halmaz reprezentatív legyen, azaz pl. a teszthalmaz ne tartalmazzon doménen kívüli szövegeket.

A függőségi elemzők jelenleg két menetben végzik a beadott nyers szöveg elemzését. Első lépésként morfoszintaktikai annotációt (és esetleg lemmatizálást) végeznek a nyers szövegen, majd ennek eredményét használják a függőségi annotációt végző parser bemeneteként. Jelen kutatásunkban nem vizsgáltuk az elemzők morfoszintaktikai annotációt végző címkéző (tagger) komponensének teljesítményét, kizárólag magára a szintaktikai elemzőre koncentráltunk. Mindegyik elemző a gold standard morfoszintaktikai annotációt használta bemenetként.

## 5. Eredmények

### 5.1. A függőségi elemzők eredményeinek összevetése

Először a módosítás nélküli eredményeket vizsgáltuk meg, azaz a Szeged Treebank eredeti annotációjával tanítottuk az elemzőket. A teljesítményt megvizsgáltuk a morfoszintaktikai jegyek nélkül is, tehát úgy, hogy csak a szót, a szótövet és a szófajcímkét adtuk oda az elemzőnek a függőségi paramétereken kívül, és úgy is, hogy ezek mellé a morfoszintaktikai jegyeket is hozzáfűztük (1. táblázat).

| | MateParser | | | SyntaxNet | | | Parser v2.0 | | |
|---|---|---|---|---|---|---|---|---|---|
| | LA | UAS | LAS | LA | UAS | LAS | LA | UAS | LAS |
| Jegyek nélkül | 0,931 | **0,921** | **0,882** | 0,848 | 0,866 | 0,768 | **0,969** | 0,895 | 0,877 |
| Morfosz. jegyekkel | 0,955 | **0,932** | **0,908** | 0,907 | 0,916 | 0,845 | **0,973** | 0,900 | 0,884 |

1. táblázat. A három függőségi elemző teljesítménye az eredeti annotáción három metrika alapján

Az 1. táblázatban láthatók a három függőségi elemző által elért pontosságok három különböző metrika alapján. Azok a modellek, amelyek a morfoszintaktikai jegyeket nem használták a tanításhoz, minden esetben rosszabb pontosságot értek el, mint a morfoszintaktikai jegyeket használó modellek. Az eredményekből az is megállapítható, hogy összességében véve – azaz LAS metrika alapján – a jegyek nélküli és a jegyeket használó modellek esetén is a MateParser érte el a legjobb teljesítményt. Az él helyét illetően is a MateParser bizonyult a legpontosabb elemzőnek, azonban érdekes, hogy a címkék szerinti metrika alapján a Parser v2.0 érte el a legjobb eredményeket mind jegyek nélkül, mind azokkal. A SyntaxNet teljesítménye elmaradt a másik két elemzőétől. Ennek fő oka az lehet, hogy a SyntaxNet nem gráfalapú elemző, és nem tud mit kezdeni az olyan nem projektív szerkezetekkel, ahol a függőségi élek keresztezik egymást.

| | Régi annotáció | | | Új annotáció | | | Relatív javulás (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | LA | UAS | LAS | LA | UAS | LAS | LA | UAS | LAS |
| MateParser | 0,955 | 0,932 | 0,908 | 0,964 | 0,934 | 0,919 | **20,00** | **2,941** | **11,96** |
| Parser v2.0 | 0,973 | 0,900 | 0,884 | 0,968 | 0,942 | 0,927 | **-18,52** | **42,00** | **37,07** |

2. táblázat. A régi és az új annotáción tanított modellek pontossága és a relatív javulás mértéke három metrika szerint

### 5.2. Az átalakított annotáción kapott eredmények

Az annotáció átalakítása után újabb modelleket tanítottunk be a teljes korpuszon. Az elemzők közül csak a MateParserrel és az Parser v2.0-val dolgoztunk a továbbiakban, mert a SyntaxNet teljesítménye elmaradt a többi elemzőétől.

A MateParserrel tanított modell eredményein látszik, hogy az új annotáció minden esetben javított a pontosságon (2. táblázat). A javulás mértéke a címkék esetén volt a legnagyobb: a címkehibák 20%-át és az élhibák 3%-át sikerült elkerülni az új annotáción tanított modellnek.

A Parser v2.0-val tanított modell esetében az élek szerinti pontosság nőtt nagy mértékben, míg a címkék szerinti pontosság csökkent (2. táblázat). Összességében véve (LAS metrika alapján) azonban a teljesítmény így is sokat javult.

A régi annotáción a Parser v2.0 a címkéket jobban jósolta, mint a MateParser, míg az éleket rosszabbul, és összességében véve kicsit pontatlanabbul jósolt. Az új annotáción azonban minden metrika szerint jobb eredményt ért el, mint a MateParser.

Érdekes, hogy a Parser v2.0-nál az élek jóslásán sikerült sokat javítani – ahogy arra számítani lehetett –, a MateParser-nél viszont pont a címkék szerinti eredmény lett jobb. Ezért érdemes megvizsgálni az elért pontosságokat címkék szerinti bontásban is.

A régi és az új annotáción tanított modellek összehasonlításához megvizsgáltuk a címkék szerinti pontosságokat is, ami itt különösen fontos volt, hiszen az átalakítások során nagymértékben módosítottunk a címkehalmazon (3. táblázat). Az érintetlenül hagyott címkék közül az új annotáció – közvetett módon – jelentősen javított a `DAT`, a `DET` és a `NEG` címkéken. A 3-3 hely- és időhatározót jelölő címkén hol sikerült javítani, hol nem, de a következetlen annotáció miatt (a korpusz legnagyobb részében csak névmások kaptak ilyen annotációt, helyenként azonban a hely- vagy időhatározói névszói csoportok feje is) ezeket a címkéket nem érdemes figyelni az automatikus kiértékelés során. Néhány címke, mint pl. az `INF` és a `ROOT`, pontossága romlott. Ezek közül a mondat fejének megtalálása különösen fontos lenne. A `ROOT` jóslása azért romolhatott, mert a régi annotációban sok helyen a külön beszúrt *VAN* token jelentette a mondat fejét, amit persze sokkal egyszerűbben meg tudott találni az algoritmus, ugyanakkor valós helyzetekben ilyen hozzáadott annotáció nem áll rendelkezésre. Ezeken kívül voltak olyan címkék, amelyeknek pontossága vagy az egyik, vagy a másik elemzőnél javult vagy éppen romlott.

Ami a szétbontott címkéket illeti, az eredmények változóak voltak. A négy eredeti címke (`COORD`, `CONJ` és `ATT`) és a `PRED` pontossága sokat romlott, ami várható volt, hiszen ezeknek teljesen meg kellett volna szűnniük (a `CONJ` kivételével), viszont az automatikus átalakítás nehézségei és az eredeti annotációban szereplő hibák miatt még maradtak ilyen címkék is a korpuszban. Ehhez képest a pontosság még így is viszonylag magas, amiből arra következtettünk, hogy az annotációban maradt címkék olyan szerkezetekben szerepelnek, amelyek vagy amelyeknek egy része valamilyen közös mintát mutat. Az alcímkéket nagyon jól sikerült megjósolnia a modellnek a birtokos, a minőség- és a mennyiségjelzős szerkezetekben. Ezzel szemben a főnévi `ATT` módosítók eredményei rosszabbak lettek, ami lehetett egyrészt azért, mert sok ilyen szerkezet valójában hibás különírást tartalmazott, aminek a szófajcímkéi és esetleg egyéb annotációi hiányosak voltak, másrészt lehetett azért, mert a modell összekeverte a tulajdonneves szerkezetekkel.[2]. A négy alcímke közül három pontossága jobb, mint az eredeti címkéé volt a régi annotáción betanított modell kimenetében. Az átalakítás során az `ATT` címkék egy másik részéből `CASE` lett (a névutós szerkezetekben), amit nagyon jól sikerült jósolnia a modellnek, de ez várható is volt. A frázis- és a

[2] Az UD magyar részkorpusza éppen abban nem felel meg az UD 2.0 specifikációnak, hogy az utóbbi szerint a szabályos szintaktikai szerkezetet tartalmazó névelemeket (például a címeket) a szokásos függőségi címkék használatával kellene annotálni – ez azonban már a Szeged Dependency Treebankben sincs így

| | MateParser | | Parser v2.0 | |
|---|---|---|---|---|
| | Régi | Új | Régi | Új |
| APPEND | 0,874 | 0,861 | **0,894** | 0,878 |
| ATT | 0,956 | 0,782 | **0,974** | 0,792 |
| ATT:A | | **0,990** | | **0,990** |
| ATT:M | | **0,989** | | **0,989** |
| ATT:N | | 0,911 | | **0,921** |
| ATT:POSS | | 0,975 | | **0,976** |
| AUX | 0,989 | **1,000** | **1,000** | 0,996 |
| CASE | | 0,989 | | **0,990** |
| CC | | 0,956 | | **0,957** |
| CONJ | 0,972 | 0,945 | **0,995** | 0,951 |
| COORD | 0,882 | 0,582 | **0,918** | 0,580 |
| COORD:C | | 0,897 | | **0,913** |
| SUBORD | | 0,905 | | **0,920** |
| COORD:P | | 0,894 | | **0,903** |
| COP | | 0,906 | | **0,928** |
| DAT | 0,889 | 0,935 | 0,933 | **0,936** |
| DET | 0,991 | 0,998 | 0,995 | **0,999** |
| FROM | 0,691 | 0,661 | **0,786** | 0,765 |
| INF | 0,989 | 0,978 | **0,993** | 0,982 |
| IS | | **0,996** | | **0,996** |
| LOCY | 0,827 | 0,819 | **0,860** | 0,836 |
| MARK | | **0,979** | | 0,978 |
| MODE | 0,895 | 0,916 | **0,925** | 0,919 |
| NE | 0,928 | 0,993 | **0,995** | 0,994 |
| NEG | 0,979 | 0,992 | **0,995** | **0,995** |
| NUM | 0,990 | 0,989 | **0,991** | 0,990 |
| OBJ | 0,973 | 0,981 | **0,987** | 0,984 |
| OBL | 0,961 | 0,973 | **0,975** | **0,975** |
| PRED | 0,862 | 0,509 | **0,908** | 0,458 |
| PREDET | | 0,965 | | **0,972** |
| PREVERB | 0,973 | 0,993 | **0,994** | 0,993 |
| PUNCT | **1,000** | **1,000** | **1,000** | **1,000** |
| QUE | **0,950** | 0,929 | 0,926 | 0,933 |
| ROOT | 0,967 | 0,949 | **0,982** | 0,962 |
| SUBJ | 0,921 | 0,947 | **0,962** | 0,958 |
| TFROM | **0,878** | 0,857 | 0,835 | 0,825 |
| TLOCY | 0,892 | 0,899 | **0,900** | 0,893 |
| TO | 0,751 | 0,790 | 0,825 | **0,827** |
| TTO | 0,773 | **0,815** | 0,787 | 0,796 |

3. táblázat. A régi és az új annotáción tanított modellek F-mértéke címkék szerinti lebontásban

mondatkoordináció felismerése is javult valamennyit, a kötőszavak megtalálása viszont romlott. Persze ezt nehéz összevetni a régebbi annotáció eredményeivel, mert megváltozott a szerkezetek elemzési logikája, viszont az látszik, hogy a tagmondatok közötti – azon belül is az alárendelő – kötőszavak jóslása könnyebben ment az elemzőnek, mint a fráziskoordináció esetében. A névszó-igei állítmány felismerése nem triviális feladat, a `COP` címke jóslási pontossága mégis viszonylag magas.

### 5.3. A szűrt részkorpuszon kapott eredmények

Az új annotáción kapott címkék szerinti pontosságokon látszik, hogy azok a címkék, amelyek a hibák miatt bennmaradtak a korpuszban, de amelyeket szeretnénk teljesen megszüntetni, nagyon sokat rontottak a pontosságokon. Ebből adódik az ötlet, hogy ki lehetne szűrni a halmazokból az ilyen módon hibásnak érzékelt mondatokat, és be lehetne tanítani egy olyan modellt, amely csak a megmaradt, látszólag jobb minőségű annotációval ellátott mondatokat használja. A kérdés az, hogy ha az egy-egy szónál előforduló rossz annotáció miatt kivesszük a teljes mondatot a tanítóhalmazból, akkor azzal inkább ártunk a modellnek, azaz lehet, hogy volt egy hiba az egyik szónál, de a mondat többi része még így is túl sok értékes információt hordozott; vagy inkább segítünk a modellnek, azaz a hibás mondatok annyira zavaróak a tanítás számára, hogy inkább zajt jelentenek, így a kihagyásukkal többet nyernénk, mint amennyi információ elvész.

A korpuszból kiszűrtük azokat a mondatokat, amelyek hiányos annotációval rendelkeztek a szófajcímkét illetően (*posTag = X, Y, Z*), vagy amelyekben az átalakítás után is szerepelt teljesen megszüntetendő függőségi címke (`ATT`, `COORD` vagy `PRED`). Az így kapott korpusz mérete (mondatszám alapján) az eredeti 83,7%-a lett. A szűrt korpuszon kapott eredmények jobbak, mint a teljes korpuszon tanított modell esetében (4. táblázat). A MateParserrel 15,7%-os, a Parser v2.0-val 13,7%-os relatív javulást értünk el. A kapott pontosságok magasabbak, mint az adott elemzővel tanított bármelyik másik modellé. Ezen belül az Parser v2.0 teljesítménye volt a legjobb, LAS metrika esetén 0,937.

| | Teljes korpuszon tanítva | | Szűrt korpuszon tanítva | | Javulás mértéke | |
|---|---|---|---|---|---|---|
| | Mate | P2.0 | Mate | P2.0 | Mate | P2.0 |
| LA | 0,964 | 0,968 | 0,970 | 0,973 | 16,67% | 18,62% |
| UAS | 0,934 | 0,942 | 0,944 | 0,950 | 15,15% | 13,80% |
| LAS | 0,919 | 0,927 | 0,930 | 0,937 | 15,71% | 13,70% |

4. táblázat. A szűrt halmazon tanított modell pontossága és relatív javulása a teljes korpuszon tanított modellhez képest

## 6. Konklúzió

Vizsgálatunk során a MateParser és a Parser v2.0 kiemelkedően jó eredményeket ért el. Az eredeti korpuszon az előbbi az élek jóslása terén volt jobb, míg az utóbbi a címkék eltalálásában. A címkék szerinti pontosságokból pedig az látható, hogy teljesítmény másképpen oszlik el a két elemző esetében. A SyntaxNet eredményei minden vizsgált esetben elmaradtak a másik két elemzőétől.

A vizsgálat során azt is megállapítottuk, hogy az automatikusan javított új annotáción tanított modellek jobban teljesítenek, mint a korábbiak. Elemzőtől függően más-más helyeken tapasztaltunk javulást, pl. a MateParsernél elsősorban a címkéken sikerült javítani, míg a Parser v2.0-nál az éleken. Összességében véve az új annotáción tanított modellek a MateParser esetében 10%-os, a Parser v2.0 esetében 37%-os relatív javulást értek el a régi annotáción tanított modellekhez képest. Az újonnan bevezetett címkék jóslása általában jobban ment a modelleknek, mint a megfelelő régi címkéké annak ellenére, hogy a lehetőségek száma nőtt. Ez igazolja azt az várakozást, hogy intuitívebb és konzisztensebb annotáció alapján a gép is könnyebben tanul meg elemezni. A korábban egybemosott kategóriák szétválasztása nem ártott a rendszernek, inkább javult a pontosság.

Hasonló eredmény olvasható ki Simkó et al. cikkéből is [6], bár abban a kutatásban épp ellenkező irányban módosították a címkekészletet, mint mi az itt leírt kísérleteinkben: a magyar UD korpusz címkekészletének elemeit összevonva általában romlottak a címkézési pontosságok (LAS, UAS) amellett, hogy a címkék összevonása még információveszteséggel is járt.

A minőségi szempontból szűrt halmazon tanított modellek teljesítménye jobb, mint bármelyik másik halmazon vagy annotációval tanított modellé (itt már csak a pontosabban működő MateParsert és Parser v2.0-t vizsgáltuk). A címkék szerinti pontosság 97 és 97,3%-os, az élek szerinti 94,4 és 95%-os, míg az élek és címkék együttes helyességét tekintve a teljesítmény 93, illetve 93,7%-os, mindhárom metrika szerint a Parser v2.0 javára. Tehát összességében véve ez a két modell érte el a legjobb teljesítményt, illetve azon belül a Parser v2.0-val tanított elemző volt a legpontosabb.

## Köszönetnyilvánítás

Jelen kutatás az FK 17 pályázati program finanszírozásában az FK 125217 számú projekt keretében a Nemzeti Kutatási Fejlesztési és Innovációs Alapból biztosított támogatással valósult meg.

## Hivatkozások

1. Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., Csirik, J.: Hungarian dependency treebank. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (2010)

2. Björkelund, A., Bohnet, B., Hafdell, L., Nugues, P.: A high-performance syntactic and semantic dependency parser. In: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations. COLING '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 33–36
3. Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. CoRR **abs/1603.06042** (2016)
4. Dozat, T., Qi, P., Manning, C.D.: Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Vancouver, Canada, August 3-4, 2017. (2017) 20–30
5. Nilsson, J., Nivre, J.: Malteval: an evaluation and visualization tool for dependency parsing. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, B.M.J.M.J.O.S.P.D.T., ed.: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, European Language Resources Association (ELRA) (2008) http://www.lrec-conf.org/proceedings/lrec2008/.
6. Simkó, K.I., Kovács, V., Vincze, V.: Szintaktikai címkekészletek hatása az elemzés eredményességére. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 20117), Szeged, SZTE (2017) 316–322

# Szerzői index, névmutató

**A digitális világ GPS-e a Neumann János Számítógép-tudományi Társaság**

**Második fél évszázadát kezdi az NJSZT**

A **Neumann János Számítógép-tudományi Társaságot** (NJSZT) 1968-ban hozták létre. Első elnöke a magyar kibernetika egyik úttörője, **Tarján Rezső** volt. Az alapításkor mindössze kilenc éve létezett számítógép Magyarországon. Az első évtizedben az NJSZT komoly, országos hatókörű tudományos szervezetté vált és bekerült az informatikai társaságok nemzetközi vérkeringésébe is. A 80-as években, elsősorban **Kovács Győző** főtitkárnak köszönhetően, a Társaság nyitott a tömegek, a fiatalok felé. Az NJSZT szorgalmazta a személyi számítógépek elterjesztését, a számítógépes klubmozgalmat, a távoktatást. Mind hangsúlyosabban jelent meg a tehetséggondozás: **azóta mintegy 300.000 honfitársunk vett részt az NJSZT tehetséggondozási rendszerében**, mérette meg magát versenyein. 1997-ben **Alföldi István, a Társaság ügyvezető igazgatója** a következőképpen fogalmazta meg az NJSZT küldetését: **Megőrizni a múlt értékeit, alkalmazkodni a jelenhez, befolyásolni a jövőt.**

**A múlt értékeinek megőrzése**: a civil szervezetek között páratlan erőfeszítéssel az NJSZT felvállalta, hogy létrehoz és fejleszt egy **informatikatörténeti állandó kiállítás**t. A **Jövő múltja** című tárlatot Neumann János lánya, **Marina von Neumann Whitman** nyitotta meg 2013-ban. Azóta folyamatosan megújulva, időszaki kiállításokkal kiegészülve várja a látogatókat. A tárlat anyaga egy, az NJSZT részvételével alapított közérdekű muzeális gyűjtemény, az **Informatika Történeti Múzeum Alapítvány** állományán alapul. A tárlat elérhetősége: **ajovomultja.hu** Az NJSZT Informatikatörténeti Fórum szakmai közössége páratlan gazdagságú informatikatörténeti adattárat gondoz, a digitális gyűjtemény elérhetősége: **itf2.njszt.hu**

**Alkalmazkodás a jelenhez:** az NJSZT a **digitális esélyegyenlőség** kérdéseinek elkötelezettje és a **digitális írástudás** vezető szervezete hazánkban. Az **ECDL** (Európai Számítógép-használói Jogosítvány) keretében eddig **több mint félmillió honfitársunk bizonyította digitális kompetenciáit az NJSZT-nek köszönhetően**. Az ECDL megújuló rendszerében az egész világon úttörő módon vezette be a Társaság az **IT biztonság** és a nem informatika szakos tanárokat megcélzó **IKT pedagógusoknak** modulokat, melyekhez ingyenes tananyagot tett hozzáférhetővé. **2018-ban mindkét modul Best Practice-díjat kapott a 150 országot számláló nemzetközi ECDL-közösségben.** A Digitális Esélyegyenlőség kérdésköreiről (**DE!**) immár tizenkét teltházas konferenciát tartott a Gellért Szállóban az NJSZT.

**Befolyásolni a jövőt:** A Társaság szakmai közösségei és területi szervezetei behálózzák az országot. **Az NJSZT mindent elkövet azért, hogy a robotika és mesterséges intelligencia kérdéseire, előnyeire és „kockázataira, mellékhatásaira" felhívja a társadalom figyelmét**, új projektje **az EDLRIS**: osztrák-magyar nemzetközi robotika és MI tananyagfejlesztés, Horizon 2020 pályázat keretében. Mindezek mellett az NJSZT egyik legfontosabb küldetése továbbra is a tehetséggondozás – melynek sikerét a nemzetközi diákolimpiákon való, eredményes magyar szereplés is bizonyítja.

Annak érdekében, hogy vállalt küldetését még hatékonyabban tudja képviselni, **a Társaság a jubileumi évében, 2018-ban megújult:** komoly erők sorakoztak fel a megújult vezetésben - köztük az új, digitalizált világ és az ipar képviselői. **A Dr. Beck György elnök és Alföldi István ügyvezető igazgató által vezetett Társaság a digitális világ GPS-eként segíti a társadalmat az eligazodásban: mert a digitalizáció megkerülhetetlen.**

njszt.hu ajovomultja.hu

Clementine

## RÓLUNK

Innovatív szemlélet, széleskörű iparági szakismeret, releváns üzleti tapasztalat, kipróbált és bevált analitikai technológiák. Csak néhány indok, ami miatt ügyfeleink minket választottak az elmúlt 15 évben, amikor olyan üzleti kihívásokkal néztek szembe, amelyek feloldásához az adat- és szöveganalitikai támogatás nélkülözhetetlennek bizonyult. Mi a Clementine-nál mindig is arra törekedtünk, hogy partnereinkkel közösen olyan megoldásokat fejlesszünk, amelyek képesek a legkomplexebb üzleti és folyamatoldali problémákra is valódi választ adni, érkezzenek azok bámely specifikus területről.

otpbank · vodafone · UNIQA · K&H · Audi Hungaria · e·on · ERSTE · AEGON · upc · NN · Garantiqa Hitelgarancia Zrt. · Rendvédelmi szervezetek

## TEVÉKENYSÉGEINK

Tevékenységi körünk három fő területre bontható: IBM SPSS termékek forgalmazása, saját fejlesztésű analitikai és mesterséges intelligencia megoldások fejlesztése és az ehhez szorosan kapcsolódó tanácsadói tevékenység, illetve statisztikai és data science témák oktatása.

### SZOFTVER

Az IBM SPSS statisztikai szoftvercsalád és piacvezető adatbányászati szoftver, valamint az i2 hálózatvizualizációs platform magyarországi forgalmazóiként teljeskörű támogatást nyújtunk ügyfeleinknek az igényeiknek legjobban megfelelő szoftver megtalálástól a terméktámogatáson keresztül egészen a support tevékenységekig.

### TANÁCSADÁS

Adat- és szöveganalitikai projektjeink során a teljes projektfolyamatban támogatjuk megbízóinkat, az üzleti probléma pontos megfogalmazásától kezdve a megfelelő analitikai eszközök kiválasztásában és testreszabásában, a hatékony stratégia kialakításában vagy a szükséges elemzések elkészítésében, üzemeltetésében.

### FEJLESZTÉS

Valós üzleti folyamatokra, illetve fejlett adat- és szöveganalitikai algoritmusokra alapozott saját fejlesztésű termékeink vállalatra szabott megoldásként segítik ügyfeleink napi munkáját a legkülönfélébb területeken, mint ügyfélszolgálat, panaszkezelés, kockázatkezelés, marketing, csalásfelderítés és bűnmegelőzés.

### KUTATÁS

Új és innovatív ötleteinkkel igyekszünk megelőzni a piaci igényeket, hogy mire azok valódi üzleti kihívásként felmerülnek, nekünk már kész megoldási javaslataink legyenek azok kezelésére. Emellett sok éve megbízhatóan működő megoldásaink folyamatos korszerűsítésére is nagy hangsúlyt fektetünk.

### OKTATÁS

A képzési rendszerünk nagyobb részét a statisztikai és adatbányászati tanfolyamaink teszik ki, melyek során magabiztos gyakorlati tudás szerezhető a szoftverek használatában. Ezek mellett téma- és üzletág specifikus workshopok, illetve bemutató szemináriumok is az érdeklődők rendelkezésére állnak.

### RENDEZVÉNY

Legnagyobb konferenciáink, a dataSTREAM és a conTEXT, az analitika és az innovatív üzleti megoldások iránt érdeklődőknek szól, ugyanakkor nagy hangsúlyt fektetünk a felsőoktatással történő együttműködésre is – számukra rendezzük az egyhetes Nyári Iskola programunkat, valamint évi 6 alkalommal megtartott meet-up eseményeinket.

## KEDVENC TÉMÁINK

#nlp #szöveganalitika #adatbányászat #automatizálás #ügyfélszolgálati hatékonyság növelés #machine learning #virtuális asszisztensek #ügyfélszolgálati robotok #mesterséges intelligencia

**Clementine** | 1115 Budapest, Bartók Béla út 105-113. I/B | info@clementine.hu | www.clementine.hu

## Szegedi Tudományegyetem
## Informatikai Intézet

Intézetünk 1990-ben alakult Informatikai Tanszékcsoport néven a Szegedi Tudományegyetem, Természettudományi és Informatikai Karának részeként. Alapfeladatunk a modern informatikai és számítástudományi ismeretek oktatása, valamint a tudományos életben és innovatív fejlesztésekben való aktív részvétel. Hallgatóink számára számos lehetőség nyílik az utazásra, munkára, és kutatásra is. Lehetőség szerint támogatjuk a hallgatók külföldi részképzését. Ennek érdekében külföldi partnerekkel tartunk fent kapcsolatot közös projektek útján, aminek eredményeképpen számos hallgatónk élhetett külföldi részképzésben. Természetesen a hallgatóink bekapcsolódhatnak a tanszékeken folyó kutatásokba is. Ennek eredményeként hallgatóink a mai napig is számos színvonalas kutatási eredményt mutattak be a rendszeresen rendezett Helyi és Országos Tudományos Diákköri Konferenciákon. Munkánkban az oktatás mellett számos intézményi, és ipari partnerrel állunk kapcsolatban kutatás-fejlesztési és innovációs projektek megvalósításában is.

### Szoftverfejlesztés Tanszék

A Szoftverfejlesztés tanszék fő oktatási és kutatási területei a szoftverfejlesztési folyamattal kapcsolatos témákhoz kapcsolódnak. A tanszék által oktatott tárgyak keretében a hallgatók megismerkednek a különböző fejlesztési módszerekkel, programozási nyelvekkel és környezetekkel, illetve a fejlesztés során alkalmazott módszerek és eszközök elméleti és technológiai hátterével. A tanszék kutatói aktívan közreműködnek a szoftverek minőségével, az M2M és beágyazott rendszerekkel, a webtechnológiákkal, fordítóprogramokkal és végrehajtómotorokkal, az információbiztonsággal, a mesterséges intelligencia alkalmazásaival, a telemedicina különböző területeivel, illetve a nyílt forráskódú szoftverek fejlesztésével kapcsolatos nemzetközi kutatásokban. Rendszeresen publikálnak rangos nemzetközi folyóiratokban, és vesznek részt a témákkal foglalkozó konferenciákon. A tanszéken folyó kutatási munka elismertségét jelzi az is, hogy a tanszék szervezésében Szeged adhatott otthont a nemzetközi viszonylatban is nagyon rangos ESEC/FSE 2011 és CSMR 2012 konferenciáknak. A tanszék számos kutatás-fejlesztési projektben vesz részt hazai és nemzetközi ipari szereplőkkel.

### Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék

A Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék jelen formájában és profiljával a SZTE Informatikai Intézetének szerkezeti átalakítása során jött létre 2003-ban. A tanszék oktatói az algoritmusok és a mesterséges intelligencia területéhez kapcsolódó tárgyakat oktatják, kutatási tevékenységet az algoritmusok elméletének és a mesterséges intelligenciának több területén folytatnak. A tanszék dolgozói szervezték meg 2003-ban az algoritmusok elméletének egyik legjelentősebb európai konferenciáját, az ALGO 2003 szimpóziumot. A tanszék munkatársainak kutatási területei felölelik a fuzzy rendszereket, a többtényezős döntéseket, ütemezési és ládapakolási feladatokat, a természetesnyelv-feldolgozást, az önszervező rendszereket, a beszédfelismerést, valamint a mély tanulást és annak különböző alkalmazásait.

### MTA-SzTE Mesterséges Intelligencia Kutatócsoport

A Mesterséges Intelligencia Kutatócsoport a Magyar Tudományos Akadémia és a Szegedi Tudományegyetem közös kutatócsoportja. A kutatócsoport vezetője Gyimóthy Tibor professzor. A csoport az 1969-ben Kalmár László által alapított Automataelméleti Kutatócsoportból alakult át Csirik János professzor vezetésével 1996-ban. A kutatócsoport jelenlegi kutatásai hét alkalmazási terület: a nyelvfeldolgozás, a beszédtechnológia, a szoftverfejlesztés, az információbiztonság, az önszervező rendszerek és a gépi tanulás elmélete köré csoportosulnak. Ezek az interdiszciplináris területek gyakran alkalmaznak gépi tanulási technikákat.