Értsük meg a magyar entitásfelismerő rendszerek viselkedését!

Farkas Richárd[1], Nemeskey Dávid Márk[2], Zahorszki Róbert[1], Vincze Veronika[3]

[1]Szegedi Tudományegyetem, Informatika Intézet

[2] Eötvös Lóránd Tudományegyetem, Digitális Bölcsészet Központ

[3] MTA-SzTE Mesterséges Intelligencia kutatócsoport

rfarkas@inf.u-szeged.hu

Kivonat: A nyelvtechnológiai megoldásokat hagyományosan egy valós életből származó szöveghalmaz tanító és tesztadatbázisra bontott verzióján szokás kiértékelni, e módszer azonban több buktatóval is rendelkezik. A CheckList egy újfajta kiértékelési módszertan, mely különböző nyelvi jelenségeket definiál, továbbá az egyes jelenségekre külön tesztkörnyezeteket állít fel, melyek az adott alkalmazás viselkedését hivatottak tesztelni. Ebben a tanulmányban a magyar nyelvű névelem-felismerési (NER) feladatra alkalmazzuk a CheckList módszertanát. Ehhez 9 nyelvi jelenséget1 definiálunk, mondatsablonokon keresztül 27 tesztkörnyezetet állítunk fel és három magyar névelem-felismerő rendszert értékelünk ki a CheckList módszertanában. Elemzésünk megmutatja, hogy ez a módszertan közelebb visz minket ahhoz, hogy megértsük a magyar entitásfelismerők viselkedésének megértését.

1 Bevezetés

Az elmúlt évtizedekben a nyelvtechnológiai megoldásokat szinte minden esetben egy valós életből származó szöveghalmaz, kézzel jelölt, tanító és kiértékelő adatbázisra vágott verzióján értékelték ki. A tanító adatbázison gépi tanult rendszerek (vagy az alapján kézzel épített szabályrendszereket) pontosságát a kiértékelő adatbázison mérjük meg és ezt egyetlen számmal (pl. accuracy, F1-érték vagy BLEU score) írjuk le. Az elérhető adatbázisokon mindig verseny indul, és aki a kiértékelési metrikában akár csak fél százalékponttal jobb eredményt ér el, mint a korábban publikált legjobb eredmény, az már publikálható eredménynek számít. Ezt a tudománytörténeti jelenséget *leaderboard paradigmá*nak nevezi Ethayarajh és Jurafsky (2020). A leaderboard paradigma számos problémát vet fel, amelyek orvoslására az elmúlt két-három évben több javaslat is megjelent a *ACL és EMNLP konferenciákon.

1 Az eredeti CheckList módszertanban használt “capability” fogalmát ‘nyelvi jelenségként’ fordítjuk, elkerülve a 'nyelvi képesség' fogalmának túlterhelését, mivel utóbbit a magyar nyelvészeti szakirodalom főként a nyelvelsajátítás és idegennyelv-tanulás területein alkalmazza (a 'linguistic ability', illetve a 'language skill' megfelelőjeként).

A kiértékelő adatbázisokon való kiértékelés természetesen hasznos, de mivel az adatbázis eloszlását követi, ezért számos torzítást tartalmazhat - például mert egy szűk téma, zsáner vagy stílus dominálja -, sokszor az adott modell túláltalánosít a tanító adatbázis alapján és egy másik kiértékelő adatbázison már kevésbé jó eredményt ad. Továbbá, ha egyetlen számmal írjuk le a rendszer teljesítményét, akkor abból nem tudjuk meg, hogy hol és miért hibázik a rendszerünk, azaz nem értjük meg, hogy hogyan viselkedik a vizsgált rendszer. Ez pedig elengedhetetlen ahhoz, hogy egy rendszer, egy konkrét valós életbeli feladatra való alkalmazhatóságáról dönteni tudjon az alkalmazásfejlesztő.

Az egyes feladatok megoldásához számos nyelvi jelenség kezelésére szükség lehet, és ezek gyakorisága és fontossága igencsak eltérő lehet egymástól. Megeshet, hogy az adott módszer a legfontosabb, legalapvetőbb példákon jól teljesít, azonban a nehezebb, bonyolultabb példákon elbukik, vagy esetleg ennek a fordítottja: néhány alapvető példát elront, de a nehezebbeken jól teljesít (például mert ezek túl vannak reprezentálva a tanító adatbázisban), a számszerű eredményekben azonban e különbségek nem mutatkoznak meg.

A fenti problémák kiküszöbölésére Ribeiro és mtsai (2020)[2] bevezették a „CheckList" tesztelés fogalmát, melyet részben a szoftverfejlesztésben használatos tesztelési módszertan inspirált. A CheckList egy újfajta kiértékelési módszertan, mely különböző nyelvi jelenségeket definiál, amelyeket a rendszernek az adott feladat (és nem adatbázis!) megoldásához bizonyítani kell. Az egyes jelenségekre külön tesztkörnyezeteket állít fel, melyek az adott alkalmazás viselkedését hivatottak tesztelni. Ez a fajta diagnosztikus tesztelés jól kiegészíti a kiértékelő adatbázison számolt metrikákkal kapott minőségellenőrzést.

Ribeiro és mtsai (2020) az angol nyelv vonatkozásában mutatják be módszerüket a szentimentelemzés, duplikált kérdések azonosítása és a gépi szövegértés területére alkalmazva. Ebben a tanulmányban a magyar nyelvű névelem-felismerési (NER) feladatra alkalmazzuk a CheckList módszertanát. Ehhez 9 nyelvi jelenséget[3] definiálunk, mondatsablonokon keresztül 27 tesztkörnyezetet állítunk fel és három magyar névelem-felismerő rendszert értékelünk ki a CheckList módszertanában. Elemzésünk megmutatja, hogy ez a módszertan közelebb visz minket ahhoz, hogy megértsük a magyar entitásfelismerők viselkedésének megértését.

2 Kapcsolódó munkák

Az elmúlt néhány évben számos munka kérdőjelezi meg a nyelvtechnológiai kutatások leaderboard paradigmáját (Ethayarajh és Jurafsky, 2020). Ethayarajh és Jurafsky (2020) a végfelhasználói alkalmazások fejlesztőinek (NLP practitioners) szempontjából tárgyalja, hogy a pontosság metrikák mellett milyen szempontok fontosak egy feladatra adott megoldás szempontjából. Például javasolja a futásidők és energiafelhasználás (Green AI) feltüntetését minden publikációban, hiszen a valós életben, ha két modell közül az egyik néhány százalékponttal pontosabb, de nagyságrendekkel erőforrásigényesebb,

[2] ACL 2020 best paper

[3] Az eredeti CheckList módszertanban használt "capability" fogalmát fordítjuk 'nyelvi jelenségként'.

mint egy másik modell, akkor az alkalmazók a valamivel pontatlanabbat fogják preferálni. Egy másik fő kritika a tanító- és kiértékelő adatbázisokon való mérésekkel szemben a *robusztusság* megismerésének hiánya, ugyanis egyetlen adatbázison kiértékelve, nem tudjuk, hogy a rendszerek mennyire jól viselkednek a tanító adatbázis eloszlásától eltérő példákon, mennyire tűrik a bemenet változásait, illetve mennyire torzítanak egyes demográfiai tulajdonságok irányába (ML fairness).

Ebben a munkában, a robusztusság témakörébe tartozó CheckList (Ribeiro és mtsai, 2020) kiértékelési módszertant használjuk. A CheckList az úgynevezett *black box diagnosztikus tesztek* közé sorolható, hiszen célja annak felmérése, hogy hol és miért hibázik a tesztelt rendszer, valamint feltesszük, hogy a rendszer belső működéséhez nem férünk hozzá, az fekete dobozként - egy bemenetre visszaad egy eredményt - áll rendelkezésre (Paroubek és mtsai, 2007).

A CheckList egy általánosított keretrendszert ad nyelvtechnológiai alkalmazások különböző viselkedési tesztjének definiálására. Például invarianca típusú tesztekkel tudjuk a zajjal - például elírásokkal - szembeni robusztussági teszteket definiálni, vagy más típusú tesztekkel tudjuk a rendszer logikai konzisztenciáját tesztelni. A CheckList kifejezetten végfelhasználói nyelvtechnológiai alkalmazások kiértékelést célozza meg, és olyan eszközt ad, amit a nyelvtechnológiában járatlan, de az adott célalkalmazás szakértő felhasználói is tudnak használni. Ez fontos különbség egyéb javaslatokkal szemben. Például a köztes modulok robusztusságának kiértékelésére használt extrinzikus tesztelés - amikor a modult különböző ráépülő alkalmazásoknak nyújtott hasznosság szerint értékeljük - nem alkalmas célalkalmazások tesztelésére.

A CheckList célkitűzése, hogy megértsük a black box rendszer viselkedését. Ebben az aspektusban a megmagyarázható MI (eXplainable AI) tárgykörébe is besorolható. Ezen algoritmusok közül is kiemelkedik azonban egyszerűségének és univerzalitásának köszönhetően. Például minden feladathoz más és más interpretációs algoritmusokra van szükség (Arrieta és mtsai, 2020), míg a CheckList keretrendszerben bármilyen feladatot kiértékelhetünk. Hasonlóan a neurális modellek megértését célzó ún. próbák módszere (probes) is minden nyelvi jelenség tesztelésére külön algoritmust követel meg (Hewitt és Manning, 2019), míg a CheckListtel bárki tesztelhet bármilyen nyelvi jelenséget.

3 Magyar NER checklist

Figyelembe véve a magyar nyelv tulajdonságait és a névelem-felismerésben fontos nyelvi jegyeket, összeállítottunk egy olyan nyelvi tesztsorozatot, mely segítségével célirányosan tudjuk tesztelni a NER-rendszerek teljesítményét, továbbá meg tudjuk állapítani, mik az egyes rendszerek erősségei és gyenge pontjai. Alább bemutatjuk e jellemzőket, valamint az egyes teszttípusokat.

3.1 Teszttípusok

Minimális működés tesztje (Minimum Functional Test, MFT): Azon alapvető példák tartoznak ide, melyeknek helyes kezelését elvárjuk egy tulajdonnév-felismerő

rendszertől. Például az *X. Y., Magyarország köztársasági elnöke* példában bármi/bárki is álljon *X.Y.* helyén, az személynév (PERSON) címkét kell hogy kapjon.

Invariancia (INV): Ha megváltoztatjuk bizonyos módon a bemeneti mondatot, az nem okozhat változást a rendszer predikciójában. Például egy szórendi csere általában nem befolyásolja a címkézést (*London*$_{LOC}$ *mellett ülésezett a NOB*$_{ORG}$ vs. *A NOB*$_{ORG}$ *London*$_{LOC}$ *mellett ülésezett*).

Elvárt változás (DIR): Ribeiro és mtsai (2020) definálnak egy harmadik típusú tesztet is, ahol a bemenet változtatásával a predikció irányának megváltozását tesztelik. Az eredeti definíció alapján, az INV és DIR teszteket el lehet végezni jelöletlen szövegeken is - míg az MFT-hez annotált példák szükségesek -, hiszen ezeknél csak azt vizsgáljuk, hogy megváltozik a predikció, és azt nem teszteljük, hogy az eredeti szövegen helyes volt-e ez a predikció. Ribeiro és mtsai (2020) erre egy szentiment elemzési példát hoznak, ahol egy negatív töltetű mondattal kiegészítve a szöveget, elvárjuk, hogy a pozitív osztály valószínűsége ne növekedjen. Mivel a tesztelt magyar NER rendszereink alapesetben nem adják meg az egyes címkék valószínűségét, ezért igazi DIR típusú tesztet nem használunk jelen munkában. Megjegyezzük azonban, hogy magyar névelem-felismeréshez is lehet olyan változásokat eszközölni kontrollált - azaz kézzel címkézett esetekben - ahol a bemenet változtatásától egy entitás osztályának megváltozását várjuk el. Például ha egy helynév névelőt kap, akkor bizonyos kontextusokban szervezetnév lesz belőle:

Manchesterben$_{LOC}$ *játszott Ronaldo*$_{PER}$ vs. *A Manchesterben*$_{ORG}$ *játszott Ronaldo*$_{PER}$

Mivel ez az elvárt változás nem felel meg pontosan az eredeti DIR definíciónak, ezért az ilyen jellegű teszteket MFT-ként fogalmazzuk meg, a fenti példából két darab MFT teszt lesz:

Egy helynév: *Manchesterben*$_{LOC}$ *játszott Ronaldo*$_{PER}$
Névelős helynév: *A Manchesterben*$_{ORG}$ *játszott Ronaldo*$_{PER}$

3.2 Nyelvi jelenségek

A magyar nyelv morfológiailag gazdag volta miatt több morfológiai, illetve szintaxis alapú nyelvi jelenségre is építhetünk a névelem-felismerés hatékonyságának tesztelése terén. Ezek mellett néhány szemantikai jellegű változásra épülő jelenséget is bemutatunk.

Szókincs: Az adott tulajdonnévre jellemző legtipikusabb szókészletet reprezentáló példamondatok tartoznak ide, például: *A szomszédomat Fekete*$_{B\text{-}PER}$ *Péternek*$_{I\text{-}PER}$ *hívják*.

Névelő: Ha névelőt kap az adott tulajdonnév, akkor adott irányú változást mutat (vagy nem mutat változást) a címkézésben, például: *Fordnál*$_{PER}$ *járt a szépségkirálynő* vs. *A Fordnál*$_{ORG}$ *járt a szépségkirálynő*.

Toldalékolás: Eltérő toldalékolás (pl. esetrag) esetén adott irányú változást mutat (vagy nem mutat változást) az adott tulajdonnév, például: *A Gyulában*$_{ORG}$ *focizott Feri* vs. *A Gyulával*$_{PER}$ *focizott Feri*.

Névutó: A névutó cseréje esetén adott irányú változást mutat (vagy nem mutat változást) az adott tulajdonnév, például: *London$_{LOC}$ mellett ülésezett a NOB$_{ORG}$* vs. *London$_{MISC}$ után ülésezett a NOB$_{ORG}$*.

Többes szám: Ha az adott tulajdonnevet többes számba tesszük, adott irányú változást mutat (vagy nem mutat változást) a címkéje, például: *Az autóversenyt Ford$_{PER}$ nyerte* vs. *Az autóversenyt Fordok$_{MISC}$ nyerték*.

Predikátum cseréje/szemantikai szerepek változása: Más predikátum esetén adott irányba változik (vagy változatlan marad) az adott tulajdonnév címkéje, például: *A cég felvásárolt még egy gyárat a Mercedes$_{ORG}$ mellett* vs. *A cég megvásárolt még egy telket a Mercedes$_{LOC}$ mellett*.

Taxonómia: Szinonimák, antominák, hipernimák stb. cseréje esetén adott irányú változást mutat (vagy nem mutat változást) az adott tulajdonnév, például: *A Manchesterben$_{ORG}$ futballozott Ronaldo$_{PER}$ vs. A Manchesterben$_{ORG}$ játszott Ronaldo$_{PER}$*.

A fenti nyelvi jelenségek mellett külön megvizsgáltuk azokat az eseteket is, amikor többtagú tulajdonneveket kell azonosítani, valamint a magyar nyelv szórendi jellemzői miatt külön figyelmet fordítottunk azokra az esetekre is, amikor szórendi okok miatt két azonos típusú, ámde különálló névelem került egymás mellé. Lásd az alábbi példákat:

Többtagú tulajdonnevek: *Megalakult a Magyar$_{B\text{-}ORG}$ Nemzeti$_{I\text{-}ORG}$ Bank$_{I\text{-}ORG}$*.

Egymást követő azonos típusú tulajdonnevek: *Évi$_{B\text{-}PER}$ Pétertől$_{B\text{-}PER}$ egy könyvet kapott.*

3.3 Nyelvi variációk

A fenti nyelvi jelenségeken túl a CheckList módszertana lehetőséget ad arra is, hogy további variációs lehetőségeknek vessük alá a tesztmondatainkat. Míg Ribeiro és mtsai (2020) a tagadást és szórendi variációkat szintén nyelvi jelenségként kezelik, mi a magyar nyelv sajátosságai miatt indokoltabbnak látjuk e két variációs lehetőséget különkülön alkalmazni a nyelvi jelenségekre. Így tehát a fenti jelenségeket megháromszorozhatjuk, a fent felsorolt alapesetek mellett beszélhetünk tagadott változatokról és szórendi variánsokról is, amelyek szintén elvárt viselkedést támasztanak a tulajdonnévi címkék esetében. Példaként véve az egyik névutós tesztesetet, itt minden címke változatlan marad az alapesethez képest:

Alapeset:
London$_{MISC}$ után ülésezett a NOB$_{ORG}$.

Tagadott variációk:
London$_{MISC}$ után nem ülésezett a NOB$_{ORG}$.
Nem London$_{MISC}$ után ülésezett a NOB$_{ORG}$.
London$_{MISC}$ után nem a NOB$_{ORG}$ ülésezett.

Szórendi variációk:
A NOB$_{ORG}$ London$_{MISC}$ után ülésezett.
London$_{MISC}$ után a NOB$_{ORG}$ ülésezett.
A NOB$_{ORG}$ ülésezett London$_{MISC}$ után.

4 Magyar NER checklist kiértékelés

4.1 A tesztadatbázis létrehozása

Mindegyik nyelvi jelenségre kézzel állítottunk össze példamondatokat és sablonokat, melyekre aztán kiterjesztettük a nyelvi variancia szintjeit is, így 9*3 mondatcsoportot kaptunk, melyeken tesztelni tudjuk a NER-modellek viselkedését. A kézzel összeállított sablonokból automatikusan generáltuk a tesztmondatokat, összesen 14649 mondatot és 125442 tokent eredményezve.

4.2 Tesztelt névelem-felismerő rendszerek

Három magyar nyelvű névelem-felismerő rendszert választottunk[4] és értékeltünk ki ebben a munkában:

- A SzegedNER egy klasszikus jellemzőkinyerésen alapuló Conditional Random Field (CRF) névelem-felismerő (Szarvas és mtsai, 2006a)
- Az mBERT (többnyelvű BERT) egy, a 104 legnagyobb Wikipédián tanított, többnyelvű BERT-Base model (Devlin és mtsai, 2018). A BERT egy kétirányú nyelvmodellen alapuló ún. kontextualizált szóbeágyazás, ami az egyes szavakhoz egy kontextusfüggő jellemzővektort rendel. A modellt az `emBERT` könyvtárral finomhangolták névelem-felismerésre (Nemeskey, 2020a).
- A huBERT egy magyar BERT modell, amit a Webcorpus 2.0-n és a magyar Wikipédián tanítottak (Nemeskey, 2020b). Mérete (a szótár kivételével) megegyezik az mBERT-évl, viszont kizárólag magyar szövegeken lett előtanítva, ezért kapacitása nem oszlik szét több nyelv között.

Mindhárom névelem-felismerő a Szeged NE (Szarvas és mtsai, 2006b) teljes korpuszán lett betanítva.

4.3 Kiértékelési metrikák

A 1. táblázat tartalmazza az egyes teszteken, az egyes névelem-felismerő rendszerek hibaarányát (százalékban). A hibaarány pontos definíciója:

- MFT típusú tesztek esetén, azt mérjük, hogy kitüntetett frázisokat milyen arányban címkéz helytelenül a névelem-felismerő. Ennek mérésére, a névelem-felismerésben elfogadottan használt, frázisszintű kiértékelő szkriptet használunk és $\text{hiba\_arány}_{\text{MFT}} = 1 - \text{micro\_fedés}$

[4] Nem volt célunk az összes magyar tulajdonnév-felismerő rendszer vizsgálata, viszont a szóbeágyazás-alapú és klasszikus jellemzőkinyerés alapú rendszereket össze akartuk hasonlítani.

ahol micro_fedés az egyes névelem osztályok fedésének (recall) súlyozott átlaga.

- INV típusú tesztek esetén, azt mérjük, hogy kitüntetett frázisoknál milyen arányban változik meg a predikció, ha az alapmondatokon tagadást vagy szórendi változtatásokat hajtunk végre. Megjegyezzük, hogy itt az a hiba, ha megváltozik a címkézés (sérti az invariancia elvárást), függetlenül attól, hogy egyébként az alap mondatban helyes vagy helytelen volt-e a predikció. Azaz az is INV hibának számít, ha az alap mondatban helytelen címkézés, míg a módosított mondatban helyes a címkézés, hiszen változás történt.

Megjegyezzük, hogy ezeknél a teszteseteknél félrevezető a konkrét értékeket vizsgálni vagy összehasonlítani, hiszen a példamondat-sablonokon nagyon sok múlik. Konkrét értékek helyett csak a nagyságrendeket érdemes nézni, azaz, hogy átment-e vagy elbukott az adott teszten az adott rendszer.

5 Eredmények

Az 1. táblázat tartalmazza a három rendszer kilenc nyelvi jelenségen elért eredményeit. A kísérletek megismételhetősége kedvéért, a tesztmondat-sablonok, a generáló és kiértékelő szkriptek[5] elérhetőek a www.github.com/szegedai/hun_ner_checklist oldalon.

1. táblázat. Hibaarányok százalékban kifejezve (minél kisebb, annál jobb).

Nyelvi jelenség	Nyelvi variációk	SzNER	mBERT	huBERT	példa
Névelő	MFT	58	58	33	A Manchesterben játszott Ronaldo.
	INV tagadás	20	10	7	A Manchesterben nem játszott Ronaldo.
	INV szórend	60	21	4	Ronaldo a Manchesterben játszott.
Toldalékolás	MFT	45	45	20	A Hamburggal játszott Messi.
	INV tagadás	40	14	15	A Hamburggal nem játszott Messi.
	INV szórend	58	32	4	Messi a Hamburggal játszott.
Névutó	MFT	49	38	33	Rio után ülésezett a MOB.
	INV tagadás	16	0	0	Rio után nem ülésezett a MOB.
	INV szórend	51	6	0	A MOB Rio után ülésezett.

[5] Ribeiro és mtsai (2020) egy tesztelő felhasználói felületet is implementáltak (https://github.com/marcotcr/checklist). A munkánk megkezdésekor úgy tűnt, hogy egyszerűbb saját szkripteket implementálnunk, mint integrálni mindent a checklist eszközbe. A munka végére ebben elbizonytalanodtunk, ezért a jövőben tesztelni tervezzük magát a checklist felhasználói felületet is.

Többes szám	MFT	74	75	47	Fordok nyerték az autóversenyt.
	INV tagadás	48	30	8	Nem Fordok nyerték az autóversenyt.
	INV szórend	48	38	8	Az autóversenyt Fordok nyerték.
Predikátum cseréje	MFT	54	54	23	A Madridban énekelt Beckham.
	INV tagadás	33	2	0	Nem a Madridban énekelt Beckham.
	INV szórend	36	20	4	Énekelt Beckham a Madridban.
Taxonómia	MFT	53	53	26	A Madridban focizott Beckham.
	INV tagadás	31	3	1	Nem a Madridban focizott Beckham.
	INV szórend	37	19	2	Focizott Beckham a Madridban.
Többtagú tulajdonnevek	MFT	3	3	0	Megalakult az Arab Állami Egyetem.
	INV tagadás	0	1	1	Nem alakult meg az Arab Állami Egyetem.
	INV szórend	0	1	0	Az Arab Állami Egyetem megalakult.
Egymást követő azonos típusú tulajdonnevek	MFT	94	94	91	Gabi Gézától kapott egy csomagot.
	INV tagadás	1	3	6	Gabi Gézától nem kapott egy csomagot.
	INV szórend	1	0	4	Gabi Gézától egy csomagot kapott.
Szókincs	MFT	50	50	48	Szlovénia tengerparton helyezkedik el.
	INV tagadás	1	0	0	Szlovénia nem tengerparton helyezkedik el.
	INV szórend	2	7	0	Tengerparton helyezkedik el Szlovénia.

6 Diszkusszió

A legfontosabb következtetés, amit a 1. táblázatból levonhatunk, hogy míg mindhárom rendszer 95-97% F_1 értéket ér a SzegedNER korpusz tanító-kiértékelő részekre bontásán, a minimális működési tesztjeink (MFT) felén nem megy át, még a legjobb névelem-felismerő rendszer sem (kilencből öt MFT teszt esetén hiba_arány(huBERT) >= ⅓). Azt is kijelenthetjük, hogy egyik rendszer sem képes kezelni az 'egymást követő azonos típusú tulajdonnevek' esetét.[6] Hangsúlyozzuk, hogy a teszteket úgy állítottuk össze, hogy egyszerű, az ember számára egyértelmű feladatok legyenek, amelyeket

[6] Megemlítjük ugyanakkor, hogy a szórendi variációk egyik esetét, amikor az ablativusban álló főnév előzi meg az alanyt (*Pétertől Évi kapott egy könyvet*), a huBERT már képes helyesen azonosítani, a másik két rendszernek azonban ez is nehézséget jelent.

minden névelem-felismerőnek illene teljesíteni (a szoftverfejlesztésben ez a *unit test*-nek felel meg). Ennek oka valószínűleg a nem megfelelő tanító adatbázis rendelkezésre állása, ugyanis a Szeged NE korpusz gazdasági rövidhírekből áll (Szarvas és mtsai, 2006b), míg a teszteseteink tartalmaznak hétköznapi életbeli (pl. *Megittam egy Soproni*) és sport (pl. *Xavi a Barcelonában futballozott*) tematikájú mondatokat is.

Az invariancia teszteken (INV) azonban nagyon jól teljesít a huBERT, kijelenthetjük, hogy azokon mind átmegy (egyedül a toldalékolásos tesztek tagadásos variánsán változik az esetek több, mint 10 százalékában a predikció).

Ha a három rendszert összehasonlítjuk, akkor is a SzegedNER-es kiértékelésnél jóval árnyaltabb kép nyerhető az 1. táblázatból. A SzegedNER tanító-kiértékelő részekre bontásán alapuló kiértékelésekben a klasszikus gépi tanuláson alapuló rendszerek, mint a SzegedNER (Szarvas és mtsai, 2006a) vagy hunner (Varga és Simon, 2007) 95% körüli F_1 értéket, míg a BERT alapú rendszerek - mind az mBERT, mind a huBERT - 97% körüli F_1 értéket érnek el (Nemeskey 2020). Míg a SzegedNERen nincs szignifikáns különbség az mBERT és huBERT között, a fenti teszteken egyértelműen jobban teljesít a huBERT, hat MFT teszten felezi az mBERT hiba arányát és lényegében minden INV teszten átmegy, míg az mBERT-nél legalább négy esetben mondhatjuk, hogy elbukik (hiba_arány(mBERT) >= ⅓)).

Ha a tesztjeinken elért eredményeket vizsgáljuk, azt mondhatjuk, hogy az mBERT viselkedése közelebb áll a SzegedNERéhez, mint a huBERTéhez, ami ellentmond a SzegedNER korpuszon mért F_1-értékek által festett képnek. Az mBERT csak a névutó MFT teszten teljesít jobban, mint a SzegedNER, igaz, robusztusabb a tagadás és szórendi változásokra (minden INV teszten, amin a SzegedNER elbukik, sokkal jobban teljesít az mBERT). Ez utóbbinak valószínűleg az a magyarázata, hogy a SzegedNER jellemzőkészletében fontos jellemzők az ún. ablakolt jellemzők, azaz pl. a címkézendő szót kettővel megelőző szó jellemzői, míg a BERT transzformer modellje az egész bemenetet figyelembe tudja venni.

Az eredmények részletesebb, nyelvi szinteket is figyelembe vevő elemzéséből az is kiviláglik, hogy - az egymást követő azonos típusú tulajdonnevek esetét leszámítva - a többes szám jelenségét, azaz egy morfológiai változást a legnehezebb kezelni a rendszereknek, hiszen itt láthatók a legmagasabb hibaarányok. Ezzel szemben egy másik morfológiai jelenség, a toldalékolás tesztjén viszonylag kevesebb hibát láthatunk: úgy tűnik tehát, hogy modelljeink fel vannak készítve a névelemek ragozott alakjainak kezelésére a magyar nyelvben. Érdekességképpen megjegyezzük, hogy míg utóbbi jelenség elsődlegesen a morfológiailag gazdag nyelvekre jellemző, addig a tulajdonnevek többes számba tétele (pl. márkanevek használata esetén) a nyelvek szélesebb körében ismert jelenség, így a jövőben mindenképpen hasznos lenne e nyelvi jelenségek vizsgálata más nyelvek CheckList-tesztjeiben is.

Ami a szintaktikai jellegű teszteket illeti, a névelő tesztjén rosszabbul teljesít a SzegedNER és az mBERT, mint a névutó esetében, a huBERT azonban azonos eredményt ér el. Úgy tűnik, az mBERT kevésbé ismeri fel a mondatkezdő pozícióban szereplő tulajdonneveket (pl. *Athén után ülésezett a NOB*), ami részben okozhatja a gyengébb teljesítményt a névutós tesztmondatok esetében.

A szemantikai jellegű tesztek esetében (predikátum cseréje, taxonómia) a SzegedNER és az mBERT egyaránt 50 körüli hibaarányt mutat, míg a huBERT 23-26-ot. Úgy tűnik tehát, hogy a szemantikai változásokra is robusztusabb a huBERT a másik két rendszernél. A szókincs tesztjén viszont mindhárom rendszernél gyakorlatilag azonos

hibaarányt láthatunk, noha jellemzően más hibákkal: míg az mBERT a terméknevek felismerésénél mutat hibákat, addig például a huBERT a szervezetnévként funkcionáló országneveknél hibázik többet.

Végül elmondhatjuk, hogy tesztjeink közül kimagaslóan a legjobb teljesítményt érték el a rendszerek a többtagú tulajdonnevek azonosításában, minimális hibaarányokkal, ugyanakkor a legtöbb hibát pedig az egymást követő azonos típusú tulajdonnevek kezelésében érhettük tetten. Ez arra utal, hogy a közvetlenül egymás mellett látott névelemek felismerése viszonylag nehéz feladat, egyben annak is jele, hogy mindegyik rendszer hajlamos az egymás mellett látott, azonos típusú névelemnek vélt elemet összevonni. Utóbbi sajátosság megint csak elsődlegesen a szabad szórendű (morfológiailag gazdag) nyelvekre jellemző, így hasonló nyelvek CheckList-es vizsgálata e téren is hozzájárulhat a névelem-felismerés kiértékelésének módszertani újragondolásához.

7 Összegzés

Cikkünkben bemutattunk magyar névelem-felismeréshez kilenc nyelvi jelenséget, amit 27 darab CheckList teszttel tudunk ellenőrizni. A három névelem-felismerő rendszer tesztelése fontos betekintést nyújt a rendszerek viselkedésébe.

Hangsúlyozzuk, hogy a CheckList kiértékelést kiegészítésként és nem alternatívaként, ajánljuk a klasszikus tanító- és kiértékelő adatbázisra bontáson számolt pontosság metrika mellett. Továbbá a kilenc nyelvi jelenség mellett, még számos más nyelvi jelenség tesztelhető a magyar névelem-felismerésben és minden feladat és alkalmazásnak saját nyelvi jelenségei vannak, azokat specifikusan kell definiálni. Cikkünk fő célja az, hogy minden olvasót motiváljunk arra, hogy értse meg jobban a nyelvtechnológiai alkalmazásainak viselkedését, amihez a CheckList keretrendszer egy hasznos eszköz.

Köszönetnyilvánítás

Farkas Richárd kutatási 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.

Zahorszki Róbert munkáját a "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.

A publikációban szereplő kutatást az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

Hivatkozások

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. Information Fusion, Volume 58, pp 82-115 (2020)

Devlin, J., Chang, M-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL, pp. 4171–4186 (2019)

Ethayarajh, K., Jurafsky, D.: Utility is in the Eye of the User: A Critique of NLP Leaderboards. In: EMNLP (2020)

Hewitt, J., Manning, C.D.: A Structural Probe for Finding Syntax in Word Representations. In: NAACL (2019)

Nemeskey D. M.: Egy `emBERT` próbáló feladat. In: MSZNY (2020a)

Nemeskey, D. M.: Natural Language Processing Methods for Language Modeling. PhD disszertáció (2020b)

Paroubek, P., Chaudiron, S., Hirschman, L.: Principles of Evaluation in Natural Language Processing. Traitement Automatique des Langues, ATALA 48 (1), pp.7-31 (2007)

Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In: ACL (2020)

Szarvas, Gy., Farkas, R., Kocsor, A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: Discovery Science, 9th International Conference, pp. 268–278 (2006a)

Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.: Highly accurate Named Entity corpus for Hungarian. In: International Conference on Language Resources and Evaluation (2006b)

Varga, D., Simon, E.: Hungarian Named Entity Recognition with a Maximum Entropy Approach. In: Acta Cybernetica 18(2), pp. 293–301 (2007)