XIX. Magyar Számítógépes Nyelvészeti Konferencia

Szerkesztette:
Berend Gábor
Gosztolya Gábor
Vincze Veronika

Szeged, 2023. január 26–27.

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

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

ISBN: 978-963-306-912-7

Nyomtatta:
Innovariant Nyomdaipari Kft.
6750 Algyő, Ipartelep 4.

Szeged, 2023. január

Az MSZNY 2023 konferencia szervezője:
ELKH-SZTE Mesterséges Intelligencia Kutatócsoport

Előszó

2023. január 26–27-én már tizenkilencedik alkalommal kerül sor a Magyar Számítógépes Nyelvészeti Konferencia megrendezésére. Ebben az évben először hibrid formátumban kerül lebonyolításra a konferencia, lehetőséget teremtve mind a személyes részvételre, mind pedig a konferencia élő közvetítésének nyomon követésére.

A konferencia fő célkitűzése a kezdetek óta állandó: lehetőséget biztosítani a nyelv- és beszédtechnológia területén végzett kutatások eredményeinek ismertetésére és megvitatására, ezen felül a különféle hallgatói projektek, illetve ipari alkalmazások bemutatására. A hagyományokat követve a konferencia idén is nagyfokú érdeklődést váltott ki az ország nyelv- és beszédtechnológiai szakembereinek körében. A 43 beküldött cikkből gondos mérlegelést követően 35 cikk került elfogadásra, melyek témája számos szakterületre terjed ki a legújabb nyelvi modellek bemutatásától kezdve a beszédtechnológia eredményein keresztül a gépi fordításig.

Nagy örömet jelent számunkra, hogy Tikk Domonkos elfogadta meghívásunkat, aki plenáris előadását *A Netflix Prizetól a Tabooláig: a Gravity és az ajánlórendszerek fejlődése* címmel fogja megtartani.

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

Köszönettel tartozunk továbbá a Neumann János Számítógéptudományi Társaság Mesterséges Intelligencia Szakosztályának a konferencia sikeres lebonyolításához nyújtott anyagi hozzájárulásáért.

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

Tartalomjegyzék

Orvosi nyelv- és beszédtechnológia **1**

3 Klinikai leletek strukturálása mondatszintű címkézéssel
Szabó Ledenyi Klaudia, Pusztai Ágnes, Kicsi András, Vidács László

17 Fokozás szkizofréniában
Szabó Martina Katalin, Vincze Veronika, Guba Csenge, Dam Bernadett, Solymos Adrienn, Bagi Anita, Szendi István

33 Sclerosis multiplex felismerése spontán beszédből wav2vec 2.0 modellekből kinyert jellemzőkkel
Gosztolya Gábor, José Vicente Egas-López, Svindt Veronika, Bóna Judit, Hoffmann Ildikó

45 A borderline személyiségzavar felismerése a nyelvhasználat lexikai és grammatikai jellemzői alapján
Felletár Fanni, Babarczy Anna

Szemantika **61**

63 Magyar melléknevek poliszém jelentéseinek automatikus kinyerése gráfokkal
Héja Enikő, Ligeti-Nagy Noémi

77 Mondatszám-meghatározás hatása a magyar nyelvű jogi szövegek extraktív kivonatainak minőségére
Csányi Gergely Márk, Gadó Krisztián, Bajári Lúcia, Megyeri Andrea, Fülöp Anna, Egri Erika, Vági Renátó, Nagy Dániel, Vadász János Pál, Üveges István

91 Data Augmentation for Machine Translation via Dependency Subtree Swapping
Attila Nagy, Dorina Lakatos, Botond Barta, Patrick Nanys, Judit Ács

107 Neurális entitásorientált szentimentelemző alkalmazás magyar nyelvre
Yang Zijian Győző, Laki László János

119 Koreferenciafeloldás magyar szövegeken BERT-tel
Vadász Noémi, Nyéki Bence

Beszédtechnológia **133**

135 „Feeding the BEAST” – A BEA Speech Transcriber továbbfejlesztése és integrálása neurális nyelvmodellel
Kádár Máté Soma, Dobsinszki Gergely, Mády Katalin, Mihajlik Péter

145 Magyar nyelvű neurális beszédszintézis vizsgálata dialógus helyzetben
Zainkó Csaba, Csapó Tamás Gábor, Bartalis Mátyás, Németh Géza, Németh Norbert, Szász Gábor Krisztián, Szviridov István

159 Effects of emotional speech on forensic voice comparison using deep speaker embeddings
Mohammed Hamzah Abed, Dávid Sztahó

171 Cross-lingual dysphonic speech detection using pre-trained speaker embeddings
Aziz Dosti Ali Hama Salih, Dávid Sztahó

Korpuszok, adatbázisok **185**

187 HunEmPoli: magyar nyelvű, részletesen annotált emóciókorpusz
Ring Orsolya, Vincze Veronika, Guba Csenge, Üveges István

203 MILQA kérdés-válasz benchmark adatbázis
Novák Attila, Novák Borbála

217 Hát te mekkorát nőttél! – A HuLU első életéve új adatbázisokkal és webszolgáltatással
Ligeti-Nagy Noémi, Héja Enikő, Laki László János, Takács Dávid, Yang Zijian Győző, Váradi Tamás

231 HunSum-1: an Abstractive Summarization Dataset for Hungarian
Botond Barta, Dorina Lakatos, Attila Nagy, Milán Konor Nyist, Judit Ács

Nyelvmodellek **245**

247 Jönnek a nagyok! BERT-Large, GPT-2 és GPT-3 nyelvmodellek magyar nyelvre
Yang Zijian Győző, Dodé Réka, Ferenczi Gergő, Héja Enikő, Jelencsik-Mátyus Kinga, Kőrös Ádám, Laki László János, Ligeti-Nagy Noémi, Vadász Noémi, Váradi Tamás

263 Látens szemantikus eloszlások használata a nyelvi modellek előtanítása során
Berend Gábor

275 Magyar nyelvű időjárásjelentések nyelvi modell alapú automatizált generálása
Knap Árpád, Dömsödi L Bíborka, Mogyorósi Pálma, Szigeti Péter, Tóth Andor, Virág Attila, Kmetty Zoltán

Morfológia, előfeldolgozás **289**

291 Korpusztisztítás és sorvégi kötőjelek kezelése karakteralapú neurális nyelvmodellel
Pethő Gergely, Sass Bálint, Simon László, Lipp Veronika

305 Transformer-alapú HuSpaCy előelemző láncok
Szabó Gergő, Orosz György, Szántó Zsolt, Berkecz Péter, Farkas Richárd

319 Hybrid lemmatization in HuSpaCy
Péter Berkecz, György Orosz, Zsolt Szántó, Gergő Szabó, Richárd Farkas

331 Neural Morphological Generators for Hungarian
László János Laki, Noémi Ligeti-Nagy, Noémi Vadász, Zijian Győző Yang

341 Gondolatok a gondola-tokról. Morfológiai annotációt javító módszerek tesztelése gold standard korpuszon
K. Molnár Emese, Dömötör Andrea

Poszter, laptopos bemutató **355**

357 A beszéd artikulációs mozgásának predikciója agyi jel alapján – kezdeti eredmények
Csapó Tamás Gábor, Arthur Frigyes Viktor, Nagy Péter, Boncz Ádám

369 Magyarcentrikus többnyelvű gépifordító rendszerek létrehozása
Laki László János, Yang Zijian Győző

381 Többnyelvű modellek és PEGASUS finomhangolása magyar nyelvű absztraktív összefoglalás feladatára
Yang Zijian Győző

395 Inzulinrezisztencia betegség jelenségének felismerése és osztályozása orvosi dokumentumokban
Yang Zijian Győző

405 huBERT alapú sziámi neurális háló architektúrák elemzése ügyfélszolgálati emailek klasszifikációjára
Vándor Péter, Csáki Csaba

417 HuBERTUSz: Alacsony paraméterszámú transzformer modellek létrehozása és kiértékelése magyar nyelvre
Ficsor Tamás, Berend Gábor

433 A new ParlaMint corpus for Hungarian – 30m tokens of annotated parliamentary data
Noémi Ligeti-Nagy, Réka Dodé, Kinga Jelencsik-Mátyus, Zsófia Varga, Enikő Héja, Tamás Váradi

447 Korpuszépítés és -feldolgozás learatott webes tartalomból
Kalcsó Gyula, Mihály Eszter, Szűcs Kata Ágnes

457 Tagmondatokra bontás és NP-chunking függőségi alapon
Dömötör Andrea, Nemeskey Dávid

Szerzői index, névmutató **471**

Orvosi nyelv- és beszédtechnológia

Klinikai leletek strukturálása mondatszintű címkézéssel

Szabó Ledenyi Klaudia[1], Pusztai Ágnes[1], Kicsi András[1], Vidács László[1]

[1]Szegedi Tudományegyetem, Szoftverfejlesztés Tanszék
Szeged, Dugonics tér 13.
{ledenyik,pusztai,akicsi,lac}@inf.u-szeged.hu

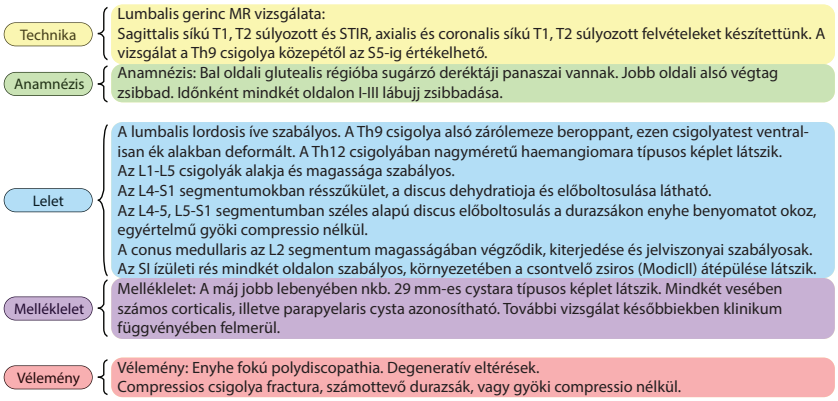
Kivonat A szöveges orvosi leletek különböző jellegű információkat tartalmaznak, mint például a vizsgálati technika, a páciens panaszai, az orvos által megállapított elváltozások, vagy az orvos véleménye. Az információ strukturálása nagyban megkönnyítheti a leletek értelmezését, adminisztrációját, és gépi feldolgozását is. Cikkünkben egy huBERT modellen alapuló megoldással címkézzük fel magyar nyelvű gerincleletek szövegét mondatonként. A modell egy egyszerű utófeldolgozás után 98% fölötti pontossági értéket produkált. Eredményeink alapján a modell jól általánosít más anatómiai régiók (97%), és más intézmények (93%) leleteire is, jó alternatívát nyújtva az anatómiai régiókra vagy intézményenkre optimalizált szabályalapú módszerek helyett.
Kulcsszavak: leletezés, lelet-strukturálás, mondat címkézés, nlp

1. Bevezetés

Az egészségügyi vizsgálatok során számos szöveges leírás keletkezik, amellyel a szakemberek leírják a páciens panaszait, korábbi klinikai információit, az orvosok által kért vizsgálatokat, a vizsgálatok során tapasztalt rendellenességeket, az orvosok véleményét, a javasolt kezeléseket, vagy egyéb más adatot a páciensről. A leletek formátumában gyakran vannak megkötések, a leletek struktúrájának fenntartására irányuló törekvések, ám ezek formája intézményenként, alkalmazása pedig orvosonként változhat. Ha van is ilyen kikényszerített és megfelelően fenntartott struktúra, az ilyen leíró mezők szövege akkor is általában kötetlen természetesnyelvű szöveg.

A strukturált leletezés az irodalom szerint nagyobb pontosságot és konzisztenciát eredményez az orvosok részéről, és a leleteket olvasó klinikusok is nagyobb elégedettséget és jobb értelmezési képességet mutattak (Bosmans és mtsai (2011),Ganeshan és mtsai (2018)). A leletek gépi feldolgozásához is nagyban hozzájárulhat az, ha legalább az adott dokumentáció típusát meg tudjuk különböztetni, hiszen például a vizsgálati technika, vagy a beteg korábbi vizsgálatainak leírásához kifejezetten más célzott módszerek lehetnek szükségesek, mint a tényleges leletszöveg, vagy vélemény elemzéséhez.

Magyarországon az intézmények többségében a lelet formátuma nem teljesen kötött, az információ többségét általában egy szövegdobozban rögzítik. Ezen belül ugyan a különböző szakaszokat gyakran tagolni szokták bekezdésekkel, de ez



1. ábra: Egy példalelet, releváns szakaszaival címkézve

egyáltalán nem kötelező, és egy-egy szakasz megléte is opcionális, nem feltétlen kötött a sorrendjük, illetve egy szakasz több bekezdésből is állhat. Számos intézményben a szakaszt felcímkézik a szövegben, de ezek a címkék akár intézményen belül is változhatnak, és szintén nem kötelezőek. Egy példaleletet láthatunk az 1. ábrán, amely a későbbiekben általunk is tárgyalt szakaszokat szemlélteti. Kutatásunk során radiológiai leletekhez volt hozzáférésünk, így a továbbiakban ezen területet fogjuk részletesebben vizsgálni.

Tekintsük át röviden a példa szakaszait: Az első szakaszban a vizsgálati technika leírását látjuk. Ez megadja, hogy milyen vizsgálat készült, mi az, ami alapján a lelet le fogja írni a megállapításokat. A vizsgálatok általában az érintett testrészek szempontjából is fontosak. Szintén gyakori azonban, hogy több vizsgálat eredménye is bekerül egy leletbe, ilyenkor az új vizsgálat vizsgálati technikáját is megadják, a lelet egy későbbi bekezdésében. Az anamnézis szakasz a páciens előéletét adja meg, panaszait írja le, illetve korábbi vizsgálatait vagy kezelését. Ezt a szakaszt jelentősen nagyobb szókincs jellemzi, hiszen tartalma kevésbé szakmai (például „három hónappal korábban elesett a kádban”). A részletes lelet leírása képezi a tartalom legfontosabb részét, ebben az orvos részletesen leírja a vizsgálat során, vagy a felvételeken látott összes tapasztalatát, akár a korábbi felvételekhez is viszonyítva. Általában nem csak a kóros eltéréseket jegyzi meg, hanem azt is, ha valamit a normálisnak megfelelő állapotúnak lát. A leletekben szerepelhet melléklelet is, ez olyankor releváns, ha a felvételeken a specialista olyan kóros elváltozást is lát, amely ugyan nem szigorúan része a vizsgálatnak, de mégis fontos megjegyezni, hiszen a páciensnek ugyanúgy problémát okozhat, mint a célzott területen lévő elváltozások. A leletet általában vélemény zárja, amely szintén az orvos tapasztalatain alapszik, ez szinte minden esetben a lelet által már leírt információt tartalmazza, ám gyakran szűkszavúbb és szakmaibb megfogalmazásban, illetve a lelet lényeges tartalmát foglalja össze. Szintén itt szoktak megjelenni az orvos javaslatai is a további kezelésre, esetleg újabb vizsgálatokra. A leírtakból valójában csak a részletes lelet a kötelező, de a technika és vélemény jelenléte szintén igen gyakori.

A szakaszok megfelelő elkülönítése különösen érdekes a későbbi automatizált feldolgozás szempontjából. A szakaszok különböző információt tartalmaznak, ennek megfelelő kinyerése hozzájárulhat a jobb és átláthatóbb adminisztrációhoz, és minőségibb leletezéshez. A leletszöveg és vélemény például egymáshoz képest redundanciát is tartalmaz, könnyű belátni, hogy ez hogyan eredményezhet később zajt. A páciensek szemszögéből valós alkalmazás lehet a leletek megértését célzó elektronikus lelet, ehhez szintén szükség van a lelet megfelelő tagolására. Ilyen esetben értékes funkciót jelenthet az is, ha nyomtatott leletet is feltölthetnek, ehhez viszont kritikus lehet a tartalmat megkülönböztetni a papíron található többi metaadattól, mint az intézmény, vagy a páciens adatai.

Jelen írásunkhoz valós leleteken végeztünk mondatszintű annotációt a szakaszok azonosításával. Fókuszunkban radiológiai gerincleletek voltak, a tanítást ezeken végeztük, egy intézmény leletein. Felmerül azonban a kérdés, hogy mennyire lehet általánosítható az így tanított modell másik intézményből származó, vagy az emberi test egy másik területét érintő leletek esetére. Kísérleteinkben ezért a következő három kutatási kérdést fogjuk vizsgálni:

- **1. kérdés**: Képes-e egy gépi tanulási modell pusztán a leletszövegből – bekezdések és egyéb címkék megkövetelése nélkül – megfelelően szakaszokra bontani a leleteket?
- **2. kérdés**: Egy terület és intézmény leletein tanított modell hogyan teljesít más területről/intézményekből származó adatok strukturálásában?
- **3. kérdés**: Hogyan teljesít az általános módszer az intézményenkénti szabályalapú megközelítésekkel szemben?

2. Háttér

A szabadszöveges leletek feldolgozását nagyban elősegítik a természetesnyelvfeldolgozó (NLP) módszerek. Ezek fejlődésével, a különböző nyelvfeldolgozási feladatokra használt, kezdeti szabályalapú és heurisztikus módszereket a szószintű reprezentációk váltották fel. A reprezentációk létrehozásának fejlődésével azok egyre kifinomultabbá váltak, és egyre több információt foglaltak magukba. A kezdeti reprezentációk, mint például a word2vec (Mikolov és mtsai (2013)), gloVe (Pennington és mtsai (2014)) és a fastText (Joulin és mtsai (2016)) a szavak egyéni, karakterszinű tulajdonságait használták fel, azonban a szövegkörnyezet által rejtett információkat nem tartalmazták. A szövegkörnyezeti információk felhasználására az ELMo (Peters és mtsai (2018)) és BERT modellek (Devlin és mtsai (2019)) nyújtottak megoldást.

A nyelvfeldolgozás területén történő fejlődés az orvosi leletek szemantikus szekciókra való felbontását (section segmentation) is pozitívan befolyásolta. A kezdeti kísérletekben statisztikai és gépi tanuló módszereket alkalmaztak. A statisztikai módszerekben a kézzel megtervezett jellemzők játszottak nagy szerepet, köztük az olyan szóalakjellemzők, mint a nagy kezdőbetűk, számok, üres sorok jelenléte és pozíciója. Taira és társai publikációjukban (Taira és mtsai (2001)) a

szabályalapú stukturális elemzés után, több lépésben történt a szekciók címkéinek meghatározása. Egy későbbi kutatás szerzői a heurisztikus határazonosítások után egy support vector machine (SVM) modellel próbálták meghatározni az egyes szekciókhoz tartozó címkéket (Apostolova és mtsai (2009)). Tepper és társai (Tepper és mtsai (2012)) a mondatszintű klasszifikáció helyett sorszinten dolgoztak, a címkék meghatározása pedig a MaxEnt elnevezésű modellel történt. Publikációjukból kiderült, hogy a sorszintű azonosítás hátránya, hogy a külön bekezdések hiánya negatívan befolyásolja az eredményt.

Egy 2019-es áttekintő kutatás (Pomares-Quimbaya és mtsai (2019)) szerint az elektronikus orvosi leletekben (EHR) a statisztikai és szabályalapú módszerek domináltak, és a legjobb eredményeket ezek kombinációjával tudták elérni. Ezek az eredmények azonban még mindig nagyban függtek a kiépített kézi jellemzők minőségétől. A statisztikai és szabályalapú módszerek időigényesek, és nem általánosíthatók, ami további megoldások keresését vonta maga után.

A szóbeágyazások és neurális hálók elterjedésével a szekciók szegmentációjában is megjelentek az ezzel kapcsolatos kutatások (Sadoughi és mtsai (2018)). Az ELMo és a BERT sikere után Rosenthal és munkatársai (Rosenthal és mtsai (2019)) 400 CTPA (Computed Tomography Pulmonary Angiography) lelet alapján, összesen több mint 4 000 mondat felhasználásával készítettek mondatszintű, szövegkörnyezetet is figyelembe vevő, terület-specifikus modellt. Előre adott szekciók klasszifikációjában a szerzők egy RNN modellel a terület-specifikus leleteken 93%-os F1-mértékkel sikeresen klasszifikálnak, és a BERT ezt a teljesítményt majdnem tökéletes, 99%-ra emeli. Azonban a gyakorlatban ezek a szekciók nem előre adottak, így ismeretlen szekcióhatárokkal, mondatklasszifikációval is bemutatják a szerzők a BERT fölényét: a 63%-os RNN modell eredményét egy 74%-os BERT modellel javítják.

Kuling és társai elkészítették a BI-RADS BERT-et (Kuling és mtsai (2022)), mely egy 155 000 radiológiai leleten előtanított BERT modell. Az előtanított modellt többek között szakasz-szegmentációs feladatra is finomhangolták, úgy, hogy a BERT reprezentációit egy 3-rétegű teljesen összekötött és egy Tanh aktivációs függvénnyel ellátott hálóval egészítették ki. A második háló kézzel szerkesztett jellemzők (az előző mondat klasszifikációja, a mondat sorszáma, és a leletben található összes mondat száma) alapján egy 128 elemű kiegészítő jellemzővektort készített. Az egyéb jellemzőket tartalmazó vektor BERT reprezentációhoz adása több százalékos javulást eredményezett. Az előtanított BI-RADS BERT további 900 leleten való szekció szegmentációra való fingomhangolás után 89,5%-os F1, és 97,8%-os pontosságú eredményt mutatott. A szerzők által bemutatott kiegészítő adatokon tanult hálóval ezt az F1-mértéket 94,8%-ra emelték, 94,1%-os pontossággal.

A magyar nyelvű leletek tekintetében a szakaszokra bontás egyelőre nem kapott sok figyelmet. Ugyan az orvosi leletekkel foglalkozó kutatások már elterjedtek (Siklósi és Novák (2014), Kicsi és mtsai (2019)), ezek általában csak a cikkünkben részletes leletnek vagy véleménynek nevezett szakasszokkal foglalkoznak. Siklósi és Novák (Siklósi és Novák (2015), Siklósi és Novák (2016), Siklósi és Novák (2015)) foglalkoztak a leletek automatikus strukturálásával, szemészeti

ambuláns leletekre fókuszálva, ezen túlmutató munkájuk egy nyelvi modelleken alapuló automatizált javító módszer, és keresőrendszer volt, a strukturálásban a leletek formázását és szózsák modellre alapozó indikátorkifejezéseket felhasználva. Tizenöt, számos esetben szemészet-specifikus címkéjükkel strukturálásuk 81,99%-ban volt helyes. Korábbi munkánk során a radiológiai leletek gépi értelmezésével foglalkoztunk (Kicsi és mtsai (2020)), ennek során a részletes lelet és vélemény szakaszokat használtuk fel entitások detektálására és azonosítására, kapcsolataik feltálrására.

A jelen cikkben bemutatott módszerünk magyar nyelvű klinikai leleteken a mondatok szintjén végez címkézést 5 szakaszba kategorizálással, a szöveg formázásának felhasználása, vagy indikátorkifejezések megállapítása nélkül.

3. Módszerek

3.1. Adatok

A feladat során az elsődleges célunk gerincről készült MR-leletek szakaszokra bontása volt. A feladatban egy nagyobb projekt során használt leletekkel dolgoztunk, mely a Budai Egészségközponttal - a továbbiakban [*A intézménnyel*] - közösen kerül megvalósításra. Az ebből a forrásból származó adatokra *A_teszt, és A_NG (nem gerinc-vizsgálat)* néven fogunk hivatkozni. Jelen munkához 600 gerinclelet került véletlenszerű kiválasztásra, melyből 500 leletet használtunk fel a modellek tanításához, és további 100 leletet a különböző adatokon tanult modellek kiértékeléséhez. Rendelkezésünkre álltak továbbá nem gerincről készült leletek is, ezekből 25 került kiválasztásra, melyeket csak a modellek teszteléséné használtunk fel.

Az annotációs munka több lépésben zajlott. Az első lépés a különböző szakaszok címkéinek kidolgozása volt. Ennek eredményeként öt címke használata mellett döntöttünk, amelyeket bevezetésünk ismertetett bővebben, ezek a következők:

- **Technika**: Minden olyan mondat, mely a vizsgálat típusát (pl. T1 súlyozott, T2, kontrasztanyagos) illetve a vizsgálati régiót írja le.
- **Anamnézis**: Ezen címke alá kerültek a jelenlegi panaszokat, korábbi betegségeket, műtéteket leíró részek, mely azért szükséges, hogy az orvos tágabb kontextusban lássa a páciens problémáját.
- **Részletes lelet**: Ebben a szakaszban kerülnek feljegyzésre az orvosok által megállapított rendellenességek, megállapítások.
- **Melléklelet**: Ebbe a csoportba azok a mondatok kerülnek, melyek nem a fő vizsgálati területről tartalmaznak megfigyeléseket.
- **Vélemény**: Tekinthető a részletes lelet összefoglalójának, melyben a leletben szereplő fontosabb információk kerülnek megemlítésre.

A tanításhoz használt első 100 lelet kézi felcímkézése után a következő 200 lelet a már meglévő leleteken tanított modellel került előannotálásra, majd az

annotátorok feladata az előannotációk javítása volt. A további annotálandó leletek pedig ezen a 300 leleten tanított modellel kerültek előannotálásra. A leleteket jelen cikk három szerzője annotálta, a kérdéses esetek közös egyeztetésével.

A korábban említett korpusz mellett egy további intézmény, a Szegedi Tudományegyetem - a továbbiakban [*B intézmény*] - klinkáján dolgozó radiológusok által kibocsájtott leletek is rendelkezésünkre álltak. Az ebből a forrásból származó leletekre *B_teszt* néven fogunk hivatkozni. Tapasztalataink szerint ezek nagymértékben strukturált leletek (legalábbis a szakaszok szintjén), azonban ezek eltérő kifejezésekkel rendelkeznek az [*A intézmény*] adataihoz képest. Ebből a csomagból 25 gerinclelet került felcímkézésre, annak érdekében, hogy más klinikáról származó, eltérő felépítéssel, szókinccsel rendelkező leleteken is le tudjuk mérni a modell pontosságát.

3.2. Szabályalapú módszerek

A szakaszok felismerésének feladatára könnyű megoldásnak tűnhetnek a szabályalapú módszerek. A két intézménytől kapott leletek strukturálására egy-egy szabályrendszer került kidolgozásra. Mindkét esetben az első lépés a leletek felépítésének megismerése volt.

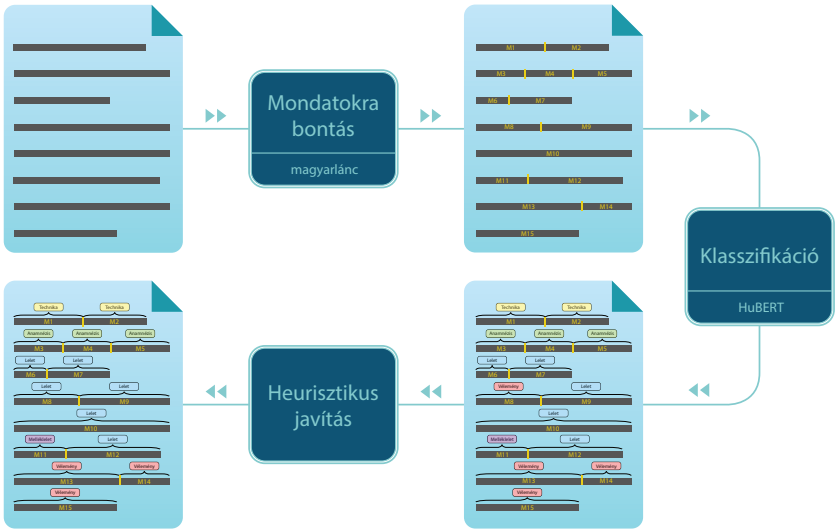
Az [*A intézménytől*] kapott szövegekben gyakoriak voltak az üres sorok, illetve a mondatonkénti újsorok. A leletek sok esetben tartalmaztak bizonyos címkéket a szakaszok elején, mint az anamnézis, vélemény, illetve melléklelet. Ezek elkülönítése könnyű olyan esetekben, amikor például dupla sortörés jelezte a megkezdett címkéhez tartozó rész végét. Azonban azokban az esetekben, amikor az egész lelet nem tartalmazott ilyen formázást, egyszerű szabályokkal próbáltuk megtalálni a szekciók közötti határokat. A címkével nem rendelkező részek meghatározásához a szövegek elemzése során készült szócsoportokat használtuk, melyen belül az egyes csoportok valamely egység gyakori szavait tartalmazták.

A [*B intézménytől*] származó leletek többsége erősen strukturált formával rendelkezett. Itt nem voltak jellemzők a dupla sortörések, azonban szinte minden esetben jelzésre került az új szekció kezdete a szekció nem feltétlenül egységes megnevezésével. A szekcióelnevezések gyűjtése, szinonimák meghatározása után ezen leletek strukturálása már egyszerűen elvégezhető volt a szekciónevek mentén való vágással.

A szabályalapú módszerek fejlesztése általában időigényes, mivel ehhez a szövegek hosszadalmas elemzése szükséges, illetve nehezen alkalmazható más témájú, más intézménytől származó, eltérő felépítésű leletek strukturálásához. A statisztikai módszerek, például szózsák modellek segíthetnek ugyan, de rugalmasságuk így is korlátozott.

3.3. Hibrid módszer

Módszerünkben a strukturálást mondatonként végeztük, melyben minden mondat egy címke alá sorolható. A kiosztott címkék alapján történik a különböző szekciók meghatározása. Ezt a folyamatot a 2. ábra illusztrálja. A feladat során



2. ábra: Módszerünk illusztrációja

egy hibrid módszer került kidolgozásra, melynek első lépése a leletek megtisztítása és a magyarlánc (Zsibrita és mtsai (2013)) nyelvi elemzővel való mondatokra bontása. A mondatrabontást követően minden mondatnak elkészítettük a reprezentációját, mely után egy klasszifikációs réteg segítségével kerültek meghatározásra az egyes mondatokhoz tartozó címkék. A modell által létrehozott címkék meghatározását követő lépés a címkék javítása a kiosztott címkék sorrendiségének figyelembevételével, ezt az ábrán heurisztikus javításnak neveztük.

Az egyes mondatokhoz tartozó reprezentációkat a BERT modell felhasználásával készítettük el. A modell egyetlen kimeneti réteg hozzáadásával finomhangolható, így state-of-the-art eredményeket elérve több természetesnyelvfeldolgozási feladaton. Minden, a modell után a folyamatba épített specifikusabb feladathoz (downstream task) külön finomhangolt modell tartozik, így minimális a különbség az előre tanított és a downstream modell architektúrája között. Ma már számos különböző nyelvű és témájú szövegeken előtanított BERT modell létezik, azonban a magyar nyelven tanított BERT modellek választéka kicsi, és az előtanított elérhető modellek általános szövegen kerültek előtanításra. A szabadon használható magyar nyelvű modellek közül a huBERT (Nemeskey (2021)) modell használata mellett döntöttünk. Modellünk architektúrájában a huBERT által készített mondatszintű reprezentációk ([CLS] token) képezték a klasszifikációs réteg bemenetét, mely lehetővé teszi a mondatszintű címkék előállítását.

A következő lépésben a huBERT modellre helyezett klasszifikációs réteggel előállított címkék szabályalapon történő javítására került sor. A szabályokat a kimenetek elemzésével hoztuk létre, és a leggyakoribb hibák javítására fektettük a hangsúlyt. A korábban ismertetett csoportok közül a leghasonlóbb szókinccsel a részletes lelet és a vélemény rendelkezik. A szövegkörnyezet nélkül ezek gyakran

szabad szemmel is megkülönböztethetetlenek. Utófeldolgozási szabályainkban figyelembe vesszük a mondatot körülvevő kiosztott címkéket is. Amennyiben például két azonos címke közé bekerül egy másik típusú címke, a közrevett címkét megváltoztatjuk az őt körülvevőre. Törekedtünk csak olyan minták megalkotására, melyek javítanak a kimeneten, összetett leletek esetében azonban előfordul, hogy helytelenül írjuk át a címkét. Tapasztalataink szerint ezen egyszerű szabályok alkalamzása a pontosság javulását eredményezte. Gyakran előfordult, hogy egy szöveg több leletet is tartalmaz, arról nem is beszélve, hogy más intézményeknél teljesen mások a szokások, ezért a szakaszok sorrendje nem elég jó indikátor.

4. Eredmények

Kísérleteink során három adathalmazon történt kiértékelés. Az *A_teszt* halmaz 100 darab [*A intézménytől*] származó, gerincről készült leletet tartalmaz, az *A_NG* halmazban 25 darab, szintén az [*A intézménytől*] származó csípő, mellkas vagy térd lelete található, a *B_teszt* csoportba pedig a [*B intézménytől*] származó gerincleletek kerültek. Kísérleteinkhez pontossági (accuracy) eredményeket közlünk, amelyek a helyesen címkézett mondatok arányát adják meg.

Az 1. táblázat felső szegmensében a tisztán szabályalapú módszerrel elért eredmények láthatók. Az *A_szabályok* alatt az [*A intézménytől*] kapott leletek elemzése során létrehozott szabályok értendők, míg a *B_szabályok* a [*B intézménytől*] kapott leletek elemzésével keletkeztek. Ezek is az első bekezdésében említett tesztadatokon lettek kiértékelve.

Az 1. táblázat második szegmensében látható eredmények azt mutatják, hogy milyen hatással volt a tanítóadat mennyiségének bővítése a különböző lelethalmazokon elért pontosságokra, valamint, hogy mekkora mértékű különbség van a csak a modellel előállított, és a heurisztikákkal javított modellkimenetek eredményei között. A táblázatban a modell_ kezdetű sorok csak a huBERT segítségével előállított modell, a hibrid_ kezdetű sorok pedig a huBERT segítségével előállított modell és az ezen alkalmazott szabályalapú utófeldolgozás eredményeit mutatják. Sorra 100, 300 majd 500 leletből álló tanítóadatot készítettünk. A táblázatból megállapítható, hogy a betanított modellek közül mindhárom tesztdhalmaz esetében az 500 leletből álló adaton finomhangolt modellel tudtuk elérni a legnagyobb pontosságot.

A következő lépésben az 500 leletből álló tanítóadatot módosítottuk két különböző módon, az így előállt modellek és a hibrid módszerek kiértékelését pedig a tanítóadathoz hasonlóan módosított, korábban ismertetett teszteken végeztük. Először a leletekből kivágásra kerültek azok a címkéző kifejezések, melyekre a szabályalapú módszerek is építenek, ilyen például az *anamnézis:, vélemény:, melléklelet:*. Az így előálló tanítóadaton kapott modell a *modell_500_1* elnevezést kapta. A másik kísérlet során a korábbi módosítás mellett nem különböztettük meg a részletes lelet és a vélemény címkét, hanem ezeket egy közös címke alá vettük. Ennek magyarázata, hogy általában ezek a valós orvosi megállapítások, a lelet valódi tartalma. A két szakasz mondatai sokszor szabad szemmel

1. táblázat. Különböző adathalmazokon mért pontosság a kialakított szabályalapú módszerrel, a modellel és a hibrid rendszerünkkel

	A_teszt	A_NG	B_teszt
A_szabályok	81,99	79,72	66,85
B_szabályok	0	0,35	93,26
modell_100	95,83	93,30	86,48
hibrid_100	97,74	95,60	87,32
modell_300	96,04	93,65	81,69
hibrid_300	97,81	96,47	81,13
modell_500	97,10	95,06	91,83
hibrid_500	98,23	97,00	92,96

sem különböztethetők meg. Az így készített modell a modell_500_2 elnevezést kapta.

A leírtak alapján előállított két korpusz eredményei a 2. táblázatban láthatók. Az 1. táblázat megfelelő soraival összehasonlítva látszik, hogy az így kapott *modell_500_1* modell, és az ehhez tartozó hibrid rendszer is alacsonyabb pontosság elérésére képes, míg a másik modell, melyben nem különböztettük meg a leletet és a véleményt az *A_teszt* halmazon jobb eredményt mutat.

2. táblázat. Az 500 leletből álló tanítóhalmazzal végzett különböző kísérletek

	A_teszt	A_NG
modell_500_1	95,34	90,12
modell_500_1_heur	96,54	90,83
modell_500_2	98,45	95,94
modell_500_2_heur	98,45	95,94

5. Az eredmények megvitatása

Bevezetésünkben három kérdés megválaszolását tűztük ki célul. Az első kérdés a mondatonkénti címkézéssel foglalkozott, ennek során megvizsgáljuk a tanítóadat változásának hatásait az eredményekre. Második kérdésünk a nem gerincleleteken, és más intézménytől származó leleteken végzett hatást tárgyalja. Harmadik kérdésünk tárgyát a hibrid és a szabályalapú módszerek összehasonlítása képezte.

Munkánk során egy hibrid módszert készítettünk, melynél azt vizsgáltuk, hogy képesek vagyunk-e olyan általános szabályrendszer megalkotására, amely javítani tudja a modell által elért eredményeket. A huBERT modell kifejezetten jól teljesített a feladaton, és az 1. tábázatból az látható, hogy néhány egyszerű

utófeldolgozási szabály alkalmazása szinte minden esetben közel 1%-os javulást eredményezett. Például a *modell_500* és a *hibrid_500* sorokat összehasonlítva a hibrid módszer sorra 1,13, 1,94 és 1,13%-os javulást eredményezett. Az eredményekből az is látható, hogy a tanító adathalmaz megfelelő bővítésével jól skálázódik a modell pontossága is. Mivel eltérő tanítóadat esetén a leválaszott 20% tesztadat is eltérne, a modelleket ugyanazon a 100 annotált leleten értékeltük ki, így valóban összehasonlíthatók ezek a számok. Ennek eredményei az 1. táblázat Tesztlelet oszlopához tartozó *modell_100, modell_300 és modell_500* sorokban láthatók. Kísérletünből kiderül tehát, hogy kevés tanítóadattal készített modellel is jó eredmények érhetők el, melyek tovább javíthatók a modell kimeneteinek elemzése után kialakított szabályokkal.

A 2. táblázat modell_500_1 soraiból az is látható, hogy a leletben lévő, orvosok által beírt címkézést figyeli a huBERT is, de ilyen esetekben is 96% feletti eredményt értünk el. Tehát a címkék figyelése nélkül is sikeresen szétválaszthatók a szakaszok.

Válasz az 1. kérdésre: A huBERT-alapú, mondatonként címkéző módszer utófeldolgozással kombinálva, 100 valós gerincleleten kiértékelve 98,23-as, a leletekben lévő címkék nélkül 96,54-es pontosságot produkált.

Második kérdésünk központjában az állt, hogy más adatokon is képes-e jól teljesíteni a módszer, melynek eredményei az 1. táblázat sorainak *A_NG* és *B_teszt* oszlopaiból olvashatók le. Az itt látható eredményeken azt láthatjuk, hogy módszerünkkel nem csak az [*A intézmény*] gerincleletein, hanem más régiókról készült leleteken, és más intézmény leletein is képes volt a strukturálásra. Érdekes, hogy a 300 leletes tanítás esetében a [*B intézmény*] leletein jelentős romlás látszik a kisebb tanítóhalmazhoz képest. Ez vizsgálatunk alapján valószínűleg annak tudható be, hogy a tanítóhalmaz random bővítésével csökkentettük az anamnézist tartalmazó címkék arányát a többi címkéhez képest, mely negatív hatással volt a [*B intézmény*] leleteire. Az 500 leletes tanításhoz a 300 leletet olyan leletekkel bővítettük, melyek tartalmaznak anamnézis és melléklelet szekciókat is, így javítva a [*B intézmény*] leleteinek strukturálását.

Válasz a 2. kérdésre: Módszerünk egy intézmény gerincleletein finomhangolva más anatómiai régiókon (97,00), és más intézmény gerincleletein (92,96) is nagy pontossággal működik, láthatóan jól általánosítva.

A harmadik kérdésre, a pusztán szabályalapú módszer eredményeit a modell és a hibrid eljárás eredményeivel összehasonlítva kapjuk meg a választ. Ehhez az 1. táblázat sorainak vizsgálata szükséges. Megfigyelhető, hogy az [*A intézménytől*] származó gerincleletekre kialakított szabályalapú *(A_szabályok)* módszer eredményei minden esetben elmaradnak még a legkevesebb adaton készített hibrid rendszer eredményeihez képest is. A [*B intézmény*] leleteinél az intézményhez kialakított szabályalapú módszert a modellnek nem sikerült felülmúlnia, de nagyon kis különbséggel megközelítette. Az is látható, hogy annak ellenére, hogy ebben az adathalmazban jobban strukturált leletek vannak, a 100%-os eredmény szabályokkal is elmarad, mivel rosszul címkézett, vagy címkézetlen leletek esetén a kialakított szabályok nem alkalmazhatók. A 0, és közel 0%-os eredményből lát-

szik, hogy más intézmény leletein, mely más címkéket tartalmaz, vagy teljesen strukturálatlan leletek esetén ez a szabályrendszer alkalmazhatalan.

Válasz a 3. kérdésre: Kísérleteink alapján az általános módszer megközelíti, vagy túl is szárnyalja az intézményekhez optimalizált, meglévő címkékre alapozó szabályokat.

Az utófeldolgozás, azaz a modell kimenetét a környezet alapján korrigáló szabályok sokkal egyszerűbben és gyorsabban létrehozhatók, mintha teljesen szabályalapon szeretnénk a leletek strukturálását végezni. Emellett egy modellből és szabályokból álló hibrid módszer jobb általánosításra képes, amennyiben más területet leíró illetve más felépítésű leletek strukturálására is szükség van. Az utófeldolgozás (hibrid módszer) az összes mérésnél javítást eredményezett, kivéve a modell_500_2 esetében, ahol az eredmény változatlan. Ez a kivétel azzal magyarázható, hogy a legtöbb jól javítható hiba a lelet és vélemény elkülönítésében volt, mivel ezek valóban nagyon hasonló mondatok, és ennél a halmaznál ezt a két címkét nem különböztettük meg. Megjegyezzük, hogy ez is a valós használatnak is megfelelő eset lehet, mert ezek a leletezés legfontosabb kimenetei, a módszer jól szegmentálja ezeket a szakaszokat a többitől.

6. Összegzés

Cikkünkben magyar nyelvű gerincleletek automatikus strukturálásával foglalkoztunk, a leletet mondatonkénti címkézéssel szakaszokra bontva. Kidolgoztunk két szabályalapú módszert intézményekre optimalizálva, valamint egy huBERT reprezentációt felhasználó modellt, és utófeldolgozással hibrid módszert készítettünk. Az eredmények azt mutatják, hogy a huBERT-alapú klasszifikációs modell 98,23% pontossággal állapítja meg a szakaszokat, jó eredményeket adva akkor is, ha az adat nem is tartalmaz címkéket az orvosoktól (96,54% ebben az esetben): A módszer jól általánosít más intézményekre (92,96%) vagy anatómiai régiókra (97.00%), illetve megközelíti, vagy felül is múlja az intézményekre optimalizált szabályalapú módszereket. A jelenlegi megoldás több fejlesztési lehetőséget kínál, mint például a tanítóadat más intézményekből származó leletekkel való bővítése, illetve a hibrid rendszer szabályainak további pontosítása. Munkánknak több felhasználása lehet, alkalmas például más kutatási/adatbányászati alkalmazások készítéséhez szükséges adatok tisztítására, vagy a leletek jobb rendszerezésére intézményeken belül.

Köszönetnyilvánítás

A kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében. A kutatás továbbá a Kulturális és Innovációs Minisztérium ÚNKP-22-4-SZTE-500 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.

Hivatkozások

Apostolova, E., Channin, D., Demner-Fushman, D., Furst, J., Lytinen, S., Raicu, D.: Automatic segmentation of clinical texts. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. vol. 2009, pp. 5905–5908 (09 2009)

Bosmans, J.M.L., Weyler, J.J., De Schepper, A.M., Parizel, P.M.: The radiology report as seen by radiologists and referring clinicians: Results of the cover and rover surveys. Radiology 259(1), 184–195 (Apr 2011)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. p. 4171–4186 (2019)

Ganeshan, D., Duong, P.A.T., Probyn, L., Lenchik, L., McArthur, T.A., Retrouvey, M., Ghobadi, E.H., Desouches, S.L., Pastel, D., Francis, I.R.: Structured reporting in radiology. Academic Radiology 25(1), 66–73 (Jan 2018)

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification (Aug 2016), arXiv preprint, arXiv:1607.01759

Kicsi, A., Pusztai, P., Szabó Ledenyi, K., Szabó, E., Berend, G., Vincze, V., Vidács, L.: Információkinyerés magyar nyelvű gerinc MR leletekből. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). p. 177–186. Szeged (2019)

Kicsi, A., Szabó Ledenyi, K., Pusztai, P., Németh, P., Vidács, L.: Entitások azonosítása és szemantikai kapcsolatok feltárása radiológiai leletekben. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020). p. 15–28. Szeged (2020)

Kuling, G., Curpen, B., Martel, A.L.: Bi-rads bert and using section segmentation to understand radiology reports. Journal of Imaging 8(5), 131 (May 2022)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (Sep 2013), arXiv preprint, arXiv:1301.3781

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021). p. 3–14. Szeged (2021)

Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations (Mar 2018), arXiv preprint, arXiv:1802.05365

Pomares-Quimbaya, A., Kreuzthaler, M., Schulz, S.: Current approaches to identify sections within clinical narratives from electronic health records: a systematic review. BMC Medical Research Methodology 19(1), 155 (Jul 2019)

Rosenthal, S., Barker, K., Liang, Z.: Leveraging medical literature for section prediction in electronic health records. In: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). p. 4863–4872. Association for Computational Linguistics, Hong Kong, China (2019)

Sadoughi, N., Finley, G.P., Edwards, E., Robinson, A., Korenevsky, M., Brenndoerfer, M., Axtmann, N., Miller, M., Suendermann-Oeft, D.: Detecting section boundaries in medical dictations: Toward real-time conversion of medical dictations to clinical reports. In: International Conference on Speech and Computer. p. 563–573. Springer International Publishing, Cham (2018)

Siklósi, B., Novák, A.: Rec. et exp. aut. Abbr. mnyelv. KLIN. szöv-ben – Rövidítések Automatikus Felismerése és Feloldása Magyar Nyelvű Klinikai Szövegekben. In: X. Magyar Számítógépes Nyelvészeti Konferencia. p. 167–176. Szeged (2014)

Siklósi, B., Novák, A.: Nem Felügyelt Módszerek Alkalmazása Releváns Kifejezések Azonosítására és Csoportosítására Klinikai Dokumentumokban. In: XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015). p. 237–248. Szeged (2015)

Siklósi, B., Novák, A.: Digitális Konzílium – Egy Szemészeti Klinikai Keresőrendszer. In: XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). p. 230–240. Szeged (2016)

Siklósi, B., Novák, A.: Restoring the intended structure of hungarian ophthalmology documents. In: Proceedings of BioNLP 15. pp. 152–157. Association for Computational Linguistics, Beijing, China (July 2015)

Taira, R.K., Soderland, S.G., Jakobovits, R.M.: Automatic structuring of radiology free-text reports. RadioGraphics 21(1), 237–45 (Jan 2001)

Tepper, M., Capurro, D., Xia, F., Vanderwende, L., Yetisgen-Yildiz, M.: Statistical section segmentation in free-text clinical records p. 2001–2008 (2012)

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A toolkit for morphological and dependency parsing of hungarian. In: Proceedings of RANLP. p. 763–771 (2013)

Fokozás szkizofréniában

Szabó Martina Katalin[1,2,3], Vincze Veronika[4], Guba Csenge[5], Dam Bernadett[6], Solymos Adrienn[7], Bagi Anita[7], Szendi István[2,8]

[1]Tokyo University of Foreign Studies, Institute of Global Studies
3-11-1, Asahi-cho, Fuchu-shi, Tokió 183-8534, Japán
[2]Szegedi Tudományegyetem, Informatikai Intézet
6720 Szeged, Árpád tér 2.
[3]Társadalomtudományi Kutatóközpont, CSS-RECENS
1097 Budapest, Tóth Kálmán utca 4.
[4]ELKH-SZTE Mesterséges Intelligencia Kutatócsoport
6720 Szeged, Tisza Lajos körút 103.
[5]Szegedi Tudományegyetem, Nyelvtudományi Doktori Iskola
6722 Szeged, Egyetem utca 2.
[6]Szegedi Tudományegyetem, Általános Nyelvészeti Tanszék
6722 Szeged, Egyetem utca 2.
[7]Szegedi Tudományegyetem, Magyar Nyelvészeti Tanszék
6722 Szeged, Egyetem u. 2-6.
[8]Kiskunhalasi Semmelweis Kórház, a Szegedi Tudományegyetem Oktató Kórháza, Pszichiátriai osztály
{martina,vinczev}@inf.u-szeged.hu
csenge.guba@gmail.com
dam.bernadett@stud.u-szeged.hu
solymos.adrienn98@gmail.com
bagianita88@gmail.com
szendi@inf.u-szeged.hu

Kivonat Dolgozatunkban szkizofrén betegek nyelvhasználati sajátságait vizsgáljuk a nyelvi fokozásra fókuszálva. Mivel a különböző mentális betegségek hatással vannak az egyén kommunikációjára, a nyelvi produktumaik (akár írott, akár beszélt formájú) elemzése fontos közvetett eszköze lehet a betegségek vizsgálatának és esetleges diagnosztizálásának. A jelen kutatásban a nyelvi sajátságok közül a fokozást tettük a fókuszba. Azt elemezzük, hogy a fokozás milyen kvantitatív és kvalitatív sajátságokkal jelenik meg a vizsgált betegek kommunikációjában. Korábbi érzelemkutatások alapján a szkizofréniával élőknél nehézség mutatkozik az érzelmi események megérzése, előrejelzése, az érzelmi benyomások és a kontextus integrálása, valamint az érzelmi élmények gazdagsága és fenntartása terén (Kring és mtsai, 2013). A háttérben álló agyi aktivitások a kognitív kontrollért felelős hálózatok működésében mutattak deficitet az érzelmek és a kogníció integrálásának elégtelenségét jelezve (Kring és mtsai, 2013). A mentális betegségek és az érzelemszabályozás ezen összefüggései okán, a fokozó elemek csoportján belül külön figyelmet fordítunk az ún. negatív emotív fokozókra, amelyek prior szemantikai tartalma valamely negatív érzelemhez kapcsolódik, azonban fokozó

elemekként funkcionálhatnak a nyelvben. A kutatás végső kérdése, hogy mutatkozik-e olyan eltérés a fokozók használatában a vizsgált betegek és a kontrollcsoport között, amely prediktív sajátságként felhasználható. Vizsgálatunkhoz beszélt nyelvi szövegeket használunk, amelyek három különböző narratív feladatban keletkeztek. Eredményeink megmutatják, hogy bár a nyelvi fokozás tekintetében a két csoport hasonlóan viselkedik, a negatív emotív fokozók használata jelentősen eltérő: a betegcsoport alig használ negatív emotív fokozókat a kontrollcsoport tagjaihoz képest, illetve rendre negatív kontextusban.

Kulcsszavak: fokozás, negatív emotív fokozók, beszélt nyelvi korpusz, magyar, szkizofrénia, mentális betegségek

1. Bevezetés

Ebben a dolgozatban szkizofrén (a továbbiakban csak: SZ) betegek nyelvhasználati sajátságait vizsgáljuk a fokozásra fókuszálva egy beszélt nyelvi korpusz alapján, számítógépes nyelvészeti (NLP) eszközök és módszerek segítségével. Bár a szakirodalomban találni említést arról, hogy a fokozók használata eltérést mutat szkizofréniában (Strous és mtsai, 2009), a jelenséget szisztematikusan még egyetlen szerző sem vizsgálta.

*Fokozók*nak tekintjük a nyelvben azokat a szavakat, amelyek képesek egy skálán felfelé vagy lefelé módosítani a melléknevek, igék, határozószók, ritkábban főnevek jelentését, pl. *nagyon nagy, elképesztő fejlődés, alig hallatszik* stb. (Bolinger, 2013; Méndez-Naya, 2008; Chang és Bin, 2020). A fokozás – a szoros nyelvészeti tárgyú vizsgálódás mellett – szociolingvisztikai és pszicholingvisztikai szempontból is érdekes problémakör. Több tényező, így a nem (Jespersen, 1922; Lakoff, 1973; Fuchs és Gut, 2012), az életkor (Fuchs és Gut, 2012), az iskolai végzettség (Xiao és Tao, 2007), az adott kultúra (Sardabi és Afghari, 2015), illetve a szociális státusz is befolyásolja a használatukat (Rasekh és Saeb, 2015). Athanasiadou (2007) alapján a fokozók a beszélői szubjektivitás nyelvi markerei, és egyenesen azok elsődleges funkciójának tekinti, hogy jelölik a beszélő nézőpontját, attitűdjét.

A szubjektivitás, illetve az érzelmek szövegszintű megjelenése szempontjából a fokozó elemek csoportján belül különösen érdekesek az ún. *negatív emotív fokozók* (a továbbiakban csak: NEF). Balogh (2009) megközelítésében a fokozók között vannak, amelyek elsődleges funkciójuk szerint töltik be a fokozó szerepet, például *nagyon, felettébb* stb., valamint azok, „amelyeknek van elsődleges jelentése, és csak másodlagosan szolgálnak fokozásra”, például *mértéktelenül, kimondhatatlanul* stb. A NEF-ek szempontjából számunkra az utóbbiak az érdekesek. Szabó és Bibok (2019) definíciójának megfelelően a NEF-ek lexikai szinten negatív szemantikai tartalommal rendelkeznek, ugyanakkor képesek más szavak fokozó szerepű módosítóiként funkcionálni, és ilyenkor ezt a tartalmat részben vagy egészben elveszítik (Szabó és Bibok, 2019). Ilyen például az *iszonyú* és a *rettenetesen* szó az *iszonyú szép* és a *rettenetesen ronda* kifejezésekben. A jelenség számos nyelvben adatolható, valószínűleg nyelvfüggetlen sajátság (Nemesi,

1998; Jing-Schmidt, 2007; Andor, 2003, 2011; Szabó és Bibok, 2019; Szabó és Otani, 2022).[1]

A fokozók, különösen a NEF-ek fenti sajátságai indokolják, hogy górcső alá vegyük azok viselkedését különböző mentális betegségek esetében. Ami kimondottan a SZ-betegeket illeti, azok a korábbi érzelemkutatások alapján az emóció érzékelésében és kifejezésében deficitet mutatnak, illetve agyi aktivitásuk az érzelmekkel kapcsolatosan eltér a kontrollszemélyek agyi aktivitásától (Kring és mtsai, 2013).

Az elemzéshez egy beszélt nyelvi korpuszt használtunk fel, amely 3 különböző feladattípus leiratait tartalmazza. A leiratokat magunk készítettük el a kutatómunka első lépéseként. A jelen dolgozathoz felhasznált korpuszunk összesen 341 szöveget tartalmaz.

A dolgozatban a fokozókra rendre FOK-ként, a negatív emotív fokozókra pedig a már fentebb ismertetetteknek megfelelően NEF-ként fogunk utalni. Az egyértelműség kedvéért azonban hangsúlyozzuk, hogy a NEF-ek – összhangban a korábbiakkal – ugyancsak fokozói szerepűek, hiszen a fokozóknak egy sajátos alcsoportját képezik. Ezzel összefüggésben a dolgozatban néhol a FOK és a NEF csoport elemeire együtt is utalunk, amikor a fokozás jelenségét, nyelvhasználati sajátságait általánososságban tárgyaljuk. Ilyenkor a „fokozó elem” vagy a „fokozószó” kifejezéseket használjuk, és ez alatt mind a FOK, mind pedig a NEF elemeket értjük, együttesen.

A dolgozatban a következő sajátságokat elemezzük: milyen a fokozó elemek használati gyakorisága a szkizofrén betegeknél a kontrollcsoporttal összehasonlítva? Van-e érdemleges eltérés a NEF-ek, valamint a FOK-ok gyakorisága tekintetében? Mely FOK-okat használják a leggyakrabban a szkizofrén betegek és a kontrollcsoport tagjai? Van-e eltérés abban a tekintetben, hogy milyen szófajú szavak módosítóinként használják a FOK-okat és a NEF-eket? Végül, van-e eltérés abban a tekintetben, hogy milyen szentimentértékű szavak fokozására használja a vizsgált szavakat a két csoport?

2. Kapcsolódó irodalom

2.1. A nyelvi fokozásról dióhéjban

A fokozók csoportja az egyik leggyorsabban változó elemkészlete a nyelvnek, és számos olyan szó létezik, amely képes fokozóként funkcionálni (Ito és Tagliamonte, 2003; Dressler és Barbaresi, 2011). Több altípusuk van, és a nemzetközi terminológia igen változatos képet mutat abban a tekintetben, hogy hogyan csoportosítja őket, illetve nevezi el az egyes csoportokat (pl. *intensive adverbs* ((Stoffel, 1901), idézi Bolinger (2013)), *intensives* (Ito és Tagliamonte, 2003), *amplifiers* és *downtoners* (Quirk és mtsai, 1985)). A jelen dolgozatban e problémakört nem tárgyaljuk, most csupán annyit említünk, hogy, mivel a fokozók

[1] Megjegyezzük, hogy Shmelev (2012) alapján az erős emocionális értékelést kifejező szavak gyakran nyernek ezzel ellentétes szemantikai tartalmat (vö. az enantioszémia jelensége, bővebben l. pl. Klégr (2013); Szabó (2019)).

alapvetően „lefelé és felfelé módosítják” a szavak jelentését, mindenekelőtt *növelők* (an. *amplifiers*) és *csökkentők* (an. *downtoners*) osztályára bonthatóak (Fuchs és Gut, 2016; Strohm és Klinger, 2018). A jelen kutatásban kizárólag az előbbi, azaz a növelő értelmű fokozókkal foglalkozunk. Közöttük találhatóak a NEF-ek is.

E helyütt a NEF-ekre vonatkozó szakirodalom részletes áttekintésére nincs mód, ezért csak néhány, a jelen kutatás szempontjából releváns sajátságról teszünk említést. Egyrészt, a magyar nyelv NEF-eivel foglalkozó tanulmányok azok prior szemantikai tartalma kapcsán az érzelmek közül kizárólag a félelmet említik (Nemesi, 1998; Kugler, 2014); Nemesi (1998) szerint például ezek a kifejezések „az ijesztés, ijedés képzetkörébe tartoznak”. Ugyanakkor Szabó és Guba (2022) alapján a NEF-ek lexikai szintű szemantikai tartalma helyenként nem a félelemmel, hanem az undor, a düh vagy a bánat érzelemmel függ össze, sőt, némely esetben egyszerre több emócióval is. Másrészt, a NEF-ek fokozóvá válását sajátos időbeli szemantikai változás jellemzi. E folyamat során a „kezdetben igen expresszív jelentésű fokozó kifejezések” fokozatosan elveszítik lexikai jelentésüket, azaz deszemantizálódnak (Lorenz (2002), idézi Dér (2015)), és grammatikalizálódnak is, hiszen új grammatikai szerephez jutnak: fokozóvá válnak (Dér, 2015). Ugyanakkor fokozó szerepük ellenére szótári szinten nem veszítik el prior szemantikai tartalmukat, hiszen a negatív emotív jelentés megfelelő kontextusban (sőt gyakorta magában a fokozó szerepben is) előhívható. Így például a *borzasztóan vékony* kapcsolat annak ellenére, hogy benne a *borzasztóan* fokozó szereppel bír (l. *nagyon*), egyben kifejezésre juttathatja a beszélő szubjektív értékelő véleményét, attitűdjét is (bővebben l. Dér (2015); Szabó (2018)). Ugyanakkor ez a szemantikai-pragmatikai „többlettartalom” fokozatosan csökken az újabb kollokációs mintázat hatására: mivel egyre gyakrabban kapcsolódik össze ellentétes, azaz pozitív töltetű elemekkel az adott NEF, annak affektív értéke folyamatosan elhomályosodik. Azaz „minél kiterjedtebbé válik az alkalmazásuk (...), annál ártalmatlanabbak is lesznek az érzelmi működésünk szempontjából” (Kugler, 2014).

2.2. A szkizofrén betegek nyelvhasználatával kapcsolatos kutatások

A pszichotikus rendellenességek, köztük a szkizofrénia jellemző tünete a kognitív működés zavara, amely a verbális és non-verbális képességeket egyaránt érinti (Van Rheenen és mtsai, 2016; Little és mtsai, 2019). Mivel a mentális egészség befolyásolja az emberi kommunikáció módját, a nyelvi adatok feldolgozása és elemzése lehetőséget nyújthat a nyelv és a psziché közötti kapcsolat feltárására, sőt a mentális állapotok megismerésére, a mentális és agyi működés progresszív károsodásának azonosítására a nyelven keresztül.

Az utóbbi évtizedekben megszaporodott azoknak a vizsgálatoknak a száma, amelyek különböző mentális rendellenességek esetében a nyelvi produkció és a metakogníció közötti kapcsolat mérésére igyekeznek felhasználni az újabb elemzési technológiák (NLP és gépi tanulás) kínálta lehetőségeket, azokat klinikai és nem klinikai környezetben egyaránt alkalmazva (Calvo és mtsai, 2017). A területen belül kiemelt figyelmet kapott a SZ-kutatás (Mitchell és mtsai, 2015). Csupán

néhányat említve a dolgozatok közül, Corcoran és mtsai (2018) szerint például a számítógépes nyelvészeti módszerek segíthetnek a pszichózis fennállásának prediktálásában. Willits és mtsai (2018) SZ-val diagnosztizált személyek hangzó szövegeit elemzik olyan sajátságok után kutatva, amelyek eltérnek az egészséges kontrollszemélyek nyelvi sajátságaitól, majd ezek alapján próbálják meg automatikusan klasszifikálni a szövegeket. Mitchell és mtsai (2015) SZ-betegek Twitter-bejegyzéseit vizsgálják, és gépi tanulás (SVM) segítségével sikerült a betegeket 82,3%-os pontossággal elkülöníteniük. Mindezek mellett Minor és mtsai (2019) kutatása egyenesen arra mutat, hogy az NLP-megközelítés eredményesebb lehet a klinikai vizsgálatokban alkalmazott skáláknál. Összehasonlítva az automatikus elemzés eredményeit a klinikusok minősítéseivel ugyanis azt találták, hogy az előbbi sikeresebben mutatott ki szignifikáns neurokognitív és metakognitív eltéréseket.

Ami a magyar anyanyelvű betegeket illeti, ugyancsak számos dolgozat vizsgálja a SZ-val élők spontán beszédét (pl. Kéri és mtsai (2001); Réthelyi és mtsai (2010); Inczédy-Farkas és mtsai (2010); Kocsis-Bogár és mtsai (2016); Döme és mtsai (2005)). A kutatások többsége a szövegek temporális sajátosságait vizsgálta (Gosztolya és mtsai, 2018; Bagi és mtsai, 2019), de végeztek kutatásokat a rekurzió előfordulására irányulóan is (Kárpáti és mtsai, 2018). Ezek mellett a metafora és irónia megértését is vizsgálták SZ-val élő betegeknél (Bagi és mtsai, 2016).

A szövegalapú vizsgálatok egyik legfontosabb elemzési eszközét az NLP-módszerekkel és eszközökkel elemezhető beszélt nyelvi szövegkorpuszok jelentik. Az elmúlt évtizedekben több, pszicholingvisztikai vizsgálatokat támogató beszélt nyelvi korpusz keletkezett több nyelven is (pl. Calvo és mtsai (2017); Corcoran és mtsai (2018); Little és mtsai (2019), és közöttük magyar nyelvet reprezentálókat is találunk (pl. Gosztolya és mtsai (2018); Vincze és mtsai (2021); Kálmán és mtsai (2022). A SZ tekintetében többek között Minor és mtsai (2019), Willits és mtsai (2018) és Bagi és mtsai (2019) korpuszait említhetjük meg.

3. A jelen vizsgálat módszere és eszközei

Kutatásunkban hangzó szövegek leiratait NLP-módszerekkel és eszközökkel vizsgálva azt igyekszünk feltárni, hogy milyen, kifejezetten nyelvhasználati sajátságokban tér el a szkizofrén betegcsoport a kontrollcsoporttól. A nyelvi sajátságok közül a jelen dolgozatban egyetlen sajátságra, a fokozásra fókuszálunk.

3.1. A korpusz bemutatása

Az elemzéshez egy beszélt nyelvi korpuszt használtunk fel, amely a Szegedi Tudományegyetemen működő, Szendi István által vezetett Mentális Betegségek Megelőzése Interdiszciplináris Kutatócsoport által készített hangfelvételeket tartalmaz.[2] Az adatbázis különféle mentális betegségekben szenvedők, valamint kontrollszemélyek spontán beszédfelvételeit tartalmazza. Spontán beszéd esetében a

[2] Az adatrögzítést a Szegedi Tudományegyetem Etikai Bizottsága hagyta jóvá, és az a Helsinki Nyilatkozatnak megfelelően történt. A kutatók a kutatási projektben részt

tervezett beszéddel ellentétben a beszélőknek nincs idejük a beszéd előkészítésére, ily módon a nyelvi sajátosságok valódi megismerésére adnak lehetőségek ezek a nyelvi anyagok.

A jelen munka során a korpusz azon részeire fókuszálunk, amely a szkizofrén betegek és a kontrollcsoport három különböző feladattípusban keletkezett nyelvi produktumait tartalmazzák. Az első feladat (a továbbiakban: Narr1) három részből állt. Az interjúztató először arra kérte az adatközlőt, hogy beszéljen saját magáról, majd megkérte, hogy mutassa be néhány szóban az édesanyját, illetve az édesapját. Néhány esetben előfordult, hogy az adatközlő nem szeretett volna valamelyik szülőről beszélni, ilyenkor egy másik, hozzá közel álló személyt választhatott a monológja tárgyául. A következő feladat (a továbbiakban: Narr2) első részében az adatközlő feladata az volt, hogy néhány mondatban elevenítse fel tanulmányai utolsó vagy a munkába állásának első éveit. Ezt követően az interjúztató azt kérte, hogy mutassa be egy hozzá közel álló személynek ugyanezt az időszakát. A harmadik feladat (a továbbiakban: TegnN) egyetlen részből állt: az adatközlőnek „a tegnapi napját”, azaz az interjút megelőző napját kellett összefoglalnia.

A hanganyagok átiratait magunk készítettük el a kutatómunka első lépéseként úgy, hogy azokat manuálisan leiratoztuk. Mivel kutatásunk nem fonológiai tárgyú, a szövegek egyszerű írott formája megfelelt a számunkra a hangzó sajátságok jelölése, valamint speciális szoftver használata nélkül. Ugyanakkor fontosnak találtuk, hogy jelöljük a megakadásokat és újraindításokat, téves szókezdéseket, valamint a néma és kitöltött szüneteket is.

A felvételeket egyszerű szövegfájlokba, UTF-8 kódolással jegyeztük le. Összesen 341 önálló leiratot készítettünk, ebből 183 tartozott a SZ, 158 pedig a kontrollcsoporthoz. A kutatásban összesen 58 adatközlő vett részt, ebből 31 SZ, 27 pedig kontroll személy volt. A demográfiai adatokat az 1. táblázat mutatja be.

	Csoportok		
	Kontroll	**SZ**	**mind**
Adatközlők	27	31	58
Szövegek	158	183	341
Életkor; *M(SD)*	36,28(10,3)	38,00(9,78)	37,20(9,84)
Iskolázottság; *M(SD)*	14,67(2,99)	14,26(2,86)	14,45(2,90)
Nemi arány; nő:férfi	13:14	10:21	23:35

1. táblázat. A két csoport demográfiai adatai (M = átlag; SD = szórás).

Amint a táblázat alapján is látható, a két csoport demográfiai adatai kiegyensúlyozottak. Az életkor és az elvégzett tanévek átlaga szinte megegyezik, az elemszámok között pedig csekély az eltérés. A két csoport életkorának és

vevő összes adatközlőtől írásos beleegyezést kaptak, továbbá a felvételek kutatásban történő felhasználására a kutatóknak hivatalos írásbeli engedélye van.

iskolázottságának összevetésére Mann-Whitney próbát alkalmaztunk, amely kimutatta, hogy a két csoport között nincs szignifikáns eltérés sem az életkor ($p = 0{,}399$), sem az elvégzett tanévek száma ($p = 0{,}572$) esetében.

3.2. A korpusz elemzése

A leiratokat alapvető számítógépes nyelvészeti feldolgozásnak vetettük alá (tokenizálás, lemmatizálás, morfológiai elemzés, szófaji egyértelműsítés és dependenciaelemzés) a magyarlanc eszköz (Zsibrita és mtsai, 2013) segítségével. Mivel a vizsgált fokozók kollokációs sajátságait is vizsgálni szerettük volna, a stopszavakat és központozást nem töröltük. Az alapvető feldolgozást követően egy általunk készített FOK-szótár, valamint egy 225 elemű NEF-szótár (Szabó és Guba, 2022) szavait automatikusan a szóalakokra illesztettük. A FOK-szótárat egy korábbi szótárból (Szabó és mtsai, 2022) a jelen munkához magunk fejlesztettük tovább, a következőképpen: a meglevő szóanyagot kézi vizsgálatok során tovább bővítettük, majd az így kialakított teljes anyagot hárman, külön-külön végigellenőriztük és valós példák alapján döntést hoztunk arról, hogy valóban fokozónak kellene-e tekintenünk az adott szót vagy sem. Végül csupán azokat a szavakat tartottuk meg a szótárban, amelyekről konszenzusra jutottunk (a szótár elérhetőségéről l. lentebb).

A szótárakat azért a szóalakokra illesztettük a lemmák helyett, mert a szótáraink is különböző szóalakokat tartalmaznak, és nem csupán az azokhoz tartozó lemmákat (pl. a *borzasztó* és a *borzasztóan* egyaránt előfordul benne). Ennek az az oka, hogy a kutatás során a különböző szóalakok korpuszbeli sajátságait külön-külön is vizsgáljuk, tehát azokat az egyes lemmákban nem kívánjuk automatikusan összevonni (Szabó és Guba, 2022). Azokat a mondatokat, amelyekben legalább egy találat volt, listáztuk. Mivel az eredmény nagyon zajos volt, kézzel egyértelműsítenünk kellett a további vizsgálatok elvégzése előtt. Ez a következőt jelentette: a NEF esetében minden találatot egyenként megvizsgáltunk és döntést hoztunk arról, hogy az adott NEF az adott kontextusban fokozó szerepben áll, vagy sem. Ez a munka rövid időn belül elvégezhető volt az összes NEF-találat esetében, azonban a FOK-ok találatainak nagy száma miatt azokat szűkítenünk kellett a kézi ellenőrzés előtt. Ennek céljából a gépi szófaji egyértelműsítés kimenetére támaszkodtunk: mivel a fokozók jelentős része melléknevet, határozószót vagy igét módosít (Ito és Tagliamonte, 2003), a közvetlen kollokátumok szófaja alapján nagy mennyiségű irreleváns adatot ki tudunk szűrni, könnyebbé és gyorsabbá téve az ezt követő kézi elemzést. Összesen 1460 előfordulást találtunk, ahol a vizsgált szó a fenti szófajok valamelyikével állt a korpuszban; ezeket az eseteket annotáltuk kézzel.

Arra is kíváncsiak voltunk, hogy milyen szentimentértékű szavakat módosítanak a vizsgált szavak. Ez utóbbi feladathoz tehát egy szentimentszótárra volt szükségünk, és az elemzési eredménynél a kvantitatív sajátságokra fókuszáltunk, azaz gyakorisági adatokat gyűjtöttünk az egyes kollokációkról. Kollokációelemzésünkhöz két, korábban publikált magyar nyelvű szentimentszótárat egyesítettünk (Liu (2012) magyar nyelvű fordítása, valamint Szabó (2015)).

A szentimentszótárat, akárcsak a fokozószótárakat egyszerű szótárillesztéses megoldással alkalmaztuk a korpuszon. A szentimentelemzést ugyanakkor már a lemmákon végeztük el, ugyanis ez lehetőséget adott a számunkra ahhoz, hogy a szótári lemmák minden egyes toldalékolt alakját megtalálhassuk a korpuszban (l. pl. az *aggasztó* lemma *aggasztóak* és *aggasztót* szóalakjai).

4. Eredmények

A kontrollcsoport feldolgozott alkorpusza központozás nélkül összesen 60279 tokent, a SZ-csoporté pedig 34670 tokent tartalmaz.

A kézi annotáció után a két csoportban összesen 1046 FOK és 68 NEF maradt, amelyről azt állapítottuk meg, hogy valóban fokozó funkciója van az adott esetben.

Tekintsük először a FOK-okra vonatkozó elemzési eredményeket! A FOK-ok számát és gyakoriságát a 2. táblázat mutatja be. Az „arány” alatt azt adjuk meg, hogy az összes szótáras találatból milyen arányban találtunk ténylegesen fokozó funkciójú szót.

csoport	**össz token**	**össz/valódi FOK**	**össz/valódi FOK%**	**arány%**
SZ	34670	450/357	1,3/1,03	79,33
Kontroll	60279	972/689	1,61/1,14	70,88

2. táblázat. A FOK-okra vonatkozó alapvető adatok.

A táblázat adatai alapján azt látjuk, hogy a kontrollcsoport valamivel magasabb arányban használja a FOK-szótárunk szavait, és ez igaz akkor is, ha leszűkítjük az adatokat a kézzel egyértelműsített előfordulásokra (Kontroll: 1,61% és 1,14%, SZ: 1,3% és 1,03%). Ez alapján azt mondhatjuk, hogy a kontrollcsoportra jellemzőbb a FOK-ok használata, bár az eltérés nem kiugróan magas.

Nézzük meg, mely FOK-okat használják az adatközlők a leggyakrabban a két csoportban! A 3. táblázatban azt láthatjuk, hogy mely az öt leggyakrabban előforduló FOK a két csoportban, valamint milyen azok gyakorisági megoszlása az összes FOK számához viszonyítva. Itt most csupán a kézi egyértelműsítés eredményeit vizsgáljuk.

A leggyakoribb FOK-okat illetően a két csoportban nincs jelentősebb eltérés, illetve olyan különbség, amely bizonyosan ne lenne magyarázható egyéni nyelvhaszálati eltérésekkel. Ugyanakkor ha összevetjük a két táblázat gyakorisági arányokra vonatkozó adatait, azt láthatjuk, hogy a *nagyon* FOK a SZ-csoport esetében nagyobb frekvenciával szerepel a többi FOK-nál a kontrollcsoport adataival összevetve. Úgy is fogalmazhatunk, hogy a kontrollanyag az egyes FOK-ok használata tekintetében kiegyenlítettebb képet mutat, lényegében nyelvhasználatuk ebben a tekintetben valamelyest változatosabb a SZ-csoporténál.

Vizsgáljuk meg, hogy a FOK-ok milyen szentimentértékű szavakat módosítanak a korpuszban! Ahogyan korábban ismertettük, a feladathoz szentimentszó-

SZ	-	-	**Kontroll**	-	-
FOK	**szám**	**gyak%**	**FOK**	**szám**	**gyak%**
nagyon	247	0,69	nagyon	408	0,59
elég	22	0,06	elég	65	0,09
annyira	22	0,06	annyira	51	0,07
igazán	14	0,04	teljesen	30	0,04
olyan	12	0,03	olyan	30	0,04

3. táblázat. A leggyakoribb öt FOK a két csoportban.

tárat használtunk, amelynek segítségével a fokozó szavak közvetlen kollokátumainak szentimentértékét dolgoztuk fel. Ahhoz, hogy az eredményeket értékelni tudjuk, nézzük meg azt is, hogy milyen a szentimentértékek száma és gyakorisági megoszlása a két csoportban a teljes korpuszra nézve (azaz függetlenül a fokozás jelenségétől)! Az adatokat, mind összességében, mind a FOK-ra nézve a a 4. táblázat mutatja be alább.

Csoport	**POS**	**POS%**	**NEG**	**NEG%**	**NEUT**	**NEUT%**
SZ_all	2299	6,63	943	2,72	31428	90,65
SZ_fok	190	53,22	52	14,57	115	32,21
Kontroll_all	3609	5,99	1471	2,44	55199	91,57
Kontroll_fok	340	49,35	80	11,61	269	39,04

4. táblázat. A teljes korpusz, valamint az alaptagok szentimentelemzési eredményei a két csoportban a FOK esetében.

A tábla adatai alapján megállapítható, hogy mind a pozitív, mind a negatív töltetű szavak a SZ-csoportban a gyakoribbak, és ugyanez a megoszlás jellemzi a FOK-ok melletti alaptagok értékének gyakorisági megoszlását is. Általánosságban elmondhatjuk tehát, hogy bár a FOK + pozitív töltetű szó kombináció, valamint a FOK + negatív töltetű szó kombináció gyakrabban szerepel a SZ-csoport monológjaiban, a SZ-sok egyébként is nagyobb arányban használnak szentimentszavakat a kontrollcsoport tagjainál; a pozitív és negatív szentimentértékű kifejezések használata nem változik a FOK-okkal összefüggésben.

Végül nézzük meg azt is, hogy a FOK-ok melletti alaptag három szófaja, azaz a melléknév, a határozószó és az ige milyen megoszlást mutat a két csoportban. Az adatokat az 5. táblázat mutatja be. Az „arány” azt mutatja meg, hogy milyen az egyes szófajok gyakorisága a kollokátumok összes, a két alkorpuszban előforduló szófajához képest.

Az eredményekből az látszik, hogy a SZ-csoport jelentősen több igét fokoz, mint a kontrollcsoport; a kontrollszemélyek több melléknevet és határozószót. Ugyanakkor ha megnézzük a három vizsgált szófaj gyakorisági megoszlását a teljes korpuszban (a fokozástól függetlenül), azt látjuk, hogy a SZ-csoport eleve több igét használ a kontrollcsoportnál: a SZ-soknál az igék előfordulása összesen 7123, ami az összes szófaj 20,54%-át teszi ki, míg a kontrollszemélyeknél ez

	SZ		**Kontroll**	
POS	**szám**	**arány%**	**szám**	**arány%**
ADJ	167	0,47	344	0,5
VERB	131	0,37	195	0,28
ADV	59	0,17	150	0,22

5. táblázat. Az alaptagok szófaji megoszlása a két korpuszban.

a szám 11018, ami az összes szófaj 18,28%-a. A határozószóknál hasonló jelenséget látunk; a kontroll használ többet belőlük nem csupán a FOK-ok mellett, hanem összességében is. Azonban a melléknevekénél már mást tapasztalunk; belőlük alapvetően a SZ-csoport használ valamivel többet (SZ: 2532, 7,3%, kontroll: 4278, 7,1%), a FOK-ok módosított alaptagjaként mégis, amint fentebb láttuk, a kontrollcsoportban jelennek meg valamivel gyakrabban. Az eltérés lehetséges magyarázatához további vizsgálatokat tartunk szükségeknek.

Tekintsük most a NEF-re vonatkozó adatokat! A NEF-ek számát és gyakoriságát a 6. táblázat mutatja be. Itt is közöljük mind az egyszerű szótáras elemzési, mind a kézi egyértelműsítési eredményeket. Az „arány” alatt itt is, mint a FOK-ok esetében tettük, azt adjuk meg, hogy az összes szótáras találatból milyen arányban találtunk ténylegesen fokozó funkciójú szót.

csoport	**össz token**	**össz NEF**	**valódi NEF**	**össz NEF%**	**valódi NEF%**	**arány%**
SZ	34670	20	5	0,06	0,01	25
Kontroll	60279	48	28	0,08	0,05	58,33

6. táblázat. A NEF-ekre vonatkozó alapvető adatok.

Ha az egyszerű szótáras elemzés eredményeit vizsgáljuk, azt látjuk, hogy a SZ-csoport tagjai a teljes tokenszámhoz viszonyítva kevesebb szót használnak a NEF-szótárunk elemei közül, mint a kontrollcsoport tagjai. Ugyanakkor ez az eredmény még nem elég informatív, lévén, hogy ezek a szavak alapvetően poliszémek, tehát ez az eredmény nem jelenti azt, hogy fokozó szerepben is ritkábban használják őket a SZ-betegek. Azonban ha a kézi annotáció eredményeit vizsgáljuk, jelentős aránybeli eltérést látunk: a betegcsoport kimondottan ritkán használ NEF-eket (valóban fokozó szerepben) a kontrollcsoport tagjaihoz képest: a teljes korpuszban összesen 5 előfordulást találtunk, ami a tokenszámhoz viszonyítva 0,01-ot tesz ki. Az arány, amelyet a teljes előfordulás és a valódi fokozói használat adatából számoltunk ki jelentős nyelvhasználati eltérésre mutat rá; amíg a kontrollszemélyek NEF-előfordulásainak többsége (valamivel több mind 58,33%-a) valóban fokozói szerepben áll, addig a SZ-csoport esetében ez mindössze 25%-ot tesz ki. Bár az adatmennyiség nem engedi meg a messzemenő következtetéseket, azért érdemes megjegyezni, hogy eredményeink egy irányba mutatnak azokkal a kutatási eredményekkel, amelyeket a SZ-betegek figuratív nyelvhasználattal kapcsolatban végzett vizsgálatai tártak fel (l. többek között

Rossetti és mtsai (2018)). Azt reméljük tehát, hogy tapasztalataink adalékul szolgálnak ezekhez a vizsgálatokhoz is.

Ami a kollokátumok szentimentértékét illeti, bár rendkívül kevés adattal dolgozunk, azért figyelemre érdemes, hogy a SZ-csoport esetében mind az öt adat negatív alaptagot módosít: *kegyetlenül rossz, iszonyú érvágás, hulla fáradt* (kétszer), valamint *rettenetes hányinger*. Ezzel szemben a kontrollcsoport korpuszában a szentimentértékek megoszlása kiegyenlített, majdnem azonos gyakorisággal módosítanak a NEF-ek pozitív, semleges és negatív szavakat; közöttük 10 pozitív (pl. *önzetlen, jó, tehetséges*), 9 semleges (pl. *nagy, sok, erős*) és 9 negatív (pl. *idegesítő, szabályellenes, utál*) szót találunk. E vizsgálati tapasztalatok különösen figyelemre méltóak, ha meggondoljuk a NEF-ek már ismertetett sajátságait azok időbeli szemantikai változási folyamataikra vonatkozóan (tudniillik, hogy prior negatív tartalmuk elhalványul, l. fentebb). Azt látjuk ugyanis, hogy a SZ-csoport esetében a NEF-ek negatív jelentéskomponense nyilvánvalóan befolyásolja azok kollokációs preferenciáját; prior szemantikai tartalmuk visszaköszön kollokációs kapcsolódási mutatóikban (l. még: kongruencia jelensége, Dilts (2010)).

Együtt tekintve a két vizsgált elemcsoportot megállapíthatjuk, hogy bár a FOK-oknál nem láttunk jelentős eltérést a SZ- és a kontrollcsoport között (eltekintve az egyes FOK-ok gyakorisági megoszlásától), a NEF-ek esetében van érdemi különbség. Ez alapján úgy véljük, a NEF-ek további vizsgálata fontos és hasznos lehet a mentális betegségek nyelvi sajátságainak kutatásában.

5. Összegzés

Ebben a dolgozatban, egy beszélt nyelvi korpuszt felhasználva azt vizsgáltuk, hogy a nyelvi fokozás tekintetében milyen eltérések fedezhetőek fel a SZ és a kontrollcsoport nyelvhasználata között. A fokozó elemek csoportján belül külön figyelmet fordítottunk a negatív emotív fokozókra (amelyekre a dolgozatban NEF-ekként utaltunk, ezzel elkülönítve azokat a másik szótárunk, a FOK-lista elemeitől).

Legfontosabb eredményeink alapján a következő megállapításokat tettük: A kontrollcsoport FOK-használata az egyes elemek gyakoriságát tekintve kiegyenlítettebb képet mutat, azaz valamelyest változatosabb a nyelvhasználatuk a SZ-nél. Ezen kívül azonban semmilyen fontosabb eltérést nem mutattak az adatközlői adatok. A NEF-ek használata azonban fontos aránybeli eltérést tárt fel: a betegcsoport ritkán használ NEF-eket a kontrollcsoport tagjaihoz képest, és ezekben az esetekben is rendre negatív környezetben. Bár ismételten hangsúlyozzuk, hogy az adatmennyiség nem engedi meg a messzemenő következtetéseket, az eredményeink összecsengenek a SZ-betegek figuratív nyelvhasználattal kapcsolatos jellemzőivel, és reményeink szerint adalékul szolgálhatnak az arra vonatkozó kutatásokhoz.

A munka következő lépéseként a vizsgálatainkat szeretnénk elvégezni a korpusz két másik betegcsoportjának szövegein is, azaz a bipoláris és a szkizoaffektív

betegek monológjainak alkorpuszain, majd összevetni az eredményeket az itt feltárt sajátságokkal.

A kutatásunkhoz készített FOK-szótárat kutatási célra szabadon hozzáférhetővé tesszük az alábbi linken: github.com/SzaboMartinaK/Hungarian-Intensifiers. A NEF-ek listáját l. a következő dolgozatban: Szabó és mtsai (2022). A korpuszt a kutatómunka zárását és az anonimizálást követően ugyancsak elérhetővé tesszük további kutatási célokra.

Köszönetnyilvánítás

A kutatást, amelyet a Szegedi Tudományegyetem és a Társadalomtudományi Kutatóközpont (Eötvös Loránd Kutatási Hálózat, MTA Kiváló Kutatóhely) valósított meg, az EFOP-3.6.1-16-2016-00008 a. sz. pályázat Szkizofrénia – Nyelv – Innovatív gyógyítás című alprojektje, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

Szabó Martina Katalin kutatásait a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal OTKA posztdoktori kiválósági programja (NKFI-azonosító: 132312), valamint az International Research Fellowship Program of Japan Society for the Promotion of Science (JSPS, Postdoctoral Fellowships for Research in Japan, Standard) támogatta.

Hivatkozások

Andor, J.: De durva ez a téma!–megfigyelések a melléknévi polaritásváltásról. Hungarológiai Évkönyv 12(1), 33–42 (2011)

Andor, J.: Functional studies in the polarity and gradation of amplifier adjectives and adverbs in english. Studies in English theoretical and applied linguistics pp. 43–59 (2003)

Athanasiadou, A.: On the subjectivity of intensifiers. Language sciences 29(4), 554–565 (2007)

Bagi, A., Gosztolya, G., Szalóki, S., Szendi, I., Hoffmann, I.: Szkizofrénia azonosítása spontán beszéd temporális paraméterei alapján–egy pilot kutatás eredményei. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) Magyar Számítógépes Nyelvészeti Konferencia 15. pp. 189–201. Szegedi Tudományegyetem, Informatikai Intézet (2019)

Bagi, A., Hoffmann, I., Szendi, I., Toth, L.: Metaphor in a Different Way: The Understanding of Metaphor and Irony in Schizophrenia, pp. 7–18. Palacký University, Olomouc (01 2016)

Balogh, P.: Gender-markerek a nyelvben. Wiener Elektronische Beiträge des Instituts für Finno-Ugristik (WEBFU) pp. 1–7 (2009)

Bolinger, D.: Degree words. De Gruyter Mouton (2013)

Calvo, R.A., Milne, D.N., Hussain, M.S., Christensen, H.: Natural language processing in mental health applications using non-clinical texts. Natural Language Engineering 23(5), 649–685 (2017)

Chang, Y., Bin, S.: Literature Review on Negative Emotive Intensifiers in English. Higher Education of Social Science 18(2), 55–59 (2020)

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), 67–75 (2018), https://onlinelibrary.wiley.com/doi/abs/10.1002/wps.20491

Dér, Cs.I.: Iszonyatosan/rettenetesen/őrületesen jó! A fokozószók grammatikalizációjáról. In: Gecső, T., Sárdi, Cs. (szerk.) Az interkulturális kommunikáció elmélete és gyakorlata. pp. 71–76. Kodolányi János Főiskola – Tinta, Székesfehérvár–Budapest (2015)

Dilts, P.: Good nouns, bad nouns: what the corpus says and what native speakers think. In: Corpus-linguistic applications, pp. 103–117. Brill Rodopi (2010)

Döme, P., Rihmer, Z., Gonda, X., Pestality, P., Kovács, G., Teleki, Z., Mandl, P.: Cigarette smoking and psychiatric disorders in Hungary. International Journal of Psychiatry in Clinical Practice 9(2), 145–148 (2005)

Dressler, W.U., Barbaresi, L.M.: Morphopragmatics: Diminutives and intensifiers in Italian, German, and other languages, vol. 76. Walter de Gruyter (2011)

Fuchs, R., Gut, U.: Do women use more intensifiers than men?-Investigating gender and age-specific language use with the International Corpus of English. ICAME332012 (2012)

Fuchs, R., Gut, U.: Register variation in intensifier usage across Asian Englishes. Discourse-pragmatic variation and change: Insights from English pp. 185–210 (2016)

Gosztolya, G., Bagi, A., Szalóki, S., Szendi, I., Hoffmann, I.: Identifying schizophrenia based on temporal parameters in spontaneous speech. In: Proc. Interspeech 2018. pp. 3408–3412. International Speech Communication Association (ISCA) (2018)

Inczédy-Farkas, G., Benkovits, J., Balogh, N., Álmos, P., Scholtz, B., Zahuczky, G., Török, Z., Nagy, K., Réthelyi, J., Makkos, Z., és mtsai: Schizobank – the Hungarian national schizophrenia biobank and its role in schizophrenia research. Orvosi Hetilap 151(35), 1403–1408 (2010)

Ito, R., Tagliamonte, S.: Well weird, right dodgy, very strange, really cool: Layering and recycling in English intensifiers. Language in society 32(2), 257–279 (2003)

Jespersen, O.: Language: Its nature, development and origin, vol. 68. H. Holt (1922)

Jing-Schmidt, Z.: Negativity bias in language: A cognitive-affective model of emotive intensifiers. Cognitive Linguistics 18(3), 417–443 (2007)

Kálmán, J., Devanand, D.P., Gosztolya, G., Balogh, R., Imre, N., Tóth, L., Hoffmann, I., Kovács, I., Vincze, V., Pákáski, M.: Temporal speech parameters detect mild cognitive impairment in different languages: validation and comparison of the speech-gap test® in English and Hungarian. Current Alzheimer Research 19(5), 373–386 (2022)

Kárpáti, E., Bagi, A., Szendi, I., Tóth, L.B., Janacsek, K., Hoffmann, I.: Rekurzió egy szkizoaffektív zavarral élő személy diskurzusaiban–esettanulmány. ISKOLAKULTÚRA: PEDAGÓGUSOK SZAKMAI-TUDOMÁNYOS FOLYÓIRATA 28(5-6), 40–54 (2018)

Kéri, S., Kelemen, O., Benedek, G., Janka, Z.: Different trait markers for schizophrenia and bipolar disorder: a neurocognitive approach. Psychological medicine 31(5), 915–922 (2001)

Klégr, A.: The limits of polysemy: enantiosemy. Linguistica Pragensia 23(2), 7–23 (2013)

Kocsis-Bogár, K., Nemes, Z., Perczel-Forintos, D.: Factorial structure of the hungarian version of oxford-liverpool inventory of feelings and experiences and its applicability on the schizophrenia-schizotypy continuum. Personality and Individual Differences 90, 130–136 (2016)

Kring, A.M., Elis, O., és mtsai: Emotion deficits in people with schizophrenia. Annual review of clinical psychology 9(1), 409–433 (2013)

Kugler, N.: A nyelvi polaritás kifejezésének egy mintázata, avagy milyen a félelmetesen jó? Magyar Nyelvőr 138(2), 129–139 (2014)

Lakoff, R.: Language and woman's place. Language in society 2(1), 45–79 (1973)

Little, B., Gallagher, P., Zimmerer, V., Varley, R., Douglas, M., Spencer, H., Çokal, D., Deamer, F., Turkington, D., Ferrier, I.N., és mtsai: Language in schizophrenia and aphasia: the relationship with non-verbal cognition and thought disorder. Cognitive Neuropsychiatry 24(6), 389–405 (2019)

Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)

Lorenz, G.: Really worthwhile or not really significant? a corpus-based approach to the delexicalization. New reflections on grammaticalization 49, 143–163 (2002)

Méndez-Naya, B.: Special issue on English intensifiers. English Language & Linguistics 12(2), 213–219 (2008)

Minor, K.S., Willits, J.A., Marggraf, M.P., Jones, M.N., Lysaker, P.H.: Measuring disorganized speech in schizophrenia: automated analysis explains variance in cognitive deficits beyond clinician-rated scales. Psychological Medicine 49(3), 440–448 (2019)

Mitchell, M., Hollingshead, K., Coppersmith, G.: Quantifying the language of schizophrenia in social media. In: Proceedings of the 2nd workshop on Computational linguistics and clinical psychology: From linguistic signal to clinical reality. pp. 11–20 (2015)

Nemesi, A.: Társalgási divatkifejezések a benyomáskeltés szolgálatában. Magyar Nyelvor 122, 24–35 (1998)

Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: A comprehensive English grammar. London and New York: Longman (1985)

Rasekh, A.E., Saeb, F.: Gender differences in the use of intensifiers in Persian. International Journal of Applied Linguistics and English Literature 4(4), 200–204 (2015)

Réthelyi, J.M., Bakker, S.C., Polgár, P., Czobor, P., Strengman, E., Pásztor, P.I., Kahn, R.S., Bitter, I.: Association study of nrg1, dtnbp1, rgs4, g72/g30, and pip5k2a with schizophrenia and symptom severity in a Hungarian sample. American Journal of Medical Genetics Part B: Neuropsychiatric Genetics 153(3), 792–801 (2010)

Rossetti, I., Brambilla, P., Papagno, C.: Metaphor comprehension in schizophrenic patients. Frontiers in Psychology 9, 670–685 (2018)

Sardabi, N., Afghari, A.: Gender differences in the use of intensifiers. Journal of Applied Linguistics and Language Research 2(7), 203–213 (2015)

Shmelev, A.: Cognitive and communicative sources of enantiosemy. In: Cantero, C. (szerk.) Proceedings of the 10th World Congress of the International Association for Semiotic Studies (IASS/AIS). Spain: Universidade da Coruña. pp. 837–844. Universidade da Coruña (2012)

Stoffel, C.: Intensives and down-toners: A study in English adverbs. C. Winter's universitätsbuchhandlung (1901)

Strohm, F., Klinger, R.: An empirical analysis of the role of amplifiers, downtoners, and negations in emotion classification in microblogs. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 673–681. IEEE (2018)

Strous, R.D., Koppel, M., Fine, J., Nachliel, S., Shaked, G., Zivotofsky, A.Z.: Automated characterization and identification of schizophrenia in writing. The Journal of nervous and mental disease 197(8), 585–588 (2009)

Szabó, M.K.: Egy magyar nyelvű szentimentlexikon létrehozásának tapasztalatai és dilemmái. In: Segédkönyvek a nyelvészet tanulmányozásához 177, pp. 278–285. Tinta, Budapest (2015)

Szabó, M.K.: A szentimentérték módosulásának a problémája a magyar nyelvű szövegek szentimentelemzésében, különös tekintettel az értékvesztésre és az értékváltásra. Ph.D.-értekezés, SZTE (2018)

Szabó, M.K.: Az értékváltás jelensége a magyar nyelvben. A negatív emotív elemek egy sajátos használatáról. Magyar Nyelv 115(3), 309–323 (2019)

Szabó, M.K., Bibok, K.: Értékvesztésre és értékváltásra képes lexémák újabb vizsgálata. Argumentum 15, 639–649 (2019)

Szabó, M.K., Guba, Cs.: Analysis of Negative Emotive Intensifiers in Hungarian Tweets (2022), kézirat

Szabó, M.K., Otani, N.: Corpus Analysis of Japanese and Hungarian Negative Emotive Words from a Discourse Interactional Perspective. The 25th Annual Conference of the Pragmatics Society of Japan The 25th Annual Meeting of Pragmatic Society of Japan (2022), Kyoto, Japan

Szabó, M.K., Vincze, V., Bibok, K.: "Thank you for the terrific party!" – An Analysis of Hungarian Negative Emotive Words, with Special Regard to The Phenomenon of Polarity Shift. Corpus Linguistics and Linguistic Theory (2022), in press

Szabó, M.K., Vincze, V., Ring, O., Guba, Cs.: Nagyot mondó képviselők? : fokozás a politikai kommunikációban. In: Magyar Számítógépes Nyelvészeti Konferencia 18. pp. 325–341 (2022)

Van Rheenen, T.E., Bryce, S., Tan, E.J., Neill, E., Gurvich, C., Louise, S., Rossell, S.L.: Does cognitive performance map to categorical diagnoses of schizophrenia, schizoaffective disorder and bipolar disorder? a discriminant functions analysis. Journal of Affective Disorders 192, 109–115 (2016)

Vincze, V., Szabó, M.K., Hoffmann, I., Tóth, L., Pákáski, M., Kálmán, J., Gosztolya, G.: Linguistic parameters of spontaneous speech for identifying mild

cognitive impairment and alzheimer's disease. Computational Linguistics pp. 1–34 (2021)

Willits, J.A., Rubin, T., Jones, M.N., Minor, K.S., Lysaker, P.H.: Evidence of disturbances of deep levels of semantic cohesion within personal narratives in schizophrenia. Schizophrenia Research 197, 365–369 (2018), https://www.sciencedirect.com/science/article/pii/S0920996417307107

Xiao, R., Tao, H.: A corpus-based sociolinguistic study of amplifiers in British English. Sociolinguistic studies 1(2), 241–273 (2007)

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In: Proceedings of RANLP. pp. 763–771 (2013)

Sclerosis multiplex felismerése spontán beszédből wav2vec 2.0 modellekből kinyert jellemzőkkel

Gosztolya Gábor[1,2], José Vicente Egas-López[1], Svindt Veronika[3], Bóna Judit[4], Hoffmann Ildikó[3,5]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]ELKH-SZTE Mesterséges Intelligencia Kutatócsoport, Szeged
[3]ELKH Nyelvtudományi Kutatóközpont, Budapest
[4]Eötvös Loránd Tudományegyetem,
Alkalmazott Nyelvészeti és Fonetikai Tanszék, Budapest
[5]Szegedi Tudományegyetem, Pszichiátriai Klinika
ggabor @ inf.u-szeged.hu

Kivonat A sclerosis multiplex (SM) a központi idegrendszer krónikus gyulladásos megbetegedése. Mivel az SM többek között az alanyok beszédét is befolyásolja, az automatikus beszédelemzés egyszerű, relatíve olcsó és találkozásmentes (távoli) módot kínálhat a beszédprodukció változásainak detektálására. Egy ilyen automatikus elemző eljárás fejlesztésének során azonban kritikusnak bizonyulhat, hogy milyen jellemzőket nyerünk ki a beszédproduktumból. Cikkünkben tíz wav2vec 2.0 modell segítségével számítunk jellemzőket, az így kapott osztályozási eredményeket pedig nagymennyiségű adaton tanított publikus, valamint kevesebb, de magyar nyelvű adaton magunk által tanított x-vektor neurális hálók használatával kapott eredményekkel is összevetjük. Kísérleteinkben a többnyelvű fonetikus készletre tanított wav2vec 2.0 modellek hatékonyabbnak bizonyultak, mint az alap („base") modellek. A legfontosabb attribútumnak ugyanakkor a modell paraméterszáma tűnik: a legjobb eredményt az egymilliárd tanítható paraméterrel bíró modell adta. Emellett azt találtuk, hogy a modell finomhangolása a célnyelvre (esetünkben a magyarra) javít az eredményeken, ugyanakkor (legalábbis kísérleti eredményeink alapján) más nyelvre finomhangolni nem érdemes. Meglepő módon nem sikerült viszont túlszárnyalnunk az x-vektorok teljesítményét, mely véleményünk szerint valószínűleg a keretszintű beágyazások bevett, de talán túlságosan egyszerű felvételszintű aggregációjának tudható be.
Kulcsszavak: sclerosis multiplex, wav2vec 2.0, beágyazások

1. Bevezetés

A sclerosis multiplex (SM) a központi idegrendszer krónikus gyulladásos megbetegedése. A motorikus képességek különféle károsodásai az SM egyik központ diagnosztikai jellemzőinek számítanak, míg a motorikus képességek változásai utalhatnak a beteg állapotának súlyosbodására is (Szirmai, 2006). Mivel a nyelvi, kognitív és motorikus funkciók az agyban egy elválaszthatatlan hálózatot

alkotnak, egyik tényező változása változást idézhet elő az összes többiben is. Emiatt az alanyok beszédprodukciójának változásának automatikus ellenőrzése, követése egy hatékony eszközt nyújthat egy SM beteg állapotának vizsgálatához, a betegség előrehaladásának monitorozásához is.

Az SM alanyok mintegy 60-70%-a számol be valamilyen kognitív károsodásról (csökkent információföldolgozási sebességről, krónikus fáradtságról vagy tájékozódási zavarokról). Emellett az SM betegek több mint egyharmadánál jelentkezik átmeneti vagy tartós beszédzavar is (Laakso és mtsai, 2000; Renauld és mtsai, 2016), ami szintén az automatikus beszédelemzés potenciális hasznosságát támasztja alá. A leggyakoribb nyelvi- és beszédtünetek a motoros beszédzavarok (dizartria, diszfónia), szótalálási nehézségek, a verbális fluencia csökkenése (Delgado-Álvarez és mtsai, 2021), mondatismétlési problémák és a magasabb szintű nyelvi folyamatok korlátozottsága (Darley és mtsai, 1972; Hartelius és mtsai, 2000). Az automatikus beszédelemzés használata fényt deríthet olyan tünetekre is, melyek a dizartria szintjét még el nem érő motorikus beszédzavarra utalhatnak (Mulfari és mtsai, 2021).

Az elmúlt évtizedben igen népszerű kutatási területté vált a valamilyen mentális vagy szervi betegséggel élők beszédének automatikus elemzése, számos tanulmány foglalkozott például Alzheimer-kórral (Pérez-Toro és mtsai, 2022), Parkinson-kórral (Moro-Velazquez és mtsai, 2020) vagy éppen depresszióval (Jenei és Kiss, 2020) élők beszédének automatikus földolgozásával. Mivel ezekben a problémákban egyetlen alany felel meg egy (gépi tanulási értelemben vett) példának, az alacsony elemszám miatt jellemzőbb Support Vector Machine (SVM) használata az osztályozási lépésre, míg a mély neurális hálók (Deep Neural Networks, DNN) inkább a jellemzőkinyerés során kapnak szerepet (Pérez-Toro és mtsai, 2022; Jenei és mtsai, 2022). Az ilyen orvosi jellegű korpuszok mérete ugyanakkor még azt sem igazán teszi lehetővé, hogy a jellemzőkinyerésre használt DNN-eket a vizsgált alanyok hangfelvételein tanítsuk, így ez a lépés a legtöbbször általános célú beszédadatbázisokon történik. Az így előálló DNN-ek úgy is tekinthetők, mint a „szokásos beszéd” valamiféle modelljei; a konkrét jellemzőkinyerési lépés pedig annak felelhet meg, hogy a „szokásos beszéd” és az aktuális alany hangfelvétele közötti különbséget számszerűsítjük.

Jelen dolgozatunkban nagyméretű korpuszokon előtanított wav2vec 2.0 neurálisháló-modellekkel végzünk kísérleteket. Ezeket a modelleket sclerosis multiplex és kontroll alanyok beszédfelvételein értékeljük ki, majd a kapott aktivációkból (*embedding*) számított felvételszintű jellemzők alapján, osztályozó eljárással próbáljuk besorolni a beszélőket a két kategória valamelyikébe. Mivel wav2vec 2.0 modellekből máris számos variáció érhető el, tíz különböző modellt tesztelünk: az általános „alap” (*base*) modellek mellett többnyelvi (*cross-lingual*) hálókat (melyek több nyelv fonetikai készletének felismerésére lettek tanítva) és egy-egy konkrét nyelvre finomhangolt modelleket is kipróbálunk. Az elért eredményeket előtanított és saját x-vektor neurális hálók (Snyder és mtsai, 2018) által kinyert jellemzők használatával kapott eredményekkel is összevetjük, melyek szintén népszerű jellemzőkinyerő módszernek számítanak az orvosi beszédfeldolgozás területén (ld. pl. Moro-Velazquez és mtsai, 2020; Jenei és mtsai, 2022).

2. Hangfelvételek

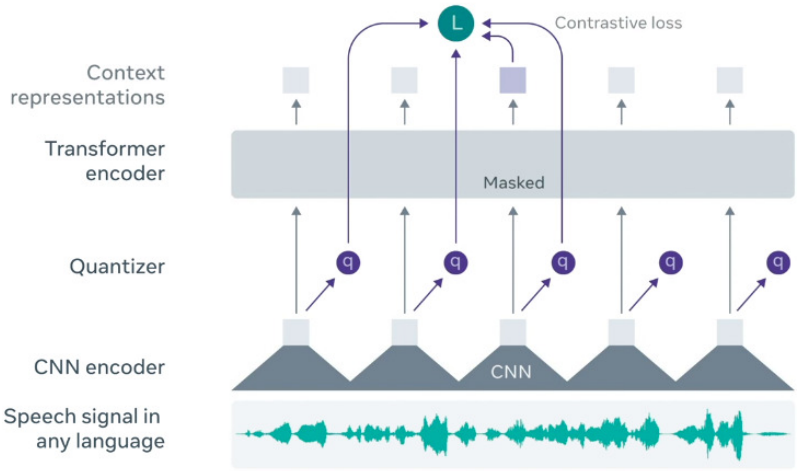
A vizsgálatokra a budapesti Uzsoki Utcai Kórház Neurológiai Osztályán és az Eötvös Loránd Kutatóhálózat Nyelvtudományi Kutatóközpontjában került sor. A vizsgálatot az Uzsoki Utcai Kórház etikai bizottsága hagyta jóvá, és a Helsinki Nyilatkozatnak megfelelően végeztük el. Kísérleteinket 23 SM alany (18 nő és 5 férfi) és 22 kontroll beszélő (16 nő és 6 férfi) felvételein végeztük. Az SM alanyok mindegyike a relapszáló-remittáló (relapsing-remitting, RRMS) altípusba tartozott. A két csoport tagjainak demográfiai jellemzőit az életkor és az iskolázottság esetében ANOVÁ-val, a beszélők nemének eloszlását χ^2-próbával vizsgáltuk; a két csoport tagjai egyik vizsgált jellemzőjükben sem tértek el statisztikailag szignifikáns mértékben (azaz $p > 0,05$).

A felvételi protokoll során összesen 17 különféle feladatot rögzítettünk az alanyokkal. Jelen tanulmányunkban kísérleteinket a *szövegösszefoglalás* feladat hangfelvételein végeztük; ennek során az alanyoknak egy kétperces, számukra korábban ismeretlen tudományos ismeretterjesztő szöveg meghallgatása után minél pontosabban el kellett azt mesélniük. A feladat végrehajtása számos kognitív funkció (figyelem-összpontosítás, munkamemória, időbeli orientáció, rendszerezés) összehangolt működését igényli (Mar, 2004). Az alanyok válaszait egy Sony PCM-A10 digitális diktafonnal, valamint csíptetős mikrofonnal rögzítettük. Az eredetileg sztereó, 48 kHz mintavételű felvételeket a feldolgozás előtt 16 kHz mintavételezésű, monó formátumra konvertáltuk.

3. Wav2vec 2.0

A beszédfelismerő rendszerek tanításához jellemzően nagymennyiségű címkézett (azaz szöveges átirattal is rendelkező) tanítóadat szükséges (Amodei és mtsai, 2016). Ennek áthidalására nyújt egy hatékony megoldást az, ha a rendelkezésre álló, de címkézetlen adaton előtanítunk egy mély neurális hálót (jellemzően felügyelet nélküli vagy önfelügyelt (self-supervised) módon). Ennek a hálónak a feladata gyakorlatilag egy hatékony reprezentáció megtanulása lesz közvetlenül a beszédjelből. Az így betanított háló súlyai ezután alkalmasak lehetnek arra, hogy egy másik DNN súlyait inicializáljuk velük; ezt a második hálót már a célfeladatra fogjuk tanítani (amelyhez általában korlátozott, de legalábbis lényegesen kisebb mennyiségű tanítóadat áll rendelkezésre, mint amit az első lépésben használhattunk).

A **wav2vec** architektúra egy konvolúciós neurális háló, melynek bemenete a nyers beszédhang-jel, kimenete pedig egy olyan reprezentáció, amely alkalmas egy beszédfelismerő rendszer bemenetének. Tanítása során a hálót az aktuális hangfelvétel következő szegmensének predikciójára tanítjuk (Schneider és mtsai, 2019); vegyük észre, hogy ehhez a fajta tanításhoz nem szükséges semmiféle annotáció (önfelügyelt tanítás). A **wav2vec 2.0** architektúra ezen a megközelítésen azzal javított, hogy a tanítás során – a modell zajtűrésének javítása érdekében – *maszkolást* (a bemenet egyes részeinek kinullázását) is alkalmazott. További különbség a *kontrasztív tanítás* (*contrastive learning*) használata: a tanítás során



1. ábra: Egy előtanított wav2vec 2.0 modell struktúrája. Forrás: `https://ai.facebook.com/blog`

a modell célja annak felismerése is, hogy két különbözőféleképpen transzformált reprezentáció (esetünkben a kontextualizált réteg kimenete, valamint a konvolúciós réteg kimenetének diszkretizált (kvantált) formája) azonos bemenethez tartozik-e.

A wav2vec 2.0 architektúra az 1. ábrán látható. Három részét különböztethetjük meg: a bejövő, nyers audiójelet konvolúciós rétegek (CNN) dolgozzák föl; kimenetüket látens reprezentációnak (*latent representation*) szokás nevezni. Ezt a reprezentációt transzformer rétegek dolgozzák föl, így kapjuk a kontextualizált reprezentációt (*contextualized representation*). Ebből a kimenetet a finomhangolással (*fine-tuning*), azaz a célfeladatra tanítással kapott lineáris projekciós réteggel kapjuk (Baevski és mtsai, 2020). Az önfelügyelt tanítási lépés során kizárólag az alsó két blokkot, tehát a konvolúciós és a kontextualizált rétegeket tanítjuk, míg az utolsó két blokk tanítása jellemzően egy beszédfelismerési feladaton történik (általában egy Connectionist Temporal Classification (CTC) veszteségfüggvénnyel, közvetlenül az átiratra). (Látható tehát, hogy a kontextualizált blokkot mind az előtanítás, mind a finomhangolási lépés során tanítjuk.)

3.1. Többnyelvi reprezentációk

Az XLSR (Cross-lingual Speech Representations) a wav2vec 2.0 többnyelvi továbbfejlesztése (Conneau és mtsai, 2021). Az XLSR megközelítésben az előtanítás többnyelvű korpuszon (vagy több, eltérő nyelvű adatbázison) történik. A különböző nyelvekhez tartozó tanítópéldák az előtanítás során egy-egy batchben is vegyesen (és az adott nyelvhez rendelkezésre álló tanítópéldák gyakoriságát fi-

gyelembe véve) fordulnak elő. Így a konvolúciós blokk utáni kvantálási lépésben előálló beszéd-reprezentációkat több nyelv modellezése során is használja, amely fokozott mértékben jelentkezik rokon nyelvek esetében (Conneau és mtsai, 2021). Az XLSR előtanítás használata kimondottan hatékonynak bizonyult olyan nyelvek esetén, amelyekre még címkézetlen adatból is kevés áll rendelkezésre.

3.2. Jellemzőkinyerés wav2vec 2.0 hálókból

Egy wav2vec 2.0 struktúrájú hálóból beágyazásokat kinyerni két ponton kézenfekvő: egyrészt a konvolúciós blokk utolsó rétegéből (erre *konvolúciós* jellemzőkként fogunk hivatkozni), másrészt pedig a második (a kontextualizált) blokk utolsó rejtett rétegéből (erre a névhasonlóságot elkerülendő nem *kontextualizált*, hanem *rejtett* aktivációként fogunk hivatkozni). Mindkét beágyazásvektorsorozat hordozhat fontos információt mind a beszélőről (Lin és Mak, 2020), mind a beszédjelben jelen lévő további jelenségekről (Fan és mtsai, 2021). Technikai problémát okoz ugyanakkor, hogy a wav2vec 2.0 architektúrát beszédfelismerési célból fejlesztették ki, így a beágyazásokból nyert (keretszintű) jellemzővektorok száma arányos a hangfelvétel hosszával. Hogy felvételszintű osztályozásra alkalmassá tegyük őket, valamiféle aggregációs stratégiát kell alkalmaznunk. Erre a szokásos, bevált megközelítést követtük: az időtengely mentén kiátlagoltuk a vektorokat, illetve az egyes értékek szórását is kiszámítottuk.

4. Kísérleti paraméterek

A wav2vec 2.0 modellek beágyazásait a HuggingFace keretrendszer segítségével nyertük ki. A jellemzőkinyerési lépés után a beszélőket SVM (Schölkopf és mtsai, 2001) alkalmazásával soroltuk a két beszélőcsoport (SM és kontroll) valamelyikébe, a libSVM csomagot (Chang és Lin, 2011) használva. Korábbi tapasztalataink alapján (ld. pl. Egas-López és Gosztolya, 2021; Gosztolya és mtsai, 2022) lineáris kernelt használtunk; az így adódó egyetlen hiperparamétert (C, complexity) a $10^{-5}, 10^{-4}, \ldots, 10^{1}$ értékek közül választottuk ki. Beágyazott keresztvalidációt alkalmaztunk (Cawley és Talbot, 2010); minden csoportban (foldban) egy-egy SM beteg és egy kontroll alany volt (egy fold kivételével, amely egyetlen SM betegből állt), így 23 csoportot kaptunk. A C hiperparamétert minden tanítás esetén egy további (belső, 22-szeres) keresztvalidációs lépés segítségével választottuk ki, a legjobb ROC görbe alatti terület (AUC) érték alapján. A modellek összehasonlítására is az AUC metrikát használtuk.

5. Eredmények

5.1. Az alap wav2vec 2.0 modellekkel kapott eredmények

Első kísérleteinkben alap wav2vec 2.0 modelleket alkalmaztunk. Az első ilyen modell a `wav2vec2-base` (Baevski és mtsai, 2020) volt, melyet a LibriSpeech

wav2vec 2.0 modell	Beágyazás	Átlag	Átlag + szórás
wav2vec2-base	Konvolúciós	0,745	0,781
	Rejtett	0,731	0,802
wav2vec2-base-960h	Konvolúciós	0,745	0,781
	Rejtett	0,785	0,731
wav2vec2-large-960h	Konvolúciós	0,700	0,795
	Rejtett	0,771	0,763

1. táblázat. Az alap wav2vec 2.0 modellekkel kapott AUC értékek.

korpusz kb. 53 ezer órányi hangfelvételén előtanítottak, a finomhangolási lépés nélkül. A másik két tesztelt modellnek (`wav2vec2-base-960h` és `wav2vec2--base-960h`) mind az előtanítása, mind a finomhangolása 960 órányi (címkézett) adaton történt. Az utóbbi két modell között a fő különbség a tanítható paraméterek száma: a *base* modell 95 millió, míg a *large* modell 317 millió paraméterrel rendelkezik. Az ezen modellek felhasználásával kapott osztályozási eredmények az 1. táblázatban láthatóak.

A kapott AUC értékek versenyképesnek számítanak ugyan, viszont, mivel általában a $0,745 \ldots 0,802$ intervallumba esnek, semmiképpen sem kiemelkedőek. A kipróbált két felvételszintű aggregációs stratégia esetében a szórásértékek használata javította az osztályozási eredményeket (tehát az AUC értékek magasabbak lettek, mint amelyeket kizárólag az átlagvektorokkal kaptunk); ez alól kivételt képeztek a finomhangolt modellek (tehát `base-960h` és `large-960h`) kontextualizált („rejtett”) beágyazásai. A két tesztelt beágyazástípus hasznosságát vizsgálva azt találjuk, hogy az „alap” modell (`wav2vec2-base`) esetében kontextualizált reprezentációk használatával kicsit jobb eredményeket kaptunk, mint a konvolúciós blokk kimeneteire támaszkodva. Ugyanakkor a két finomhangolt (960h) modell esetén a konvolúciós rétegek hasonló eredményre vezettek, mint korábban, míg a „rejtett” reprezendációk teljesítménye nagymértékben romlott az „Átlag + szórás” esetben. Ennek oka valószínűleg az, hogy a finomhangolási lépés eredményeképpen ezek a rétegek jobban fókuszálnak a beszéd fonetikai tartalmára, így aktivációik kevésbé voltak alkalmasak az SM detektálására.

5.2. A többnyelvi modellekkel kapott eredmények

Következőleg a többnyelvi modelleket vizsgáltuk meg. Az `XLSR-53` modell 53 különböző nyelvű (kb. 56 ezer órányi) adaton lett előtanítva, míg az XLSR utódja (*XLS-R*) esetén Babu és munkatársai körülbelül félmillió órányi felvételt használtak erre (mely 128 nyelvet ölelt föl) (Babu és mtsai, 2022). Három különböző méretű modell érhető el; technikai okokból eltekintünk a legnagyobb modell használatától (mely kétmilliárd súlyból áll), és a két kisebbre koncentrálunk (`wav2vec2-XLS-R-300M` 300 millió, míg `wav2vec2-XLS-R-1B` egymilliárd tanítható paramétert tartalmaz). A 2. táblázatban találhatóak az ezen modellek használatával kapott AUC-értékek.

wav2vec 2.0 modell	Beágyazás	Átlag	Átlag + szórás
XLSR-53	Konvolúciós	0,700	0,729
	Rejtett	0,763	0,787
XLS-R-300M	Konvolúciós	0,741	0,743
	Rejtett	0,773	0,706
XLS-R-1B	Konvolúciós	0,670	0,696
	Rejtett	0,862	0,872

2. táblázat. A többnyelvű wav2vec 2.0 modellekkel kapott AUC értékek.

Meglepő módon, a konvolúciós rétegek aktivációinak használatával alacsonyabb AUC értékeket értünk el, mint az alap wav2vec 2.0 modellek esetén ($0,670 \ldots 0,743$). Az utolsó rejtett rétegből nyert beágyazások esetén az `XLSR-53` modellel hasonló eredményeket kaptunk, mint a `wav2vec2-base` modell esetén, és valamivel jobbakat, mint a két finomhangolt („960h") alap modellel (főleg amikor a beágyazások szórásait is fölhasználtuk). Ez alapján a többnyelvi wav2vec 2.0 megközelítés hatékonyabbnak tűnik, mint az egynyelvű előtanítás és finomhangolás, legalábbis mikor a célfeladatunk SM alanyok és egészséges kontrollok megkülönböztetése. Az `XLS-R` megközelítés nem bizonyult lényegesen hatékonyabbnak, mint az „eredeti" `XLSR-53`, legalábbis a 300 millió súlyból álló modell esetén. Ezzel szemben az egymilliárd paraméteres, nagyobb modell láthatólag jobb eredményekhez vezetett: a rejtett rétegek aktivációinak átlagát és szórását is fölhasználva $0,872$-es AUC-értéket értünk el.

5.3. Egy nyelvre finomhangolt XLSR modellekkel elért eredmények

A következő lépésben arra voltunk kíváncsiak, hogyan alakulnak az osztályozási eredményeink, ha egy konkrét nyelvre finomhangolt XLSR-53 modellt használunk a jellemzőkinyerési lépés során. Négy nyelvet (és így négy modellt) vizsgáltunk: mivel az SM adatbázisunk nyelve is magyar, első fölhasznált modellünk magyar nyelvű volt. A következő vizsgált modell (a nyelvrokonság miatt) a finn nyelvre finomhangolt XLSR-53 modell volt. A fennmaradó két nyelvet úgy választottuk, hogy azokra (relatíve) nagymennyiségű tanítóadat álljon rendelkezésre, így egy német és egy spanyol nyelvre finomhangolt wav2vec 2.0 hálót teszteltünk még. Mind a négy XLSR-53 modellt ugyanaz a csapat finomhangolta (`jonatasgrosman`) a Mozilla Common Voice 6.1 adatbázis megfelelő részhalmazán, ehhez 8 (magyar), 1 (finn), 777 (német) és 579 (spanyol) órányi adatot használva. Mivel a finomhangolás nem érintette a háló konvolúciós blokkját, csak a kontextualizált („rejtett") rétegből vett beágyazásokból nyert jellemzőkkel kapott osztályozási eredményeket adjuk meg.

A 3. táblázatban olvashatók a nyelvfüggő, finomhangolt wav2vec 2.0 modellekkel kapott eredmények (viszonyításként az eredeti XLSR-53 modellel elért értékeket is feltüntettük). A magyar modell egyértelműen jelentős javuláshoz

wav2vec 2.0 modell	Átlag	Átlag + szórás
XLSR-53	0,763	0,787
XLSR-Hungarian-53 (magyar)	0,816	0,820
XLSR-Finnish-53 (finn)	0,749	0,767
XLSR-German-53 (német)	0,791	0,783
XLSR-Spanish-53 (spanyol)	0,761	0,737

3. táblázat. A finomhangolt többnyelvű wav2vec 2.0 modellek kontextualizált („rejtett") rétegéből nyert beágyazásokkal kapott AUC-értékek.

x-vektor modell	Jellemzők	AUC
SRE-16 (angol)	MFCC	0,876
Saját (magyar)	MFCC	0,850
	FBANK	0,793
	Spektrogram	0,798

4. táblázat. Az x-vektor jellemzőkkel kapott AUC-értékek.

vezetett. Ugyanakkor a fennmaradó három modell esetében nem találtunk jelentős különbséget az eredeti többnyelvi XLSR-53 modellhez képest: a keretszintű beágyazásvektorok átlagával $0,749 \ldots 0,791$-es AUC-értékeket kaptunk (XLSR-53: $0,763$), míg a szórás aggregációt is használva $0,737 \ldots 0,783$-at (XLSR-53: $0,787$). Ez véleményünk szerint azt jelzi, hogy az alkalmazástól eltérő nyelvre finomhangolni az eredeti XLSR-53 wav2vec 2.0 modellt nem igazán kifizetődő, viszont a célnyelvre adaptálása hatékonyabb SM-detektáláshoz vezethet még akkor is, ha a célnyelvi tanítóadat mennyisége kimondottan csekély (esetünkben mindössze 8 órányi). Ez a stratégia mind az „Átlag" aggergáció esetén (ahol az AUC-érték $0,763$-ról $0,816$-ra nőtt), mind az „Átlag + szórás" esetben (ahol pedig $0,787$-ről $0,820$-ra emelkedett) tapasztalható.

5.4. Összevetés az x-vektor jellemzőkkel elért eredményekkel

Utolsó kísérletünkben, hogy az elért osztályozási eredményeket kontextusba helyezzük, összevetjük azokat az x-vektor jellemzőkinyerő eljárással kapottakkal. Ehhez egyrészt Snyder és munkatársai szabadon elérhető modelljét használjuk (`SRE-16 Snyder és mtsai, 2018)`, mely a Switchboard (28 ezer felvétel) és a NIST SRE (63 ezer felvétel) korpuszok egy részén lett betanítva, másrészt saját x-vektor hálókat is tanítottunk a BEA Spontánbeszéd-adatbázis (Neuberger és mtsai, 2014) egy részhalmazán (165 beszélő, 60 órányi hangfelvétel). Utóbbi esetben keretszintű jellemzőként MFCC-t, FBANK-ot és spektrogramot is kipróbáltunk. Az SM-detektálási feladaton kapott AUC-értékek a 4. táblázatban találhatóak.

Bár az x-vektor neurális hálók architektúrája lényegesen egyszerűbb, mint egy wav2vec 2.0 hálóé, és sokkal kevesebb (kb. tízmillió) paraméterből is állnak, a kapott AUC-értékek igen magasak. Még a saját tanítású hálók használatával is olyan értékeket értünk el, melyeket csak két wav2vec 2.0 modell tudott túlszárnyalni (`XLS-R-1B` és `XLSR-Hungarian-53`). Ugyanakkor ezek a modellek véletlen súlyinicalizálással („from scratch") lettek tanítva, csupán 60 órányi hanganyag felhasználásával (mely különféle zajok hozzáadásával és visszhangosítással 240 órányira nőtt az FBANK és spektrogram jellemzők esetén). Az SRE-16 előtanított x-vektor modell által számított jellemzők pedig még ezeknél is magasabb AUC-értékhez vezettek (0,876), melyhez hasonlóan magas osztályozási teljesítményt kizárólag az egymilliárd súlyból álló wav2vec 2.0 modell volt képes nyújtani. Ez a kétségkívül meglepő eredmény véleményünk szerint a wav2vec 2.0 hálók esetén használt, bevett, mégis igen szimpla felvételszintű aggregációs technikának tudható be. Bár az x-vektor hálók összegző rétege szintén csupán a keretszintű aktivációs értékek átlagát és szórását számítja ki, az x-vektor struktúrában ezt még két további rejtett réteg követi, és az x-vektor jellemzőket általában az utolsó rejtett réteg aktivációi adják. Ezt a két utolsó, felvételszintű réteget ugyanúgy (sőt, együtt) tanítjuk, mint a mélyebb, keretszintű rétegeket, és így összességében a keretszintű aktivációk szofisztikáltabb aggregálási módját érhetjük el. Ezen felbuzdulva a közeljövőben az igen gyakran használt átlag- és szórásszámításon túlmutató aggregálási technikákat is meg tervezünk vizsgálni.

6. Összegzés

Jelen tanulmányunkban sclerosis multiplex (SM) beszédhangból történő azonosítását vizsgáltuk. Ennek érdekében tíz különböző, transzformer-alapú wav2vec 2.0 modellt használtunk jellemzőkinyerésre; a kapott keretszintű beágyazásvektorokat átlag és szórás segítségével aggregáltuk felvételszintű jellemzőkké. Azt találtuk, hogy az önfelügyelt tanulással előtanított hálókhoz képest a többnyelvi előtanítás nem hoz érdemi előrelépést az SM felismerésében, az igen nagyméretű (`XLS-R-1B`) `XLS-R` modell kivételével. Emellett a többnyelvi hálók finomhangolása sem javított az osztályozás teljesítményén, kivéve amikor a finomhangoláshoz használt tanítóadat nyelve megegyezett az alkalmazás nyelvével (azaz mindkettő magyar volt). Meglepő módon a wav2vec 2.0 modellek használatával elért osztályozási teljesítmény nem múlta felül a standard `SRE-16` előtanított x-vektor modellel kapott osztályozását sem, miközben az x-vektor neurális háló paramétereinek száma a wav2vec 2.0 háló paraméterszámának csupán töredéke. Mivel ennek oka véleményünk szerint a keretszintű beágyazásvektorok túlságosan egyszerű aggregálási módja lehetett, a közeljövőben a wav2vec 2.0 beágyazások kifinomultabb aggregálási módszereire tervezünk fókuszálni.

Köszönetnyilvánítás

A kutatást részben a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta (NKFIH, K-132460, NKFIH-1279-2/2020, ELTE TKP2020-IKA-06 és

TKP2021-NVA-09). A kutatást (amelyet a Szegedi Tudományegyetem valósított meg) az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal is támogatta a Mesterséges Intelligencia Nemzeti Laboratórium (MILAB, RRF-2.3.1-21-2022-00004) keretében.

Hivatkozások

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J., Fan, L.J., Fougner, C., Hannun, A.Y., Jun, B., Han, T., LeGresley, P., Li, X., Lin, L., Narang, S., Ng, A., Ozair, S., Prenger, R., Qian, S., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Sriram, A., Wang, C.J., Wang, Y., Wang, Z., Xiao, B., Xie, Y., Yogatama, D., Zhan, J., Zhu, Z.: Deep speech 2: End-to-end speech recognition in English and Mandarin. In: ICML. pp. 173–182 (2016)

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A.: XLS-R: Self-supervised cross-lingual speech representation learning at scale. In: Interspeech. pp. 2278–2282 (2022)

Baevski, A., Zhou, H., Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in Neural Information Processing Systems 33, 12449–12460 (2020)

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(Jul), 2079–2107 (2010)

Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011)

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: Unsupervised cross-lingual representation learning for speech recognition. In: Interspeech. pp. 2426–2430 (2021)

Darley, F.L., Brown, J.R., Goldstein, N.P.: Dysarthria in multiple sclerosis. Journal of Speech and Hearing Research 15(2), 229–245 (1972)

Delgado-Álvarez, A., Matias-Guiu, J.A., Delgado-Alonso, C., Hernández-Lorenzo, L., Cortés-Martínez, A., Vidorreta, L., Montero-Escribano, P., Pytel, V., Matias-Guiu, J.: Cognitive processes underlying verbal fluency in multiple sclerosis. Frontiers in Neurology 11, 629183 (2021)

Egas-López, J.V., Gosztolya, G.: Using the Fisher Vector approach for cold identification. Acta Cybernetica 25(2), 223–232 (2021)

Fan, Z., Li, M., Zhou, S., Xu, B.: Exploring wav2vec 2.0 on speaker verification and language identification. In: Interspeech. pp. 1509–1513 (2021)

Gosztolya, G., Tóth, L., Svindt, V., Bóna, J., Hoffmann, I.: Sclerosis multiplex hangalapú felismerése akusztikai alapú beágyazások használatával. In: MSZNY. pp. 151–160. Szeged (2022)

Hartelius, L., Runmarker, B., Andersen, O.: Prevalence and characteristics of dysarthria in a multiple-sclerosis incidence cohort: Relation to neurological data. Folia Phoniatrica et Logopaedica 52(4), 160–177 (2000)

Jenei, A.Z., Kiss, G.: Depresszió detektálása korrelációs struktúrán alkalmazott konvolúciós hálók segítségével. In: MSZNY. pp. 59–71. Szeged (2020)

Jenei, A.Z., Kiss, G., Sztahó, D.: Detection of speech related disorders by pretrained embedding models extracted biomarkers. In: SPECOM. pp. 279–289. Gurugram, India (2022)

Laakso, K., Brunnegård, K., Hartelius, L., Ahlsén, E.: Assessing high-level language in individuals with multiple sclerosis: A pilot study. Clinical Linguistics & Phonetics 14(5), 329–349 (2000)

Lin, W., Mak, M.W.: Wav2spk: A simple DNN architecture for learning speaker embeddings from waveforms. In: Interspeech. pp. 3211–3215 (2020)

Mar, R.A.: The neuropsychology of narrative: Story comprehension, story production and their interrelation. Neuropsychologia 42(10), 1414–1434 (2004)

Moro-Velazquez, L., Villalba, J., Dehak, N.: Using x-vectors to automatically detect Parkinson's disease from speech. In: ICASSP. pp. 1155–1159 (2020)

Mulfari, D., Meoni, G., Marini, M., Fanucci, L.: Machine learning assistive application for users with speech disorders. Applied Soft Computing 103(May), 107147 (2021)

Neuberger, T., Gyarmathy, D., Gráczi, T., Horváth, V., Gósy, M., Beke, A.: Development of a large spontaneous speech database of agglutinative Hungarian language. In: TSD. pp. 424–431 (2014)

Pérez-Toro, P.A., Klumpp, P., Hernandez, A., Arias, T., Lillo, P., Slachevsky, A., García, A.M., Schuster, M., Maier, A.K., Nöth, E., Orozco-Arroyave, J.R.: Alzheimer's detection from English to Spanish using acoustic and linguistic embeddings. In: Interspeech. pp. 2483–2487. Incheon, Dél-Korea (2022)

Renauld, S., Mohamed-Said, L., Macoir, J.: Language disorders in multiple sclerosis: A systematic review. Multiple Sclerosis and Related Disorders 10, 103–111 (2016)

Schneider, S., Baevski, A., Collobert, R., Auli, M.: wav2vec: Unsupervised pretraining for speech recognition. In: Interspeech. pp. 3465–3469 (2019)

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), 1443–1471 (2001)

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust DNN embeddings for speaker verification. In: ICASSP. pp. 5329–5333. Calgary, Alberta, Kanada (2018)

Szirmai, I.: Neurológia. Medicina, Budapest (2006)

A borderline személyiségzavar felismerése a nyelvhasználat lexikai és grammatikai jellemzői alapján

Felletár Fanni[1,2], Babarczy Anna[1,2]

[1] Nyelvtudományi Kutatóközpont, Kísérleti Szemantika és Pragmatika Kutatócsoport
{felletar.fanni, babarczy.anna}@nytud.hu
[2] Budapesti Műszaki és Gazdaságtudományi Egyetem, Kognitív Tudományi Tanszék
babarczy.anna@ttk.bme.hu

Kivonat A borderline személyiségzavarban (borderline personality disorder; BPD) érintett személyeket az identitás, a társas kapcsolatok valamint az érzelem- és indulatszabályozás zavarának tartós mintázata jellemzi. Az átlagpopuláció heterogenitása és a tradicionális diagnosztikai módszerek szubjektív jellege miatt a kórkép felismerése nehéz feladatot ró a klinikusokra. Számítógépes nyelvészeti módszerekkel kimutatták, hogy a BPD-vel élők – a kontroll személyekhez (healthy controls; HC) viszonyítva – szignifikánsan gyakrabban használnak egyes számú és tagadást kifejező nyelvi formákat valamint negatív valenciájú szavakat a narratíváikban. Emellett előző kutatásunkban arra az eredményre jutottunk, hogy a beszédprodukció nyelvi enkódolás fázisához köthető hibáinak aránya jelentős magyarázóerővel bír a BPD-ben érintett személyek megjóslásakor. Ezek alapján azt feltételeztük, hogy a BPD és a HC csoport elkülöníthető a nyelvi enkódolási hibák, az egyes számú és a tagadást kifejező nyelvi formák valamint a negatív valenciájú szavak alapján. A manuálisan és a NarrCat segítségével kinyert faktorokat egy logisztikus regressziós modellben használtuk fel, melynek AUC értéke 0,907 lett. A modellből arra következtethetünk, hogy minél magasabb az említett faktorok aránya egy narratívában, annál nagyobb a valószínűsége, hogy az adott személy a BPD csoportba tartozik.

1 Bevezetés

1.1 A borderline személyiségzavar jellemzői és felismerésének nehézségei

A borderline személyiségzavarral (borderline personality disorder; BPD) élő személyeket az identitás, a társas kapcsolatok valamint az érzelem- és indulatszabályozás zavarának tartós mintázata jellemzi. A DSM-5-TR (American Psychiatric Association [APA], 2022) dimenzionális modellje alapján a diagnózis felállításához az egyénnek a következő 7 személyiségvonásból legalább 4–gyel kell rendelkeznie, melyek közül legalább az egyik impulzivitás, kockázatvállalás vagy ellenségesség: érzelmi labilitás, depresszivitás, szorongásosság, szeparációs bizonytalanság, impulzivitás, kockázatvállalás és ellenségesség. A kórkép központi komponense továbbá az identitászavar, amely üresség érzés, alacsony szorongástűrés, valamint társas interakció hatására változó énkép, célok és értékrend formájában jelenik

meg (APA, 2022). Mivel a nyelvhasználat során az egyén gondolatait, érzelmeit, intencióit fejezi ki, a gondolatokat, érzelmeket, intenciókat formáló személyiségvonások is tükröződnek a nyelvhasználatban (Du és Sun, 2022; Hirsh és Peterson, 2009; Ireland és Mehl, 2014; Pennebaker és mtsai, 2003). Bár a BPD-nek nem centrális tünete az atipikus nyelvhasználat, a fentiek alapján feltételezhető, hogy a BPD személyiségvonásai megjelennek valamilyen módon a nyelvhasználatban.

Tekintve, hogy a BPD gyakran komorbid egyéb zavarokkal (pl. hangulatzavarok, szorongásos zavarok, függőségek, figyelemhiányos hiperaktivitás-zavar [ADHD], más személyiségzavarok), valamint a DSM – klinikai gyakorlatban továbbra is elterjedtebb – kategoriális modellje szerint a diagnózis felállítható 9–ből bármely 5 diagnosztikai kritérium teljesülése esetén (APA, 2022), a BPD-ben érintettek populációja meglehetősen heterogén (Cavelti et al, 2012; Millon és mtsai, 2004; Smits és mtsai, 2017), így a diagnosztizáció komoly klinikai kihívást jelent. További nehezítő tényező, hogy a tradicionális diagnosztikai módszerek (interjú, tesztek) a páciens belátásán, őszinteségén és emlékezeti képességén valamint a klinikus szubjektív megállapításain alapulnak, így nem teszik lehetővé a tünetek közvetlen és objektív mérését (Cwik és mtsai, 2016; Heinz és mtsai, 2022; Walsh és mtsai, 2020). Ezek a módszerek elengedhetetlenek, hiszen a páciens szubjektív megélése és a klinikus szakértelme a diagnosztizáció központi komponensei, azonban a BPD-vel élők nyelvhasználatával foglalkozó kutatások (Carter és Grenyer, 2012; Lyons és mtsai, 2018; Rosenbach és Renneberg, 2015; Wang és mtsai, 2020) eredményei rámutattak, hogy a számítógépes nyelvészeti módszerek diagnosztikai alkalmazása hasznos kiegészítő eleme lehet a jelenlegi hagyományos eljárásnak.

1.2 A BPD-ben érintett személyek nyelvhasználatának jellemzői

A BPD-vel élő személyek nyelvhasználatának vizsgálata viszonylag új kutatási iránynak számít, azonban néhány jó kiindulópontként szolgáló tanulmány már megjelent a témában. A tanulmányok egy része a diskurzusszinttel foglalkozik, ennek is két típusával: a narratív monológok koherenciaelemzésével (Adler és mtsai, 2012; Bendstrup és mtsai, 2021; Lind és mtsai, 2019) és a párbeszédek szerveződésének vizsgálatával (Wang és mtsai, 2020, 2021). A mondatok szintjén Carter és Grenyer (2012) a mondatok hosszát és komplexitását vizsgálta. A szavak szintjén a lexikai denzitást és diverzitást mérték, valamint szógyakorisági elemzéseket végeztek a Linguistic Inquiry and Word Count (LIWC; Pennebaker és mtsai, 2001) szoftver segítségével a következő kategóriákban: szófajok, beszédtéma (pl. család, munka, halál), személyjelölés (E/1, E/3, T/1, T/3), tagadás, pszichés folyamatok (pl. kogníció, érzelmek), “abszolút” szavak (pl. minden, semmi) (Carter és Grenyer, 2012; Lyons és mtsai, 2018; Rosenbach és Renneberg, 2015; Spruit és mtsai, 2022; Wang és mtsai, 2020).

A BPD-ben érintett személyek szóhasználatára fókuszáló kutatások jellemzően a személyjelölést és a szavak valenciáját vizsgálták. Az E/1-es személyes névmásokat tekintve azt találták, hogy a BPD-vel élő személyek narratíváiban szignifikánsan gyakrabban fordulnak elő ezek az alakok, mint az egészséges kontroll személyek (healthy controls; HC) személyek narratíváiban (Carter és Grenyer, 2012; Lyons és mtsai, 2018; Rosenbach és Renneberg, 2015; Spruit és mtsai, 2022). Spruit és

munkatársai (2022) egy különböző mentális zavarokkal (ADHD, autizmus spektrum zavar, bipoláris zavar, BPD, evészavar, generalizált szorongásos zavar, major depresszió, obszesszív-kompulzív zavar [OCD], poszttraumás stressz szindróma, skizofrénia) élő személyeket magában foglaló klinikai csoportot és egy HC csoportot összevetve azt fedezte fel, hogy az E/1-es névmások használata a mentális zavarral élők csoportjának általános jellemzője. Az E/3-as személyes névmásokat azonban nemcsak a HC személyekhez viszonyítva (Carter és Grenyer, 2012; Lyons és mtsai, 2018), hanem a generalizált szorongásos zavarral, a major depresszióval, az OCD-vel és a skizofréniával élő személyekhez mérten is szignifiánsan gyakrabban használják a BPD-ben érintett személyek (Lyons és mtsai, 2018). A személyes névmások gyakorisága arról ad információt, hogy milyen személy(ek) áll(nak) az adott személy figyelmi fókuszában (Lyons és mtsai, 2018). Így az E/1-es névmások gyakori használatáról befelé irányuló figyelemre, depresszióra, ruminációrа következhetünk (Edwards és Holzman, 2017; Zimmerman et al, 2016), az E/3-as névmások gyakori megjelenése pedig egy az egyén életében kitüntetett helyet elfoglaló személyre való fókuszáltságot jelez (Lyons és mtsai, 2018). Mivel a depresszivitás és a kitüntetett személy irányában érzett szeparációs szorongás a BPD személyiségvonásai közé tartoznak (APA, 2022), ezek az eredmények jól tükrözik a BPD-vel élő személyek működését.

A személyjelölésen túl a BPD-ben érintett személyek nyelvhasználatának meghatározó nyelvi markere az érzelmeket és értékelést kifejező szavak valenciája. Az ezzel kapcsolatos kutatások egybehangzóan arra az eredményre jutottak, hogy a BPD-vel élő személyek narratívái szignifikánsan több negatív valenciájú szót tartalmaznak, mint a HC személyeké (Carter és Grenyer, 2012; Lyons és mtsai, 2018; Spodenkiewicz és mtsai, 2013). Mivel a negatív valenciájú szavak megnövekedett száma a depresszió jele (Baddeley és mtsai, 2011; Newell és mtsai, 2018), és a BPD-re jellemző a depresszivitás (APA, 2022), ez az eredmény érvényesnek tekinthető. A negatív valenciájú szavakhoz hasonlóan a tagadás is egyfajta negativizmust, negatív orientációt fejez ki. Rosenbach és Renneberg (2015) major depressziós, BPD-ben érintett és HC személyeket összehasonlítva azt mutatták ki, hogy a BPD csoport mindkét csoporthoz képest gyakrabban használt tagadást kifejező nyelvi formákat. Wang és munkatársai (2020) kutatása szerint a tagadószavak a bipoláris személyekkel szemben is elkülönítik a BPD-vel élő személyeket.

Előző tanulmányunkban a BPD és a megakadásjelenségek gyakorisága közötti összefüggést vizsgáltuk (Felletár és mtsai, 2022). Számos jellemzőt figyelembe véve, a beszédprodukció nyelvi enkódolás (Levelt, 1989) fázisához köthető hibáinak (grammatikai hibák, kontamináció) aránya bírt a legnagyobb magyarázóerővel a BPD csoport függvényében. A grammatikai hibák morfológiai és szintaktikai szabályoknak ellentmondó jelenségek. A kontaminációk szó- és szerkezetvegyülések, amelyek egyszerre aktiválódó, egymással versenyző szerkezetek eredményei (Gósy, 2002).

1.3 A kutatás célja és menete

Jelen kutatásunk célja a BPD-ben érintett személyek nyelvhasználatának modellálása a HC személyek nyelvhasználatához viszonyítva, számítógéppel támogatott tartalomelemzési módszerrel. A fenti kutatások eredményeire támaszkodva azt feltételezzük, hogy a BPD és a HC csoport jól elkülöníthető a nyelvi enkódolási hibák,

az egyes számú nyelvi formák, a negatív valenciájú szavak valamint a tagadást kifejező nyelvi formák alapján. Tudomásunk szerint kutatásunk az első, amely magyar anyanyelvű mintán foglalkozik a témával.

A grammatikai hibákat, a kontaminációt és az E/3-as nyelvi formákat manuálisan, a többi jellemzőt pedig gépi módszerrel, a NarrCat (Ehmann és mtsai, 2014; László és mtsai, 2013) segítségével annotáltuk. A szógyakoriság alapján kinyert, standardizált, majd kategóriánként aggregált jellemzőket egy logisztikus regressziós statisztikai modellben használtuk fel a két csoport osztályozása céljából. Mivel kutatásunk hosszú távú célja a BPD felismerésének számítógépes nyelvészeti támogatása, a modell kiértékelésekor főként a modell fedését, F-értékét és AUC értékét vettük figyelembe.

2 Módszer

2.1 Résztvevők

A BPD-ben érintett és a HC személyek esetében is kényelmi és hólabda mintavételt alkalmaztunk. Mindkét csoport meghatározó részét a közösségi médiában jelen lévő BPD-vel élő személyeknek létrehozott és egyéb (pl. kísérleteket hirdető) csoportokból toboroztuk. A kutatásban már részt vett személyeket arra kértük, hogy osszák meg a részvételi lehetőséget ismerőseikkel. Mindkét csoportnál alapkritérium volt, hogy a résztvevők magyar anyanyelvűek legyenek, illetve ép hallással és beszédszervekkel rendelkezzenek. A BPD-ben érintett személyek esetében a kutatásban való részvétel feltétele a BPD diagnózisát igazoló klinikai dokumentum bemutatása volt, amely tartalmazza az F6030 – Érzelmileg labilis személyiségzavar BNO-kódot (BNO-11, 2022). A komorbid zavarok nem számítottak kizáró oknak, mivel az átlagpopulációban is gyakran előfordulnak. A HC személyként való részvétel feltétele az volt, hogy az egyének ne rendelkezzenek az alábbi zavarok egyikével sem: nyelvi vagy beszédzavar (pl. afázia), agyi vagy idegrendszeri betegség, sérülés (pl. Parkinson-kór, agydaganat), fejlődési zavar (pl. ADHD), pszichoaktív szerhasználat (gyógyszer-, alkohol-, drogfüggőség), mentális zavar (pl. személyiségzavar, skizofrénia). A HC személyek önbevallás alapján adtak számot az egészségi állapotukról. Végül 27 BPD-vel élő és 27 HC személy vett részt a kísérletben. A két csoport illesztésénél a résztvevők nemét, életkorát és sikeresen elvégzett tanéveinek számát vettük figyelembe, amelyek adatait az 1. táblázat foglalja össze.

Jellemzők	**BPD**	**HC**
N	27	27
nemi arány; nő:férfi	23:4	23:4
életkor; *M*(*SD*)	27,26(7,18)	26,59(7,28)
elvégzett tanévek száma; *M*(*SD*)	14,85(2,38)	15,52(2,85)

1. táblázat. A résztvevők demográfiai adatai.
BPD = borderline személyiségzavar; *HC* = egészséges kontroll; *N* = elemszám; *M* = átlag; *SD* = szórás.

A táblázat alapján láthatjuk, hogy a két csoport adatai kiegyensúlyozottak. Az elemszám (N = 27) és nemi arány (23:4) teljesen megegyezik, az átlagéletkor és az elvégzett tanévek átlagos száma pedig közel azonos. A nők ugyan felülreprezentáltak, azonban a BPD-ben érintettek klinikai populációját is hasonló arány jellemzi (APA, 2022). A BPD és a HC csoport életkorának és elvégzett tanévei számának összevetésére – az adatok nem normális eloszlása miatt – Mann-Whitney próbát alkalmaztunk, amely azt mutatta, hogy sem az életkor (p = 0,590), sem az iskolázottság (p = 0,437) tekintetében nincs szignifikáns eltérés a két csoport között. A hasonló demográfiai adatok lehetővé teszik a valós különbségek megállapítását a statisztikai elemzés során.

2.2 Adatgyűjtés

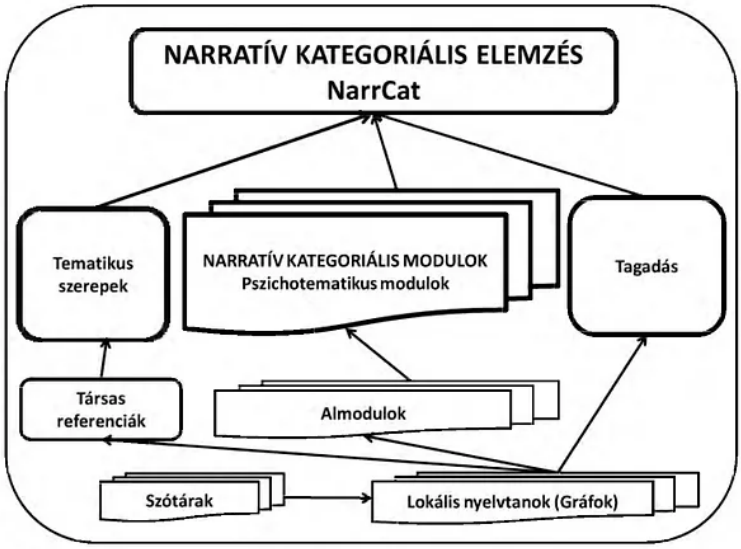
Az adatgyűjtéshez a narratív interjú módszerét választottuk, amely egy személyes életeseményekre fókuszáló, strukturálatlan interjúzási stratégia, melyben az adatrögzítő határozza meg a narratíva témáját, majd hagyja az adatközlőt minél szabadabban kibontakozni. Azért esett a választásunk erre a módszerre, mert 1) a pszichiátriai vizsgálatok során jellemzően strukturálatlan interjút alkalmaznak (Bruchmüller és mtsai, 2011; Cwik és mtsai, 2016; Tringer, 2010), 2) a BPD-ben érintett személyek a HC személyekhez hasonlóan teljesítenek strukturált körülmények között (Edell, 1987; Kernberg, 1975; Millon és mtsai, 2004), 3) így megfelelő közlésmennyiséget kapunk a jellemzők méréséhez. Az adatközlőket arra kértük, hogy meséljék el a tegnapi napjukat: *Kérlek, meséld el részletesen a tegnapi napodat a felébredésedtől kezdve! Hogyan telt a napod? Mit csináltál? Mi történt veled? Hogy érezted magad?* Nem szabtunk meg minimum és maximum időkeretet, ezzel is elősegítve az adatközlő minél spontánabb megnyilvánulását. A résztvevőket a kísérlet menetéről tájékoztattuk, a beszédfelvételhez és annak tudományos célú felhasználásához írásos hozzájárulásukat adták.

2.3 Adatok előfeldolgozása

Az adatgyűjtés lezárultával a hangfelvételeket megvágtuk és felhangosítottuk az Audacity program segítségével (Audacity® 3.1.3; https://audacityteam.org/). A hangfelvételeket ezután lefuttattuk az Alrite automatikus beszédfelismerő programon (Alrite©; https://alrite.io/ai/hu/), amelynek eredményeként egy összesen 20.139 tokenből álló szöveges korpuszt kaptunk. Az átiratok tartalmi és nyelvi helyességét manuálisan ellenőriztük, szükség esetén a hibákat kijavítottuk.

Mivel fontosnak tartottuk, hogy kutatásunk módszertanában is konzisztens legyen a bemutatott tanulmányokkal, a nyelvhasználat lexikai jellemzőinek kinyeréséhez a hivatkozott irodalomban alkalmazott LIWC mintájára fejlesztett NarrCat narratív kategoriális tartalomelemző szoftver (Ehmann és mtsai, 2014; László és mtsai, 2013) társas referencia, tagadás (Hargitai, 2004), érzelem (Fülöp és László, 2006) és értékelés (Bigazzi és mtsai, 2006) moduljait használtuk. A NarrCat a Nooj korpusznyelvészeti fejlesztő környezetben (Silberztein, 2005) fut, és a Magyar Nemzeti Szövegtár (Váradi, 2002), a Szeged Korpusz (Csendes és mtsai, 2005) valamint különböző történelmi és

többek között BPD-vel élőkkel gyűjtött pszichológiai tematikájú korpuszokon lett betanítva. A rendszer alapját a felsorolt korpuszokból épített annotált szótárak és a lokális nyelvtanok képezik, amelyek együttesen szolgáltatnak bemenetet az ún. pszichotematikus modulokhoz (1. ábra). A szoftver a narratívákat tagmondatokra bontja, majd az egyes tagmondatokat adatredukció révén vektorokká transzformálja, melyek értékei a pszichológiai kategóriákba tartozó nyelvi formák gyakoriságát reprezentálják (Ehmann és mtsai, 2014).



1. ábra: A NarrCat felépítése (Ehmann és mtsai, 2014).

A társas referencia modulból az E/1-es formákat, az érzelem modulból a negatív érzelmet jelölő szavakat, az értékelés modulból pedig a negatív értékelést jelölő szavakat mérő jellemzőket vettük figyelembe. Érzelmet jelölő szavaknak nevezzük azon főneveket, mellékneveket, igéket és határozókat, amelyek bármely morfológiai formájukban érzelmi (pl. *dühös*), hangulati (pl. *jókedvű*) vagy izgalmi (pl. *fáradt*) állapotot, állapotváltozást (pl. *beleszeret*), illetve ezek viselkedéses megnyilvánulását (pl. *sír*) fejezik ki, melynek az elbeszélő experiense (pl. *izgul*) vagy páciense (pl. *megbánt*) (Fülöp és László, 2006). Értékelést kifejező szavaknak a cselekvést, történést, állapotot (pl. *fekszik*) értelmező főneveket (pl. *munkakerülő*), mellékneveket (pl. *lusta*), igéket (pl. *pihen*) és határozókat tekintjük (Bigazzi és mtsai, 2006). Mivel a társas referencia modul az E/1-es és a T/1-es nyelvi formákat azonosítja, az E/3-as alakokat manuálisan annotáltuk. Az E/3-as formák kézi annotációja során a társas referencia modul nyelvi markerein (személyes névmások, személyragok) túl a közneveket és a tulajdonneveket is jelöltük. Abban az esetben is jelöltük az E/1-es és E/3-as alakokat, ha implicit argumentumként szerepeltek. A lexikai jellemzők annotálását a 2. táblázat szemlélteti.

Szövegszegmens	**E/1**	**E/3**	**NEZ**	**NET**	**TAG**
Szóval nagyon-nagyon kemény volt,	0	0	0	1	0
és minden egyes szombatom után ilyen nagyon idegesen megyek haza.	2	0	1	0	0
Jól érzem magam a barátnőimmel,	3	0	0	0	0
tök jót nevetgélünk meg dumálunk,	0	0	0	0	0
de belül ideges vagyok,	1	0	1	0	0
mert nem tanultam semmi újat,	1	0	0	0	2
mert leszólták a filozófiát,	0	0	0	1	0
mert a témaköröm a szexualitás,	1	0	0	0	0
azt is leszólják meg kiröhögik,	0	0	0	2	0
túl nyíltnak tartanak,	1	0	0	1	0
az is vagyok,	1	0	0	0	0
az nem zavar,	1	0	0	0	1
de hogy kicsit úgy kezelnek így a tanárok is,	1	0	0	0	0
hogy én leszek az,	2	0	0	0	0
akivel nehéz dolguk lesz.	0	0	0	1	0

2. táblázat. Lexikai jellemzők annotálása.
NEZ = negatív érzelem; *NET*= negatív értékelés; *TAG* = tagadás.

A nyelvi enkódolás fázisához köthető hibák elemzéséhez a megakadásjelenségek két típusát, a grammatikai hibákat és a kontaminációkat annotáltuk. A grammatikai hibák helyét [GRM], a kontaminációkét pedig [KNT] címkével jelöltük manuálisan a szövegben. Ezt követően a jegyek gépi összeszámolásával kiszámítottuk mindkét jelenség gyakoriságát. A grammatikai jellemzők annotálását az alábbi példa szemlélteti:

...gondoltam, hogy így elé megyek, akkor jövünk [GRM] *együtt ha- jöjjünk együtt haza, meg ilyesmi, vele voltunk* [KNT] (vele voltam × együtt voltunk)*, aztán én még elmentem…*

Miután kinyertük az összes jellemző gyakoriságát, a kapott értékeket arányosítottuk a szószám alapján. A hatásméret növelése és a dimenziócsökkentés érdekében a jellemzőket (grammatikai hiba, kontaminació, E/1, E/3, negatív érzelem, negatív értékelés) faktorokba aggregáltuk: nyelvi enkódolási hibák, egyes szám, negatív valenciájú szavak. A tagadás jellemzőt eredeti formájában hagytuk meg. A jellemzők rendszerét a 3. táblázat foglalja össze.

Jellemzők	Nyelvi markerek	Példák
Nyelvi enkódolási hibák		
grammatikai hiba	morfológiai és szintaktikai hibák	*"...azt néztem, ahogy Jamie Oliver scrampie hamburgereket készül."*
kontamináció	szerkezeti vegyülés	*"...ha most nem fogod be ezt a faszságot, akkor meg foglak ütni!"*
Egyes szám		
E/1	Névmások	*én, engem, magát, neki*
E/3	Személyragok	*jött, órái*
	Köznevek	*a srác, a barátnőm*
	Tulajdonnevek	*Rodrigo*
Negatív valenciájú szavak		
negatív érzelem[a]	Főnevek	*szégyenérzet[a], függőség[b]*
	Melléknevek	*szomorú[a], bántalmazott[b]*
negatív értékelés[b]	Igék	*stresszelek[a], utálom[b]*
	Határozók	*idegesen[a], kínosan[b]*
Tagadás[c]	Tagadószók	*nem, se*
	Igék	*nincs, sincsen*
	Határozószók	*sosem, sehol*
	Névmások	*senki*
	Névutók	*nélkül*
	Szuffixumok	*névtelen, hibátlan*
	Prefixumok	*unintelligens, antipatikus*

3. táblázat. Kinyert jellemzők, a BPD csoport narratíváiból vett példákkal.
[a] = negatív érzelmet kifejező szavak; [b] = negatív értékelést kifejező szavak;
[c] = Hargitai (2004) példái.

3 Eredmények

Az adatok előfeldolgozását követően leíróstatisztikai elemzést majd hipotézistesztelést végeztünk az R programozási nyelvre épülő JASP statisztikai szoftver segítségével (JASP Team (2022), JASP (Version 0.16.3); https://jasp-stats.org/).

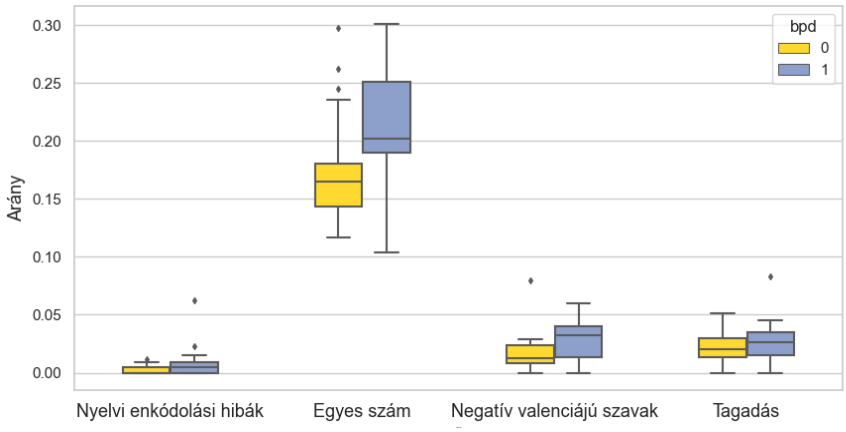
3.1 Leíróstatisztikai elemzés

Első lépésként a jellemzők és a faktorok átlagát, szórását és a BPD csoporttal való összefüggését vizsgáltuk meg, melynek eredményeit a 4. táblázat foglalja össze.

Jellemzők	BPD M(SD)	HC M(SD)	r^2
Nyelvi enkódolási hibák	**0,008(0,012)**	**0,002(0,003)**	**0,137**
grammatikai hiba	0,006(0,012)	0,002(0,003)	0,084
kontamináció	0,002(0,004)	0,000(0,000)	0,017
Egyes szám	**0,213(0,046)**	**0,172(0,043)**	**0,336**
E/1	0,170(0,052)	0,134(0,041)	0,260
E/3	0,043(0,025)	0,038(0,029)	0,014
Negatív valenciájú szavak	**0,028(0,017)**	**0,017(0,015)**	**0,185**
negatív érzelem	0,012(0,011)	0,006(0,006)	0,096
negatív értékelés	0,016(0,013)	0,011(0,011)	0,068
Tagadás	**0,027(0,016)**	**0,021(0,013)**	**0,048**

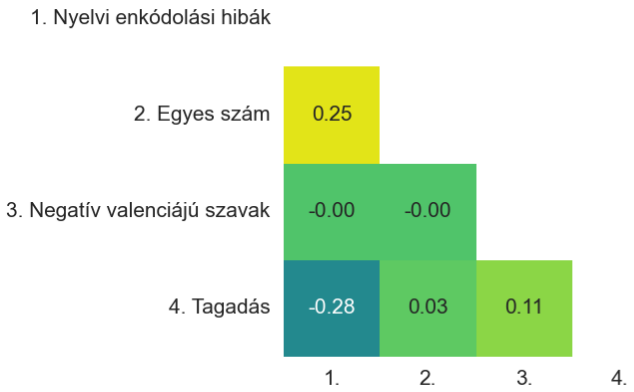
4. táblázat. Leíróstatisztikai eredmények. *BPD* = borderline személyiségzavar; *HC*= egészséges kontroll; *M* = átlag; *SD* = szórás; r^2 = determinációs együttható.

A BPD csoport narratíváiban az összes jellemző átlagosan nagyobb arányban fordul elő, mint a HC csoportában. A nyelvi enkódolási hibák esetében ez az arány (M_{BPD} = 0,008) a HC csoportának (M_{HC} = 0,002) a négyszerese. A determinációs együttható alapján, a két csoport közötti különbség az egyes szám tekintetében a legnagyobb (r^2 = 0,336), amelyet az alábbi ábra (2. ábra) jól szemléltet a többi faktorhoz viszonyítva.



2. ábra: A faktorok arányának helyzeti középértékei a BPD (kék) és HC (sárga) csoportban.

A hipotézistesztelésre használt logisztikus regressziós modell előkészítéseként, a kollinearitás minimalizálása érdekében megvizsgáltuk, a faktorok páronként mennyire korrelálnak egymással. A faktorok kiszűréséhez a korrelációs erősség kritikus szintjét |0,40|-nél húztuk meg, ez alatt ugyanis hagyományosan legfeljebb csak közepes erősségű összefüggésről beszélhetünk (pl. Dancey és Reidy, 2004). Az alábbi korrelációs mátrix alapján láthatjuk, hogy nincs erős korreláció a faktorok között (3. ábra).



3. ábra: A faktorok korrelációs mátrixa.

3.2 Hipotézistesztelés

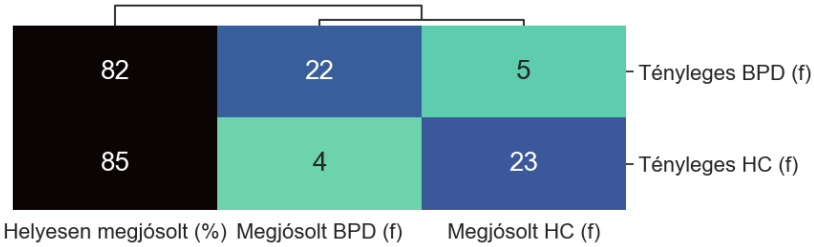
A hipotézistesztelést egy logisztikus regressziós modellben, "stepwise" módszerrel végeztük el, melyben az eredményváltozó a csoport volt, a prediktorok pedig a nyelvi enkódolási hibák, az egyes szám, a negatív valenciájú szavak és a tagadás. A első modell csak az interceptet tartalmazta, amelyhez a második modellben (R^2 = 0,233; $\chi^2(52)$ = 10,392; p = 0,001) az egyes szám, a harmadikban (R^2 = 0,371; $\chi^2(51)$ = 7,216; p = 0,007) a nyelvi enkódolási hibák, a negyedikben (R^2 = 0,527; $\chi^2(50)$ = 9,581; p = 0,002) a negatív valenciájú szavak, az ötödikben (R^2 = 0,624; $\chi^2(49)$ = 6,909; p = 0,009) pedig a tagadás adódott hozzá (ld. 5. Táblázat).

Modell	BIC	R^2	χ^2	df	p
1. Csak intercept	78,849			53	
2. Egyes szám	72,446	0,233	10,392	52	0,001
3. Egyes szám Nyelvi enkódolási hibák	69,219	0,371	7,216	51	0,007
4. Egyes szám Nyelvi enkódolási hibák Negatív valenciájú szavak	63,627	0,527	9,581	50	0,002
5. Egyes szám Nyelvi enkódolási hibák Negatív valenciájú szavak Tagadás	60,707	0,624	6,909	49	0,009

5. táblázat. A modellépítés folyamata.
BIC = Bayesi információs kritérium; R^2 = pszeudo determinációs együttható; χ^2 = a χ^2-teszt eredménye; *df* = szabadságfok; *p* = szignifikanciaérték.

Bár az összes modell erősen szignifikáns, az utolsó modell bejósló ereje a legpontosabb (*BIC* = 60,707), és a varianciát is ez magyarázza a legmagasabb arányban

(R^2 = 0,624). A végső modell pontossága 0,833, precizitása 0,846, fedése 0,815, F-értéke 0,830, AUC értéke pedig 0,907 lett. Az alábbi konfúziós mátrixban (4. ábra) láthatjuk, hogy a modell mindkét csoport tagjait közel egyforma, 80% körüli arányban osztályozza helyesen, a HC személyeket azonban valamivel pontosabban kategorizálja.



4. ábra: Az osztályozás konfúziós mátrixa a kimenetelek gyakoriságával (*f*) és a helyes kategorizációk arányával (%).

A Khí-négyzet teszt alapján nem mutatható ki szignifikáns összefüggés a kategorizáció helyessége és a nem között (φ_c = 0,093; $\chi^2(1)$ = 0,470; p = 0,493). A nők közel 85%-át és a férfiak 75%-át osztályozta helyesen a modell. A modell megfelel a prediktorok egymástól való függetlenségét mérő multikollinearitás kritériumainak, és mind a négy prediktorának jelentős mértékű és szignifikáns a hozzájárulása a modell bejósló erejéhez (ld. 6. táblázat).

Változók	b	SE	z	p	Tolerancia	VIF
Intercept	0,485	–2,827	–3,312	<,001		
Egyes szám	1,700	0,599	2,839	0,005	0,630	1,588
Nyelvi enkódolási hibák	3,325	0,349	2,869	0,004	0,625	1,599
Negatív valenciájú szavak	1,261	0,449	2,808	0,005	0,692	1,445
Tagadás	1,294	0,560	2,311	0,021	0,696	1,436

6. táblázat. Az utolsó modell változói.
b = standardizált regressziós együttható; *SE* = a standardizált regressziós együttható standard hibája; *z* = a z-teszt eredménye; *p* = szignifikanciaérték; *VIF* = Variance Inflation Factor.

4 Összegzés

Jelen tanulmányunkban azt feltételeztük, hogy a BPD-ben érintett személyek jól elkülöníthetőek az egészséges személyek csoportjától a nyelvi enkódolási hibák, az egyes számú nyelvi formák, a negatív valenciájú szavak és a tagadást kifejező nyelvi formák alapján. Egy logisztikus regressziós modell segítségével hatékonyan be tudtuk sorolni a résztvevőket a BPD és a HC csoportba. A szakirodalomban tárgyalt megállapításokkal összhangban, eredményként azt kaptuk, hogy minél magasabb a nyelvi enkódolási hibák, az egyes számú nyelvi formák, a negatív valenciájú szavak és

a tagadást kifejező szavak aránya egy narratívában, annál nagyobb a valószínűsége, hogy az adott személy a BPD csoportba tartozik.

Kutatásunk hosszú távú célja a BPD felismerésének számítógépes nyelvészeti támogatása. A gyakorlatban is hasznosítható modell építéséhez azonban a BPD populáció nyelvhasználatának átfogóbb megismerésére lesz szükség, amit többféle annotációs és elemzési módszer (pl. HuSpaCy, Orosz és mtsai, 2022; emóció- és szentimentszótárak, Szabó, 2014) keresztezésével, változatos szövegtípusok elemzésével, és más pszichiátriai zavarokban érintett populációk bevonásával remélhetünk megvalósítani a középtávú jövőben.

Köszönetnyilvánítás

Felletár Fanni kutatását a Magyar Tudományos Akadémia fiatal kutatói ösztöndíjprogramja támogatta. Köszönjük Kneszl Beáta Carmennek, BPD aktivistának és írónak, hogy részt vett a kutatásban, és emellett népszerűsítette a részvételi lehetőséget a BPD közösségben, Gosztolya Gábornak, az ELKH-SZTE Mesterséges Intelligencia Kutatócsoport munkatársának, hogy közreműködött a jellemzők kinyerésében, Pólya Tibornak, a Természettudományi Kutatóközpont Narratív és Történeti Pszichológiai Kutatócsoportja vezetőjének, hogy biztosította számunkra a NarrCat használatát, valamint a kísérlet összes résztvevőjének, hogy narratíváikkal hozzájárultak az adatgyűjtés sikerességéhez.

Hivatkozások

Adler, J. M., Chin, E. D., Kolisetty, A. P., Oltmanns, T. F.: The distinguishing characteristics of narrative identity in adults with features of Borderline Personality Disorder. An empirical investigation. In: Journal of Personality Disorders 26(4). pp. 498–512 (2012) https://doi.org/10.1521%2Fpedi.2012.26.4.498

American Psychiatric Association: Diagnostic and statistical manual of mental disorders (5th ed., text rev.). American Psychiatric Association, Washington, DC (2022)

Baddeley, J. L., Daniel, G. R., Pennebaker, J. W: How Henry Hellyer's use of language foretold his suicide. Crisis 32(5). pp. 288–92 (2011) https://doi.org/10.1027/0227-5910/a000092

Bendstrup, G., Simonsen, E., Kongerslev, M. T., Jørgensen, M. S., Petersen, L. S., Thomsen, M. S., Vestergaard, M.: Narrative coherence of autobiographical memories in women with borderline personality disorder and associations with childhood adversity. Borderline Personality Disorder and Emotion Dysregulation 8(18). (2021) https://doi.org/10.1186/s40479-021-00159-5

Bigazzi S., Csertő I., Nencini A.: A személy- és csoportközi értékelés pszicholingvisztikája. In: IV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 267–277. Szegedi Tudományegyetem, Szeged (2015)

Carter, P. E., Grenyer, B. F. S.: Expressive language disturbance in borderline personality disorder in response to emotional autobiographical stimuli. Journal of

Personality Disorders 26(3). pp. 305–321 (2012) https://doi.org/10.1521/pedi.2012.26.3.305

Cavelti, M., Lerch, S., Ghinea, D., Fischer-Waldschmidt, G., Resch, F., Koenig, J., Kaess, M.: Heterogeneity of borderline personality disorder symptoms in help-seeking adolescents. Borderline Personal Disord Emot Dysregul. 8(1):9. (2021 Feb 26) https://doi.org/10.1186/s40479-021-00147-9

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference on Text, Speech and Dialogue. pp. 123-131. Karlovy Vary, Czech Republic (2005 September 12-16)

Cwik, J. C., Papen, F., Lemke, J-E., Margraf, J.: An Investigation of Diagnostic Accuracy and Confidence Associated with Diagnostic Checklists as Well as Gender Biases in Relation to Mental Disorders. Front. Psychol. 7(1813). (2016) https://doi.org/10.3389/fpsyg.2016.01813

Dancey, C., Reidy, J.: Statistics without Maths for Psychology. Using SPSS for Windows. Prentice Hall, London (2004)

Du, X., Sun, Y.: Linguistic features and psychological states. A machine-learning based approach. Front. Psychol. 13(955850). (2022) https://doi.org/10.3389/fpsyg.2022.955850

Edwards, T., Holtzman, N. S.: A meta-analysis of correlations between depression and first person singular pronoun use. Journal of Research in Personality 68. pp. 63–68 (2017) https://doi.org/10.1016/j.jrp.2017.02.005

Ehmann, B., Csertő, I., Ferenczhalmy, R., Fülöp, É., Hargitai, R., Kővágó, P., Pólya, T., Szalai, K., Vincze, O., László, J.: Narratív kategoriális tartalomelemzés. A NARRCAT. In: X. Magyar Számítógépes Nyelvészeti Konferencia. pp. 136–147. Szegedi Tudományegyetem, Szeged (2014)

Felletár, F., Gosztolya, G., Hoffmann, I., Babarczy, A.: A megakadásjelenségek és a temporális paraméterek szerepe a borderline személyiségzavar felismerésében. Beszédtudomány – Speech Science 3(1). pp. 207–243 (2022) DOI: 10.15775/Besztud.2022.207-243

Fülöp, É., László, J.: Az elbeszélések érzelmi aspektusának vizsgálata tartalomelemző program segítségével. In: IV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 296–304. Szegedi Tudományegyetem, Szeged (2006)

Gósy, M.: A megakadásjelenségek eredete a spontán beszéd tervezési folyamatában. Magyar Nyelvőr 126(2). pp. 192–204 (2002)

Hargitai, R.: A LAS-Vertikum tagadás és self-referencia modulja. In: II. Magyar Számítógépes Nyelvészeti Konferencia. pp. 261–264. Szegedi Tudományegyetem, Szeged (2004)

Heinz, M. V., Thomas, N. X., Nguyen, N. D., Griffin, T. Z., Jacobson, N.C.: Technological Advances in Clinical Assessment. In: Asmundson, G. J. G. (szerk.) Comprehensive Clinical Psychology, 2nd edition, Vol. 4. pp. 301–320. Elsevier (2022) https://dx.doi.org/10.1016/B978-0-12-818697-8.00171-0

Hirsh, J. B., Peterson, J. B.: Personality and language use in self-narratives. Journal of Research in Personality 43. pp. 524–527 (2009) https://doi.org/10.1016/j.jrp.2009.01.006

Ireland, M. E., Mehl, M. R.: Natural language use as a marker of personality. In: Holtgraves, T. M. (szerk.), The Oxford handbook of language and social psychology. pp. 201–218. Oxford University Press (2014)

László, J., Csertő, I., Fülöp, É., Ferenczhalmy, R., Hargitai, R., Lendvai, P., Péley, B., Pólya, T., Szalai, K., Vincze, O., Ehmann, B.: Narrative Language as an Expression of Individual and Group Identity. The Narrative Categorical Content Analysis. SAGE Open. pp. 1–12 (2013 April-June) https://doi.org/10.1177/2158244013492084

Levelt, W. J. M.: Speaking. From intention to articulation. MIT Press, Cambridge, MA (1989)

Lind, M., Vanwoerden, S., Penner, F., Sharp, C.: Inpatient Adolescents With Borderline Personality Disorder Features. Identity Diffusion and Narrative Incoherence. Personality Disorders 10(4). pp. 389–393 (2019) https://doi.org/10.1037/per0000338

Lyons, M., Aksayli, N. D., Brewer, G.: Mental distress and language use. Linguistic analysis of discussion forum posts. Computers in Human Behavior 87. pp. 207–211. (2018) https://doi.org/10.1016/j.chb.2018.05.035

Millon, T., Grossman, S., Millon, C., Meagher, S., Ramnath, R.: Personality Disorders in Modern Life (2nd ed.). Wiley, Hoboken, NJ (2004)

Newell, E. E., McCoy, S. K., Newman, M. L., Wellman, J. D., Gardner, S. K.: You Sound So Down. Capturing Depressed Affect Through Depressed Language. Journal of Language and Social Psychology 37(4). pp. 451–474 (2018)

Orosz, Gy., Szántó, Zs., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 59–73. Szegedi Tudományegyetem, Szeged (2022) https://doi.org/10.48550/arXiv.2201.01956

Pennebaker, J. W., Mehl, M. R., Niederhoffer, K. G.: Psychological aspects of natural language. use. Our words, our selves. Annu Rev Psychol. 54. pp. 547-77 (2003) https://doi.org/10.1146/annurev.psych.54.101601.145041

Rosenbach, C., Renneberg, B.: Remembering rejection. Specificity and linguistic styles of autobiographical memories in borderline personality disorder and depression. Journal of Behavior Therapy and Experimental Psychiatry 46. pp. 85–92 (2015) https://doi.org/10.1016/j.jbtep.2014.09.002

Silberztein, M.: NooJ. A Linguistic Annotation System for Corpus Processing. In: Proceedings of HLT/EMNLP 2005 Interactive Demonstrations. pp. 10–11. Association for Computational Linguistics, Vancouver, British Columbia, Canada (2005)

Smits, M. L., Feenstra, D.J., Bales, D. L., de Vos, J., Lucas, Z., Verheul, R., Luyten, P.: Subtypes of borderline personality disorder patients: a cluster-analytic approach. Borderline Personal Disord Emot Dysregul 4(16). (2017) https://doi.org/10.1186/s40479-017-0066-4

Spodenkiewicz, M., Speranza, M., Taïeb, O., Pham-Scottez, A., Corcos, M., Révah-Levy, A.: Living from day to day. Qualitative study on borderline personality disorder in adolescence. J Can Acad Child Adolesc Psychiatry 22(4). pp. 282–289 (2013 Nov) PMID: 24223047

Spruit, M., Verkleij, S., de Schepper, K., Scheepers, F.: Exploring Language Markers of Mental Health in Psychiatric Stories. Applied Sciences 12(2179). (2022) http://dx.doi.org/10.3390/app12042179

Szabó, M. K.: Egy magyar nyelvű szentimentlexikon létrehozásának tapasztalatai. Nyelv, kultúra, társadalom. Budapest, Hungary (2014)

Váradi, T.: The Hungarian National Corpus. Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002). pp. 385–389. Las Palmas de Gran Canaria (2002)

Walsh, C. G., Chaudhry, B., Dua, P., Goodman, K. W., Kaplan, B., Kavuluru, R., Solomonides, A., Subbian, V.: Stigma, biomarkers, and algorithmic bias. Recommendations for precision behavioral health with artificial intelligence. JAMIA Open 3(1). pp. 9–15 (2020) https://doi.org/10.1093/jamiaopen/ooz054

Wang, B., Wu, Y., Taylor, N., Lyons, T., Liakata, M., Nevado-Holgado, A. J., Saunders, K. E. A.: Learning to Detect Bipolar Disorder and Borderline Personality Disorder with Language and Speech in Non-Clinical Interviews. INTERSPEECH 2020. pp. 437–441. Shanghai, China (2020) https://doi.org/10.48550/arXiv.2008.03408

Wang, B., Wu, Y., Vaci, N., Liakata, M., Lyons, T., Saunders, K. E. A.: Modelling Paralinguistic Properties in Conversational Speech to Detect Bipolar Disorder and Borderline Personality Disorder. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7243-7247 (2021) https://doi.org/10.48550/arXiv.2102.09607

Zimmermann, J., Brockmeyer, T., Hunn, M., Schauenburg, H., Wolf, M.: First-person Pronoun Use in Spoken Language as a Predictor of Future Depressive Symptoms. Preliminary Evidence from a Clinical Sample of Depressed Patients. Clin Psychol Psychother 24(2). pp. 384-391 (2017 Mar) https://doi.org/10.1002/cpp.2006

Szemantika

Magyar melléknevek poliszém jelentéseinek automatikus kinyerése gráfokkal

Héja Enikő, Ligeti-Nagy Noémi

Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
{heja.eniko, ligeti-nagy.noemi}@nytud.hu

Kivonat A cikk egy kutatás első fázisát mutatja be, amelynek célja interpretálható poliszém melléknévi jelentések automatikus kinyerése egynyelvű korpuszból egy felügyelet nélküli tanulási keretben. Kiindulásként 4 kritériumot határoztunk meg a jelentések elkülönítésére. A melléknéveket statikus szóbeágyazásokkal reprezentáltuk, majd ezekből egy szemantikai hasonlósági gráfot állítottunk elő. A jelentések elkülönítésére szolgáló kritériumokat ezen gráf részgráfjaival modelleztük. Végül egy részletes kvalitatív kiértékelés következett. Kutatásaink hosszabb távon hozzájárulnak a lexikográfusok és a nyelvészek munkájához, de a lexikális szemantikai információt tartalmazó NLP-célú *benchmark* adatbázisok létrehozását is segítik.
Kulcsszavak: poliszémia, WSI, gráf, klikk, főnévi kontextusok

1. Bevezetés

A kutatás célja a jelentés-megkülönböztetés mögött meghúzodó nyelvi intuíció modellezése és az elkülönített jelentések lehorgonyzása megfigyelhető adatokhoz. A kutatás jelentőségét az adja, hogy számos szerző szerint (pl. Véronis, 2003; Adamska-Sałaciak, 2006; Kuti és mtsai, 2010) a beszélők intuíciója nagy eltérést mutat az egyes szavakhoz tartozó jelentések particionálásában, amely komoly következményekkel bír a lexikográfiában, a lexikális szemantikában és a természetes nyelvfedolgozásban egyaránt. Különösen igaz ez a *poliszémia* esetében, hiszen a szójelentések automatikus detekcióját (*word sense induction* – WSI) célzó kutatások is elsősorban a homoním jelentések elkülönítésére koncentrálnak.

Így a jelen kiséreletben célunk az eltérő jelentések fogalmát úgy definiálni, hogy az elégséges kritériumként szolgáljon a jelentések megkülönböztetésére. Elvárásaink szerint ezek a kritériumok nemcsak az egyes jelentések kontextusokhoz való lehorgonyozását teszik majd lehetővé, hanem az így elkülönített jelentéseket szemantikai kategóriákhoz is kötik, hogy azok az emberi intuíció számára is megragadhatóvá váljanak. A kritériumokat felügyelet nélküli tanulási módszerrel modellezzük, hogy az introspekció szerepét minimalizáljuk.

A vizsgált mellékneveket statikus szóbeágyazásokkal reprezentáljuk. Bár ezen reprezentáció egyik fő hátrányaként azt szokás kiemelni, hogy nem alkalmas egy szóalak egyes jelentéseinek az elkülönítésére (vö. 'meaning conflation deficiency'

Camacho-Collados és Pilehvar, 2018) első eredményeink azt mutatják, hogy a gráf-alapú módszerek ezt a problémát képesek kiküszöbölni – legalábbis a poliszém jelentések esetében.[1]

A 2. fejezetben részletesen kifejtjük hipotézisünket és javaslunk 4 kritériumot a jelentések megkülönböztetésére, a 3. fejezetben bemutatjuk az alkalmazott módszert, a 4. fejezetben ismertetjük, hogyan validáltuk az automatikusan kinyert eredményeket főnévi kontextusaikkal. Az 5. fejezet témája a kiértékelés, végül a cikk egy összefoglalással zárul a 6. fejezetben.

2. A jelentések elkülönítésére szolgáló kritériumok

A jelentés hagyományos definíciója a jelentések azonosságát veszi alapul: a szinonímia definíciója[2] hosszú múltra tekint vissza (pl. Frege, 1892). Ez alapján azokat a jelentéseket tekintjük különbözőknek, amelyek nem szinonimák. A jelen gondolatmenetben megfordítjuk a definíciós sorrendet: nem a szinonímia fogalmából indulunk ki, hanem abból, hogy mikor eltérő két jelentés. Ez a döntés összhangban van azzal a ténnyel, hogy a szinonímia fogalma előfeltételezi az igazságfeltételek fogalmát, amely azonban a disztribúciós szemantika számára közvetlenül nem megragadható. Ezért nehéz valódi szinonima-osztályokat kinyerni a szövegből pusztán disztribúciós alapon: az automatikusan kinyert szinonima-osztályok sok esetben egyéb szűk szemantikai osztályokat is lefednek, pl. nemzetek neveit, színneveket, sőt hasonló disztribúciós tulajdonságokat mutató antonimákat is.

2.1. A majdnem-szinonímia fogalma

Ehhez bevezetjük a 'majdnem-szinonímia' (*near-synonymy* vö. Ploux és Victorri, 1998) fogalmát, amely a szinonímia egy kiterjesztett változata: akkor nevezünk két kifejezést majdnem-szinonimáknak, ha van a kontextusoknak egy olyan *korlátozott halmaza*, ahol a két kifejezés felcserélhető egymással az eredeti mondatok jelentésének megváltozása nélkül. Például, a *finom* és a *lágy* szinonimák számos zenéhez kapcsolódó főnév előtt, mint például *dallam*, *ritmus* vagy a *zene* maga, de nem szinonimák pl. ételek nevei előtt (pl. *lágy kenyér* $\neq$ *finom kenyér*). A fenti definíción túl a szűk szemantikai osztályok elemeit is majdnem-szinonimáknak tekintjük. A definíció ilyen irányú kiterjesztését az indokolja, hogy az eltérő szemantikai osztályokhoz való tartozás elégséges feltétele a jelentés megkülönböztetésnek, még akkor is, ha a szűk szemantikai osztályok elemeit egymással felcserélve nem feltétlenül őrizzük meg a mondatok igazságértékét.[3]

[1] Bár számos WSI-t célzó kutatás alapul a célszavakból készített gráf valamilyen lokális tulajdonságán (pl. Dorow és mtsai, 2004; Véronis, 2004; Biemann, 2006), tudomásunk szerint csak kevesen vizsgáltak statikus sűrű szóbeágyazásokból épített gráfokat (pl. Pelevina és mtsai, 2016).

[2] Két kifejezés akkor szinonima, ha minden kontextusban felcserélhetőek egymással úgy, hogy a mondatok igazságértéke nem változik.

[3] Pl. *fekete kalap* $\neq$ *szürke kalap*; *Fekete István* $\neq$ *Német István*.

2.2. A jelentés megkülönböztetés kritérumai

A fentiek alapján egy melléknév két jelentését akkor kell megkülönböztetni, ha:

1. Mindegyik melléknévi jelentéshez találunk (min. 1) majdnem-szinonimát.
2. Vannak olyan főnevek, amelyek grammatikus konstrukciókat alkotnak az eredeti melléknévvel és ennek majdnem-szinonimáival is.
3. Az egyes jelentéseket karakterizáló főnevek halmazai nem átfedőek.
4. A nem-átfedő főnév-halmazok egy vagy több szemantikai kategóriát képeznek „amelyek tükrözik a melléknevek szubszelekciós tulajdonságait" (vö. Pustejovsky, 1995).

A kritériumokat az 1. példával illusztráljuk, amely a *napfényes* melléknév automatikusan kinyert két jelentését tartalmazza. Láthatjuk, hogy mindkét jelentéshez tartozik egy-egy majdnem-szinonima: a *napsütéses* az első jelentéshez és a *napsütötte* a második jelentéshez. Listáztuk a majdnem-szinonima halmaz elemeivel grammatikus szerkezeteket alkotó főneveket is: *napfényes/napsütéses vasárnap*, *napfényes/napsütéses nap* stb., és *napfényes/napsütötte terület*, *napfényes/napsütötte terasz* stb. A két főnévhalmaz nem átfedő: nincs olyan, hogy *napsütötte nap* vagy *napsütéses terasz*. Végezetül, a szóban forgó főnevek egy szemantikai kategóriát alkotnak: az első jelentésre sepecifikus főnevek időintervallumokat, míg a második jelentésre specifikus főnevek fizikai kiterjedéssel rendelkező dolgokat jelölnek.

(1) 1. jelentés: *napfényes*, *napsütéses*
Főnévi kontextusok: *vasárnap*, *nap*
2. jelentés: *napfényes*, *napsütötte*
Főnévi kontextusok: *terület*, *sziget*, *oldal*, *terasz*

A kutatás jelen szakaszában azt vizsgáljuk, hogy a fenti kritériumok mennyiben alkalmasak a melléknévi jelentések elkülönítésére. A következő fejezetben egy gráf-alapú megközelítést javaslunk a kritériumok modellezésére.

3. Módszer

Módszerünk alapjául Ah-Pine és Jacquet (2009) szolgált, amennyiben a jelentésmegkülönböztetéseket **teljes részgráfokkal**, vagyis **klikkekkel** reprezentáljuk. Két fő különbséget azonban érdemes kiemelni: egyfelől, míg Ah-Pine és Jacquet (2009) kísérleteik során a névelemekre fókuszáltak, mi kizárólag az attributív mellékneveket vizsgáljuk. Ennek elsődleges oka, hogy előfeltevésünk szerint az attributív pozicióban lévő melléknevek jelentése egy viszonylag egyszerű jegykészlettel – a módosított főnévi csoport feje – leírható. Másfelől, a célszavakat statikus sűrű beágyazásokkal reprezentáljuk a gyakoriság-alapú ritka vektorok helyett.

3.1. Bemeneti adatok

A vizsgálandó melléknevek körét a 180 millió szavas MNSZ (Váradi, 2002) alapján határoztuk meg, ahol csak az alanyesetű mellékneveket vettük figyelembe,

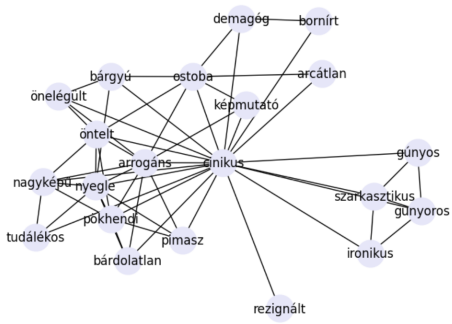
feltételezve, hogy a MN + FN szerkezetekben a MN mindig alanyesetű. Abból kiindulva, hogy a gyakoribb szavaknak több jelentése van, mint a ritka szavaknak (Zipf, 1949), csak a legalább 200-szor előforduló melléknevеket vizsgáltuk.

3.2. Reprezentációk

A melléknevek reprezentációja Bár Ah-Pine és Jacquet (2009) gyakoriság alapú vektorokkal reprezentálták a célkifejezéseket, mi statikus sűrű szóbeágyazásokat használtunk e célra.

Egy word2vec nyelvmodellt[4] (Mikolov és mtsai, 2013a,b) tanítottunk a Webcorpus 2.0 (Nemeskey, 2020) első 999 fájlján (21GB nyers szöveg). Kb. 170 M mondat,[5] a szövegek normalizált változata szolgált a tanítás bemenetéül. A 300 dimenziós CBOW vektorokat a Gensim Python csomag (Rehurek és Sojka, 2011) segítségével tanítottuk, ahol az ablak-méret 6 volt és a minimum gyakoriság 3. A tanítás során feltételeztük, hogy a szóalak által kódolt morfoszintaktikai információ hozzájárulhat a melléknevek jelentésének karakterizálásához. Ezt a hipotézisünket Novák és Novák (2018) is alátámasztja, akik különböző statikus szóbeágyazások szóhasonlósági feladatokon mért teljesítményének a vizsgálata során azt találták, hogy a melléknévi jelentéseket legjobban a szóalakokon tanított beágyazások reprezentálják. A tanítás eredményeképpen kb. 8,5M szóalakhoz rendeltünk szóbeágyazást.

A szemantikai hasonlóság reprezentációja Ahogy azt az 1. ábrán is láthatjuk, a melléknevek jelentéshasonlóságának gráf-alapú reprezentációja során olyan irányítatlan gráfokat generáltunk, amelyek csúcsai a melléknevek, élei (vagy azok hiánya) azt jelzik, hogy a szemantikai hasonlóság fennáll-e két melléknév között (vagy sem).

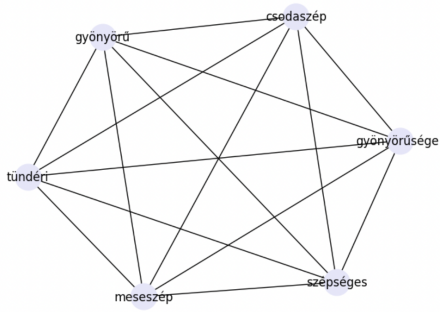


1. ábra: A *cinikus*-hoz szemantikailag hasonló melléknevek és azok közötti élek

[4] Elérhető itt: `https://nlp.nytud.hu/word2vec/cbow_3.tar.gz`

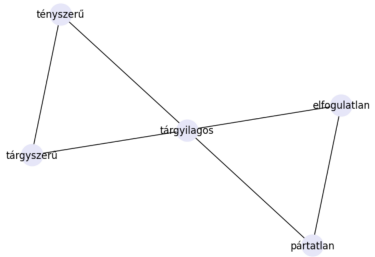
[5] Ugyanezekkel a hiperparaméterekkel tanítottunk egy word2vec modellt a teljes Webcorpus 2.0-n (kb. 591,4M mondat) is. Mindkét modellt kiértékeltük a Google Analogy Test Set magyar fordításán (Makrai, 2015). Mivel a nagyobb modell csak kisméretű javulást eredményezett a kisebb modellhez képest, a kisebb modellt használtuk a kisérleteink során.

A majdnem-szinonímák reprezentációja klikkekkel A 'különböző jelentés' fogalmát a majdnem-szinonima fogalmára támaszkodva kívánjuk megragadni. Ah-Pine és Jacquet (2009) nyomán a majdnem-szinonimákat klikkekkel reprezentáljuk, ezért a melléknévi gráfban teljes részgráfokat keresünk. Feltételezésünk szerint a klikkekkel jól modellezhetőek a majdnem-szinonima osztályok, mivel minden klikkhez tartozó melléknév disztribúciósan „nagyon hasonló" a klikk összes többi melléknevéhez (vö. 2. ábra).



2. ábra: Egy majdnem-szinonima halmaz klikk reprezentációja

A jelentés-elkülönítés reprezentációja: egy melléknév több klikk Összefoglalva az eddigieket: ha egy melléknévi lexéma több majdnem-szinonima osztályhoz tartozik, akkor több jelentéssel rendelkezik, és a majdnem-szinonima osztályokat klikkekkel modellezzük. Ebből következik, hogy egy melléknévnek akkor van több jelentése, ha több klikkhez is tartozik (vö. 3. ábra). Az így kinyert jelentéseket explicit kontextuális információhoz is le tudjuk horgonyozni, hiszen minden klikkhez kinyerhetőek azok a főnevek, amelyek a klikkben található összes melléknévvel előfordulnak.



3. ábra: A *tárgyilagos* két jelentésének reprezentációja klikkekkel

3.3. A klikkek kinyerése

M_{sim} $(n \times n)$ hasonlósági mátrix Az első lépésben egy M_{sim} hasonlósági mátrixot (ld. 4. ábra) hoztunk létre, amelynek soraiban és oszlopaiban is $A_{1...n}$ melléknevek word2vec reprezentációi $(a_{1...n})$ szerepelnek, ahol n a vizsgált melléknevek száma. A melléknevek szemantikai hasonlóságát a word2vec reprezentációjuk közötti koszinusz hasonlósággal számoltuk ki: $M_{sim}(a_i, a_j) = sim_{cos}(a_i, a_j)$

$$sim_{cos}(a_i, a_j) = \frac{\mathbf{a_i} \cdot \mathbf{a_j}}{||a_i|||a_j||} \tag{1}$$

	mesés	*káprázatos*	*varázslatos*
mesés	1	0,76	0,84
káprázatos	0,76	1	0,77
varázslatos	0,84	0,77	1

4. ábra: M_{sim} hasonlósági mátrix: $M_{a_i,a_j} = sim_{cos}(a_i, a_j)$.

M_a szomszédsági mátrix A következő lépésben M_{sim} hasonlósági mátrixból előállítottuk M_a szomszédsági mátrixot (ld. 5. ábra), amely azt reprezentálja, hogy A_i és A_j melléknevek hasonlóak-e vagy sem: M_{sim}-ben szereplő $sim_{cos}(a_i, a_j)$ hasonlósági értékeket egy alkalmas K küszöbérték mellett "vágtuk": $sim_{cos}(a_i, a_j) \geq K$ esetén $M_a(a_i, a_j) = 1$, egyébként pedig $M_a(a_i, a_j) = 0$.

	mesés	*káprázatos*	*varázslatos*
mesés	1	0	1
káprázatos	0	1	0
varázslatos	1	0	1

5. ábra: M_a szomszédsági mátrix: $K \geq 0.8$.

Eredményül tehát M_a-t kaptuk, amely egy 0-t és 1-t tartalmazó négyzetes $(n \times n)$ és szimmetrikus mátrix, amelynek a főátlójának minden eleme 1. Ezek a tulajdonságok tükrözik a hasonlósági reláció reflexív és szimmetrikus voltát.

Klikkek M_a-ban Ezentúl a M_a szomszédsági mátrixra úgy tekintünk, mint a melléknevek gráf reprezentációjára: a gráf csúcsai a melléknevek; $M_a(a_i, a_j) = 1$ azt jelöli, hogy a_i és a_j csúcsok éllel vannak összekötve, vagyis A_i és A_j melléknevek szemantikailag hasonlóak. Majd a gráf reprezentációban olyan mellékneveket keresünk, amelyek több klikkhez is tartoznak.

3.4. A főnévi kontextusok kinyerése – a klikkek validálása

Ebben a lépésben a melléknévi klikkek validálása történik azon klikk-specifikus főnevek halmazának előállításával, amelyekkel a klikkbe tartozó melléknevek előfordulnak. Várakozásaink szerint egy melléknév különböző jelentéseit a majdnemszinonimákon túl a vele abban a jelentésben együtt előforduló főnevek is jellemzik. Ezek a nem-átfedő főnévhalmazok explicit információval szolgálnak a jelentés-megkülönböztetésben. A klikk-specifikus főnévhalmazokat a következőképpen állítjuk elő:

1. Első lépésként összegyűjtjük az összes főnevet, amellyel egy melléknév előfordul. Ezt egy klikk minden melléknevére végrehajtjuk.[6]
2. A következő lépés a fent leírt módszerrel előállított halmazok metszetének meghatározása: ezek azok a főnevek, amelyek egy klikk minden melléknevével előfordulnak. Ha egy klikkre legalább egy ilyen főnév létezik, akkor az adott klikket jelentés-jelöltnek tekintjük.
3. Az 1. és 2. lépéseket minden klikkre elvégezzük, amelyekben az adott melléknév szerepel. Így mindegyik klikkre előáll egy főnévhalmaz.
4. Végül vesszük ezeket a halmazokat, és a metszetük tartalmát kidobjuk: csak azokat a főneveket tartjuk meg egy klikkhez, amelyek az adott klikkre kizárólagosak; nem fordulnak elő más klikkek főnévhalmazaiban.

3.5. Kiértékelés

Végül az eredményeket különböző paraméterbeállításokkal értékeltük ki. Mivel nem létezik hasonló adatbázis magyarra, kvalitatív kiértékelést végeztünk.

A kiértékelés során két célunk volt. Egyfelől, alá kívántuk támasztani, hogy az általunk javasolt jelentés-megkülönböztetést célzó kritériumok ténylegesen használhatóak erre a feladatra. Másfelől, az elkülönített melléknévi jelentéseket egy adatbázisban listáztuk a releváns kontextusaikkal együtt. Az evaluálás két lépésben történt: először az automatikusan kinyert melléknévi klikkek szemantikai tulajdonságait vizsgáltuk meg, majd a főnévi kontextusokra fókuszáltunk.

3.6. Paraméter-beállítás

A kiértékelés során azt találtuk, hogy 3 paraméternek van komoly hatása az eredményekre:

(i) a vizsgált melléknevek minimum-gyakorisága az MNSZ-ben,[7]
(ii) a K küszöbérték,
(iii) a főnevek melléknevekkel való együttes előfordulásának minimum gyakorisága a klikkvalidációs lépésben.

[6] Mindehhez a korpuszt a Magyar nemzeti szövegtár (Oravecz és mtsai, 2014) egy 91.4 millió tokenes részkorpusza jelentette, amelyet kifejezetten ennek a vizsgálatnak a céljából állítottunk össze.

[7] Ezt a paramétert jelen cikkben nem tárgyaljuk, de nyilvánvalóan nagy hatása van az eredmények fedésére.

A K küszöbérték hatása
Érdekes módon a K küszöbérték nemcsak a kinyert klikkek számára volt hatással, hanem a klikkbe tartozó melléknevek jelentésére is. Például, a legalább 200-szor előforduló melléknevek esetében a $K = 0,9$ beállítás csak egy pár melléknévi klikket eredményezett: 8 melléknév tartozott legalább 2 klikkhez és csak 2 klikket tudtunk validálni. A klikkek kizárólag számokra, hónapok és napok neveire referáltak, így nem különösebben érdekesek a kutatási célunk szempontjából. Másfelől alacsonyabb K küszöbértéket alkalmazva ($K = 0,7$) 187 különböző melléknevet kaptunk eredményül, amelyek mindegyike több validált klikkhez is tartozott. A $K = 0,7$-es küszöbérték mellett 3847 izolált csúcsot kaptunk, és 1085 olyan csúcsot, amelyekre igaz, hogy pontosan egy szomszédos csúcsa van. Ez azt jelenti, hogy a melléknevek 82%-át már eleve kizártuk a vizsgálatból. Így a $K = 0,7$-es küszöbérték valószínűleg túl magas.
A főnévi kontextus gyakoriságának a hatása
A főnévi kontextus minimum elvárt gyakoriságát ($Freq_n$) szintén figyelembe vettük a kísérleteink során. $Freq_{ADJ} = 200, K = 0,7$ rögzített paraméterek mellett három értékét vizsgáltuk. Az első esetben egy klikket akkor tekintettünk validnak, ha volt legalább 1 olyan főnév, amely a validáló korpuszban legalább 5-ször előfordult a klikkbe tartozó minden melléknévvel ($Freq_n \geq 5$). Mivel így csak kevés validált klikket kaptunk eredményül, ezt az értéket túl magasnak ítéltük. A fedés lehető legnagyobb növelése céjából a $Freq_n = 2$ értéket választottuk.[8] Ebben az esetben a 6042 input melléknévből 446 olyan melléknév maradt, amely több validált klikkhez is tartozott. Mivel ebben a kutatási szakaszban elsősorban a módszer alkalmazhatóságnak, pontosságnak a vizsgálata a cél, az alacsony fedés ellenére a további kiértékelési lépéseket ezeken a melléknévi klikkeken fogjuk végezni.

4. A releváns jelentések

A jelen fejezetben azokat a nyelvészeti megfontolásokat ismertetjük, amelyeket a kiértékelés során figyelembe vettünk.

4.1. Produktivitás

Az egyes jelentések elkülönítése során megkülönböztethetjük egy kifejezés kollokációs és produktív(abb) használatait. A jelen kutatásban a produktivitásra egy skálaként tekintünk. A skála egyik végén a szigorú értelemben vett kollokációkat találjuk, ahol az attributív melléknév és a módosított főnév egyaránt rögzítve vannak lexikailag. Ebben az esetben a kifejezés jelentése nem kompozicionális: egyik komponens sem helyettesíthető egy majdnem-szinonimával, úgy hogy az eredeti kifejezés jelentése (i) ne változzon, (ii) előre megjósolható módon változzon (pl. *fehér zaj*, *fekete doboz*).

[8] Vizsgáltuk a $Freq_n = 1$ paraméterbeállítást is: ez a nagyobb fedés mellett nehezebben interpretálható eredményeket adott – feltételezésünk szerint azért, mert ez az érték nem alkalmas a szisztematikus és véletlen hiányok szétválasztására.

Annak ellenére, hogy a kollokációk szolgálhatnak az elkülönítendő jelentések forrásául, a jelen WSI feladatban a 'félig-kompozicionális' szerkezetekre fókuszáltunk, ahol a kompozicionalitás a melléknevek és főnevek egy meghatározott körén működik (pl. *fehér/szürke/fekete gazdaság*). Ezek a kifejezések nem tekinthetőek szigorú értelemben kollokációnak, hiszen ezek a színnevek a *gazdaság* szó kontextusában egy új dimenziót vezetnek be: annak a mértékét, hogy egy gazdasági ág mennyiben kerüli ki a legális üzleti forgalmat.

4.2. Szubkategorizáció

A legérdekesebb esetek azok, ahol a főnevek egy vagy több szemantikai osztályt alkotnak, és a klikket alkotó melléknevek szinonímák a szemantikai osztályba tartozó főnevek előtt. Ezekben az esetekben a melléknév szubkategorizálja a főnévi kontextust (vö. Pustejovsky, 1995). Például a *könnyű*, *komoly*, *szép*, *éles*, *finom* melléknevek egyes jelentései elkülöníthetőek az alapján, hogy milyen szemantikai főnévcsoportokat szubkategorizálnak. A *könnyű* különböző jelentéseket hordoz a fizikai tárgyakra (*könnyű táska*), a ruhákra (*könnyű kabát*), az ételekre referáló (*könnyű vacsora*) főnevek előtt, és mást jelent az olyan főnevek előtt, mint *válasz*, *feladat*, *megoldás*.

Az egyes főnévi szemantikai osztályok mérete eltérő lehet: a produktivitási skála másik végén találjuk azokat a melléknévi majdnem-szinoníma osztályokat, amelyek sokkal több elemet tartalmazó szemantikai osztályokat szubkategorizálnak; ezek szintén fontosak a kutatás szempontjából. Például az automatikusan kinyert klikkek alapján a *szomorú* különböző jelentésekkel bír attól függően, hogy a módosított főnév emberekre vagy időtartamokra referál-e.[9] A kinyert klikkek alapján mondhatjuk azt, hogy *szomorú* [*időszak, év, nap*] és *gyászos* [*időszak, év, nap*], de nincs olyan, hogy *bánatos* [*időszak, év, nap*] és olyan sem, hogy *gyászos* [*lány, ember*].

(2) 1. klikk: *szomorú*, *gyászos*
Főnévi kontextusok: *időszak*, *év*, *nap*
2. klikk: *szomorú*, *bánatos*
Főnévi kontextusok: *lány*, *ember*

5. Kiértékelés

5.1. Az automatikusan kinyert melléknévi klikkek szemantikai osztályozása

Szűk szemantikai osztályok Az első szembetűnő probléma az volt, hogy nem minden melléknév bizonyult egyformán relevánsnak a jelentés-elkülönítés szempontjából. Például a dátumok (napok, hónapok) és mértékegységek legtöbb esetben nem mutattak semmilyen érdekes tulajdonságot. Ha több klikkhez is

[9] Érdekes, hogy a *vidám*-nak is két jelentése különült el: pontosan ugyanazokat a főnévi szemantikai osztályokat szubkategorizálta, mint a *szomorú*.

tartoztak egyszerre, akkor is általában ugyanazzal a jelentéssel szerepeltek. Hipotézisünk szerint ezen szűk szemantikai osztályok eltérő mérete és az elemeik közötti eltérő távolságok miatt a szűk szemantikai osztályokhoz tartozó melléknevek nem csoportosíthatóak következetesen egy-egy klikkbe függetlenül a paraméterek aktuális értékeitől. Egy másik indok, hogy ezeket a mellékneveket kizárjuk a további vizsgálódásból az, hogy az ilyen melléknevek jelentése általában szemantikailag egyértelmű, egyjelentésűek.[10] Néhány kivételtől eltekintve azt találtuk, hogy a számnevek, nap- és hónapnevek, színnevek, mértékegységek és különféle nemzeti valuták esetében a főnévi kontextusok nem szolgáltatnak elég alapot a jelentések elkülönítésére.

Névelemek A melléknevek egy másik szemantikai osztályát a névelemek alkották: elsősorban ország- és városnevek valamint vezetéknevek. Az elgondolkodtató csoportosítások ellenére a jelen vizsgálatban ezeket a mellékneveket is figyelmen kívül hagytuk, hiszen érdeklődésünk központjában a lexikális jelentés áll, és a névelemek esetében a klikkbe tartozást elsősorban faktuális tényezők határozzák meg (pl. az *egri* melléknév két klikkhez tartozott: [*egri*, *soproni*, *veszprémi*] illetve [*egri*, *esztergomi*], ahol az első klikk vélhetően borvidékekre, míg a második érsekségekre referáló névelemeket tartalmaz). A kézi kiértékelés egy érdekes eredménye az volt, hogy a $6k$ ablakméretű word2vec reprezentáció meglehetősen hatékony eszköznek bizonyult a szűk szemantikai osztályok és a névelemek klikkjeinek kinyerésében: 446 több klikkhez is tartozó melléknévből 99 volt névelem, 28 tartozott a mértékegységek csoportjába és 11 melléknév referált számnevekre.

Negatív emotív szemantikai tartalmú fokozó elemek Az emotív fokozó elemekkel kapcsolatban is azt találtuk, hogy a klikkek nem tükröznek eltérő jelentéseket.

(3) 1. klikk: *borzalmas iszonyatos rettenetes*
Főnévi kontextus: *szenvedés*, *kép*, *körülmény*

2. klikk: *borzalmas*, *félelmetes*, *rettenetes*, *szörnyű*
Főnévi kontextus: *látvány*, *nap*, *érzés*

3. klikk: *borzalmas*, *borzasztó*, *rettenetes*, *szörnyű*, *rémes*
Főnévi kontextus: *emlék*, *élmény*

Míg a klikkek azt támasztják alá, hogy az emotív fokozó elemek egy koherens szemantikai osztályt alkotnak a mellékneveken belül, sem a klikkek maguk, sem a főnévi kontextusok nem nyújtanak elég alapot arra, hogy az egyes klikkek jelentéseit elkülönítsük egymástól.

5.2. A főnévi kontextusok kvalitatív kiértékelése

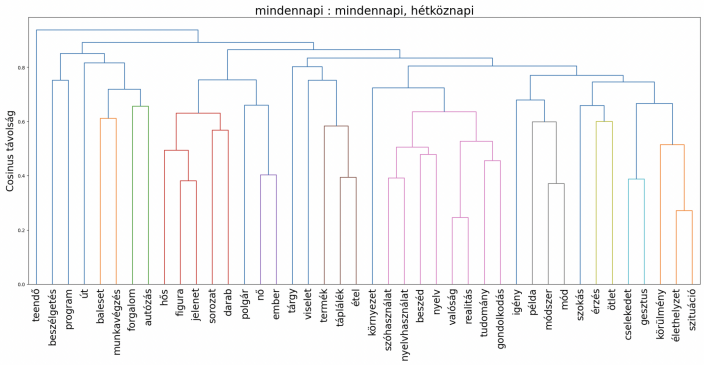
Miután kizártuk a nem-releváns eseteket (kb. 240 melléknevet), egy részletesebb kiértékelést is végeztünk, amelynek célja egy olyan melléknévi adatbázis létreho-

[10] Bár bizonyos esetekben a szűk szemantikai osztályokhoz tartozó melléknevek is lehetnek poliszémek (pl. a *fekete*, *szürke*, *fehér*).

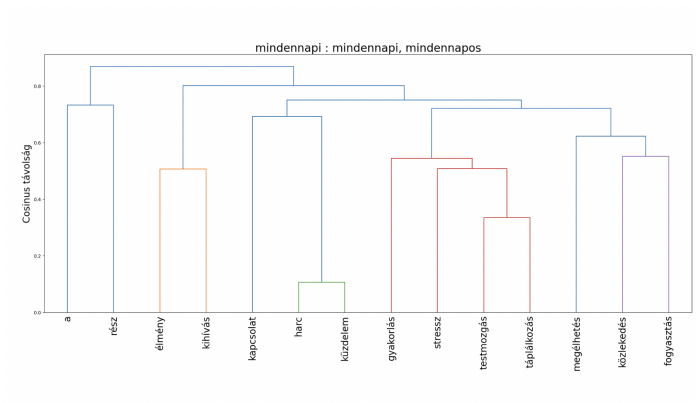
zása volt, amelyben a jelentések elkülönítése a majdnem-szinonimák által motivált, és az egyes jelentések expliciten megadott a főnévi kontextusokhoz vannak kötve. Azt is vizsgáltuk, hogy a szóban forgó főnevek segítenek-e a jelentéskülönbségek fogalmi megragadásában. Így a maradék klikkeket kézzel ellenőriztük, a készülő adatbázisba klikkenként legfeljebb 5 főnév került, úgy hogy törekedtünk arra, hogy a választott főnevek jellemzőek legyenek az adott jelentésre az emberi észlelés számára. A klikkek validálása és a releváns főnevek kiválasztása során az alábbi lépéseket követtük:

1. A főnévi kontextusok word2vec reprezentációját használtuk, amelyeket a 3.1 fejezetben leírtak alapján hoztunk létre.
2. A főnévi vektorokat klasztereztük egy hierarchikus agglomeratív klaszterezési eljárással (*average linkage*, koszinusz távolság), melynek célja a szubkategorizációs mintázatok automatikus kinyerése volt.

Például a *mindennapi* melléknév két klikkhez is tartozott: a 6. és a 7. ábrák ábrázolják a főnévi kontextusok dendogramjait. A majdnem-szinonimák már megvilágítják a *mindennapi* melléknév két jelentését (*hétköznapi* ill. *mindennapos*), amelyek közül az egyik a *normális/bevett/megszokott* míg a másik jelentés szabályos időközönként űzött tevékenységekre utal. A dendogramok alapján levonhatjuk azt a következtetést, hogy például a nyelvhez kapcsolódó dolgok (pl. *szóhasználat*, *nyelvhasználat*) inkább normálisak vagy megszokottak, mintsem periódikusan mindennap ismétlődőek, míg a *gyakorlás* vagy a *testmozgás* szabályos időközönként végzett (végzendő) tevékenységek, és nem szükségszerűen megszokott vagy bevett dolgok. Így a dendogram ágai azt jelzik, hogy a melléknév egyes jelentései milyen szemantikai osztályba tartozó főneveket szubkategorizálnak.



6. ábra: A [*mindennapi*, *hétköznapi*] melléknévi klikk főnévi kontextusainak klaszterei



7. ábra: A [*mindennapi*, *mindennapos*] melléknévi klikk főnévi kontextusainak klaszterei

A 446 melléknévből 53 olyan melléknév maradt, amelyek validált klikkekhez tartoztak: 118 klikkhez összesen. Ez azt jelenti, hogy az 53 melléknévnek 118 jelentését különítettük el. Ezek elérhetőek a GitHub-on; `https://github.com/nytud/HuWiC`. A kvalitatív kiértékelés során azt találtuk, hogy a javasolt felügyelet-nélküli tanuláson alapuló módszer meglepően érdekes jelentés megkülönböztetéseket tesz, amelyek az emberi intuíció számára nem közvetlenül megragadhatóak. Így az alacsony fedés ellenére azt gondoljuk, hogy az ez irányú kutatásokat érdemes a jövőben folytatni.

6. Összegzés és további feladatok

A cikkben egy folyamatban lévő kutatás első lépését ismertettük, amelynek célja, hogy egy felügyelet-nélküli tanítási algoritmust alkalmazva magyar melléknevek interpretálható poliszém jelentéseit nyerje ki egynyelvű korpuszból. Ez a munka hozzájárulhat a lexikográfusok, nyelvészek munkájához, és segíti a kapcsolódó benchmarkok építését is NLP célokra. Első lépésben 4 kritériumot támasztottunk a jelentések elkülönítésére, amelyeket a következő lépésben implementáltunk. Végül egy részletes kvalitatív kiértékelés következett, amely egyfelől az automatikusan kinyert jelentés-jelöltek relevanciáját vizsgálta, másfelől a főnévi kontextusok szerepét vizsgálta a jelentések elkülönítésében. A kiértékelés azt mutatta, hogy a kinyert jelentések sok esetben nagyban segítettek a jelentések interpretálható elkülönítésében, méghozzá az alkalmazott módszertan miatt oly módon, amely véleményünk szerint sem introspektív, sem korpusz-alapú módszertan számára nem közvetlenül hozzáférhető. Az első kísérletek azt mutatták, hogy a fedés alacsony, így a jövőben elsődleges célunk ennek növelése. Ezt első körben a paraméterek változtatásával kívánjuk elérni, de vizsgálni kívánunk a klikkeknél kevésbé megszorított részgráfokat is. További feladat a poliszémiában szerepet játszó főnévi kontextusok szemantikai osztályainak feltárása is.

Hivatkozások

Adamska-Sałaciak, A.: Meaning and the Bilingual Dictionary. The Case of English and Polish. (Polish Studies in English Language and Literature 18). Peter Lang, Frankfurt am Main (2006)

Ah-Pine, J., Jacquet, G.: Clique-Based Clustering for Improving Named Entity Recognition Systems. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). pp. 51–59. Association for Computational Linguistics, Athens, Greece (Mar 2009), https://aclanthology.org/E09-1007

Biemann, C.: Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing. pp. 73–80. Association for Computational Linguistics, New York City (Jun 2006), https://aclanthology.org/W06-3812

Camacho-Collados, J., Pilehvar, M.T.: From Word to Sense Embeddings: A Survey on Vector Representations of Meaning (2018), https://arxiv.org/abs/1805.04032

Dorow, B., Widdows, D., Ling, K., Eckmann, J.P., Sergi, D., Moses, E.: Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination (2004), https://arxiv.org/abs/cond-mat/0403693

Frege, G.: Uber Sinn und Bedeutung. In: Textor, M. (szerk.) Funktion - Begriff - Bedeutung, Sammlung Philosophie, vol. 4. Vandenhoeck & Ruprecht, Göttingen (1892)

Kuti, J., Héja, E., Sass, B.: Sense disambiguation - 'Ambiguous sensation'? In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). p. 23–30. la Valetta, Malta (2010)

Makrai, M.: Comparison of distributed language models on medium-resourced languages. In: Tanács, A., Varga, V., Vincze, V. (szerk.) XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015) (2015)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013a), https://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems 26 (10 2013b)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)

Novák, A., Novák, B.: POS, ANA and LEM: Word embeddings built from annotated corpora perform better. In: Computational Linguistics and Intelligent Text Processing: 17th International Conference, (CICLing 2018). Springer International Publishing, Cham, Hanoi, Vietnam (2018)

Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014), http://www.lrec-conf.org/proceedings/lrec2014/pdf/681_Paper.pdf

Pelevina, M., Arefiev, N., Biemann, C., Panchenko, A.: Making Sense of Word Embeddings. In: Proceedings of the 1st Workshop on Representation Learning for NLP. pp. 174–183. Association for Computational Linguistics, Berlin, Germany (Aug 2016), https://aclanthology.org/W16-1620

Ploux, S., Victorri, B.: Construction d'espaces sémantiques a l'aide de dictionnaires de synonymes. Traitement automatique des langues 1(39), 146–162 (1998)

Pustejovsky, J.: The Generative Lexicon. MIT Press, Cambridge, MA (1995)

Rehurek, R., Sojka, P.: Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3(2) (2011)

Váradi, T.: The Hungarian National Corpus. In: Proceedings of the Second International Conference on Language Resources and Evaluation, Las Palmas. pp. 385–389 (2002)

Véronis, J.: Sense tagging: does it make sense? In: Wilson, A., Rayson, P., McEnery, T. (szerk.) Corpus Linguistics by the Lune: a festschrift for Geoffrey Leech. Peter Lang, Frankfurt (2003)

Véronis, J.: HyperLex: lexical cartography for information retrieval. Computer Speech & Language 18(3), 223–252 (2004), http://dblp.uni-trier.de/db/journals/csl/csl18.html#Veronis04

Zipf, G.K.: Human Behaviour and the Principle of Least-Effort. MA Thesis (1949)

Mondatszám-meghatározás hatása a magyar nyelvű jogi szövegek extraktív kivonatainak minőségére

Csányi Gergely Márk[1], Gadó Krisztián[1], Bajári Lúcia[1], Megyeri Andrea[2], Fülöp Anna[2], Egri Erika[2], Vági Renató[1,3], Nagy Dániel[1], Vadász János Pál[1,4], Üveges István[1,5]

[1]MONTANA Tudásmenedzsment Kft., 1029 Budapest, Hársalja utca 32.
[2]Wolters Kluwer Hungary Kft., 1117 Budapest, Budafoki út 187-189.
[3]Eötvös Loránd Tudományegyetem Állam- és Jogtudományi Doktori Iskola, 1053 Budapest, Egyetem tér 1–3.
[4]Nemzeti Közszolgálati Egyetem, Információs Társadalom Kutatóintézet, 1083 Budapest, Ludovika tér 2.
[5]Szegedi Tudományegyetem Nyelvtudományi Doktori Iskola, 6722 Szeged, Egyetem utca 2.
{csanyi.gergely,gado.krisztian,bajari.lucia,vagi.renato, nagy.daniel,vadasz.pal,uveges.istvan}@montana.hu
{andrea.megyeri,anna.fulop}@wolterskluwer.com

Kivonat Az egyes dokumentumok tartalmi összefoglalása során a cél egy dokumentum rövidebb változatának előállítása úgy, hogy annak fő információtartalma a kivonatban megőrződjön. Cikkünkben az anonimizált bírósági határozatokhoz készült extraktív kivonatoló rendszer fejlesztése során szerzett tapasztalatokat ismertetjük, különös tekintettel a kivonatok hosszával (mondatszám) kapcsolatban felmerült kérdésekre, és az azokra adott válaszainkra. A kivonatokkal egy jogi adatbázis felhasználóinak találati listában való könnyebb orientációját kívántuk támogatni.
Kulcsszavak: extraktív szövegösszefoglalás, kivonatolás

1. Bevezetés

Egy szöveg kivonatán a dokumentum tartalmának lehetőség szerint rövid, de pontos reprezentánsát értjük. Módszertanát tekintve beszélhetünk extraktív vagy absztraktív szövegösszefoglalásról attól függően, hogy az adott szöveg tartalmi kivonatát milyen módszerrel állítjuk elő. Extraktív esetben a kivonat részét képező mondatok szó szerint, az eredeti szövegből kiemelve kerülnek a kivonatba, míg absztraktív esetben a cél olyan szintaktikaliag és szemantikailag is helyes mondatok generálása, amelyek ugyancsak helyes összefoglalását adják a kiindulási szövegnek (Nenkova és McKeown, 2011).

A megfelelően elkészített kivonatokkal jelentősen javítható például jogi keresőrendszerek teljesítménye azáltal, hogy egy keresésre válaszul nem pusztán relevánsnak ítélt dokumentumok egy halmazát, hanem azok tartalmi összefoglalását

is visszaadjuk. Az ilyen összefoglalások megfelelő választ nyújthatnak például a jogi keresőket gyakran jellemző homogenitás problémájára, amikor is a visszakapott találati listákban a technikai adatokon túl nem jelenik meg olyan érdemi információ, amely alapján a felhasználó a dokumentumba való beleolvasás nélkül is megítélhetné annak tényleges relevanciáját (Vági, 2022).

A célunk egy olyan modul elkészítése volt, amely által előállított kivonatok a fent említett módon képesek orientálni a felhasználókat egy jogi keresőrendszer használata közben. A konkrét implementációt a bírósági határozatok kivonatolásának automatizálása céljából valósítottuk meg. A kivonatkészítést extraktív alapokon készítettük el, mivel elvárás volt, hogy a kivonatok csakis nyelvileg helyes mondatokat tartalmazzanak. Felügyelet nélküli gépi tanulással kísérleteztünk, tekintettel arra, hogy a megbízhatóbb megoldások ezek között találhatók (Schluter, 2017).

Általános esetben egy kivonat minőségét felügyelet nélküli, extraktív kivonatolás esetén befolyásolhatjuk az abba kerülő lehetséges szövegegységek rangsorolási módjának változtatásával, a vektorizálási forma megválasztásával, a kivonat szempontjából nem releváns szövegrészek kiszűrésével, valamint a kivonat hosszának változtatásával is. Míg előbbi három viszonylag gyakran előkerülő kérdéskör a szakirodalomban, addig az utóbbi azonban egy jelentősen alulreprezentált problémakör, ezért is szenteltünk különös figyelmet a vizsgálatának. A kivonatok hossza pedig amiatt is különösen releváns, mivel az üzleti környezetben működő rendszerek esetében gyakran elvárás lehet az ezekre vonatkozó adott karakterlimit betartása (Mehdad és mtsai, 2016), amely szűkítheti az optimális összefoglalók elkészítési lehetőségeit.

A tanulmányunk felépítése a következő: a 2. pontban röviden áttekintjük a vonatkozó szakirodalmat, a 3. pontban röviden bemutatjuk a vizsgálathoz használt adathalmazt. A 4. pontban bemutatjuk a mondatszámok meghatározásához használt megközelítéseket, az 5. pontban átvesszük milyen módszereket alkalmaztunk a vektorizálás során, végül a 6. fejezetben bemutatjuk a kapott eredményeinket.

2. Szakirodalmi áttekintés

Az automatikus tartalmi összefoglalót készítő rendszereket csoportosítani lehet aszerint, hogy azok csak egyetlen dokumentumhoz készítenek összefoglalót (*single document summarization*), vagy ezt dokumentumok egy csoportjához kapcsolódóan végzik el (*multi-document summarization*). A kivonat létrehozásának módja szerint pedig megkülönböztetjük a már említett absztraktív és extraktív metódusokat alkalmazó rendszereket, illetve felügyelt- és felügyelet nélküli gépi tanulást alkalmazókat is.

Extraktív kivonatokra tekintettel (Qiang és mtsai, 2016) például a szövegbányászatban (*text mining*) korábban sikerrel alkalmazott szekvencia feltáró módszerekkel (*sequential pattern mining*) készített több dokumentumra vonatkozó összefoglalókat, (John és mtsai, 2017) pedig ugyanezt a feladatot látens szeman-

tikus analízis (LSA), valamint nem-negatív mátrix faktorizálás (NMF) segítségével oldotta meg.

Egy dokumentumra vonatkozó extraktív összefoglalókat a jogi doménben például (Anand és Wagh, 2019) készített FFNN (Feed-Forward Neural Network) alkalmazásával, míg például (Elaraby és Litman, 2022) az egyes dokumentumok absztraktív összefoglalásának elkészítésében ért el eredményeket, többek között a BART (Lewis és mtsai, 2019) nyelvmodell alkalmazásával[1].

A magyar esetében az elmúlt években szintén a neurális háló alapú megoldások vannak többségben, mind az extraktív, mind az absztraktív kivonatok készítése kapcsán (pl. Yang Zijian és mtsai (2020); Yang Zijian (2022))

A mondatszám, vagy általánosan a kivonat hosszának meghatározása alapvetően három módszerrel közelíthető meg;

- fix érték megadása valamennyi kivonat esetében (*set*),
- kézzel kivonatolt dokumentumok, valamint a belőlük készített kivonatok mondatszámát alapul vévő módszer, például egyenes illesztésével az egyes dokumentumhosszokhoz tartozó kivonatok jellemző hosszának meghatározásához (*empirikus*), vagy
- automatikus meghatározással, például a dokumentumban található témák / témacsoportok szerint, mindegyikből n mondat kiválasztásával (*automatikus*).

Az egyik legkorábbi (modern értelemben vett) kísérlet a kivonatok optimális hosszának meghatározására (Goldstein-Stewart és mtsai, 1999) kísérletéhez kötődik. Ennek során a szerzők (inkább feltáró jelleggel) hírszövegek kézzel készített összefoglalóit vizsgálták meg, és vetették össze egy automatikus kivonatoló rendszer eredményeivel, amely statisztikai és nyelvi jegyeket egyaránt figyelembe vett. A kézzel készített összefoglalókkal történt összevetés során egyik legfontosabb megállapításuk az volt, hogy a kézi összefoglalók hossza jellemzően függetlenül alakult a dokumentum hosszától, továbbá, hogy az nagyjából 75-100 szó hosszúság között kulminált, amely támpontot adhat egy lehetséges fix hosszúság optimális értékéhez.

A neurális hálókon alapuló megoldások közül a Neusum (Zhou és mtsai, 2018) esetén (amely a mondatok pontozása és szelektálása alapján választja meg a kivonatba kerülő egységeket) ismert olyan irányú továbbfejlesztés (Nathan és mtsai, 2020) amelyben a modell a tényleges működés során képes kezelni az eltérő hosszúságú optimális összefoglalók kérdését. A modelltanítás során a szerzők figyelembe vették, hogy egy-egy mondatnak a kivonathoz csatolása meddig képes javítani a kivonat ROUGE pontszámát, és ezt az információt visszavezetve a hálózatba igyekeztek kezelni a nem minden esetben azonos hosszúságú optimális összefoglalók problémáját. Éppen ezért ez a módszer egyszerre tekinthető automatikus és empirikus megoldásnak is.

A kivonat hosszának automatikus meghatározására jó példa lehet az elemi diskurzusegységek (*Elementary Discourse Unit*) alapján (Keskes és mtsai, 2014)

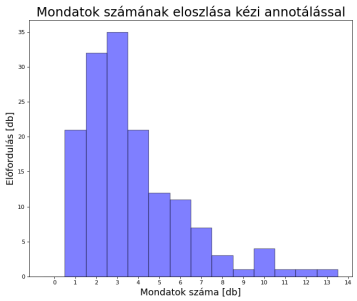
[1] A jogi doménben az elmúlt években (főleg neuronhálókkal) elért eredmények áttekintését részletesen lásd: (Shukla és mtsai, 2022).

történő szegmentálás, amely esetében az alapegységek ugyan nem mondatok, de a cél ugyancsak a szöveg tartalmának pontos reprezentációja változó hosszúságú kivonat elkészítésével.

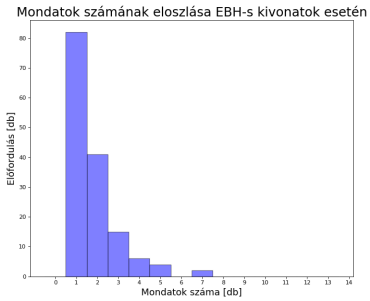
3. Adathalmaz

A vizsgálat elvégzésére 140 darab elvi bírósági határozat mellett 10 darab elvi bírósági döntést választottunk ki. Azért esett az ilyen típusú dokumentumokra a választásunk, mert ezek a dokumentumok valójában egy egyszerű bírósági határozathoz köthetők, tartalmilag lényegében megegyeznek ezekkel, azonban ki vannak bővítve egy pár mondatos kivonattal, amely kivonat az adott ügy jogi lényegét illetve relevanciáját tartalmazza. Így a dokumentum elejéről viszonylag könnyedén kinyerhetők ezek a kivonatok.

A mi célunk azonban az volt, hogy olyan kivonatot készítsünk, amely segít eldönteni, hogy az adott ügy releváns-e a felhasználó számára, vagy sem. Ezért felkértük a Wolters Kluwer Hungary Kft. munkatársait, hogy végezzenek el ilyen szemlélettel extraktív kivonatolást ezeken a dokumentumokon, megjelölve a kivonatként kiválasztott mondatokat. Nem tettünk semmilyen kikötést a mondatok számával kapcsolatban, azonban csak teljes mondatokat lehetett kijelölni. Magát a kivonatolást több ember is végezte. A kézi kivonatolás valamint az EBH-k elején található mondatok számainak eloszlását az 1. ábra mutatja be.



(a) Kézzel annotált



(b) EBH-kból származó

1. ábra. Mondatok számának eloszlása kézi illetve az EBH-k elején szereplő kivonatok esetében

Szembetűnő a kétféle megközelítés közti különbség az eloszlások alapján. Látható, hogy a kézzel annotált esetben sokkal több mondatot tartalmaztak a kivonatok, mint az EBH-k elejéről származó dokumentumok esetében. Az előbbieknél legnagyobb számossággal a 3 mondatos kivonatok bírtak, míg az utóbbiaknál az 1 mondatosak. Mindazonáltal mindkét esetben erős túlsúlyban voltak a rövid,

pár mondatos kivonatok, a kézzel annotált esetben egészen hosszú, 13 mondatos kivonatot is kaptunk, míg az EBH-k esetében a leghosszabb mindössze 7 mondatból állt.

4. Mondatszámok meghatározása

4.1. Mondatszegmentálás

A mondatokra történő bontást a *sentence-splitter*[2] nevű csomag segítségével végeztük el, amely egy többféle nyelven is használható heurisztikus mondatszegmentáló megoldás. Lehetővé teszi működés hangolását egy szólistával, amely olyan példákat tartalmaz, amely nem lehet egy mondat vége. A tapasztalataink alapján ez a megoldás jellemzően túlszegmentálta a mondatokat, így a jellemző hibákat, amelyeket nem lehetett kezelni szótár alapon, csak logikai alapon, reguláris kifejezések használatával javítottuk ki.

Három különböző megközelítést vizsgáltunk meg a mondatszámok meghatározására, melyeket alább mutatunk be részletesen.

4.2. Automatikus meghatározás

Automatikus meghatározást készítettünk oly módon, hogy a mondatokat vektorizáltuk, majd megvizsgáltuk K-Means algoritmust használva 1-10 klaszterre, hogy hogyan alakul a minták és a hozzájuk legközelebbi klaszterközéppont közötti távolságok négyzetösszege. Ezt követően (Satopaa és mtsai, 2011) munkában közöltek alapján határoztuk meg automatikusan az optimális klaszterszámot. Erre mutat be egy példát a 2. ábra.

Az automatikus meghatározás során normalizáljuk a kiszámított klaszterközéppontoktól való távolságokat (y), illetve a kipróbált klaszterszámokat is (esetünkben 1-10). Az optimális klaszterszámot a kezdeti- és végállapotot összekötő egyenes (referencia), valamint a normalizált értékek közti különbségből kaphatjuk meg oly módon, hogy ahol a kettő közti különbség (feketével jelölt az ábrán) a legnagyobb, az az automatikusan meghatározott töréspont (sárgával kiemelve).

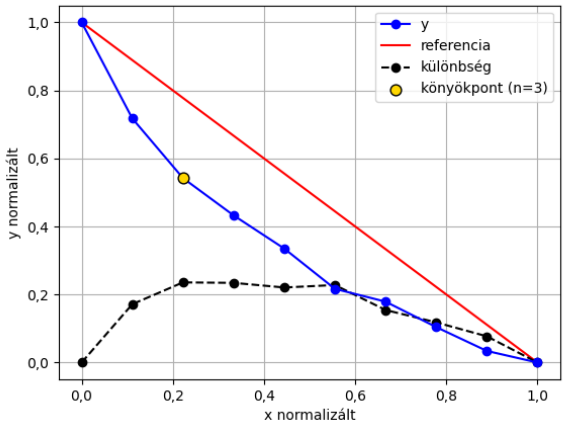
4.3. Egyenes illesztése

Egy másik lehetséges mondatszám-meghatározási módszer a kézzel annotált adat felhasználásával történő meghatározás. Megnéztük, hogy átlagosan hány mondat hosszúak az adott mondatszámú kivonathoz tartozó dokumentumok, majd ezen pontokra illesztettünk egy egyenest (lásd 3. ábra).

Látható, hogy általánosságban igaz volt, hogy az átlagos mondatszám növekedésével több mondat került a kivonatba. Ez némiképpen ellentétes a (Goldstein-Stewart és mtsai, 1999) által levont konklúzióval, miszerint ezek egymástól nagyrészt függetlenek.

A mondatszám meghatározásakor feltételként szabtuk még meg, hogy legalább 1, de maximum 10 mondatot adhat a módszer mondatszámosságként.

[2] https://pypi.org/project/sentence-splitter/



2. ábra. Klaszterszám automatikus meghatározása (Satopaa és mtsai, 2011)

4.4. Egyszerű beállítás

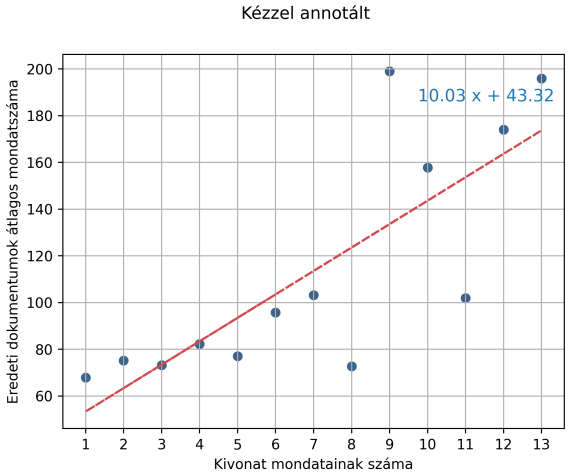
Megvizsgáltuk a legegyszerűbb megközelítést is, amely egy fix érték beállítását jelentette. Itt az emberek által készített kivonatok átlagos számához legközelebb eső értéket vizsgáltuk (4 mondat), valamint ennek közvetlen környezetét (3 illetve 5 mondat mindenhol).

5. Vektorizálás

Az automatikus módszer használatához, illetve a kivonatolás elvégzéséhez elengedhetetlen volt a szövegrészek vektorizálása. Ezt a mondatokra bontott szövegeknél a következő módszerekkel tettük meg:

- **Tf-Idf:** uni- és bigramokat tartottunk meg, kisbetűsítés, írásjel és stopszó szűrést végeztünk, valamint szótöveztünk a *hungarian-stemmer*[3] szótövezővel.
- **Doc2Vec:** 100 dimenziós modellt tanítottunk 5-ös szóablakkal, és megköveteltük a legalább 5-szöri előfordulást, minden más paramétert alapértelmezettként hagytunk (Le és Mikolov, 2014).
- **FastText:** A FastText hivatalos honlapon[4] elérhető magyar nyelvű modellt redukáltuk 100 dimenziósra, majd ennek segítségével a mondatokat vektorizáltuk (Bojanowski és mtsai, 2017; Mikolov és mtsai, 2018).
- **BERT:** A *sentence-transformers* (Reimers és Gurevych, 2019) csomagot alkalmaztuk a *huBERT* alap, nagybetűket is használó modelljével (Nemeskey, 2020, 2021; Devlin és mtsai, 2018).

[3] https://github.com/montana-knowledge-management/hungarian-stemmer
[4] https://fasttext.cc/docs/en/crawl-vectors.html



3. ábra. Átlagos mondatszámokra illesztett egyenes

6. Eredmények és következtetések

6.1. Eloszlások

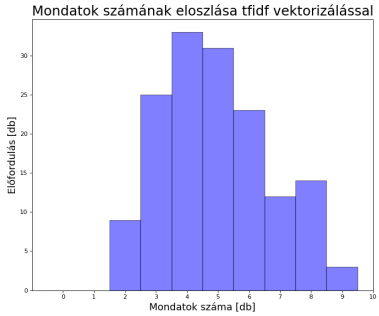
Első lépésként megvizsgáltuk az automatikus, K-Means klaszterezéssel történő mondatszám-meghatározás eredményeként kapott mondatszámok eloszlását. Az eredményeket az 1. táblázat, valamint a 4. ábra mutatja be.

1. táblázat. Különböző vektorreprezentációk hatása automatikus mondatszám-meghatározással

	Átlag	**Szórás**	**Min**	**Max**
kézi	3,72	2,41	1	13
ebh	1,78	1,15	1	7
tfidf	4,94	1,75	2	9
doc2vec	3,36	0,69	2	5
fasttext	3,82	0,82	3	7
bert	3,76	0,72	2	6

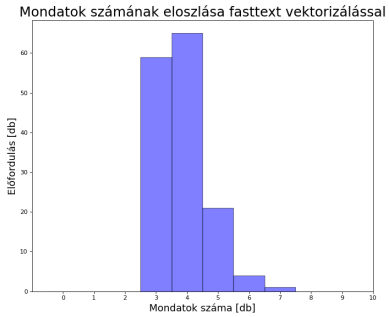
Az átlagértékek vizsgálatából kitűnik, hogy a kézi eredményekhez képest a legközelebb a BERT-tel kapott eredmények estek, másodikként a FastText, majd a Doc2Vec és végül a Tf-Idf.

A 4. ábrán bemutatott eredmények alapján is kijelenthető, hogy jelentős különbségek adódtak a különböző vektorizálási megoldások szerint.

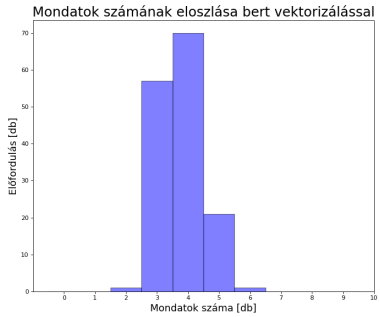


(a) Tf-Idf

(b) Doc2Vec



(c) FastText



(d) BERT

4. ábra. Mondatszám eloszlások automatikus mondatszám-meghatározás esetén

6.2. Átlagos eltérés

Összevetettük a különböző megközelítéseket a kézzel annotált mondatszámokhoz képest tapasztalt eltérések szerint is. Ehhez kiszámítottuk az átlagos négyzetes eltérés (ÁNE, Mean Squared Error, MSE) értéket a kézi annotálás eredményeihez képest. Egyenlet formában:

$$\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{1}$$

ahol x_i az i-edik dokumentumra meghatározott mondatszám, y_i pedig ugyanezen dokumentumra kézzel adott mondatok számát jelenti, N pedig a vizsgálathoz használt dokumentumok száma.

Az eredményeket a 2. táblázat tartalmazza.

2. táblázat. Átlagos négyzetes eltérések a kézzel kiválasztott kivonatok mondatszámához képest

	kézi	**fix 3**	**fix 4**	**fix 5**	**illesztett**	**automatikus**			
						tfidf	**doc2vec**	**fasttext**	**bert**
ÁNE (MSE)	0	6,32	5,88	7,44	9,64	9,38	6,4	6,22	5,76

Látható, hogy a fix beállítások közül a 4 mondatos bizonyult a legjobbnak átlagosan, míg az összes megközelítést figyelembe véve ez bizonyult a második legjobbnak. Ez nem meglepő, hiszen az 1. táblázat alapján a kézi kivonatolással kapott mondatátlaghoz (3,72) a 4 esik a legközelebb. Az abszolút legjobb eredményt a BERT esetében kaptuk az automatikus meghatározás segítségével. Kijelenthető, hogy a vektorizálás fajtája jelentős hatással volt az optimális mondatszám meghatározására. Megfigyelhető egy javuló tendencia a vektorizálási formák között, a legrosszabbnak a Tf-Idf bizonyult, majd sorban a Doc2Vec, FastText és BERT módszerek következtek, a legutóbbi az abszolút legjobb eredményt szolgáltatva. Ez a sorrend megegyezik azzal a sorrenddel, amelybe az egyes módszereket a képességük alapján soroltuk volna.

Érdekes eredmény, hogy a kézi eredményekre illesztett egyenes szolgáltatta a legnagyobb átlagos eltérést az összes vizsgált megközelítés közül. Ennek egyik oka, hogy bár mindegyik kivonatbeli mondatszámhoz a hozzájuk tartozó dokumentumok mondathosszának átlagát vettük, ezek a mondatszámosságok elég széles skálán szóródtak. Így az átlagokra illesztett egyenes is csak hozzávetőlegesen képes pontos eredményt szolgáltatni.

6.3. Mondatszámosság kivonat minőségére gyakorolt hatása

Megvizsgáltuk, hogy a kivonatok minőségében mennyire játszik jelentős szerepet a kiválasztott mondatok száma. A vizsgálathoz kiválasztottuk a két legjobban szereplő megközelítést az automatikus meghatározásból (BERT és FastText) valamint a legrosszabbul teljesítőt (Tf-Idf), a 4 mondatra fixált megoldást, valamint a legrosszabbul teljesítő illesztett megoldást.

A kivonatolást a következőképpen végeztük:

- kiszűrtük a dokumentumokból a kivonat szempontjából nem releváns részeket, ez a dokumentumok bevezető részét, valamint a dokumentum végén található perköltségekről, illetve a keltezést, és bírói neveket tartalmazó részeket jelentette,
- mondatokra bontottuk a szöveget,
- a mondatokat vektorizáltuk Tf-Idf, BERT és FastText módszerekkel,
- a TextRank (Mihalcea és Tarau, 2004) algoritmus segítségével sorba rendeztük a mondatokat,
- kivettünk a legrelevánsabb mondatok közül annyit, amennyit az adott mondatszámosság-meghatározó megközelítés adott.

Fontos megjegyezni, hogy az automatikus megközelítések esetében a vektorreprezentációk megegyeztek az optimális mondatszám-meghatározás valamint a kivonatolás közben.

Az így kapott kivonatokat összehasonlítottuk a kézi kivonatokkal a következő metrikák alapján: ROUGE-1, ROUGE-2, ROUGE-L F1 értékek, valamint kiszámítottuk a mondatok FastText és sentence BERT vektorreprezentánsait mind a gépi, mind a kézi kivonatok esetében is, melyek átlagai között mértünk cos távolságot (lásd 5. ábra). Ez utóbbi két metrikával igyekeztük a jelentésbeli közelségét

5. ábra. Kivonatok jelentéstartam szerinti összehasonlítása

mérni a kivonatoknak. A Rouge metrikák mérése előtt mindkét kivonatot kisbetűsítettük, szótöveztük, illetve kiszűrtük az írásjeleket. Ezeket az előfeldolgozási lépéseket nem végeztük el a FastText és a BERT cos távolságok számítása során.

A ROUGE metrikák olyan metrikák gyűjteményét jelentik, amelyek segítségével gépi fordítással készült szövegeket, vagy automatikus tartalmi összefoglalókat készítő szoftverek teljesítményét szokás számszerűsíteni Lin (2004). Segítségükkel meghatározható a kivonatolás kimenetének pontossága és fedése. Pontosság esetében arról kapunk visszajelzést, hogy a gép által az összefoglalóba beválogatott szavakból mennyi azok aránya, amelyek helyesen kerültek be az összefoglalóba, a fedés pedig azt mutatja meg, hogy a megtalálandó szavak közül mekkora arányban adta azokat vissza az algoritmus. Az egyes ROUGE metrikák e téren abban térnek el, hogy csak egy-egy szót tekintenek egységnek (ROUGE-1), két egymás melletti szó jelent egy egységet (ROUGE-2), vagy pedig az elvárt, és a kapott szövegek leghosszabb közös (nem feltétlenül megszakítás nélkül egymás után következő) részét (ROUGE-L).

Az eredményeket a 3. táblázat mutatja be, félkövérrel kiemelve a legjobb értékeket.

3. táblázat. Mondatszám-meghatározási módszer hatása a kivonat minőségére

Módszer	**Vektorforma**	**FastText cos**	**BERT cos**	**Rouge-1**	**Rouge-2**	**Rouge-L**
auto	BERT	0,963	0,9828	0,373	0,2008	0,259
fix 4	BERT	0,9643	0,9831	0,3825	0,2111	0,2669
illesztett	BERT	0,9552	0,9805	0,3291	0,1671	0,2262
auto	FastText	0,9689	0,9835	0,4083	0,2348	0,2873
fix 4	FastText	**0,97**	**0,9839**	**0,4142**	0,2387	0,2907
illesztett	FastText	0,9636	0,9805	0,3687	0,2034	0,259
auto	Tf-Idf	0,9563	0,98	0,4097	**0,2514**	**0,303**
fix 4	Tf-Idf	0,955	0,9795	0,4009	0,242	0,2967
illesztett	Tf-Idf	0,9481	0,9766	0,3719	0,22	0,2745

Az abszolút legjobb eredményt a cos távolságok esetén a fixen 4 mondatból álló FastText megoldás szolgáltatta, illetve ugyanez a megoldás bizonyult a legjobbnak a Rouge-1 metrika esetében is. A Rouge-2 és Rouge-L metrikáknál azonban az automatikus Tf-Idf megközelítés bizonyult a legjobbnak. Érdekesség, hogy a cos távolságok nagyon kis mértékben szórtak a BERT cos metrika esetében: a legjobb és legrosszabb értékek között a különbség mindössze 0,0073 volt, szemben a FastText cos távolságnál tapasztalt 0,0219-es értékkel. Ennek valószínű okát sikerült kiderítenünk. Kézzel egy példamondaton leellenőriztük, hogy az általunk alkalmazott `sentence-transformer` megoldás ugyanazt a vektorreprezentációt szolgáltatja-e mint amikor kézzel kiválasztjuk a [CLS] token vektorreprezentációját, és azt tapasztaltuk, hogy ez a két vektor egymástól eltérő volt.

Mindhárom vektorizálási forma esetében igaz volt, hogy a különböző mondatszám-meghatározási módszerek közül az illesztett függvény segítségével meghatározott eset bizonyult a legrosszabbnak. A 2. táblázat alapján is ez a mondatkiválasztási metódus bizonyult a legrosszabbnak.

Az összes vektorforma esetében azonban az automatikus és a fix meghatározás viszonylag hasonló eredményt szolgáltatott, de a BERT és FastText esetekben a fix 4-es mondatkiválasztás esetén kaptuk a legjobb megoldásokat minden metrikánál, a Tf-Idf vektorforma esetében a legjobb eredményeket az automatikus meghatározás szolgáltatta. Ez meglepő eredmény a 2. táblázat tükrében, ugyanis a második legrosszabbnak az automatikus Tf-Idf bizonyult.

Az automatikus Tf-Idf kiemelkedő szereplésének feltehető oka, hogy a módszer átlagosan több, mint egy mondattal több mondatot választott a kivonatba (lásd 1. táblázat), mint a többi megoldás, ily módon nagyobb eséllyel tartalmazhatott olyan szövegrészeket, amelyek a kézi kivonatban is szerepeltek.

Az eredmények alapján tehát kijelenthető, hogy a mondatszám-meghatározás hatással van ugyan a kivonatok minőségére, azonban nem olyan mértékben, mint ami a 2. táblázatból következett volna. Kiszámítva az egyes vektorformák ese-

tén a különböző mondatszám-meghatározási módszerek esetén a legjobb és legrosszabb értékek közti különbségeket FastText cos metrika alapján, a következőket kapjuk: BERT: 0,0091, FastText: 0,0064, Tf-Idf: 0,0082. Megvizsgálva azt, hogy az egyes vektorformákkal kapott eredmények átlagai között mekkora a legjobb és legrosszabb közti eltérés, 0,014-et kaptunk. Az eredmények alapján arra a következtetésre juthatunk, hogy a mondatszám-meghatározás a vektorizálással összemérhető, azonban annál kisebb hatással bír a kivonat minőségére, ezért egy automatikus kivonat készítő projekt esetében fontos hangsúlyt fektetni az ideális hosszúság meghatározásra is.

7. Összefoglalás

Cikkünkben egy gyakorlati problémának, név szerint jogi dokumentumok extraktív kivonatolásának egy részfeladatát mutattuk be. A jellemzően több ezer szavas jogi határozatokból igyekeztünk kiválasztani azokat a mondatokat, amelyek a legjobban leírják az adott ügyet, ezzel segítve a jogi adatbázis felhasználóit, hogy könnyebben eldönthessék az adott ügy számukra releváns-e, vagy sem.

Extraktív, nem felügyelt kivonatolás esetében négy fő módon van befolyásunk a kivonat minőségére, a szöveg előszűrésével, a szöveg vektorreprezentációjával, a szövegrészek fontosság szerinti sorba rendezés megoldásával, valamint a kivonat szöveghosszának változtatásával. Jelen cikkben a kivonatok mondatszámának hatását vizsgáltuk a kivonatok minőségére.

Megmutattuk, hogy a mondatszám a vektorreprezentációs formával összemérhető, azonban annál kisebb hatással van a kivonat minőségére.

Hivatkozások

Anand, D., Wagh, R.: Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences (2019)

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the association for computational linguistics 5, 135–146 (2017)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

Elaraby, M., Litman, D.: Arglegalsumm: Improving abstractive summarization of legal documents with argument mining (2022), https://arxiv.org/abs/2209.01650

Goldstein-Stewart, J., Kantrowitz, M., Mittal, V., Carbonell, J.G.: Summarizing text documents: sentence selection and evaluation metrics. In: SIGIR '99 (1999)

John, A., Premjith, P., Wilscy, M.: Extractive multi-document summarization using population-based multicriteria optimization. Expert Systems with Applications 86, 385–397 (2017)

Keskes, I., Zitoune, F.B., Belguith, L.H.: Splitting arabic texts into elementary discourse units. ACM Transactions on Asian Language Information Processing 13(2) (jun 2014), https://doi.org/10.1145/2601401

Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196. PMLR (2014)

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019), https://arxiv.org/abs/1910.13461

Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)

Mehdad, Y., Stent, A., Thadani, K., Radev, D., Billawala, Y., Buchner, K.: Extractive summarization under strict length constraints. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 3089–3093. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://aclanthology.org/L16-1493

Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. pp. 404–411 (2004)

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)

Nathan, Z., Yijun, J., Yi, L.: Sentence-level extractive text summarization (March 2020), https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/-custom/15790001.pdf, [Online; posted 20-March-2020]

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)

Nemeskey, D.M.: Introducing `huBERT`. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). p. TBA. Szeged (2021)

Nenkova, A., McKeown, K.: Automatic summarization. Foundations and Trends® in Information Retrieval 5(2–3), 103–233 (2011), http://dx.doi.org/10.1561/1500000015

Qiang, J.P., Chen, P., Ding, W., Xie, F., Wu, X.: Multi-document summarization using closed patterns. Knowledge-Based Systems 99, 28–38 (2016)

Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084

Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops. pp. 166–171. IEEE (2011)

Schluter, N.: The limits of automatic summarisation according to rouge. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 41–45 (2017)

Shukla, A., Bhattacharya, P., Poddar, S., Mukherjee, R., Ghosh, K., Goyal, P., Ghosh, S.: Legal case document summarization: Extractive and abstractive methods and their evaluation (2022), https://arxiv.org/abs/2210.07544

Vági, R.: Szemantikai keresés és predictive coding a jogi munkában. In: Ződi, Zs. (szerk.) Jogi technológiák - digitális jogalkalmazás, pp. 141–156. Ludovika Egyetemi Kiadó (2022)

Yang Zijian, Gy.: Barterezzünk! messze, messze, messze a világtól, bart kísérleti modellek magyar nyelvre. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–29. Szegedi Tudományegyetem, Szeged (2022)

Yang Zijian, Gy., Perlaki, A., Laki, L.J.: Automatikus összefoglaló generálás magyar nyelvre bert modellel. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–353. Szegedi Tudományegyetem, Szeged (2020)

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences (2018), https://arxiv.org/abs/1807.02305

Data Augmentation for Machine Translation via Dependency Subtree Swapping

Attila Nagy[1], Dorina Petra Lakatos[1,2], Botond Barta[1,2], Patrick Nanys[1], Judit Ács[2]
attila.nagy234@gmail.com, patrick.nanys2000@gmail.com, {dorinalakatos, botondbarta, acsjudit}@sztaki.hu

[1]Department of Automation and Applied Informatics
Budapest University of Technology and Economics
[2]Institute for Computer Science and Control
Eötvös Loránd Research Network

Abstract. We present a generic framework for data augmentation via dependency subtree swapping that is applicable to machine translation. We extract corresponding subtrees from the dependency parse trees of the source and target sentences and swap these across bisentences to create augmented samples. We perform thorough filtering based on graph-based similarities of the dependency trees and additional heuristics to ensure that extracted subtrees correspond to the same meaning. We conduct resource-constrained experiments on 4 language pairs in both directions using the IWSLT text translation datasets and the Hunglish2 corpus. The results demonstrate consistent improvements in BLEU score over our baseline models in 3 out of 4 language pairs. Our code is available on GitHub[1].

1 Introduction

Parallel data is a necessity for building performant neural machine translation (NMT) systems. For high- and medium-resource languages, millions of parallel data points enable researchers to build translation models of high quality. However, in a resource-constrained setting, such as low-resource or domain-specific machine translation, the lack of data must be compensated with a variety of techniques to improve performance. These include training multilingual models for zero- and few-shot learning (Johnson et al., 2017; Sharaf et al., 2020), transfer learning (Kocmi and Bojar, 2018) and a wide variety of data augmentation techniques that alter existing parallel sentences to create more training data (Sennrich et al., 2016; Zhang and Zong, 2016; Fadaee et al., 2017).

In this paper, we present a framework for generating augmented samples with the help of syntactic information for machine translation. We benchmark our augmentation method in a low-resource setting on 4 language pairs, using low-resource datasets for English-German, English-Hebrew and English-Vietnamese and a subsample of a high-resource dataset for English-Hungarian.

[1] https://github.com/attilanagy234/syntax-augmentation-nmt

2 Related work

Data augmentation (DA) involves a set of techniques that enhance the data used to train a machine learning model both in size and variety. It has been widely applied in NLP for fixing class imbalance, mitigating bias, making the model more robust through adverserial examples or increasing model accuracy (Feng et al., 2021). DA methods that incorporate syntactic knowledge have been applied to a number of NLP tasks, but not particularly in the domain of machine translation. For part-of-speech tagging, Şahin and Steedman (2018) augment dependency trees by removing dependency links and moving tree fragments around the root. This dependency tree morphing method was also shown effective for dependency parsing by Vania et al. (2019). Xu et al. (2016) improve relation classification by making use of the directionality of relationships in a dependency tree. The idea of swapping compatible subparts of datapoints to generate augmented samples has shown performance improvements in dependency parsing (Dehouck and Gómez-Rodríguez, 2020), named entity recognition (Dai and Adel, 2020) and constituency parsing (Shi et al., 2020). Shi et al. (2021) introduce a generalized framework for substructure substitution, which produces new samples by swapping same-label substructures.

For machine translation, the most common augmentation method is back-translation (Sennrich et al., 2016), where additional training data is obtained by translating a target-language monolingual corpus into the source language, using a baseline model trained with the originally available data. Fadaee et al. (2017) targets low-frequency words by generating new sentence pairs with rare words in a new context. Wang et al. (2018) propose SwitchOut, a technique, which randomly replaces words in both the source and target sentences with other random words from their corresponding vocabularies. Gao et al. (2019) introduce Soft Contextual DA, where they augment a randomly chosen word in a sentence by its contextual mixture of multiple related words. Nguyen et al. (2020) diversify the training data by using the predictions of multiple forward and backward models and then merging them with the original dataset on which the final NMT model is trained. Moussallem et al. (2019) use knowledge graphs to enhance semantic feature extraction and hence the translation of entities and terminological expressions. Chen et al. (2020) create constraint-aware training data by first randomly sampling the phrases of the reference as constraints, and then packing them together into the source sentence with a separation symbol. Wei et al. (2020) propose an uncertainty-aware semantic DA method, which explicitly captures the universal semantic information among multiple semantically-equivalent source sentences and enhances the hidden representations with this information for better translations. Sánchez-Cartagena et al. (2021) present an approach, where augmented sentence pairs are generated with simple transformations and used as auxiliary tasks in a multi-task learning framework with the aim of providing new contexts where the target prefix is not informative enough to predict the next word. Wei et al. (2022) suggest Continuous Semantic Augmentation (CSANMT), which augments each training instance with an adjacency semantic region that could cover adequate variants of literal expression under the same

meaning. A method, which uses dependency parsing for data augmentation was introduced by Duan et al. (2020). They perform simple token-level manipulations, such as blanking, dropout and replacement and use the depth of tokens in the dependency tree as guidance for computing word-selection probabilities.

3 Methodology

In this section, we introduce the data augmentation and the graph-based filtering methods used in our experiments in detail. Figure 1 gives an overview of the augmentation algorithm starting from pairs of parallel sentences or bisentences.

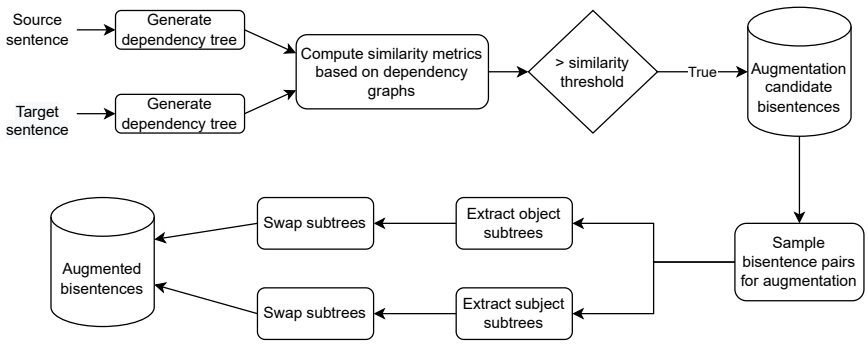


Fig. 1: Overview of our data augmentation method

3.1 Dependency subtree swapping

We propose a simple, yet effective data augmentation method for machine translation based on dependency trees. The intuition behind our approach is that elements of a translation pair - although they are of a different language - show syntactic similarities. The central idea is to extract corresponding syntactic substructures in the source and target sentences and swap these across bisentences to produce augmented datapoints. One clear advantage of this method compared to many other non-model based data augmentation methods for NMT, is that this approach alters the source and target sentences simultaneously, so it has a better chance of maintaining parallelism in the augmented samples. Depending on the language and the phrasing, associated substructures could have different composition, although we hypothesize that subjects and objects have a similar enough representation across many language-pairs to be used for this kind of augmentation. We consider object and subject subtrees in the dependency trees that correspond to OBJ and NSUBJ edges defined in the Universal Dependencies (Nivre et al., 2020). Figure 2 shows an example of the object subtrees in a

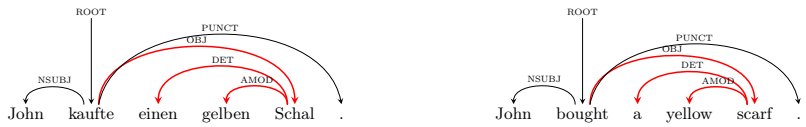


Fig. 2: Example of a object substructures in the dependency trees of a German-English translation pair. The corresponding subtrees are highlighted in red.

German-English translation pair. Figure 3 explains the technique of object and subject subtree swapping through an example bisentence pair.

Applying the method in a more generic setting to more complex syntactic substructures is possible. However, as the performance of the algorithm is dependent on both the translation quality and the quality of the dependency parser, finding correspondance between more complex subtrees could be hard and that may result in injecting too much noise into the training data. Nevertheless, this is an interesting future direction of research, which should be explored.

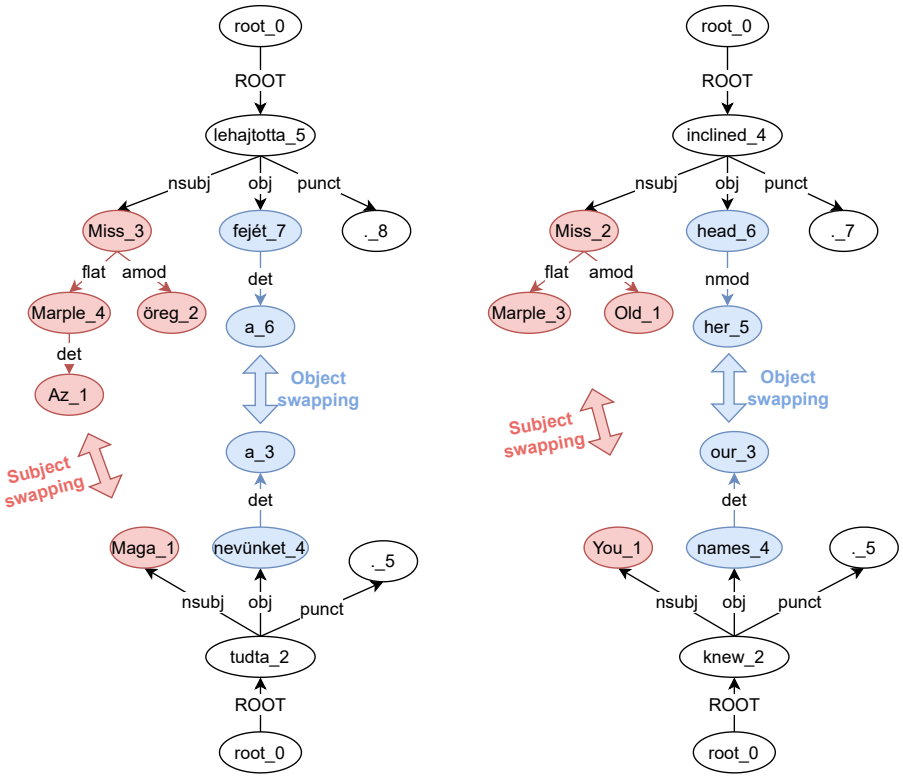


Fig. 3: Two kinds of augmentation techniques: object and subject subtree swapping.

3.2 Generating augmentation candidates

Our prior experiments have shown that although the number of samples that we can generate grows exponentially with respect to the size of the training data, the generated bisentences can be very noisy (Nagy et al., 2022). In this section, we discuss the rules and heuristics that we implemented to ensure that the generated bisentences are high quality.

Filtering heuristics As the augmentation method is based on swapping certain subtrees, it is only possible to sample from a subset of the training data, where the dependency trees contain the required edges. We consider a bisentence eligible for augmentation, if the dependency trees of both the source and target sentences have at least one NSUBJ and one OBJ edges. We also constrain that they cannot contain more than one of each dependency edge type. In order to further reduce noise, we exclude bisentences, where the root of the selected subtrees of the source and target sentences do not belong to the same part-of-speech tag. We also constrain, that every subtree eligible for swapping must contain either a noun or a proper noun.

Graph-based sampling As the method is sensitive to both the translation quality and the performance of the dependency parser, we need to ensure that the selected subtrees capture the same meaning in both the source and target sentences. We hypothesize that if the syntactic composition of the subtrees of the source and target sentences are significantly different, the subtrees likely have a different meaning. To study the syntactic similarities of the dependency trees, we explore two graph-based methods: Graph Edit Distance (GED) (Sanfeliu and Fu, 1983) and Edge Mapping (EM).

Graph Edit Distance is a generalization of the Levenshtein distance (Levenshtein et al., 1966) for graphs. The GED value represents the minimal cost of graph edit operations, which is required to transform the first graph into the second one. The operations we used for nodes and edges are insertion, deletion and substitution. A deletion or an insertion costs 1, while the substitution costs 2. To compare these numbers concerning the size of the graphs, we normalized the edit distance with the following equations:

$$\begin{aligned} d_{\max} &= 2 * |V_1| - 1 + 2 * |V_2| - 1 \\ \mathrm{sim}(G_1, G_2) &= \frac{d_{\max} - \mathrm{GED}(G_1, G_2)}{d_{\max}} \end{aligned} \tag{1}$$

where $d_{\max}$ corresponds to the maximum distance between 2 graphs, when all of the edges and nodes are deleted from G_1 and every node and edge in G_2 is inserted. V_1 and V_2 are the set of vertices in G_1 and G_2 respectively. The $\mathrm{sim}(G_1, G_2)$ is a similarity measure between G_1 and G_2, and its value is between 0 and 1, 1 if the two graphs are isomorphic.

The Edge Mapping (EM) algorithm is based on a general graph similarity algorithm (Champin and Solnon, 2003). A modified version of this is described in

Algorithm 1 Edge mapping

Require: $G_1(V_1, E_1), G_2(V_2, E_2)$
  $m \leftarrow \{\}$
  for all $e_1 \in E_1$ **do**
    cands $\leftarrow \{e_2 \mid e_2 \in E_2, e_2 \notin m, e_1 = e_2\}$
    if cands is empty **then**
      continue
    end if
    cands $\leftarrow \arg\max_{c \in \text{cands}} \text{score}(e_1, c)$
    cands $\leftarrow \arg\max_{c \in \text{cands}} \text{route_sim}(e_1, c)$
    $m[e_1] \leftarrow \text{cands}[0]$
  end for
  return m

Algorithm 1. The score(e_1, e_2) represents how many nodes are common between the two edges (2 at max). route_sim(e_1, e_2) is the Levenshtein distance between the root $- e_1$ and root $- e_2$ routes, where every route is defined by the part-of-speech tag of the visited nodes. The algorithm gives a mapping between the edges of the graphs, which we consider as the intersection. This way we can calculate a Jaccard index between the edges, which gives us another similarity measure:

$$J(G_1, G_2) = \frac{|m|}{|E_1| + |E_2| - |m|} \tag{2}$$

where m is the mapping, E_1 and E_2 are the set of edges in G_1 and G_2 respectively. Figure 4 compares the two kinds of similarity measures through the object subtree of a sentence.

4 Experiments

We conduct experiments on 4 language pairs in both directions: English–German, English–Hebrew, English–Vietnamese and English–Hungarian. We trained baseline models for all language pairs, which we benchmark against models trained using additional augmented data.

4.1 Datasets

Following other works in the field (Sánchez-Cartagena et al., 2021; Gao et al., 2019; Guo et al., 2020; Wang et al., 2018), we use datasets from the text translation track of IWSLT. The IWSLT datasets are transcripts of Ted and TedX talks and are regularly used for benchmarking data augmentation methods for NMT in a low-resource setting. For English-German and English-Hebrew, we use the IWSLT 2014 (Cettolo et al., 2014) text translation track for training and we use the *tst2013* and *tst2014* datasets for development and test sets respectively.

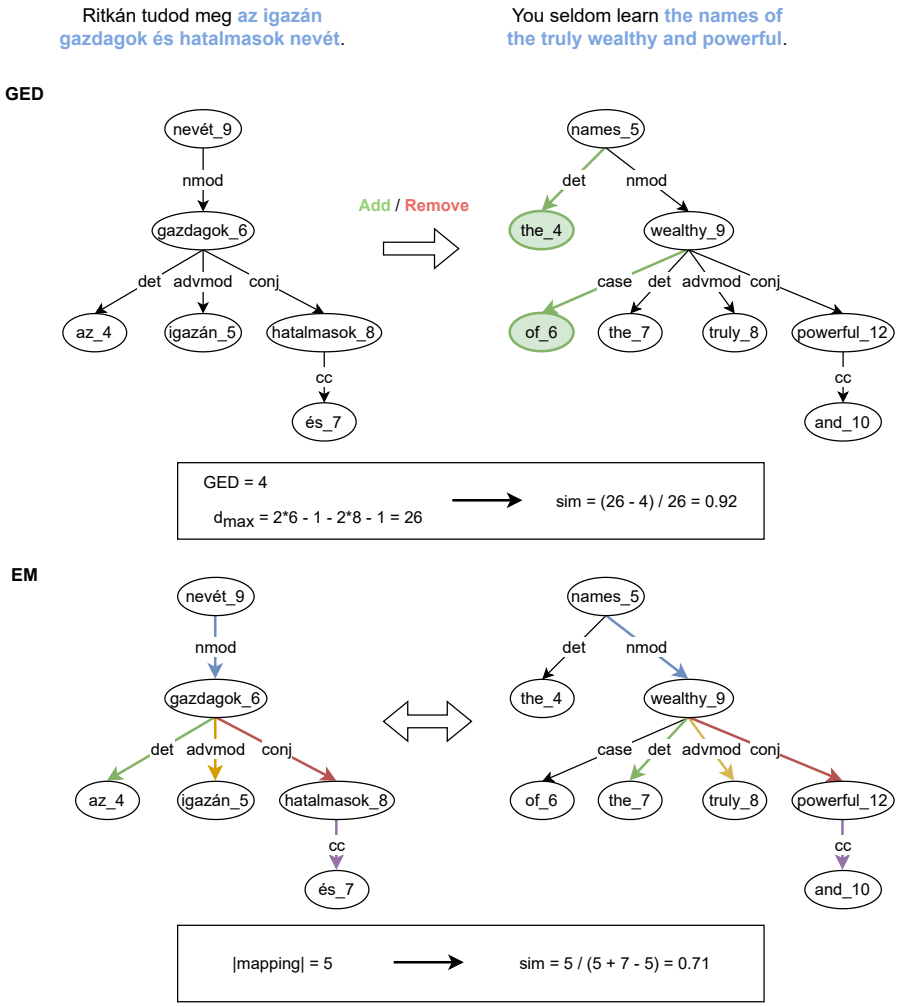


Fig. 4: The two kinds of graph similarity measure of a bisentence's object subtree. For the GED method the added and removed vertices/edges are colored green and red respectively. In the case of the EM measure, the colors of the edges represent the mapping.

For English-Vietnamese, we select the IWSLT 2015 (Cettolo et al., 2015) text translation track for training, whereas the *tst2012* and *tst2013* datasets are used for development and test sets respectively. For English-Hungarian, we perform experiments on the Hunglish2 corpus (Varga et al., 2007), which is a sentence-aligned dataset. It was constructed by scraping and aligning bilingual text in several domains, such as literature, movie subtitles, software documentation and legal text. To simulate a low-resource experiment, we created a stratified subsample of 250k sentences based on the domains and created a train-dev-test splits from this, also with stratified sampling.

We apply the same preprocessing steps for all language pairs. We remove sentences longer than 32 tokens. We also remove sentences where the difference of the source- and target-side token counts are more than 7 and their ratio is more than 1.2. We also remove leading and trailing quotation marks and dashes. We perform language detection on both the source and target sentences using *fastText* (Joulin et al., 2016) and remove datapoints whenever the languages mismatch with what is expected. We normalize punctuations with *sacremoses*[2]. The statistics of the the preprocessed datasets can be seen in Table 1.

Language pair	**train**	**dev**	**test**
En-De	127,506	993	1,305
En-He	132,105	1,382	962
En-Vi	89,188	1,553	1,268
En-Hu	212,500	37,500	21,700

Table 1: Number of bisentences in the train/dev/test sets for each language pair.

4.2 Training details

We use the same Transformer-based encoder-decoder architecture (Vaswani et al., 2017) for every experiment. The hyperparameters used in each experiment can be seen in Table 2. All of the models were trained on a single A100 GPU for 4 hours using the openNMT framework (Klein et al., 2017). We used early stopping to avoid overfitting on the training set based on the perplexity of the validation set. For tokenization, part-of-speech tagging and dependency parsing, we use HuSpacy (Orosz et al., 2022) for Hungarian and Stanza (Qi et al., 2020) for the rest of the languages.

For each experiment, we generate augmented data and shuffle it with the training sets before training. We explore data generation with different augmentation configurations, including the graph similarity metrics (GED and EM),

[2] https://github.com/alvations/sacremoses

Parameter	**Value**	**Parameter**	**Value**
batch type	tokens	batch size	12288
accumulation count	4	average decay	0.0005
train steps	150000	valid steps	5000
early stopping	4	early stopping criteria	ppl
optimizer	adam	learning rate	2
warmup steps	8000	decay method	noam
adam beta2	0.998	max grad norm	2
label smoothing	0.1	param init	0
param init glorot	true	normalization	tokens
max generator batches	32	encoder layers	8
decoder layers	8	heads	16
RNN size	1024	word vector size	1024
Transformer FF	2096	dropout steps	0
dropout	0.1	attention dropout	0.1
share embeddings	true	position encoding	true

Table 2: Hyperparameters of the models.

the subtree type that is swapped (Subj, Obj) and different augmentation ratios (0.5, 2, 3). For the graph similarity filtering, we use 0.4 as a generic threshold for all language pairs. We found during qualitative analysis of augmented data points that this value yields high quality and diverse results. Selecting a higher threshold value does not necessarily mean better quality augmentations, because in practice, a very high similarity often means that the subtrees are small and this way we cannot guarantee with high confidence that the subtrees correspond to the same meaning. The number of sentence pairs with similarity above a given threshold between the source and target sentences are shown for all language-pairs in Figure 5.

5 Results

We measure the effectiveness of the proposed method with BLEU scores of NMT models trained with and without augmented data. Figure 6 shows the BLEU scores for each dataset and each augmentation setup. It is clear that there is no *one size fits all*, but with the exception of Vietnamese, at least one configuration is always better than the baseline with a non-negligible margin. Unfortunately the Vietnamese dependency parser is not very high quality, which explains the subpar performance after our augmentation. We find an upward trend in the other three language pairs, in other words, more augmented data seems to help, although the exact augmentation ratio needs to be determined separately for

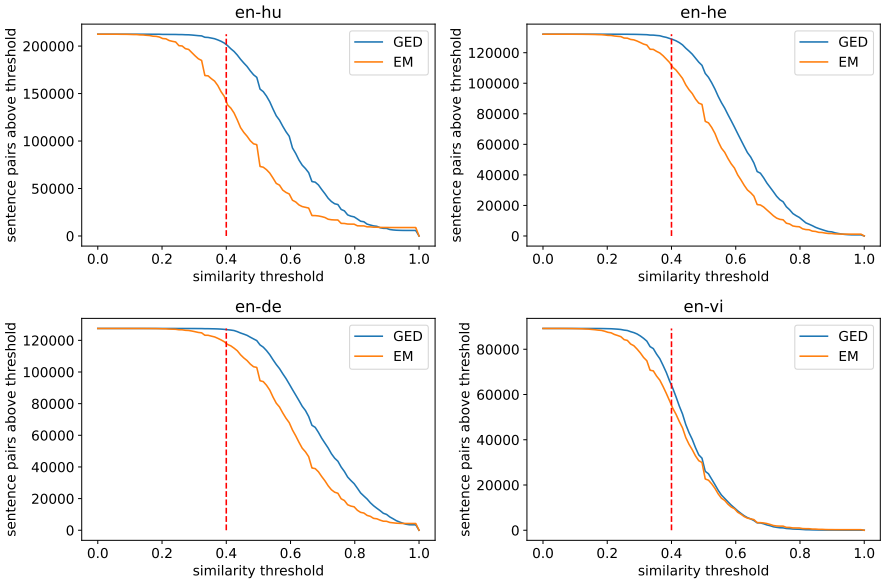


Fig. 5: The number of sentence pairs with similarity above a given threshold for both the GED and EM methods computed between the dependency parse trees of the source and target sentences.

each dataset. Table 3 reports the BLEU scores of the baseline models and the best performing models' hyperparameters and results.

Language pair	**Baseline**	**Graph sampling**	**Aug type**	**Aug ratio**	**Aug BLEU**
En-De	20.1	GED	Subj	3	21.3
De-En	23.0	GED	Obj	3	24.5
En-He	17.1	EM	Obj	3	17.5
He-En	25.2	GED	Obj	2	27.0
En-Vi	21.8	EM	Subj	3	21.6
Vi-En	22.1	EM	Obj	0.5	21.7
En-Hu	18.1	EM	Subj	2	20.0
Hu-En	21.7	EM	Subj	2	23.6

Table 3: Augmentation parameters of the best performing models for each language pair.

In general we observed that the augmentation method starts to improve the performance of the models considerably if the augmentation ratio is 2 or higher.

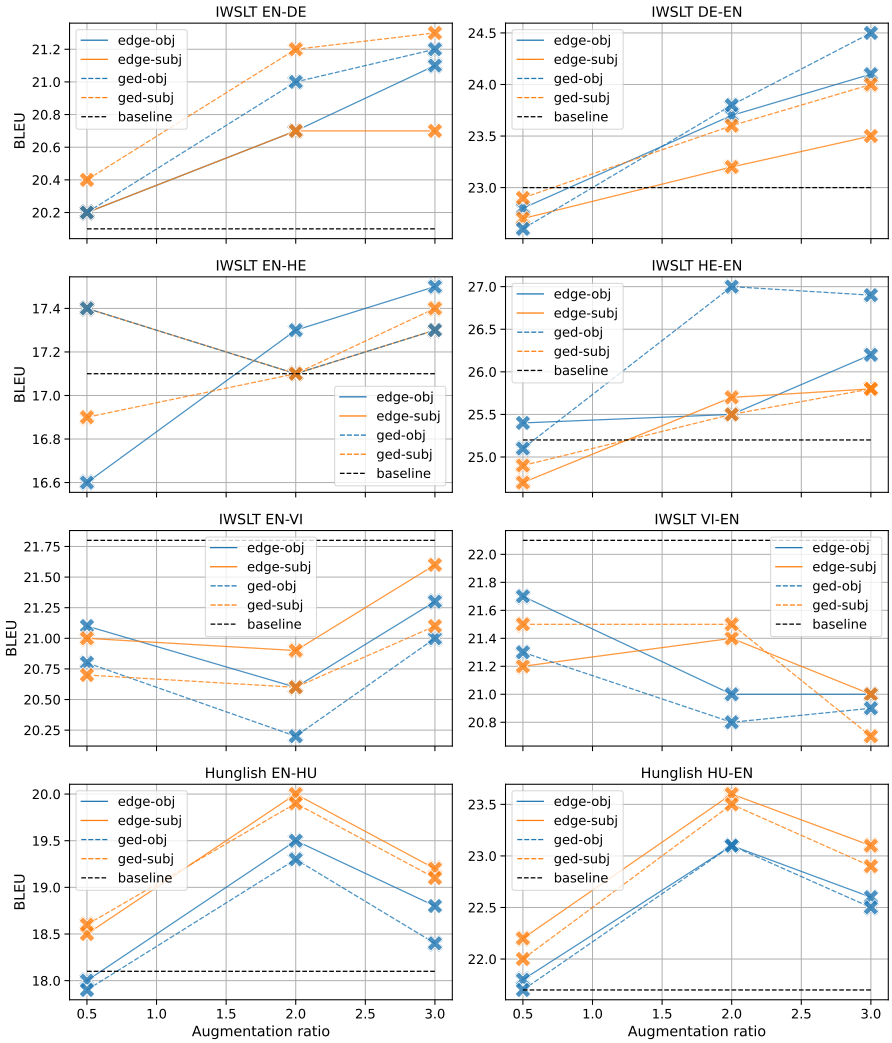


Fig. 6: Test BLEU scores for each augmentation method and augmentation ratio. We indicate the baseline results with a horizontal black dashed line.

This means augmenting a dataset of size X with at least $2X$ amount of new samples. The choice of augmentation type (Object or Subject) seems to depend on the language. Subject is usually better, particularly for Hungarian, but it is less obvious in other languages. The same can be said about the filter type (GED or EM). One trend, we see is that the same augmentation type and ratio usually yields similar results regardless of the filter type. This suggests that the filter methods we use are somewhat robust.

The augmentation methods did not improve the scores for English-Vietnamese in neither directions. We suspect that this is because Stanza has relatively poor performance on tokenization and dependency parsing for Vietnamese[3] . It would be interesting to reproduce the experiments with a better NLP pipeline for Vietnamese and compare the results.

6 Conclusion

We presented an augmentation method for machine translation based on dependency subtree swapping. We applied two types of graph distance based filtering to discard low quality augmentation. We demonstrated the effectiveness of our method by performing resource-constrained experiments on 4 language pairs in both directions. With the exception of English–Vietnamese, we see consistent improvement over the non-augmented baselines in English–German, English–Hebrew and English–Hungarian.

Acknowledgements

The authors would like to thank András Kornai and Csaba Oravecz for discussions on the augmentation methodology.

[3] https://stanfordnlp.github.io/stanza/performance.html

References

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., Federico, M.: The IWSLT 2015 evaluation campaign. In: Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign. pp. 2–14. Da Nang, Vietnam (Dec 3-4 2015), https://aclanthology.org/2015.iwslt-evaluation.1

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Federico, M.: Report on the 11th IWSLT evaluation campaign. In: Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign. pp. 2–17. Lake Tahoe, California (Dec 4-5 2014), https://aclanthology.org/2014.iwslt-evaluation.1

Champin, P.A., Solnon, C.: Measuring the similarity of labeled graphs. vol. 2689 (07 2003)

Chen, G., Chen, Y., Wang, Y., Li, V.O.: Lexical-constraint-aware neural machine translation via data augmentation. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 3587–3593. International Joint Conferences on Artificial Intelligence Organization (7 2020), https://doi.org/10.24963/ijcai.2020/496, main track

Dai, X., Adel, H.: An analysis of simple data augmentation for named entity recognition. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 3861–3867. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020), https://aclanthology.org/2020.coling-main.343

Dehouck, M., Gómez-Rodríguez, C.: Data augmentation via subtree swapping for dependency parsing of low-resource languages. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 3818–3830. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020), https://aclanthology.org/2020.coling-main.339

Duan, S., Zhao, H., Zhang, D., Wang, R.: Syntax-aware data augmentation for neural machine translation. arXiv preprint arXiv:2004.14200 (2020)

Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 567–573. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), https://aclanthology.org/P17-2090

Feng, S.Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., Hovy, E.: A survey of data augmentation approaches for NLP. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 968–988. Association for Computational Linguistics, Online (Aug 2021), https://aclanthology.org/2021.findings-acl.84

Gao, F., Zhu, J., Wu, L., Xia, Y., Qin, T., Cheng, X., Zhou, W., Liu, T.Y.: Soft contextual data augmentation for neural machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5539–5544. Association for Computational Linguistics, Florence, Italy (Jul 2019), https://aclanthology.org/P19-1555

Guo, D., Kim, Y., Rush, A.: Sequence-level mixed sample data augmentation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5547–5552. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.447

Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al.: Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics 5, 339–351 (2017)

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)

Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: Open-source toolkit for neural machine translation. In: Proceedings of ACL 2017, System Demonstrations. pp. 67–72. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), https://www.aclweb.org/anthology/P17-4012

Kocmi, T., Bojar, O.: Trivial transfer learning for low-resource neural machine translation. In: Proceedings of the Third Conference on Machine Translation: Research Papers. pp. 244–252. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), https://aclanthology.org/W18-6325

Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710. Soviet Union (1966)

Moussallem, D., Arčan, M., Ngomo, A.C.N., Buitelaar, P.: Augmenting neural machine translation with knowledge graphs. arXiv preprint arXiv:1902.08816 (2019)

Nagy, A., Nanys, P., Konrád, B.F., Bial, B., Ács, J.: Syntax-based data augmentation for hungarian-english machine translation. arXiv preprint arXiv:2201.06876 (2022)

Nguyen, X.P., Joty, S., Wu, K., Aw, A.T.: Data diversification: A simple strategy for neural machine translation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 10018–10029. Curran Associates, Inc. (2020)

Nivre, J., de Marneffe, M.C., Ginter, F., Hajič, J., Manning, C.D., Pyysalo, S., Schuster, S., Tyers, F., Zeman, D.: Universal Dependencies v2: An evergrowing multilingual treebank collection. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 4034–4043. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.497

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit (2022)

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020), https://nlp.stanford.edu/pubs/qi2020stanza.pdf

Şahin, G.G., Steedman, M.: Data augmentation via dependency tree morphing for low-resource languages. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 5004–5009. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1545

Sánchez-Cartagena, V.M., Esplà-Gomis, M., Pérez-Ortiz, J.A., Sánchez-Martínez, F.: Rethinking data augmentation for low-resource neural machine translation: A multi-task learning approach. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 8502–8516. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021), https://aclanthology.org/2021.emnlp-main.669

Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics SMC-13(3), 353–362 (1983)

Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 86–96. Association for Computational Linguistics, Berlin, Germany (Aug 2016), https://aclanthology.org/P16-1009

Sharaf, A., Hassan, H., Daumé III, H.: Meta-learning for few-shot nmt adaptation. arXiv preprint arXiv:2004.02745 (2020)

Shi, H., Livescu, K., Gimpel, K.: On the role of supervision in unsupervised constituency parsing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7611–7621. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.614

Shi, H., Livescu, K., Gimpel, K.: Substructure substitution: Structured data augmentation for NLP. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 3494–3508. Association for Computational Linguistics, Online (Aug 2021), https://aclanthology.org/2021.findings-acl.307

Vania, C., Kementchedjhieva, Y., Søgaard, A., Lopez, A.: A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 1105–1116. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1102

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., Trón, V.: Parallel corpora for medium density languages. Amsterdam Studies In The Theory And History Of Linguistic Science Series 4 292, 247 (2007)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

Wang, X., Pham, H., Dai, Z., Neubig, G.: SwitchOut: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 856–

861. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1100

Wei, X., Yu, H., Hu, Y., Weng, R., Luo, W., Jin, R.: Learning to generalize to more: Continuous semantic augmentation for neural machine translation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 (2022)

Wei, X., Yu, H., Hu, Y., Weng, R., Xing, L., Luo, W.: Uncertainty-aware semantic augmentation for neural machine translation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 2724–2735. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.216

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1461–1470. The COLING 2016 Organizing Committee, Osaka, Japan (Dec 2016), https://aclanthology.org/C16-1138

Zhang, J., Zong, C.: Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1535–1545. Association for Computational Linguistics, Austin, Texas (Nov 2016), https://aclanthology.org/D16-1160

Neurális entitásorientált szentimentelemző alkalmazás magyar nyelvre

Yang Zijian Győző, Laki László János

Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
{yang.zijian.gyozo, laki.laszlo}@nytud.hu

Kivonat Napjaink egyik kiemelkedő kutatása a szentimentelemzés. Kognitív tudományok számára fontos feladat, hogy mélyebb megértésre jussanak egy személy állapotával kapcsolatban. A nyelvtechnológia fejlődésével lehetőség nyílt a szöveges tartalmakban rejlő érzelmek, emóciók meghatározására. A szociális média elterjedésével és az online platformokon megjelenő tartalmak mennyiségének növekedésével egyre nagyobb igény támad a bennük lévő szentimentek automatikus elemzésére. A szentimentelemzés egyik feladata a szöveg érzelmi töltetét egy adott személyre nézve megvizsgálja, ezt hívjuk entitásorientált szentiment elemzésnek. Nyelvtechnológia szempontjából abban nehezedik a feladat, hogy először magát a személyt kell felismerni a szövegben és utána a hozzátartozó emóciót az adott környezetben. Kutatásunkban különböző neurális nyelvmodelleket finomhangoltunk entitásorientált szentimentelemzésre, majd adatkiterjesztési eljárásokkal növeltük a modellek teljesítményét. Továbbá tanulmányunkban egy olyan pipeline-t mutatunk be, amely először egy adott szövegben megkeresi a személyneveket, majd meghatározza a megtalált személynevekhez tartozó érzelmi töltetet az adott szövegkörnyezetben, majd végül az egész szövegre is megállapítja a szentimentet. A pipeline moduljai mind transzformer-alapú finomhangolt neurális modellek. Összesen három modell dolgozik az alkalmazásban. Egy névelemfelismerő, egy entitásorientált szentimentelemző és egy mondatalapú szentimentelemző modell. A pipeline-hoz készítettünk egy demo felületet is. Végül, de nem utolsó sorban kiértékeltük a pipeline teljesítményét.
Kulcsszavak: entitásorientált szentiment analízis, névelemfelismerés, adatkiterjesztés, gépi fordítás, transzfer tanítás, zeroshot

1. Bevezetés

A mesterséges intelligencia által támogatott megoldások megjelenése a szentiment analízisben megegyezik azokkal a tendenciákkal, amelyek megfigyelhetők a kognitív infokommunikáció területén is. Kiemelt fontossággal bír, hogy jobban érthetővé váljon az adatbányászat hátterében álló elméleti keretrendszer, amely elősegíti a felhasználói érzelmi állapotokat vagy tömegvélemények alakulását prediktáló alkalmazások kifejlesztését (Baranyi és Csapó, 2012). Az online platformokon megjelenő tartalmak mennyiségének növekedésével és a szociális

média térhódításával nagy igény alakult e tartalmak automatikus elemzése. A nyelvtechnológia fejlődésével lehetőség nyílt a szövegek magas minőségű automatikus elemzésére. Az információ világában egyre fontosabbak azok az elemzések, amelyek egy adott termékről vagy személyről adnak adott szempontból összefoglaló információt. Ezt a célt szolgálhatja egy jól működő entitásorientált szentimentelemző, amellyel kimutatást lehet vele készíteni egy adott termékről vagy személyről, hogy hogyan vélekednek róla az emberek. Másik alkalmazási irány, ahol egyre nagyobb igény mutatkozik a szentiment analízisre, az a pszichológiai kutatások, elemzések. A páciensek interjúinak és beszámolóinak szentimentelemzésével következtetni lehet az adott betegség fejlődésére. Illetve adott dokumentumokban bizonyos jelenségek, mint például stigmatizáció jelenségének keresésére is lehet használni egy ilyen eszközt.

Jelen tanulmányban először bemutatjuk a neurális entitásorientált szentimentelemző kutatásunk eredményeit. Ezután bemutatjuk a szentimentelemző alkalmazásunkat és annak teljesítményét. Az alkalmazásunk egy pipeline-t valósít meg, amely egy bemeneti szöveget elemez. Felismeri a benne lévő személyneveket, majd a megtalált személynevekre megállapítja a szentimentet az adott környezetben. Továbbá az alkalmazás egy mondatszintű szentimentet is visszaad eredményként. A legjobban teljesítő modelljeink elérhetőek a Hugging Face oldalunkon[1]: NYTK/sentiment-ohb3-hubert-hungarian, NYTK/sentiment-ohb3-xlm-roberta-hungarian.

2. Kapcsolódó irodalom

A szentiment analízis egy komplex nyelvtechnológiai feladat és számos területen alkalmazható. Többek között szociális média monitorozására és elemzésére (Neri és mtsai, 2012), üzleti döntéshozatalok támogatására, gazdasági hírek szemantikus kontextusának elemzésére (Lutz és mtsai, 2018; Saura és mtsai, 2019) vagy pszichés állapot vizsgálatára (Jo és mtsai, 2018). Jelenleg többféle irányban zajlanak a szentiment analízis kutatások. Kezdetben a szentimentelemzés során osztályozni lehetett dokumentumokat, mondatokat és a szövegeket negatív, pozitív vagy semleges osztályba sorolták. Ezek a kutatások mind a mai napig aktuálisak (Pang és mtsai, 2002; Feldman, 2013; Lutz és mtsai, 2018). Később egy új irány alakult, az aspektus-alapú, vagy entitásorientált módszer, amely kifinomultabb és célja, hogy azonosítsa egy tárgy, személy vagy bármilyen entitás olyan aspektusait, melyek az érzelmek kiváltásáért felelősek (Pontiki és mtsai, 2014), illetve a vizsgált entitásról megállapítsa a szentimentjét az adott környezetben.

Magyar nyelvre kevés szentimentelemző tanítóanyag található. A HuSent (Szabó és mtsai, 2016) egy manuálisan annotált szentiment korpusz, amely a Dívány honlapról vett magyar nyelvű véleményeket tartalmaz a különböző termékekkel kapcsolatban. A korpusz 154 véleményt tartalmaz, mintegy 17 000 mondatból és 251 000 tokenből áll. Entitásorientált szentiment analízis korpusz az OpinHuBank (Miháltz, 2013). Ez egy manuálisan annotált korpusz, amely a véleménykutatást támogatja. 10 000 mondatból áll és személyneveket tartalmaz a

[1] https://huggingface.co/NYTK

főbb magyarországi honlapokról és blogokról. Minden egyes szegmenst 5 annotátor elemzett, a szentiment polaritásra nézve lehet semleges, pozitív vagy negatív. (Steinberger és mtsai, 2011) kutatásai során előállított egy aspektus-alapú szentiment korpuszt többnyelvű párhuzamos korpuszokkal, amely tartalmaz egy magyar nyelvű alkorpuszt is. (Laki és Yang, 2021) kutatásukban mondatszintű neurális szentimentelemző modellekkel kísérleteztek. Különböző magyar és többnyelvű modelleket finomhangoltak. A modellek teljesítményét különböző adatkiterjesztési eljárásokkal, transzfer és gépi fordítás módszerekkel növelték.

Jelen kutatás keretében az entitásorientált szentimentelemzésre fókuszáltunk, valamint a szentimentelemző alkalmazás pipeline-t mutatjuk be.

3. Tanítóanyag

Az entitásorientált szentimentelemző modell tanításához a szabadon elérhető magyar nyelvű OpinHuBank (Miháltz, 2013) (OHB3) korpuszt használtuk fel. A korpusz címkéi: -1 (negatív), 0 (semleges), 1 (pozitív). Minden szegmenst öt annotátor annotált. A feladatunkhoz összevontuk és átkonvertáltuk a címkéket: 1 (negatív), 2 (semleges), 3 (pozitív). Így minden szegmenshez pontosan egy darab címke tartozik. Továbbá végeztünk transzfer és fordítással történő adatkiterjesztési kísérleteket is. Ehhez a feladathoz az acl-14-short-data (Dong és mtsai, 2014) (ACL14) angol nyelvű korpuszt használtuk fel. A fordításhoz a Laki és Yang (2022) által közzétett gépi fordító rendszert használtuk fel.

A 1. táblázatban láthatóak a korpuszok legfőbb tulajdonságai. A duplavonal felett szereplenek a fontosabb statisztikai információk, a duplavonal alatt találhatóak a tanító- és tesztanyagra vonatkozó főbb tulajdonságok. Az OHB3 tanító- és tesztanyag előállításához 90%-10% vágást végeztünk. A korpusz első 1000 szegmense alkotja a tesztanyagunkat. A 1. táblázatban továbbá láthatóak a címkék eloszlása. Jól megfigyelhető, hogy nem kiegyensúlyozott a korpusz, ezért a kutatásunk során végeztünk adatkiegyenlítési kísérleteket is.

Végül, de nem utolsó sorban megmértük az annotátorok egyetértését is, amit a 2. táblázat mutat.

A szentimentelemző modell tanításához különböző transzformer-alapú neurális nyelvi modelleket finomhangoltunk. A transzformer modellek előnye, hogy képesek a bemeneti szövegben lévő viszonyokat megtanulni. Ez az entitásorientált tanítóanyag létrehozását is megkönnyíti, hiszen csak meg kell jelölni, vagy ki kell emelni az elemezni kívánt szövegrészt. A kiemelés úgy történt a mi esetünkben, hogy a vizsgált személynevet a szegmens elejére szúrtuk be, majd egy szeparátor címkével választottuk el a szegmenstől. A különböző modell típusok különböző szeparátort használnak. A modelljeink esetében a tanítóanyag formátumát a következő kettő példa demonstrálja:

- BERT: Tamás [SEP] Milan és Tamás is jól beilleszkedett már.
- RoBERTa: Tamás </s> Milan és Tamás is jól beilleszkedett már.

		ACL14	OHB3	ACL14 fordított
Szegmens		6 940	10 005	6 940
Token		150 904	308 407	146 717
Type		16 405	47 492	23 422
Átlagos szószám		17,64	26,36	17,01
Címkék		1;2;3	1;2;3	1;2;3
Tanító	szegmens	2 468	9 005	2 468
	1 (címke)	1 560	2 253	1 560
	2 (címke)	3 127	5 205	3 127
	3 (címke)	1 561	1 548	1 561
Teszt	szegmens	269	1 000	269
	1 (címke)	173	136	173
	2 (címke)	346	565	346
	3 (címke)	173	299	173

1. táblázat. Korpuszok tulajdonságai

	OpinHuBank
Fleiss Kappa	0,6551
Krippendorff-alfa	0,6548
Scott pi	0,6548
Cohen-féle kappa	0,6549

2. táblázat. Annotátorok közötti egyetértés

4. Kísérletek

A kutatásunkban először finomhangoltunk három magyar és kettő többnyelvű neurális nyelvi modellt az entitásorientált szentimentelemzés feladatára: huBERT (Márk, 2021), HILBERT (Feldmann és mtsai, 2021), HIL-RoBERTa (Yang és Váradi, 2021), XLM-RoBERTa base (XLM-R) (Conneau és mtsai, 2020), Multilingual Cased BERT-Base (mBERT) (Devlin és mtsai, 2019).

Ezután végeztünk adatkiegyenlítés és adatkiterjesztési kísérleteket a modelljeink minőségének javítására.

Az adatkiegyenlítés eljárás során kettő kísérletet végeztünk:

- **bővítés**: vettük a legtöbb szegmenssel bíró osztálycímkéhez tartozó szegmensek számát, ami az OHB3 esetén a 2-es címke 5 205 szegmensszámmal. Majd a többi osztálycímkéhez tartozó szegmensek számát bővítettük addig, amíg el nem érte ezt az 5 205 számot.
- **elvonás**: vettük a legkevesebb szegmenssel bíró osztálycímkéhez tartozó szegmensek számát, ami az OHB3 esetén a 3-as címke 1 548 szegmensszámmal. Majd a többi osztálycímkéhez tartozó szegmensek számát csökkentettük addig, amíg el nem érte ezt az 1 548 számot.

Az adatkiterjesztéses kísérleteink során a Laki és Yang (2021) által végzett eljárásokat végeztük el. Így a kutatásunk során összesen kilenc különböző mérést végeztünk el:

- **eredeti**: minden előtanított nyelvi modell finomhangolásra került az eredeti OHB3 korpuszon. Ezt tekintjük az alapeljárásnak.
- **bővítés**: adatkiegyenlítő eljárás során történt adatbővítéssel kiegyenlített OHB3 korpuszon finomhangoltunk minden előtanított modellt.
- **elvonás**: adatkiegyenlítő eljárás során történt adatelvonással kiegyenlített OHB3 korpuszon finomhangoltunk minden előtanított modellt.
- **zeroshot**: többnyelvű modellek képesek magyar nyelvű NLP feladatok predikciójára. Az angol ACL14 korpuszt használtuk finomhangolásra és a rendszernek magyar nyelvű mondatokra kellet prediktálnia.
- **transzfer**: többnyelvű modellek finomhangolva lettek az ACL14 korpuszon, majd további finomhangolásra kerültek az OHB3 tanító korpuszon.
- **fordított**: minden előtanított modell finomhangolva lett a magyarra fordított ACL14 korpuszon (ACL14_hu), majd közvetlenül tesztelve az OHB3 tesztkorpuszon.
- **fordított+finomhangolás**: minden előtanított modell finomhangolva lett az ACL14_hu korpuszon, majd ismét finomhangolva lett az OHB3 tanító korpuszon.
- **mix**: minden előtanított modell finomhangolva lett a konkatenált és összekevert ACL14_hu és OHB3 tanító korpuszon, majd le lett tesztelve az OHB3 tesztelő korpuszon.
- **mix+finomhangolás**: minden előtanított modell finomhangolva lett a konkatenált és összekevert ACL14_hu és OHB3 tanító korpuszon, majd újabb finomhangolás következett az OHB3 tanító korpuszon

Az elvégzett kísérletek eredményeit az OHB3 tesztelő korpuszán értékeltük ki. A technikai feltételek és beállítások megegyeznek a Laki és Yang (2021) által közzétett cikkben leírtakkal.

5. Eredmények

A 3. táblázatban láthatóak a méréseink eredményei. Első megfigyelés, hogy az adatkiegyenlítési kísérletek nem jártak sikerrel. Az elvonás eljárás sikertelensége várható volt, hiszen drasztikusan lecsökkent vele a tanítóanyag mérete.

Az adatkiterjesztéses mérések eredményei hasonlóak a (Laki és Yang, 2021) kutatásaiban bemutatott mondatalapú szentimentelemzés kísérleteihez. A fordítással hozzáadott adat képes növelni a modell minőségén. Továbbá az látható még, hogy ez a feladat sokkal jobban megosztotta a modelleket, ami azt jelenti, hogy ez a feladat már nehezebbnek bizonyult. A HILBERT eredményeit nézve gyakorlatilag nem tudta megtanulni ezt a feladatot. A huBERT várhatóan kimagaslóan teljesít ebben a feladatban is. Ami még érdekes, hogy a többnyelvű XLM-R modell is igen magas eredményt ért el.

A 4. táblázatban látható három példa a legjobban teljesítő huBERT mix modelltől. Látható, hogy egy mondaton belül, a különböző személynevekhez megfelelő szentimentet prediktált. Külön kiemelendő, hogy a harmadik példában Kovácsné Erzsébet helyesen semleges szentiment értéket kapott, hiszen ő ebben az esetben csak mesélő.

	huBERT	HILBERT	HIL-RoBERTa	XLM-R	mBERT
eredeti	81,99	57,80	68,50	79,69	75,49
bővítés	81,00	57,40	68,00	78,50	75,19
elvonás	77,99	56,99	66,29	75,40	74,09
zeroshot	-	-	-	66,79	61,19
transzfer	-	-	-	79,79	75,70
fordított	63,70	52,10	54,90	63,89	41,60
fordított+finonhangolás	81,69	54,40	69,49	**80,59**	61,40
mix	**82,30**	51,49	**69,80**	79,69	75,49
mix+finonhangolás	81,49	**58,60**	66,69	77,70	**76,39**

3. táblázat. Entitásorientált szentimentelemzés eredménye

Vizsgált személynév	Mondat	Prediktált szentiment
Kovácsné Erzsébet	A Kovácsné Erzsébet nagyon jól érzi magát a Nokiánál, azonban a Németországból érkezett Kovács Péter nehezen boldogul a beilleszkedéssel.	pozitív
Kovács Péter	A Kovácsné Erzsébet nagyon jól érzi magát a Nokiánál, azonban a Németországból érkezett Kovács Péter nehezen boldogul a beilleszkedéssel.	negatív
Kovácsné Erzsébet	A Kovácsné Erzsébet azt mondta, hogy a Németországból érkezett Kovács Péter nehezen boldogul a beilleszkedéssel.	semleges

4. táblázat. Példák a huBERT mix modelltől

A végső alkalmazásunkban a legjobb eredményt elért finomhangolt huBERT mix modellt használtuk fel.

6. Névelem kinyerése

Az entitásorientált szentimentelemzés valós időben való alkalmazásához szükség van egy névelemfelismerő modellre (NER), amellyel a személyneveket kinyerjük a szövegből. A magyar nyelvre elérhető neurális névelemfelismerő modellek korlátolt számossága miatt, a feladathoz a (Yang és Váradi, 2021) által közzétett NER modellt használtuk fel. A használt NER modell az NYTK-NerKor korpuszon (Simon és Vadász, 2021) tanított, amely 90,18% F1 mértékkel tudja felismerni a névelemeket. Az entitásorientált elemző alkalmazásunkhoz csak a személyneveket (PER címkével ellátott) használtuk fel.

A névelemfelismerés pontosságától erősen függ az alkalmazás pontossága. A NER modellek minőségének hatásait mértük a következő kísérletünkben. A kísérlethez vettük a NER feladatra finomhangolt HILBERT és HIL-RoBERTa modelleket (Yang és Váradi, 2021). A 5. táblázatban mutatjuk a kiértékelés arról, hogy a NER modellek hogyan teljesítenek az összes címkére nézve (NerKor-MIND) és csak a személynevekre nézve (NerKor-PER). Továbbá láthatóak a modellek teljesítményei az OHB3 korpuszban lévő személynevek felismerésére.

Az OHB3 korpuszban csak a személynevek vannak megjelölve, ezért a vizsgálatunkban is csak a személynevekre koncentráltunk, vagyis a szervezet (ORG), helynevek (LOC) és a vegyes (MISC) címkékkel ellátott tulajdonneveket nem vettük figyelembe. Az eredményekből tisztán látszik, hogy a NerKor tesztanyagán mért 5% különbség az OHB3 korpuszban 20-40% teljesítményeltérést is mutathat. Ez azt is mutatja, hogy például a HILBERT-en finomhangolt modell a NerKor korpuszon túltanult és egy független korpuszon rendkívül gyengén teljesített.

	NerKor-MIND (F1)	NerKor-PER (F1)	OHB3 (Pontosság)
huBERT	90,18	96,25	87,4
HILBERT	84,81	92,41	46,0
HIL-RoBERTa	85,16	93,18	69,2

5. táblázat. NER modellek kiértékelése

Ezután a következő kísérletünkben a legjobban teljesítő finomgangolt huBERT NER modellt teszteltük a különböző adatkiterjesztéses kísérleteinkben. A 6. táblázatban láthatóak azok az eredmények (lásd második oszlop), ha a tesztanyagban a személyneveket a huBERT NER modellel nyernénk ki és adnánk oda a szentimentelemző modellnek. Az eredményekben meg is látszik a különbség, hiszen megmutatkozik a NER modell hibája, de a hiba mértéke a legtöbb esetben nem szignifikáns.

	Referncia entitás	huBERT NER
huBERT eredeti	81,99	81,49
huBERT fordított	63,70	51,89
huBERT fordított+finomhangolás	81,69	81,30
huBERT mix	82,30	82,20
huBERT mix+finomhangolás	81,49	81,30

6. táblázat. huBERT NER eredmények

Végül megvizsgáltuk a hibásan felismert személynevek hibatípusait (lásd 7. táblázat). A példákból látható, hogy a hibák nagyobb része referenciahiba. A két nagyobb hibakategória ezekből a hiányzó előtag és a helytelen kinyerés. Ebből látszik, hogy automatikus módszerekkel nyerték ki a tulajdonneveket. A modell hibái közül nagy része a hibásan kinyert személynevek, de találunk tokenizálási hibát is.

7. Demó applikáció

A demó applikáció egy pipelinet valósít meg, amelyben van egy NER modell és egy szentimentelemző modell. A pipeline egy mondatot vár, amelyet átad a

Hiba típusa	**Arány (darab)**	**Példa** (név a referenciában — kinyert név)
Referencia hiba	**70,63% (89)**	
Hiányzó előtag	34,83%	Johann Strauss — ifj. Johann Strauss Litavecz Anna — Dr. Litavecz Anna János Pál — II. János Pál
Nem személynév	10,12%	Szent István — Szent István templomban Puskás Ferenc — Puskás Ferenc stadion
Lemmatizált	13,48%	Magyar Kornélia — Magyar Kornéliát Dombóvári István — Dombóvári Istvánnal
Helytelen kinyerés	41,57%	Varga Miklós — Varga Miklósné Joe Bi — Joe Biden Kristin Krohn — Kristin Krohn Devold
Modell hiba	**29,37% (37)**	
Tokenizálás	24,32%	Széles-Kovács Gyula — Széles - Kovács Gyula
Sztemmelt	29,73%	Bogár Zsolttal — Bogár Zsolt Jakab Jánosék — Jakab János
Helytelen kinyerés	45,95%	Varga Miklós — Varga Miklós Európ Bándi Gyula — Bándi Gyul Boross József — ∅

7. táblázat. huBERT NER modell hibaelemzése az OHB3 tesztanyagon

NER modell számára. A NER modell a szövegben felismeri a tulajdonneveket, amelyeket meg is jelöl. Ezután a felismert névelemek közül a személyneveket (PER címkével ellátott) továbbadja a szentimentelemző modell számára, aki a névelem és a bemeneti mondat segítségével prediktálja a személynévhez tartozó szentimentet.

Az 1. ábrán látható a demó alkalmazás felülete. Egy szövegbeviteli mezőben meg lehet adni egy mondatot, majd az Elemzés gombbal az alkalmazás leelemzi a mondatot. A szövegbeviteli mező alatt jelenik meg az elemzett szöveg. A huBERT NER modellje által felismert tulajdonneveket a program különböző színekkel megjelöli: zöld: személynév; kék: szervezet; narancssárga: helynév; lila: vegyes. A NER modell képes nem csak személyneveket felismerni, azonban a jelenlegi entitásorientált szentimentelemző modellünk csak személynevekkel dolgozik.

Az elemzett szöveg alatt látható a mondatszintű szentimentelemzés eredménye a modell által prediktált valószínűséggel. A mondatszintű elemzéshez a Laki és Yang (2021) által tanított huBERT mondatalapú szentimentelemző modellt használtuk. Ezután alatta következik egy felsorolás, ahol a szövegben felismert személyneveket jeleníti meg a hozzátartozó szentimentelemzéssel és a modell által prediktált valószínüséggel. A szentimentelemzéseket különböző színű háttérrel is jelöltük: zöld: pozitív; bézs: semleges; bordó: negatív.

1. ábra: Demó alkalmazás

8. Összegzés

Kutatásunkban különböző magyar és többnyelvű neurális nyelvmodellt finomhangoltunk entitásorientált szentimentelemzésre. A modellek teljesítményét különböző adatkiterjesztési, transzfer és gépi fordítás módszerekkel növeltük. Eredményünkben bemutattuk, hogy adatkiterjesztési módszereinkkel valóban tudtuk növelni a modellek minőségét. Továbbá bemutattunk egy alkalmazást, amely képes szentimentelemezni egy bemeneti szöveget. Az alkalmazás először egy névelemfelismerő modell segítségével kiválasztja a személyneveket, majd az entitásorientált szentimentelemző modell a kiválasztott személynevekre megvizsgálja az érzelmi polaritást az adott környezetben. Végül a rendszer ad egy mondatszintű szentiment értéket is. Az alkalmazáshoz készítettünk egy demó felületet is. Végül, de nem utolsó sorban kiértékeltük az entitásorientált szentimentelemzőnk teljesítményét a névelemfelismerő modell függvényében is.

Hivatkozások

Baranyi, P., Csapó, Á.: Definition and synergies of cognitive infocommunications. Acta Polytechnica Hungarica 9(1), 67–83 (2012)

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual

Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: The 52nd Annual Meeting of the Association for Computational Linguistics (ACL). ACL (2014)

Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM 56(4), 82–89 (Apr 2013)

Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: Hilbert, magyar nyelvű bert-large modell tanítása felhő környezetben. XVII. Magyar Számítógépes Nyelvészeti Konferencia pp. 29–36 (2021)

Jo, H., Kim, S.M., Ryu, J.: What we really want to find by sentiment analysis: The relationship between computational models and psychological state (2018)

Laki, L., Yang, Z.G.: Improving performance of sentence-level sentiment analysis with data augmentation methods. In: Proceedings of 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021). pp. 417–422. IEEE, Online (2021)

Laki, L., Yang, Z.G.: Jobban fordítunk magyarra, mint a Google! In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 357–372. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Lutz, B., Pröllochs, N., Neumann, D.: Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning (2018)

Márk, N.D.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2013)

Neri, F., Aliprandi, C., Capeci, F., Cuadros, M., By, T.: Sentiment analysis on social media. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 919–926 (2012)

Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). pp. 79–86. Association for Computational Linguistics (Jul 2002), https://www.aclweb.org/anthology/W02-1011

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 27–35. Association for Computational Linguistics, Dublin, Ireland (Aug 2014)

Saura, J.R., Palos-Sanchez, P., Grilo, A.: Detecting indicators for startup business success: Sentiment analysis using text data mining. Sustainability 11(3), 917 (Feb 2019), http://dx.doi.org/10.3390/su11030917

Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Steinberger, J., Lenkova, P., Kabadjov, M., Steinberger, R., van der Goot, E.: Multilingual entity-centered sentiment analysis evaluated by parallel corpora. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2011. pp. 770–775. Association for Computational Linguistics, Hissar, Bulgaria (Sep 2011), https://www.aclweb.org/anthology/R11-1113

Szabó, M.K., Vincze, V., Simkó, K.I., Varga, V., Hangya, V.: A Hungarian sentiment corpus manually annotated at aspect level. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 2873–2878. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016)

Yang, Z.G., Váradi, T.: Training language models with low resources: Roberta, bart and electra experimental models for hungarian. In: Proceedings of 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021). pp. 279–285. IEEE, Online (2021)

Koreferenciafeloldás magyar szövegeken BERT-tel

Vadász Noémi[1], Nyéki Bence*

[1]Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
vadasz.noemi@nytud.hu, nyeki.bence96@gmail.com

Kivonat A cikk egy kísérletet mutat be, amelyben összevonva használtunk két koreferenciakorpuszt a magyar BERT modell finomhangolásához, amivel magyar szövegeken lehet koreferenciafeloldást végezni. A cikk ismerteti a kísérletünk lépéseit a korpuszok előkészítésétől és felhasználásától a BERT modell finomhangolásán keresztül az eredmények kiértékeléséig. A koreferenciafeloldót szabadon hozzáférhetővé tettük.
Kulcsszavak: BERT, finomhangolás, koreferencia

1. Bevezetés

A neurális nyelvmodellek lehetővé teszik, hogy finomhangolás segítségével magasabb szintű nyelvfeldolgozási feladatokat oldjunk meg. A finomhangolás során egy kisebb méretű, speciális annotációval ellátott korpuszt használunk. Mivel a koreferenciafeloldás feladatának megoldását tűztük ki célként, a finomhangoláshoz koreferenciaannotációval ellátott korpuszra volt szükségünk. Abban a szerencsés helyzetben vagyunk, hogy magyarra már rendelkezésre állnak nagy méretű neurális nyelvmodellek és a kitűzött feladathoz alkalmazható koreferenciakorpuszok is.

Kísérletünk során a `huBERT` (Nemeskey, 2021) modellt finomhangoltuk a SzegedKoref (Vincze és mtsai, 2018) és a KorKor (Vadász, 2020) korpuszok felhasználásával. Az általunk készített koreferenciafeloldó szabadon hozzáférhető, a GitHub-on[1] elérhető.

2. Háttér

2.1. Anafora és koreferencia

Az információkinyerés területén belül számos olyan feladat van, amelyek megoldásához anafora- vagy koreferenciafeloldásra van szükség. Anaforikus kapcsolat egy visszautaló elem (pl. egy névmás) és a szövegbeli előzménye (antecedense) között áll fenn, a koreferencia pedig azt jelenti, hogy a szövegben két elem ugyanarra a valóságbeli entitásra refereál. Számítógépes nyelvészeti szempontból általában a koreferenciafeloldás feladata egy entitás összes szövegbeli említésének az

* A kutatás ideje alatt a szerző a Nyelvtudományi Kutatóközpont munkatársa volt.
[1] https://github.com/nytud/bert_coref_hu

azonosítása, míg az anaforafeloldás az anafora előzményének, az antecedensnek a megtalálása.

Az anafora- és koreferenciaannotációt tartalmazó korpuszok különböző annotációs elvek mentén készülnek, a különböző anaforikus- és koreferenciakapcsolatok aleseteit gyakran különböző osztálycímkékkel jelölve. A névmási anafora alesetei kategorizálhatók a névmás alapján (személyes, mutató, birtokos, kölcsönös, visszaható, vonatkozó névmások), míg a főnevek között fennálló koreferenciakapcsolat is többféle lehet (pl. ismétlés, szinonima, rész-egész kapcsolat, epitheton, appozíció stb.). Emellett megjelenhetnek a nem névszói kapcsolattípusok is pl. az igei és a határozói anaforák esetében, valamint akár képzett alakok között is. Nem egységes a deixis és a katafora kezelése sem az egyes korpuszokban, valamint az ún. pro-drop nyelvek esetén a zérónévmások kezelése sem.

2.2. Feloldók

A magyar nyelvre eddig több tudásalapú, szabályokkal operáló anafora- és koreferenciafeloldó megoldás is született. Lejtovicz és Kardkovács (2007) szabályalapú anaforafeloldója a BFP algoritmust (Brennan és mtsai, 1987) implementálta és szintaktikailag elemzett mondatokat kap bemenetként, tehát szintaktikai és morfológiai információra támaszkodik a feloldás során. Miháltz és mtsai (2007) és Miháltz (2012) megoldása a mondatok mély szintaktikai szerkezete mellett a kötéselmélet tételeire és pszicholingvisztikai megállapításokra támaszkodik, valamint a Magyar WordNet (Miháltz és mtsai, 2008) ontológiában tárolt nyelvi tudást is kiaknázza. Ez utóbbi a koreferenciakapcsolatok esetében segít, míg az előbbiek az anaforikus kapcsolatok felismerésében. (Vadász, 2017) és az annak alapján készült szabályalapú anaforafeloldó, amit a KorKor korpusz (Vadász, 2020) építéséhez használtak fel szintén szintaktikai és morfológiai információkra támaszkodnak (az előbbi még a főnevek [+/-] élő szemantikai jegyére is). Mindkét megoldás első lépésként azonosítja a szintaxisfákban az egyes grammatikai funkciókat, majd az ún. Pléh-Radics algoritmus (Pléh és Radics, 1976) szabályait alkalmazva keresi az egyes szereplők antecedensét.

Munkácsy és Farkas (2016) mutatta be az első statisztikai módszert koreferenciafeloldásra magyar nyelvű szövegekben, amelyben a SzegedKoref (Vincze és mtsai, 2018) korpuszon tanították a HOTCoref (Björkelund és Kuhn, 2014) rendszert, amit aztán a magyar nyelv igényeihez igazítottak. Tudásszegény megoldást kínál még a magyar névmási anaforák antecedenskeresésére Kovács Viktória doktori disszertációja (Kovács, 2021), amelyben gépi tanulási kísérleteket végez a különböző névmástípusok esetében.

Neurális megoldás a legjobb tudásunk alapján magyarra eddig nem született a témában, az angol nyelv esetében viszont több példa is van rá, hogy neurális hálókat alkalmaztak koreferenciafeloldásra. Ezek közül kettőt emelünk ki, hiszen ezek a munkák inspiráltak bennünket a kísérletezéseink kezdetén.

A NeuralCoref[2] (Wolf, 2017) a SpaCy[3] (Honnibal, 2015) rendszer pipelinejába illeszthető koreferenciafeloldó eszköz, amit neurális háló segítségével készítettek és angol nyelvű szövegekre működik. Noha a NeuralCoref újratanítható akár más nyelvű korpuszokkal is, számos megszorítás vonatkozik a tanítóanyag formátumára és tartalmára. Emellett a rendszer összetettsége miatt bonyolult adaptálni azokra a nyelvekre, amelyeket a SpaCy nem támogat, vagy nem áll a rendelkezésre megfelelő tanítóanyag. Joshi és mtsai (2019) pedig egy magas szintű, transzformeralapú koreferenciafeloldó rendszert készítettek, amelyben a *c2f-coref* modellt(Lee és mtsai, 2018) fejlesztették tovább a BERT integrálásával.

A bemutatott angol nyelvre készült megoldásokkal szemben a kutatásunkban megpróbáltunk egy konceptuálisan egyszerűbb, könnyebben felépíthető megoldásra törekedni a finomhangolt `huBERT` modell utolsó rétege által kibocsátott kontextuális tokenbeágyazások klaszterezésével.

3. A felhasznált korpuszok

A kísérleteink során két korpuszt használtunk, amelyek többek között koreferenciaannotációt is tartalmaznak. A SzegedKoref (Vincze és mtsai, 2018) a Szeged Korpusz (Csendes és mtsai, 2005) egy részéből készült és iskolai fogalmazásokat és újságcikkeket tartalmaz. A korpusz a publikáció alapján 55 ezer tokent tartalmaz, ugyanakkor a méréseink alapján sokkal nagyobb, majdnem 124 ezer token. A másik korpusz, a KorKor[4] (Vadász, 2020) sokkal kisebb, mindössze 31 ezer tokenes (beleszámolva az írásjeleket és a zéró elemeket, mint a zéró létigék, elliptált igék, zérónévmások) és hírszövegeket és Wikipédia szócikkeket tartalmaz. Hogy minél nagyobb tanító- és tesztelőanyag álljon rendelkezésünkre, a két korpuszt egységes formátumra alakítva használtuk fel a kísérletünkhöz.

Mindkét korpusz tartalmazza a zérónévmásokat is, így azokat az anaforikus kapcsolatokat is, amelyekben a zérónévmások vesznek részt. A pro-drop nyelvek esetében nagyon fontos, hogy egy koreferenciakorpusz tartalmazza ezeket a zéró elemeket. A KorKor korpusz esetében például az összes névmás háromnegyede zérónévmás. A zérónévmások mellett minden nem testes elem, tehát a zéró létigék és elliptált igék is nagyon fontos szerepet játszanak a különböző információkinyerési feladatokban. Mindezek ellenére a jelenlegi kísérletben nem használtuk a zérónévmásokat. Ennek az az oka, hogy a koreferenciafeloldó program bemenete sem tartalmazza őket – hacsak nem egy előfeldolgozó lépés eredményeként. A jelenlegi munkában tehát csak a szövegekben testesen előforduló elemekkel dolgoztunk. A SzegedKoref elérhető egy olyan verzióban, ami nem tartalmazza ezeket a testetlen elemeket, a KorKor korpuszt pedig automatikus módszerekkel alakítottuk át ilyenné.

A két korpusz formátuma és annotációs sémája több ponton is eltér, ezért mindenekelőtt egységesítenünk kellett őket. A KorKor esetében a dependencia-elemzés mintájára a koreferenciakapcsolatban résztvevő elem feje mellett szere-

[2] https://github.com/huggingface/neuralcoref
[3] https://spacy.io/
[4] https://github.com/vadno/korkor_pilot

pel az antecedens vagy a szövegben korábban előforduló koreferens elem index-száma, míg a SzegedKorefben a konstituenselemzéshez hasonlóan vannak összezárójelezve az összetartozó elemek. Az utóbbi a szerencsésebb megoldás, mert így az egymásba ágyazott vagy mellérendelt főnévi csoportok esetében is egyértelműen kiderül, hogy melyik főnévi csoport szerepel a koreferenciakapcsolatban. A SzegedKoref esetében ez azt jelenti, hogy egy token mellett akár több index-szám is szerepelhet. Az egységesítés azonban sajnos csak úgy volt automatikusan megoldható, ha a KorKor-ban használt módszert alkalmazzuk mindkét korpuszra. Ez azt jelenti, hogy a SzegedKoref esetében néhány koreferenciakapcsolatot elveszítettünk. Ha egy token mellett több koreferens elem is meg van jelölve, akkor csak a legbelső főnévi csoporthoz tartozó index-számot tartottuk meg.

Mindkét korpusz rövid szövegekből áll (KorKor: 112 és 652 token között, SzegedKoref: 11 és 1496 token között), az előkészítés során mégis tovább kellett darabolnunk őket. A `huBERT` bemenetét ugyanis subword tokenekre kell bontani, a modell pedig csak olyan szekvenciákat dolgoz fel, amelyek legfeljebb 512 subwordöt tartalmaznak. A feldarabolás után 600 darab fájlt kaptunk.

A két korpusz méretének és a bennük szereplő szövegek műfajának az arányát megtartva tanító, fejlesztő és teszthalmazra bontottuk őket. Az 1. táblázat mutatja az egyes halmazok méretét.

forrás		train	devel	test	TOTAL
SzegedKoref		98 774	12 131	12 521	12 3426
	HVG	7 779	772	1 163	9 714
	iskolai fogalmazás (8.o.)	73 891	9 353	9 215	92 459
	iskolai fogalmazás (10.o.)	17 104	2 006	2 143	21 253
KorKor		22 222	2 837	2 817	27 876
	hír	6 970	887	803	8 660
	Wikipedia	15 252	1 950	2 014	19 216
TOTAL		120 996	14 968	15 338	151 302

1. táblázat. A két korpuszból elkülönített tanító, fejlesztő és tesztanyag mérete tokenben kifejezve, a központozási jeleket is beleszámítva.

A 2. táblázatban a korpuszokban jelölt anaforikus és koreferenciakapcsolatok száma látható. Azt is szerettük volna megjeleníteni, hogy a korpuszokban összesen hány koreferenciacsoport van. Ez nem azonos a szövegekben előforduló entitások számával, hiszen csak azokat az entitásokat számítottuk bele, amiknek legalább két említése volt egy szövegben. Ennek az az oka, hogy a csupán egyetlen említéssel rendelkező entitások nincsenek megjelölve az általunk használt korpuszokban (lásd a 4. fejezetben). Az összevont korpusz fájljai átlagosan 11-12 koreferenciacsoportot tartalmaznak.

	KorKor		SzegedKoref	
	kapcsolat	csoport	kapcsolat	csoport
tanítóadat	2 521	796	10 290	3 522
fejlesztőadat	364	113	1 294	432
tesztelőadat	406	121	1 211	431

2. táblázat. A kapcsolatok és a koreferenciacsoportok száma a korpuszokban.

4. Szófajalapú előszűrés

Az egyes koreferenciakorpuszok között különbségek lehetnek atekintetben, hogy hogyan jelölik azokat a referáló elemeket, amelyek nem állnak semmivel koreferenciakapcsolatban. (A szakirodalomban ezekre *singleton*-ként hivatkoznak, a cikkben a továbbiakban szingletonnak nevezzük.) Az általunk használt korpuszokban a szingletonok nem voltak annotálva, ezért nem állt a rendelkezésünkre tanítóadat ehhez az alfeladathoz.

A nemzetközi szakirodalom a *mention* vagy *markable* kifejezéseket használja az anaforikus- vagy koreferenciakapcsolatokban potenciálisan részt vevő elemek megnevezésére, a cikkben a továbbiakban mi jelölteknek nevezzük ezeket. A lehetséges jelöltek kiválogatása egy külön előfeldolgozási lépésként értelmezhető a tényleges koreferenciafeloldás előtt. Megoldásunk azonban más irányból közelíti meg ezt a feladatot: a jelöltek kiválogatása helyett megpróbáljuk kizárni azokat a tokeneket, amelyeknek valószínűleg nincs antecedense vagy nem vesz részt koreferenciakapcsolatban. Szerencsére egyszerű szabályok alapján könnyen szűkíthetjük a lehetséges jelöltek körét. Ehhez szófajalapú előszűrést alkalmaztunk.

Mindkét általunk felhasznált korpusz tartalmaz kézzel ellenőrzött minőségű nyelvi annotációt, így a szófajt is – igaz, eltérő annotációs sémákkal és címkekészletekkel. Az egységesítés kedvéért ezért mindkét korpuszt leelemeztük az e-magyar (Váradi és mtsai, 2018; Indig és mtsai, 2019) elemzővel. Ez azt jelenti, hogy a gold standard minőségű kézi annotációt feláldoztuk az egységesítés oltárán, ugyanakkor azt is szem előtt kell tartani, hogy az e-magyar nagyon jó minőségű morfológiai elemzést bocsát ki (Simon és mtsai, 2020).

A nyelvi jelenségek ismeretében és a korpuszok segítségével nagy biztonsággal megállapítható, hogy mely szófajokat érdemes kizárni. Ez az egyszerű előszűrési lépés nagymértékben megkönnyíti a koreferenciafeloldó munkáját, még ha feltételezzük is, hogy az előszűrés során használt szófaji címkéző esetleges hibái begyűrűzhetnek a következő feldolgozási lépésekre, így rontva a koreferenciafeloldó teljesítményét. Az előszűrés során az igéket, igekötőket, csak jelzőként használt mellékneveket, határozószókat, névelőket, névutókat, kötőszavakat, indulat- és mondatszavakat és az írásjeleket zártuk ki a jelöltek közül.

5. Finomhangolás

A kísérletünkben a huBERT (Nemeskey, 2021) által kibocsátott kontextuális tokenbeágyazásokat finomhangoltuk. A célunk az volt, hogy a tokenbeágyazásokat úgy módosítsuk, hogy megragadják a koreferenciakapcsolatokat, tehát azt szerettük volna elérni, hogy az azonos koreferenciacsoporthoz tartozó tokenek reprezentációi a lehető legközelebb legyenek egymáshoz és a lehető legtávolabb legyenek a többi token beágyazásától.[5] Ehhez a reprezentációtanulásban gyakran használt[6] ún. *triplet loss* célfüggvényt alkalmaztuk. A függvény definíciója

$$\mathcal{L}(A, P, N) = \sum_{i}^{N} max(||f(A^{(i)}) - f(P^{(i)})|| - ||f(A^{(i)}) - f(N^{(i)})|| + \epsilon, 0)$$

ahol $A^{(i)}$ a viszonyítási alap (*anchor*),[7] $P^{(i)}$ egy *pozitív* példa (egy olyan beágyazás, amelytől azt várjuk, hogy közel legyen a viszonyítási alaphoz), $N^{(i)}$ pedig egy *negatív* példa (egy olyan beágyazás, amelynek távol kell lennie a viszonyítási alaptól). $||\cdot||$ egy távolságmetrika, ϵ pedig egy nemnegatív valós szám, amit toleranciának (*margin*) nevezhetünk.

Ezt a reprezentációtanulásban használt eljárást a saját feladatunkhoz igazítottuk. Jelöljük a tanítóanyag i-edik adatpontját $X^{(i)}$-vel, amely egy koreferenciajelölésekkel annotált szöveg. Legyen n_i azon tokenek száma $X^{(i)}$-ben, amelyekre igaz, hogy valamelyik koreferenciacsoporthoz tartoznak. $X^{(i)}$-t n_i-szer másoltuk le. Az annotáció módosítása után ezeket a másolatokat használtuk tanítóadatként.

A módosítás a következőképpen zajlott: Minden új $X^{(i,j)}$ ($j \in \{1, \ldots, n_i\}$) adatpontban viszonyítási alapként jelöltük meg a j-edik olyan tokent, amely valamelyik koreferenciakapcsolathoz tartozott. Azokat a tokeneket, amelyek ugyanahhoz a koreferenciakapcsolathoz tartoztak, mint a viszonyítási alap, pozitívként címkéztük meg. Minden egyéb olyan tokent, amelyet nem zárt ki a szófaji szűrőnk, negatívnak jelöltünk, eltekintve egy kivételtől: a tanítóanyagban azokat a szingletonokat is figyelmen kívül hagytuk (azaz a szófaji alapon kiszűrt tokenekkel azonos módon kezeltük), amelyek szintaktikailag valamilyen nem szingleton tokentől függtek. Ennek az az oka, hogy nem feltétlenül akartuk megakadályozni, hogy a szingletonok beágyazásai távol kerüljenek a szintaktikai szüleik beágyazásától, ez ugyanis megnehezíthette volna a tanulást. Ez ugyan némi zajt eredményezhet a klaszterezés kimenetében (lásd a 6. fejezetben), amely azonban szintaktikai szabályokkal csökkenthető.

[5] A tokenek reprezentciója alatt a token első subwordjének a reprezentációját értjük, a többi subword reprezentációját nem vettük figyelembe a tanítás és a klaszterezés során.

[6] A triplet loss célfüggvényt használták többek között a FaceNet (Schroff és mtsai, 2015) és az SBERT (Reimers és Gurevych, 2019) tanításakor.

[7] A továbbiakban így nevezzük a viszonyítási alapnak választott tokent és annak beágyazását is, mivel a kontextusból mindig világos lesz, hogy adott esetben melyikről van szó.

A triplet loss formulát úgy módosítottuk, hogy $||f(A^{(i)}) - f(P^{(i)})||$-t a viszonyítási alapnak a pozitív tokenek beágyazásaitól mért átlagos euklidészi távolságával cseréltük ki. A $||f(A^{(i)}) - f(N^{(i)})||$ tag helyett pedig euklidészi távolságot számoltunk a viszonyítási alap és a hozzá legközelebbi negatív token beágyazása között.

A fentebb ismertetett triplet loss célfüggvény segítségével dúsított tanítóadattal finomhangoltuk a `huBERT` modellt. AdamW optimalizálót használtunk, a tanulási rátát pedig koszinuszos ütemezővel csökkentettük a tanítás alatt. A hiperparamétereket Bayes-i keresés segítségével állítottuk be a validációs veszteség minimalizálásával.

5.1. Vizualizáció

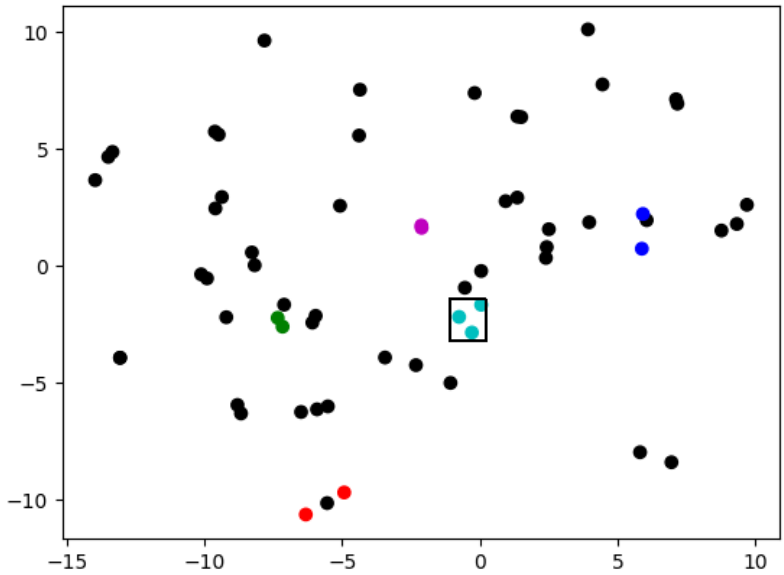
A klaszterező algoritmus bemeneteként szolgáló tokenbeágyazásokat vizualizáltuk is olyan módon, hogy a 768 dimenziós beágyazást két dimenzióra vetítettük le főkomponens-analízis (PCA) és t-elosztott sztochasztikus szomszédbeágyazás (t-SNE) segítségével.

Az 1. ábrán egy tesztfájl eredményének a vizualizációja látható. A pontok színe jelöli az egy koreferenciacsoportba tartozó elemeket a gold standard alapján, tehát minden szín egy entitást jelöl. A fekete pontok a szingleton elemek, tehát azok, amelyeket nem zárt ki a szófajalapú előszűrés, de mégsem szerepelnek koreferenciakapcsolatban. Az ábrán látható, hogy az egyes koreferenciacsoportok tagjai valóban közel vannak egymáshoz.

6. Klaszterezés

Azt vártuk, hogy az egyes koreferenciacsoportokba tartozó tokenreprezentációk klaszterekbe rendeződnek. A klaszterek felismerésére egy egyszerű algoritmust alkalmaztunk. Minden t tokenhez meghatároztunk egy s_t halmazt, ami azokat a tokeneket tartalmazza, amelyek beágyazása közelebb áll a t token beágyazásához, mint egy előre meghatározott δ küszöbérték. Minden (t, t') tokenpárra érvényes, hogy t-hez és t'-höz akkor és csakis akkor rendeltünk azonos klaszter-címkét, ha $s_t = s_{t'}$ fennállt. Úgy hangoltuk δ-t, hogy az maximalizálja a fejlesztő halmaz koreferenciacímkéi és a klaszter-címkék között mért normalizált kölcsönös információ (NMI) értékét. A legjobb eredményt úgy értük el, hogy a δ értékét 10-re állítottuk.

Az eredmények javítására egy függőségi információra támaszkodó heurisztikát használtunk. Ha egy c klaszterben a tokenek függőségi szüleinek a halmaza nem tartalmazott legalább két olyan tokent, amely nem volt része c-nek, akkor a c klasztert eltávolítottuk. Azért választottuk ezt a technikát, mert a szintaktikai fejek és jelzőik beágyazásai gyakran közel kerültek egymáshoz.



1. ábra: Egy tesztfájl eredményének 2D vizualizációval ábrázolva. A bekeretezett három pont által jelölt csoportba az alábbi szóalakok tartoztak: *itt, gazdaságok, sógazdaságok.*

7. Kiértékelés

A koreferenciafeloldó teljesítményét több szempontból is szerettük volna megítélni, ezért – és a feladat összetettsége miatt – a megoldásunkat két lépésben értékeltük ki.

7.1. A szófajalapú előszűrés kiértékelése

A koreferenciafeloldó megoldásunk első lépése a szófajalapú előszűrés, aminek a minősége nagyban befolyásolja a tényleges koreferenciafeloldás teljesítményét, így elsőként az előszűrés minőségét értékeltük ki. A szófajalapú előszűrés kimenetét összevetettük a tesztanyaggal és megvizsgáltuk, hogy hány olyan elemet zárt ki a lehetséges jelöltek közül, amit nem kellett volna, és fordítva.

Az előszűrés kiértékeléséhez mindenképpen szükség van a szingletonokra is a tesztanyagban, hiszen az, hogy egy elem nem vesz részt koreferenciakapcsolatban, még nem jelenti azt, hogy nem jó lehetséges jelölt. Ennek érdekében a tesztanyagban kézzel bejelöltük a lehetséges jelölteket.

Az összehasonlítás során a következő találati kategóriákat állítottuk fel minden tokenre:

- TP: az előszűrő kizárta és a tesztanyagban sem jelölt
- TN: az előszűrő nem zárta ki és a tesztanyagban is jelölt
- FP: az előszűrő kizárta, de a tesztanyagban jelölt
- FN: az előszűrő nem zárta ki, de a tesztanyagban nem jelölt

A fenti találati kategóriákat összesítettük, majd pontosságot, fedést és F-mértéket számoltunk. A eredmények a 3. táblázatban láthatók.

metrika	eredmény
pontosság	0,9900
fedés	0,6798
F-mérték	0,8061

3. táblázat. A szófajalapú előszűrés kiértékelése.

Az eredménynek értékelésekor figyelembe kell venni, hogy a szófajalapú előszűrés a szófaji egyértelműsítés kimenetére támaszkodik, tehát előfordulhat, hogy a POS-tagger által vétett hiba továbbgyűrűzik és rontja a szófajalapú előszűrés és a rá épülő koreferenciafeloldó teljesítményét is.

7.2. A koreferenciafeloldás kiértékelése

Ahhoz, hogy jobban felmérhessük a feloldónk teljesítményét, összevetettük egy baseline megoldással. Sajnos nem találtunk olyan elérhető, könnyen beüzemelhető koreferenciafeloldót magyarra, ami megfelelt volna az elvárásainknak. A 2. fejezetben ismertetett megoldások nagy része nem biztosított hozzáférést a forráskódhoz, így nem tudtuk reprodukálni a működésüket. Vadász (2020) szabályalapú szkriptje pedig hozzáférhető[8] ugyan, de csak a személyes névmások antecedenskeresését végzi.[9]

A fentiek miatt tehát egy saját baseline megoldást készítettünk el. A baseline megoldásban fájlonként összekapcsoltuk azokat a főneveket, amelyeknek ugyanaz volt a lemmája. A baseline tehát pusztán egy bizonyos főnévi koreferenciakapcsolatot, az ismétlést próbálja megragadni, viszont az ismétlés egy nagyon gyakori koreferenciatípus. A SzegedKoref anyagában az összes kapcsolat 23,44%-a ismétlés! Ennél csak a névmási anaforakapcsolatokból van több (az összes kapcsolat 34,37%-a), de ezek a különböző névmások eltérő viselkedésének köszönhetően

[8] https://github.com/vadno/korkor_pilot/blob/master/scripts/anafora.py

[9] Mivel a zérónévmásokat kihagytuk a megoldásunkból és így a tesztanyagból is, ezért nagyon kevés személyes névmás van a tesztanyagban. A tesztanyag 132 személyes névmásából 4 zérónévmás volt.

nehezebben megragadhatóak. Az ismétlés tehát elég gyakori, és felszíni jegyek alapján könnyen azonosítható, így jól megfelelt baseline megoldásnak.

A koreferenciafeloldás kiértékelése összetett feladat és többféle szempontból is megközelíthető. A minőségbe beleszámít az, hogy a feloldó megtalálta-e a koreferenciakapcsolatokban résztvevő elemeket, összekapcsolta-e az összetartozó elemeket, megtalálja-e egy entitás összes említését a szövegben stb. A szakirodalomban ezért többféle kiértékelési módszerrel és mérőszámmal találkozhatunk, a módszerek és a mérőszámok megbízhatósága pedig gyakran megkérdőjelezhető[10]. Az elterjedt kiértékelő metrikákhoz azonban szükség lenne a szingletonokra is, ez azonban nem állt a rendelkezésünkre.

A fentiek miatt jelen kísérletünkben nem egy standard kiértékelő metrikát alkalmaztunk, hanem egy olyan módszert, ami több szempontból is megmutatja a koreferenciafeloldó megoldásunk teljesítményét. Mivel klaszterezést alkalmaztunk az egymáshoz közel álló elemek reprezentációjára, ezért a klasztereket értékeltük ki és vetettük össze a gold standard koreferenciacímkékkel. Azokat a tokeneket nem vesszük figyelembe, amelyeket a szófajalapú előszűrés során kizártunk a jelöltek közül, valamint azokat sem, amelyeket a feloldó helyesen jelölt szingletonnak.

Két metrikát alkalmaztunk a kiértékelésre: a tisztságot (*purity*) és a normalizált kölcsönös információt (*normalized mutual information*, NMI). Az előbbi azt méri, mennyire homogének az osztálycímkék (esetünkben a korpusz annotációjában megadott címkék) a klasztereken belül, következésképpen a tisztaság könnyen növelhető a klaszterszám emelésével. Az NMI arról ad képet, hogy mennyivel csökken az osztálycímkék entrópiája a klasztercímkék ismeretében. Mindkét metrika 0 és 1 között vesz fel értékeket (a tisztaság rosszabb esetben megközelíti a nullát). Az eredmények a 4. táblázatban láthatók. A koreferenciafeloldó mindkét metrika alapján jobban teljesített a baseline megoldásnál.

metrika	baseline	**eredmény**
purity	0,7245	0,7619
NMI	0.6272	0,6794

4. táblázat. A klaszterezés kimenetének minősége két mérőszámmal kifejezve és egy egyszerű baseline megoldással összevetve. Az eredmények az összes tesztfájl átlagát mutatják.

A tesztfájlokat közelebbről megnézve kiderül, hogy a koreferenciakapcsolatok közül az ismétlés, az anaforikus kapcsolatok közül pedig a vonatkozó névmások előzményét találta meg sikeresen a feloldó.

[10] A gyakran használt koreferenciakiértékelő metrikák tulajdonságait, előnyeit és hátrányait Moosavi és Strube (2016) vette sorra, valamint bevezettek egy újat is (*LEA*), ami bekerült a CoNLL-scorerjébe is.

8. Diszkusszió

A cikkben egy kísérletet mutattunk be, amiben egy neurális nyelvmodellt, a huBERT-et finomhangoltunk két egységesített koreferenciakorpusz segítségével a koreferenciafeloldás feladatára. Összetett és bonyolult felépítésű rendszerek felállítása helyett igyekeztünk konceptuálisan átlátható megoldásra törekedni, amit kiegészítettünk egy egyszerű szabályalapú előszűréssel. A programot szabadon hozzáférhetővé tettük és törekedtünk a reprodukálhatóságra is.

Az eredmények azt mutatják, hogy a koreferenciafeloldónk bár jobban teljesített a baseline megoldásnál, van még tér a fejlődésre, ezért a későbbiekben szeretnénk még jobb eredményt elérni. Emellett a jelenlegi megoldásunkból szándékosan kihagytuk a magyarban egyébként olyan fontos zérónévmásokat, azonban a továbbiakban szeretnénk ezekkel is kísérletezni.

Hivatkozások

Björkelund, A., Kuhn, J.: Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 47–57. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014), https://aclanthology.org/P14-1005

Brennan, S.E., Friedman, M.W., Pollard, C.J.: A Centering Approach to Pronouns. In: Proceedings of the 25th Meeting of the Association for Computional Linguistics. pp. 155—162 (1987)

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference, TSD 2005. pp. 123–131. Springer, Karlovy Vary, Czech Republic (2005)

Honnibal, M.: (2015), https://explosion.ai/blog/introducing-spacy

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – The emtsv pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (2019)

Joshi, M., Levy, O., Zettlemoyer, L., Weld, D.: BERT for Coreference Resolution: Baselines and Analysis. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5803–5808. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1588

Kovács, V.: Névmási anaforafeloldási kísérletek a magyar nyelvben. Ph.D.-értekezés, Szegedi Tudományegyetem (2021)

Lee, K., He, L., Zettlemoyer, L.: Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 687–692. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), https://aclanthology.org/N18-2108

Lejtovicz, K., Kardkovács, Zs.T.: Anaphora Resolution. In: Proceedings of the 8th International Symposium of Hungarian Researchers on Computational Intelligence. Budapest (2007)

Miháltz, M.: Tudásalapú koreferencia- és birtokviszony-feloldás magyar szövegekben. Általános Nyelvészeti Tanulmányok 24, 151–166 (2012)

Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., Váradi, T.: Methods and Results of the Hungarian WordNet Project. In: Proceedings of The Fourth Global WordNet Conference. pp. 311–321 (2008)

Miháltz, M., Naszódi, M., Vajda, P., Varasdi, K.: NP-koreferenciák feloldása magyar szövegekben a Magyar WordNet ontológia segítségével. In: Tanács, A., Csendes, D. (szerk.) V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2007). pp. 138–146. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2007)

Moosavi, N.S., Strube, M.: Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-based Entity Aware Metric. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 632–642. Association for Computational Linguistics, Berlin, Germany (Aug 2016), https://aclanthology.org/P16-1060

Munkácsy, G., Farkas, R.: Statisztikai koreferenciafeloldó rendszer magyar nyelvre — első eredmények. In: Tanács, A., Varga, V., Vincze, V. (szerk.) XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). pp. 295–297. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2016)

Nemeskey, D.M.: Introducing huBERT. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021). pp. 3–14. Szegedi Tudományegyetem Informatikai Intézet, Szeged (2021)

Pléh, Cs., Radics, K.: „Hiányos mondat", pronominalizáció és a szöveg. Általános Nyelvészeti Tanulmányok 11(1), 261–277 (1976)

Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084

Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering. CoRR abs/1503.03832 (2015), http://arxiv.org/abs/1503.03832

Simon, E., Indig, B., Kalivoda, , Mittelholcz, I., Sass, B., Vadász, N.: Újabb fejlemények az e-magyar háza táján. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020). pp. 29–42. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2020)

Vadász, N.: Anaforafeloldás menet közben – névmások egy pszicholingvisztikailag motivált elemzőben. In: Ludányi, Z. (szerk.) Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből 2017: XI. Alkalmazott Nyelvészeti Doktoranduszkonferencia, pp. 192–205. MTA Nyelvtudományi Intézet, Budapest (2017)

Vadász, N.: KorKorpusz: kézzel annotált, többrétegű pilotkorpusz építése. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 141–154. Szegedi Tudományegyetem (2020)

Vincze, V., Hegedűs, K., Sliz-Nagy, A., Farkas, R.: SzegedKoref: A Hungarian coreference corpus. In: Proceedings of the 11th Language Resources and Evaluation Conference. European Language Resource Association, Miyazaki, Japan (2018)

Váradi, T., Simon, E., Sass, B., Mittelholcz, I., Novák, A., Indig, B., Farkas, R., Vincze, V.: E-magyar – A Digital Language Processing System. 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. (szerk.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (2018)

Wolf, T.: (2017), https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30

Beszédtechnológia

„Feeding the BEAST” – A BEA Speech Transcriber továbbfejlesztése és integrálása neurális nyelvmodellel

Kádár Máté Soma[1, 3], Dobsinszki Gergely[1], Mády Katalin[2], Mihajlik Péter[1, 2]

[1] Budapesti Műszaki és Gazdaságtudományi Egyetem,
Villamosmérnöki és Informatikai Kar,
Távközlési és Médiainformatikai Tanszék,
H-1117, Budapest, Magyar tudósok körútja 2.
[2] Nyelvtudományi Kutatóközpont,
H-1394, Budapest, VI. Ker. Benczúr utca 33.
[3] SpeechTex Kft.,
1181, Budapest, Madách Imre utca 47.

Kivonat: Cikkünkben a korábban BEAST néven publikált, a BEA-Base adatbázison tanított mély-neuronháló alapú beszédleiratozó modellt meghaladó struktúrát mutatunk be. A használt architektúra magába foglal egy unigram szótöredékeken tanított wav2vec2 alapú akusztikus és egy Transformer alapú nyelvi modellt. Az akusztikus modell az uráli nyelvcsaládba tartozó nyelveken (magyar, finn, észt) önfelügyelten előtanított wav2vec-large struktúrára épül, mely a BEA-Base-en történő finomhangolása és egyes hiperparaméterek optimalizálása után önmagában is felülmúlta a BEAST eredményeit: a korábbi 16.62%-os szóhibarátát 12.08%-ra csökkentette. Az akusztikus modellhez integráltuk a Magyar Nemzeti Szövegtár beszélt nyelvi alkorpuszán tanított mély neurális nyelvi modellt, mely a nyalábkeresés segítségével 10.98%-ra javította a leiratozó szóhibaarányát. Tudomásunk szerint eddig ez a legjobb beszédfelismerési eredmény ezen az adathalmazon.

Kulcsszavak: mélytanulás, automatikus beszédfelismerés, nyelvmodell, önfelügyelt tanulás, Transformer, nyalábkeresés

1 Bevezetés

Az elmúlt években a „neurális forradalom” az automatikus beszédfelismerés (ASR) területét is elérte: sorra jelennek meg újabb, konkrétan a beszédfelismerésre optimalizált neurális architektúrák. A természetes nyelvfeldolgozás (NLP) esetén már évek óta elérhető a word2vec architektúra (Mikolov és mtsai, 2013), mely segítségével az egyes tokenek (szavak és szótöredékek, akár karakterek is) beágyazhatók, azaz szemantikájuk alapján osztályozhatók, vektorizálhatók. Mióta ez a lehetőség létezik szövegekre, azóta próbálnak hasonló elven működő architektúrát létrehozni, mely képes hullámformák vektorizációjára. Erre kínál megoldást a wav2vec2 struktúra, mely önfelügyelt, kontrasztív módon tanítható, valamint számos state-of-the-art eredményt tudhat magáénak (Baevski és mtsai, 2020).

Hála a BEÁ-nak (BEszélt nyelvi Adatbázis) (Gósy, 2008, Gósy és mtsai, 2012), most már magyar nyelven is létezik egy ingyenesen elérhető, kutatási célokra használható beszélt nyelvi adatbázis. Ennek felhasználásával lehetőség adódik kurrens ASR modellek készítésére, valamint az egyes modelleket össze lehet hasonlítani leiratozási hibájuk alapján a BEA teszthalmazán (Mihajlik és mtsai, 2022).

A klasszikus - maximum likelihood - megközelítéssel szemben a tisztán mély tanuláson alapuló neurális rendszerek jóval alkalmasabbak mind a mintaillesztés végrehajtására, mind a nyelvmodellezés („Language Model(ing)") megvalósítására. A továbbiakban egy wav2vec2-t felhasználó enkóder-dekóder akusztikus és egy Transformer (Vaswani és mtsai, 2017) alapú nyelvi modellből álló, szótöredékekre épülő ASR rendszer kerül ismertetésre. A rendszer a BEA-Base 114 beszélő spontán beszédét tartalmazó hanganyagain tanult és a cikk írásának pillanatáig a legjobb eredménnyel rendelkezik a BEA-Base spontán teszthalmazán. A fejlesztéshez a SpeechBrain (Ravanelli és mtsai, 2021) névre hallgató nyílt forráskódú keretrendszert használtuk. A tanított neurális modelleket akadémiai kutatás céljából ingyenesen elérhetővé teszszük, mint a BEAST (BEA Speech Transcriber) továbbfejlesztése, BEAST2 néven.

2 A tanító és kiértékelő halmazok áttekintése

A BEA-Base adatbázis 114 beszélő hanganyagait és azok szöveges átiratait tartalmazza. Az adatbázis két felé osztható: a spontán, valamint a kötött beszédes részhalmazokra. Mivel az emberi kommunikáció természetéből adódóan spontán, így a tanított akusztikus és nyelvi modell az adatbázis spontán beszédből álló részhalmazán lett finomhangolva, illetve a hálózatok hiperparaméterei is a spontán teszthalmazra lettek optimalizálva.

A nyelvmodellek tanításához elengedhetetlen a sok adat, különösen, ha Transformer hálózatról beszélünk. Ahogy az 1. táblázatban látható, a BEA-Base 3.38M karakternyi leiratot tartalmaz, ami a nyelvmodellezési feladatokhoz meglehetősen kevés. Emiatt szükséges volt egy nagyobb tanítókorpusz használata, melyen a Transformer nyelvi modell előtanítása történt. A választás a Magyar Nemzeti Szövegtár (későbbiekben SPOK-ként hivatkozva, mint "SPOKen language" alkorpusz) beszéltnyelvi részadatbázisára esett (Oravecz és mtsai, 2014). A SPOK ezen része a MR1 Kossuth Rádió 2004–2012 évek során rögzített hír- és riportműsorainak leiratát tartalmazza.

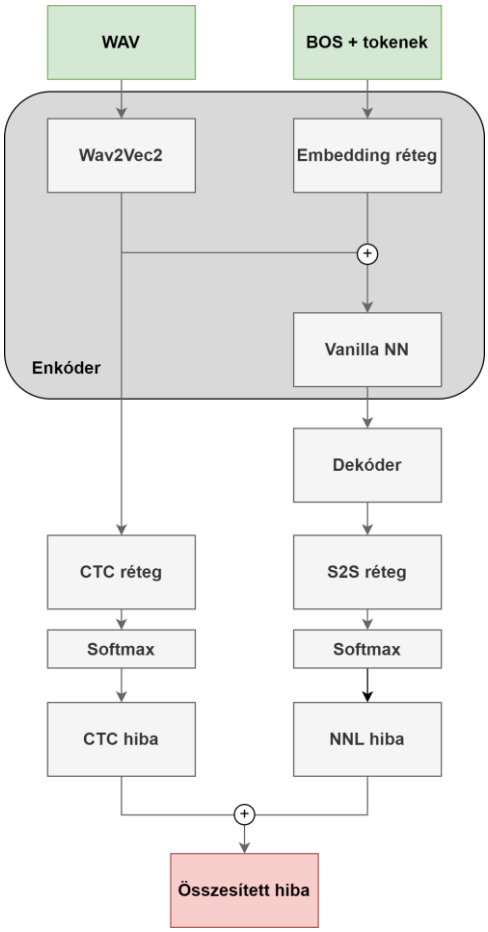
1. táblázat: A tanításhoz felhasznált adatbázisok és méreteik

Adatbázis	Karakterek (db)	Szavak (db)	Tokenek (db)	Audió (h)
BEA-Base	3.38M	0.56M	1.18M	71.2
SPOK	516.84M	56.13M	172.97M	-

Mind az akusztikus, mind a nyelvi modell szótöredék, pontosabban unigram szótöredék (Kudo és Richardson, 2018) alapon tanult. Ez esetünkben a BEA-Base spontán tanítókorpuszának önfelügyelt tokenizációját jelentette, 600-as szótármérettel (mivel a magyar nyelv 44 darab karaktert tartalmaz, így <unk> („unknown") tokenek nem keletkeznek a tokenizációkor).

3 Az akusztikus modell

A klasszikus rejtett Markov-modell alapú megközelítés helyett a korábban említett wav2vec2 alapú architektúrát alkalmaztuk, mint az akusztikus „end-to-end” modell enkóder része. A választott hálózat a VoxPopuli projekt (Wang és mtsai, 2021) keretein belül előtanított, ~317M paraméterrel rendelkező modell volt. A wav2vec egy adott hullámforma tanult reprezentációját állítja elő, melyet utána egy dekóder használ fel.



1. ábra: Az akusztikus modell architektúrája az előtanított wav2vec2-vel felszerelt enkóderrel, dekóderrel és a tanításkor használt költségfüggvényekkel

Az önfelügyelt előtanítás során felhasznált tanítóadatok különlegessége, hogy az csak uráli nyelvcsaládba tartozó nyelveket tartalmazott, ideértve a magyart is. Az előtanítás során a hálózat 17.7K órányi, leirattal nem rendelkező magyar nyelvű beszéden tanult a többi uráli nyelven (finn 14.2K, észt 10.6K óra) felül. Az előtanított

modellt finomhangoltuk a BEA-Base spontán hanganyagain. A finomhangolás klasszikus tanítási metodikával – Connectionist Temporal Classification (CTC) és attention költség minimalizálással (Graves és mtsai, 2006, Watanabe és mtsai, 2017) – történt.

3.1 Az adatok augmentációja

Tanításkor a bementi adatokon sebességperturbációt hajtottunk végre. Ez esetünkben azt jelentette, hogy a 16kHz-en mintavételezett beszédet 0.95, 1.0 vagy 1.05-szoros sebességű jelként adjuk a hálózat bementére. További augmentációként véletlenszerű időintervallum(ok)ban kimaszkoltuk a megfigyelhető jelet (Park és mtsai, 2019). A maszkolás minden köteg esetén – tehát 1.0 valószínűséggel – bekövetkezett 0–5 alkalommal mintánként.

Az augmentáció hozzájárul ahhoz, hogy a tanított hálózat generalizáltabb módon tanulhasson. Ezt kísérleteink során is igazoltnak láttuk: csökkentve a maszkolás valószínűségét 0.5-re, a tanított hálózat rosszabb WER eredményt ért el (13.19%), mint a növelt értékkel (12.08%). Az ismertetett augmentációs eljárások a túltanulás ellen is hatnak, így egyfajta regularizációként is tekinthetünk rájuk.

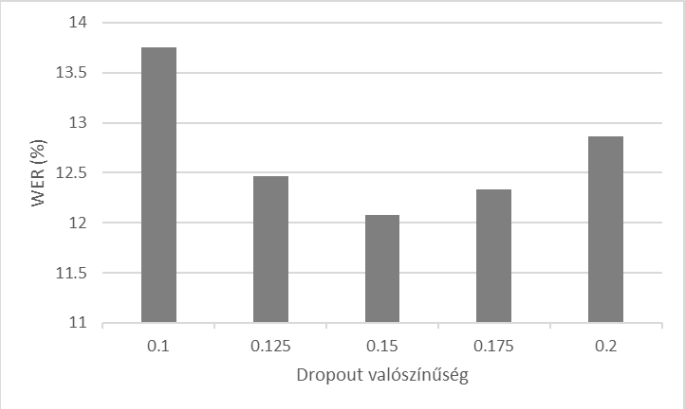
3.2 Hiperparaméterek és kísérletek

A tervezett architektúrában az enkóder magába foglalja a wav2vec2 modult és egy vanilla előrecsatolt alhálózatot, ami két rétegből, rétegenként 1024 neuronból áll. Dekóderként egy attention-ön alapuló rekurrens GRU (Cho és mtsai, 2014) hálózatot specifikáltunk (enkódolási méret: 1024, csatornák száma: 10, bemeneti méret: 128, attention dimenzió: 1024).

Az architektúra enkóder-dekóder része az Adadelta (Zeiler, 2012), míg a wav2vec2 része az Adam (Kingma és Ba, 2017) optimalizálót használta, rendre 1.0 és 1e-4 kezdeti tanulási rátával. Az Adam β_1 értéke 0.9, β_2 értéke 0.999 volt. Az Adadelta esetén a ρ értéke 0.95. A tanítási ráta ütemezése a validációs halmazon mért költségváltozáson alapult (NewBob ütemezés): amennyiben egy epoch során a költség egy bizonyos határértéknél kisebb mértékben csökkent – vagy esetleg nőtt –, úgy csökkentettük a tanulási rátát. A konkrét kötegméretet 3-ra állítottuk, ám az effektív kötegméret 12 volt. A kettőt a gradiens-akkumulációs tényező köti össze: e tényező állításával a hibavisszaterjesztés egy konstans számú lépés után hajtódik csak végre (ebben az esetben 4), nem pedig minden köteg után, így lehetőség adódik nagyobb kötegmérettel történő tanítás szimulálására.

További regularizációként a dropout technikát használtuk (Srivastava és mtsai, 2014). Észrevételeink alapján az ismertetett architektúra erre a hiperparaméterre érzékeny volt, így ezzel mélyebben kísérleteztünk. A hálózatot összesen 100 epoch-on keresztül tanítottuk. Az első 20 epoch során 0.4 súllyal vettük figyelembe a CTC költséget, 0.6-tal a „Negative Log-Likelihood” (NLL) költséget. A fennmaradó 80 epoch során tisztán a NLL alapján történt a hiba számítása. A NLL hiba kalkulálása során 0.1-es értékkel címkesimítást használtunk a túltanulás ellen (Müller és mtsai, 2020).

A tanított rendszer eredményét a szóhibaráta („Word Error Rate”, WER) határozta meg. Értelemszerűen, az alacsonyabb WER érték pontosabb átiratok létrehozását jelenti. A legjobb eredményt (12.08%) 0.15-ös globális dropout értékkel értük el (2. ábra), így a BEAST2 rendszer is ezt az akusztikus modellt foglalja magába.



2. ábra: A modell elért WER eredményei a BEA-Base spontán teszthalmazán a dropout regularizáció függvényében

4 A nyelvi modell

A számláláson alapuló N-gram nyelvmodellek N darab token figyelembevételével tudnak becslést adni a következő token valószínűségére. Elterjedt 4, illetve 5-gram modellek használata, ám belátható, hogy ennyi előtörténet még szavak esetén sem elegendő. A rekurrens neurális hálózatok ezt hivatottak kiküszöbölni: segítségükkel lehetőség adódik távolibb kapcsolatok modellezésére, ami a magyar nyelv esetén is rendkívül hasznos. Jó pár évig megkerülhetetlenek voltak a rekurrens hálók, ám szekvenciális felépítésük miatt lassú és nehézkes volt a tanításuk (Pascanu és mtsai, 2013).

A Transformer egy új megközelítést alkalmaz: a szekvenciális tanítást párhuzamosra cserélve nagyságrendekkel gyorsítja a nyelvi modellek konvergenciáinak sebességét. Ezen felül a grafikus kártyákra (GPU) is jobban illeszkedik, kihasználva azok masszív párhuzamosítási képességeit.

A tervezett hálózatot az akusztikus modellel megegyező unigram szótöredékek segítségével tanítottuk. A hálózatot a GPT (Radford és mtsai, 2018) architektúra alapján terveztük, a SPOK halmazon tanítottuk és validáltuk, majd a legjobbnak ítélt – legalacsonyabb perplexitással (PPL) rendelkező – modellt tovább hangoltuk a BEA-Base spontán átiratain. Kontrollként tanítottunk egy klasszikus 5-gram módosított Kneser-Ney simított (Chen és Goodman, 1999) nyelvmodellt is a KenLM (Heafield, 2011) eszköz felhasználásával.

4.1 Hiperparaméterek és szöveg alapú kísérletek

Annak érdekében, hogy a hálózat optimalizációja során pontosabb becslést kaphassunk a költségfüggvény gradiensére, fontos a megfelelő kötegméret megválasztása. Technikai megfontolásból, a 256 karakternél hosszabb mondatokat kivettük a SPOK adathalmazból, így a kötegméretet nem kellett irracionálisan alacsonyan tartani, ami a tanítási procedúra degradálását idézné elő.

Az előtanítás során használt adatbázis (SPOK) 97%-a adta a tanító-, 2%-a a validációs és 1%-a a teszthalmazt. Az előtanítás során 20 epoch-on át tanult az összes ismertetett hálózat. Az effektív kötegméret minden háló esetén 512 volt, a konkrét kötegméret viszont a tanított modell méretétől függően változott az elérhető memóriakapacitás miatt. Közös vonás továbbá, hogy a tanulási ráta dinamikusan változott a Noam ütemezésnek megfelelően: kezdetben lineárisan nőtt a megadott tanulási rátának függvényében, majd fordítottan négyzetesen csökkent. Az optimalizáló algoritmusnak az Adam-et választottuk, melynek β_1 paraméterét 0.9-re és β_2 értékét 0.98-re vettük. A túltanulás ellen dropout regularizációt alkalmaztunk 0.1 valószínűséggel. A további hiperparaméter-értékek a 2. táblázatban láthatók.

2. táblázat: Az egyes Transformer nyelvmodellek eltérő hiperparaméterei és az előtanítás utáni perplexitás a SPOK-ból képzett validációs halmazon

Modell / param. (db)	Tanulási ráta	Enkóderek (db)	Beágyazás (db)	Belső (db)	Fejek (db)	PPL
baseline / ~67.7M	0.1	12	768	2048	12	12.61
I. / ~100.7M	0.5	14	768	3072	12	6.29
II. / ~149.3M	0.75	14	1024	3072	16	**6.15**
III. / ~171.6M	0.5	24	768	3072	16	6.22

Az előtanítást követően a legalacsonyabb perplexitást elérő nyelvmodellt (II.) a BEA-Base spontán átiratain finomhangolva tanítottuk tovább. A finomhangolásnál elterjedt a mélyebben lévő – tehát a bementhez közelebbi – rétegek befagyasztása. A megoldásunkban az első 4 réteg került befagyasztásra, így a tanítható paraméterek száma ~149.3M-ról ~107.3M-ra csökkent. A továbbtanítás 32 epochon keresztül zajlott, kezdetben 1.75e-6 tanulási rátával. Mivel ez esetben egy előtanított modellt használtunk fel, így a bemelegítési lépésekre már nem volt szükség, helyette a korábban ismertetett NewBob ütemezést használtuk. A további hiperparaméter-értékek a korábban ismertetettekkel megegyeztek. A modell finomhangolás előtti PPL értéke a BEA-Base teszthalmazán 32.2, míg utána 18.5 lett.

A kontroll 5-gram nyelvmodell esetében a finomhangolás nem értelmezhető, így azt külön a SPOK-on és külön a BEA-Base megegyező részén tanítottuk. Előbbi esetben 13.79, utóbbi esetben 42.13 PPL értéket ért el, ami mindkét esetben rosszabb, mint a neurális hálózatok által elért eredmény.

5 Kísérleti eredmények

Az akusztikus modell kimeneteit különböző stratégiák mentén lehet dekódolni: „greedy” módon, mindig a legvalószínűbb tokeneket választva, valamint a nyelvmodell segítségével átsúlyozva. Utóbbit a „beam-search” (nyalábkeresés) segítségével érdemes implementálni.

A finomhangolt modelleket a „shallow fusion” (Toshniwal és mtsai, 2018) megközelítéssel kapcsoltuk össze: az akusztikus háló hipotézisei a nyelvi modell segítségével kerültek újrasúlyozásra.

A nyalábkeresési algoritmus hiperparaméter-optimalizációját követően a következő hiperparaméter-értékekkel történt végleges kiértékelés: a nyaláb szélességét 84-re állítottuk, külön számításba vettük a CTC költséget 0.014 súllyal. A nyelvi modell súlyát 0.285-re vettük. A számontartott hipotéziseket a hosszukkal normalizáltuk (Wu és mtsai, 2016). Ez különösen fontos, mert enélkül az eljárás a rövidebb szekvenciákat preferálná, hiszen azok log-valószínűségeinek összege nagyobb, mint a hosszabb szekvenciáknak. Mind az akusztikus, mind a nyelvi modell predikcióit 1.05 „temperature”-rel mintavételeztük. Az „end-of-sentence” határ (Hannun és mtsai, 2019) értékét 2.5-re, valamint a „coverage penalty” (Chorowski és Jaitly, 2016) értékét 3.0-ra állítottuk.

A tanítás mindkét modell esetén két darab NVIDIA RTX 3090-es, 24 GB VRAM-mal rendelkező GPU-n történt. Az egyes tanítások több napot vettek igénybe az összes tanított hálózat esetén. A könnyebb összehasonlíthatóság érdekében a 3. táblázat tartalmaz információt a korábbi és a mostani modellek teljesítményéről.

3. táblázat: Az alap BEAST, egy 440k órányi adaton előtanult és finomhangolt wav2vec alapú rendszer, az uráli nyelveken előtanított, majd finomhangolt wav2vec2 struktúra, valamint a BEAST2 WER eredményei a BEA-Base spontán teszthalmazán mérve

Modell	LM	WER (%)
BEAST[1]	-	16.62
440k-wav2vec-finetuned[2]	-	15.61
43k-wav2vec2-uralic-finetuned	-	12.08
BEAST2	neurális (II.)	**10.98**

Az eredmények tekintetében kijelenthető, hogy a wav2vec2 előtanítása uráli nyelveken, majd finomhangolása a BEA megfelelő részadatbázisán szignifikáns javulást ért el a BEAST-hez képest. Ehhez egy explicit nyelvmodellt kapcsolva (BEAST2) képesek voltunk a nyalábkeresés során összességében ~34%-kal csökkenteni a spontán teszthalmazon mért szóhibaarányt a BEAST rendszerhez képest. A továbbfejlesztett modelleket ingyenesen elérhetővé tesszük[3] kutatási célokra.

A kiértékelése használt spontán teszthalmaz 4.91 órányi hanganyagát a BEAST2 rendszer ~40 perc alatt dekódolta egy NVIDIA RTX 3090 GPU-t használva.

[1] Mihajlik és mtsai, MSZNY 2022
[2] Mihajlik és mtsai, LREC 2022
[3] https://phon.nytud.hu/bea/bea-base

6. Összegzés

A Transformer alapú beszédleiratozás ma már az első számú beszédfelismerési megközelítéssé vált világszerte. Ugyanakkor magyar nyelvre csak minimális mennyiségű felügyelt adaton tanult modellek érhetők el a BEAST rendszeren kívül. Megmutattuk, hogy az előtanított modell megfelelő megválasztásával, hiperparaméter-optimalizációjával, valamint korszerű, mély neurális nyelvmodell integrálásával jelentős javulás érhető el még a kiforrott BEAST-hez képest is – a cikk írásának pillanatában a legjobb eredményt értük el (WER=10.98%). Ez a várakozásaink szerint még tovább javítható további hiperparaméter-optimalizációval, illetve a BEA-Base adathalmaz várható bővülésével.

Köszönetnyilvánítás

Köszönettel tartozunk a munkánkat támogató, NKFIH-135038 projektazonosítójú, „Prozódiai szerkezet és mondattípusok vizsgálata nagy beszédadatbázisokon mély tanulási támogatással” című, valamint az NKFIH-143075-ös és NKFIH-828-2/2021 (MILab) projekteknek. Ezen felül köszönjük mindenkinek, aki munkájával hozzájárult a BEA adatbázis létrejöttéhez.

Bibliográfia

Baevski, A., Zhou, H., Mohamed, A., Auli, M.: wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, arXiv: 2006.11477v3 (2020)

Chen, S F., Goodman, J.: "An empirical study of smoothing techniques for language modeling," Computer Speech & Language, vol. 13, no. 4, pp. 359–393 (1999)

Cho, K., Merrienboer, B., Bahdanau, D.: On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, arXiv: 1409.1259v2 (2014)

Chorowski, J., Jaitly, N.: Towards Better Decoding and Language Model Integration in Sequence-to-Sequence models, arXiv: 1612.02695v1 (2016)

Gósy, M.: Magyar spontánbeszéd-adatbázis – BEA, Beszédkutatás, pp. 194–207 (2008)

Gósy, M., Gyarmathy, D., Horváth, V., Gráczi, T.E., Beke, A., Neuberger, T., Nikléczy, P.: BEA: Beszélt nyelvi adatbázis, In: Gósy Mária (szerk.): Beszéd, adatbázis, kutatások, Budapest, Akadémiai Kiadó, pp. 9–24 (2012)

Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, ICML 2006 – Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)

Hannun, A., Lee, A., Xu, Q., Collobert, R.: Sequence-to-sequence Speech Recognition with Time-Depth Separable Convolutions, arXiv: 1904.02619v1 (2019)

Heafield, K.: KenLM: Faster and Smaller Language Model Queries, Proceedings of the Sixth Workshop on Statistical Machine Translation, pp.: 187–197 (2011)

Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization, arXiv: 1412.6980v9 (2017)

Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, arXiv: 1808.06226v1 (2018)

Mihajlik, P., Balog, A., Gráczi, T. E., Kohári, A., Fegyó, T., Mády, K.: „Releasing the Beast" – A BEA gépi beszédleiratozási feladat, megközelítések, eredmények, XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)

Mihajlik, P., Balog, A., Gráczi, T.E., Kohári, A., Tarján, B., Mády, K.: BEA-Base: A Benchmark for ASR of Spontaneous Hungarian, In: Proceedings of the 13th Conference of LREC 2022, pp. 1970–1977 (2022)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space, arXiv: 1301.3781v3 (2013)

Müller, R., Kornblith, S., Hinton, G.: When Does Label Smoothing Help?, arXiv: 1906.02629v3 (2020)

Oravecz Cs., Váradi T., Sass B.: The Hungarian Gigaword Corpus, In: Proceedings of LREC 2014 (2014)

Park, D.S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E.D., Le, Q.V.: SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, arXiv: 1904.08779v3 (2019)

Pascanu, R., Mikolov, T., Bengio, Y.: On the Difficulty of Training Recurrent Neural Networks, arXiv: 1211.5063v2 (2013)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners, OpenAI blog 1(8), 9 (2019)

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J., Yeh, S., Fu, S., Liao, C., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R., Bengio, Y.: SpeechBrain: A General-Purpose Speech Toolkit, arXiv: 2106.04624v1 (2021)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, pp. 1929–1958 (2014)

Toshniwal, S., Kannan, A., Chiu, C., Wu, Y., Sainath, T.N., Livescu, K.: A Comparison of Techniques for Language Model Integration in Encoder-Decoder Speech Recognition, arXiv: 1807.10857v2 (2018)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention Is All You Need, arXiv: 1706.03762v5 (2017)

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., Dupoux, E.: VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation, arXiv: 2101.00390v2 (2021)

Watanabe, S., Hori, T., Kim, S., Hershey, J.R., and Hayashi T.: Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253 (2017)

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv: 1609.08144v2 (2016)

Zeiler, M.D.: Adadelta: An Adaptive Learning Rate Method, arXiv 1212.01v1 (2012)

Magyar nyelvű neurális beszédszintézis vizsgálata dialógus helyzetben

Zainkó Csaba[1], Csapó Tamás Gábor[1], Bartalis Mátyás[1], Németh Géza[1], Németh Norbert[2], Szász Gábor Krisztián[2], Szviridov István[2]

[1]Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
[2]IdomSoft Zrt. Rendvédelmi Fejlesztési Ágazat
zainko@tmit.bme.hu

Kivonat Jelen tanulmányban olyan mély neurális hálózat alapú beszédszintetizátor rendszert (DNN-TTS) mutatunk be, amely hangsorozat bemenetet vár és a beszéd hullámformáját két lépésben állítja elő, melspektrogram köztes reprezentációt használva. Részletesen bemutatjuk és összehasonlítotjuk a Tacotron2+WaveGlow és FastPitch+HiFi-GAN (tőlünk független) rendszereket és komponenseiket. A magyar nyelvű adatokon végzett saját kísérletekben három beszélővel (két női és egy férfi) generálunk szintezitált beszédmintákat. Szubjektív, MUSHRA típusú meghallgatásos tesztjeink során a tesztalanyok a DNN-TTS beszédszintetizátorral előállított mondatokat lényegesen természetesebbnek minősítették, mint a HMM-TTS alaprendszert. A szintetizált beszédminták minősége (természetessége) ugyan nem éri el a természetes beszéd szintjét, de közel áll hozzá (Tacotron2: 58%, FastPitch: 73%, természetes: 89%). Összességében a tesztelők a FastPitch rendszert preferálták a Tacotron2-vel szemben természetesség szempontjából. A ChatBot dialógusba ágyazott tesztek eredménye szerint a női beszélők preferáltak, és a DNN-TTS rendszerekkel előállított beszéd érthetőbb, természetesebb, mint a HMM-TTS alaprendszer, és tesztelők a válaszokat is relevánsabbnak és részletesebbnek érezték az alaprendszerhez képest.
Kulcsszavak: beszédtechnológia, DNN, TTS

1. Bevezetés

A gépi szövegfelolvasók (más néven szöveg-beszéd átalakítók vagy beszédszintetizátorok, Text-To-Speech, TTS) legújabb generációi gépi tanuláson alapulnak. Kb. 2015-ig az úgynevezett rejtett Markov-modelleket (HMM, Zen és mtsai, 2009; Tóth és Németh, 2009; Tóth, 2013) alkalmazó beszédszintézis volt a fő irány, melyet a legújabb rendszerekben azóta fokozatosan felváltott a mély tanuláson (deep learning) alapuló gépi szövegfelolvasás (DNN-TTS, Zen és mtsai, 2013; van den Oord és mtsai, 2016; Zainkó és mtsai, 2017). A fejlődés során a korábbi technológiák nem avulnak el, mert marad komparatív előnyük, például a HMM szintetizátorok jellemzően kisebb erőforrást (CPU, memória, betanító adatbázis) igényelnek, mint a DNN-TTS megoldások.

A jelenleg használt beszédszintetizátorok bemenete a legtöbb esetben szöveg, egyes rendszereknél pedig hangsorozat. A beszédet a bemenetből általában

két lépésben állítják elő. Az első lépésben egy köztes reprezentációt állítanak elő, amely tipikusan a széles körben elterjedt mel-spektrogram. Ezt a köztes reprezentációt utána egy másik modell alakítja át hullámformává. Vannak úgynevezett end-to-end megoldások is, amelyek ezt egyetlen lépésben valósítják meg. Ezek minimálisan jobb minőséget tudnak, de kevésbé rugalmasak. Részletes áttekintő a beszédszintézis legújabb kutatási eredményeiről Tan és mtsai (2021) összefoglaló cikkében olvasható.

Jelen tanulmányban egy olyan rendszert mutatunk be, amely hangsorozat bemenetet vár és a beszéd hullámformáját két lépésben állítja elő. A bemeneti kvázi hangsorozatot a bemutatott modellektől független szabály alapú fontetikus átíróval állítjuk elő a megszólaltatni kívánt szövegből.

2. Módszerek

2.1. Adatok

TTS célokhoz stúdióminőségű beszéd a megfelelő, jellemzően 40 dB vagy jobb jel/zaj viszony szükséges a beszédjelen, ezért a saját fejlesztésű PPBA adatbázis (Olaszy, 2013; Tóth és mtsai, 2012) férfi és női beszélőinek hanganyagával végeztük a beszédszintézis kísérleteket. A stúdióban 44 100 Hz mintavételi frekvenciával és 16 bites lineáris kvantálással rögzítettük a beszédjelet. A tanítás során – hatékonyság növelés miatt – kisebb mintavételi frekvenciával dolgozunk, ezért jelen esetben 22 050 Hz-re újramintavételeztük a beszédet.

Jelen tanulmányban fonetikailag gazdag szöveget (Olaszy, 2013) használunk, amelyre jellemző, hogy a legváltozatosabb formában tartalmazza a hangkapcsolatokat. A hangsorozatot szabály alapú fonetikus átíróval állítjuk elő. A könnyebb kezelhetőség miatt nem fonetikus ábécét használunk, hanem közelítő betűképpel jelöljük a beszédhangokat. A szöveg feldolgozásakor a rendszer feloldja a rövidítéseket, átírja a számjegyekkel megadott adatokat, például összegeket, dátumokat. Így a modell bemenetére minden esetben olyan bemenet kerül, amely már csak hangokat vagy betűsorozatokat tartalmaz, számjegyeket vagy más jeleket (pl. százalékjelet) már nem.

A tanító adatok mondatszintűek, egy elem egy mondatból áll. A felhasznált adatok esetében 10 beszélőtől, beszélőnként 2000 mondatot használunk. A 2000 mondat minden beszélő esetén azonos. A mondatokat 3 halmazra osztjuk szét a mély tanulásos eljárások tanításánál szokott módon (tanító / validációs / teszt). A modellek tanításakor a teljes adathalmaz mondatainak kb. 2%-a volt a teszthalmaz, 7,5%-a a validációs halmaz, a maradék része a tanító halmaz. Az azonos mondatok minden beszélőnél ugyanabba a halmazba kerültek, tehát ha egy mondat a validációs halmaz része volt, akkor az minden beszélő esetében a validációs halmazban volt. Keverés esetében a halmazok függetlensége nem lett volna biztosítva.

2.2. A DNN modell bemutatása

Jelen kutatás során 2-2 modellt alkalmaztunk, mindegyik hálózat mély tanulás alapú, sok réteget tartalmazó struktúra. A beszédszintézis során nem end-to-

end megoldásokat használtunk, hanem három komponens segítségével állítottuk elő a beszédet. Ennek az az előnye, hogy jobban lehet kontrollálni a folyamatokat, lehetőség van a köztes reprezentációk módosítására, ellenőrzésére, illetve az egyes modellek egyedi tanítására, optimalizálására. Az end-to-end megoldások többnyire gyorsabbak és az adott halmazon akár minimálisan jobb minőséget is képesek előállítani, használatuk végleges rendszerek esetén indokolt, kutatási, kísérleti megoldások esetében a rugalmatlanságuk miatt kevésbe kifizetődő.

A beszédszintetizátorok összeállítása során 3 komponenst használtunk:

1. Szövegfeldolgozó, betű-hang konverzió (Profivox)
2. Hang-mel spektrogram konverter (Tacotron2, FastPitch)
3. Mel spektrogram-Hullámforma konverter (WaveGlow, HiFi-GAN)

A szövegfeldolgozó és betű-hang konverziós megoldás a korábbi diád, triád hullámforma-összefűzéses megoldás moduljait használja, részletesen Olaszy és mtsai (2000) tárgyalja.

A hang-mel spektrogram konverziós modellek esetében két rendszert vizsgáltunk, mindkettőt a saját beszédadatbázissal tanítottuk be. A Tacotron2 modell egy figyelmi mechanizmust (attention mechanism) is alkalmazó seq-to-seq modell, míg a FastPitch egy gyorsabb tanítást és futtatást is biztosító transzformer alapú megoldás.

A hullámforma előállításhoz a WaveNet alapú WaveGlow modellt használtuk, illetve a GAN megoldást tartalmazó HiFi-GAN architektúrát. A WaveGlow modellhez képest a HiFi-GAN lényegesen gyorsabb, illetve rendelkezésre áll a HiFi-GAN esetében több, változó minőségű és sebességű konfiguráció, amely lehetőséget biztosít a követelményekhez jobban illeszkedő verzió kiválasztására.

Tacotron2 A Tacotron2 (Shen és mtsai, 2018) bemenete valamilyen karaktersorozat, kimenete 80 csatornás mel-spektrogram. A modell esetében a következő mel-spektrogram paramétereket alkalmaztuk:

- Mel-csatornák száma: 80
- Minimális Mel frekvencia: 0 Hz
- Maximális Mel frekvencia: 8000 Hz
- Mintavételi frekvencia: 22050 Hz
- Ablakméret: 1024 minta
- Eltolás mérete (Hop size): 256 minta

A Tacotron2 bemenetén egy karakter beágyazás található, a modellben szabadon definiálható, hogy hány különböző bemeneti karakter használható. A szövegelőfeldolgozásnak biztosítania kell, hogy az előre definiált karakterkészleten kívül más karakter ne kerülhessen a bemenetre, mert abban az esetben a modell nem futtatható, hiba keletkezik. Mivel a modellünk bemenetére már az előfeldolgozott szöveg kerül, így biztosított, hogy a nem kívánt karakterek kezelve legyenek.

Annak érdekében, hogy a kiejtés felett nagyobb kontroll álljon rendelkezésre, a bemeneti betűsorozat esetében betűképpel jelölt hangsorozatot alkalmaztunk, pl.: *"[START] sz é p i d ő l e sz [END]"*. A bemeneten megkülönböztetjük a rövid és a hosszú hangokat. A hangkódok mellett szükségesek még start és stop

karakterek, illetve szünetek és írásjelek (pl. kérdőjel). A konfigurációtól függően 90-95 különböző bemeneti karaktert használtunk a tanításnál.

A Tacotron2 módosításával lehetőség van több beszélővel együttes tanításra is, ekkor egy vektor segítségével megadható, hogy éppen melyik beszélő tartozik az adott bemeneti és kimeneti adatokhoz. A modell így több-beszélős lesz, ami a tapasztalatok alapján jobb beszédminőséget eredményez. Az adott Tacotron2 modell esetében a PPBA (Olaszy, 2013) 10 beszélőjével tanítottuk be a hálózatot. A futtatásnál a vektor megadásával választható, hogy melyik beszélő hangján szeretnénk a szintetizálást elvégezni. A 10 beszélő között található az FF1, NOI1-es hang is, ezért ezeket közvetlenül elő lehet állítani, további beszélőadaptáció nélkül. A NOI2-es hang továbbtanítás során jött létre, a betanított 10 beszélős modellt tanítottuk tovább csak a NOI2-es hang segítségével. A Tacotron2 tanítása időigényes, 3 Bi-LSTM réteget is tartalmaz, amelyekre jellemző a lassú taníthatóság. A 10 beszélős modell továbbtanításával jelentős gyorsulást értünk el, a NOI2 hang egy nagyságrenddel rövidebb idő alatt betanítható volt így, kb. 4-5 óra egy NVIDIA Titan Xp-n.

WaveGlow A WaveGlow modell (Prenger és mtsai, 2019) mel-spektrogramból állítja elő a hullámformát. A konfigurációs paraméterek megegyeznek a Tacotron2-vel. A 80 csatornás mel reprezentációból egy WaveNet alapú megoldás segítségével áll elő a beszédjel. A modell elég általános, nem feltétlenül szükséges, hogy tanításnál a célhang segítségével adaptáljunk. Egy női beszélővel betanított hálózat a hasonló hangmagasságú női hangokat is magas minőségben képes előállítani. Viszont ha lényegesen eltérő alapfrekvenciájú beszéd kell (pl. férfi hang alacsony F0-lal), már jól hallható minőségromlás áll elő. A WaveGlow számításigényes és a tanítási folyamat is lassú: az NVIDIA által publikált egybeszélős modell tanításához is 580 000 iteráció volt szükséges egy 8 GPU-s környezetben.

Rendelkezésre áll egy NVIDIA által publikált modell, amely az LJ Speech adatbázis egyetlen női beszélőjével lett betanítva [1]. Méréseink szerint ez női beszélők esetében megfelelő, de férfi beszélők esetén nem használható, mély hangú férfi hangoknál durva torzítások jelentkeznek.

Azért, hogy a férfi beszéd is megfelelő minőségben előállítható legyen, a PPBA adatbázis segítségével tanítottunk egy WaveGlow modellt. Ez a hálózat az adatbázis 5 női és 5 férfi hangjával lett tanítva, így alkalmas a férfi beszélők beszédének előállítására is. A modell 635 000 iterációig volt tanítva.

A modell futtatása is időigényes, GPU-n ad csak megfelelő sebességet.

FastPitch A FastPitch modell (Łańcucki, 2021) több lényeges elemében is eltér a Tacotron2-től. Lényeges különbség, hogy nem tartalmaz LSTM rétegeket, helyette transzformer alapú megoldást használ. Ez lényegesen gyorsabb tanítást és futtatást eredményez. További lényeges eltérés, hogy a modell elkülönítve tartalmaz az alapfrekvenciáért és a hangidőtartamokért felelős komponenseket, így

[1] `https://drive.google.com/file/d/1cjKPHbtAMh_4HTHmuIGNkbOkPBD9qwhj`

azok külön-külön történő vezérlésére is lehetőség nyílik. Ez azt jelenti, hogy ellentétben a Tacotron2-vel, a FastPitch esetében lehetőség van a sebesség, illetve az alapfrekvencia egyszerű paraméterezésére, módosítására.

A modell bemenete hasonló a Tacotron2-höz: lehet betű vagy hang alapú is. A kompatibilitás megvalósítása érdekében: a Tacotron2-höz hasonló betűképpel jelölt hangsorozat bemenetet valósítottunk meg: 90-95 bemeneti szimbólum áll rendelkezésre, a rövid és hosszú beszédhangok, néhány kiegészítő szimbólummal.

A kimenet mel-spektrogram paraméterei megegyeznek a Tacotron2-nél ismertetett paraméterekkel, így a FastPitch kimenete használható a WaveGlow modellel is.

HiFi-GAN A HiFi-GAN (Kong és mtsai, 2020) egy GAN hálózat, amely a mel-spektrogram bemenetet alakítja át hullámformává. Az általunk használt megoldás 22 050 Hz-es mintavételi frekvenciát használ, és a HiFi-GAN szerzők által használt (v1) paraméterezést használjuk (Kong és mtsai, 2020).

A HiFi-GAN kevésbé univerzális mint a WaveGlow, azaz a generálandó beszélő hangjához közelebb kell lennie a tanító hangoknak. Amennyiben eltérő hangot (akár azonos nemű, de alapfrekvenciában jobban eltérő) állítunk elő, akkor a WaveGlow-hoz képest jelentősebb minőségi degradációt tapasztalhatunk.

A HiFi-GAN tanítása több szempontból is nehéz. Egyrészt az ilyen GAN hálózatok lassan taníthatók, másfelől a hiba (loss) értéke nem teljesen korrelál a minőséggel. Megfigyelhető volt, hogy hosszabb tanítás esetén, egyre kisebb validációs hiba érték mellett is a hangminőség nem javult, hanem erősödő torzítás miatt egyre romlott.

A felhasznált modellek továbbtanítás segítségével jöttek létre, a publikált súlyokat [2] tanítottuk tovább a PPBA 10 beszélőjével. Így az egybeszélős modellt elég univerzálissá tudtuk fejleszteni ahhoz, hogy mind férfi, mind női beszélők hangjait elő tudja állítani.

2.3. Tesztkörnyezet kialakítása

A TTS-ek szubjektív értékelése során a tesztelőket a dialógus és a felhasználás környezete is befolyásolhatja. A vizsgálatunk során arra törekedtünk, hogy ezen hatásokat amennyire tudjuk közelítsünk és így a kapott szubjektív értékelések a valós felhasználásnál ténylegesen fennálló érzeteket tükrözze. A tesztmondatok előállítása egy dialógusrendszer működési logikáját és felépítését követte, a minták előállításánál a komponensek integrációja úgy valósult meg, ahogy a valós felhasználás során is megvalósult volna.

Csatoló felület A TTS szolgáltatást HTTP hívásokkal lehet elérni, a felület elrejti a mögöttes technológiát, a korábbi HMM megoldással megegyező API biztosítja a kompatibilitást. A csatolón keresztül mondatonként lehet a szöveget szintetizálni. Az eredményt a webszerver egyetlen audióban adja vissza.

[2] `http://drive.google.com/drive/folders/1-eEYTB5Av9jNql0WGBlRoi-WH2J7bp5Y`

Szövegelőfeldolgozó modul A szöveg előfeldolgozása elengedhetetlen lépése a beszéd szintézisnek. Itt történnek azok a feldolgozási lépések, amelyek tetszőleges szövegek felolvasását lehetővé teszik, pl: számok, dátumok, összegek feloldása, speciális karakterek értelmezése, rövidítések feloldása, stb. A modul szöveg be- és kimenettel rendelkezik, alapvetően szabály vagy szótár alapú megoldás. A számok feloldása jelenleg is szabály alapon a legmegbízhatóbb, mivel a legtöbb esetben nem környezetfüggő a feloldása, egyszerű logikával is megoldható.

Betű-hang konverzió Mind a Tacotron2, mind a FastPitch képes kezelni betűszintű bemenetet, de ebben az esetben az üzemeltető számára kevesebb kontroll áll rendelkezésre, kevésbé tudja befolyásolni a kiejtést. A szavak, mondatok kiejtése a tanító adathalmazon alapul, speciális karaktersorozatok (pl. történelmi személynevek) kiejtése egyáltalán nem, vagy csak nehezen megoldható. Ennek kezelésére a DNN modulok bemenetére a betűsorozat helyett az ahhoz tartozó hangkód-sorozatot adjuk meg. A hangkód-sorozatot közelítő betűképpel kódoljuk, így nem szükséges a modelleken változtatni. A hangkód-sorozat előállítására a ProfiVox beszédszintetizátor betű-hang konverziós modulját használjuk (Olaszy és mtsai, 2000). A modul segítségével nem csak a hangkód-sorozatot határozzuk meg, hanem a hangidőtartamokat és a fonológiai hanghosszúsági kategóriát is (rövid vagy hosszú hang). A hangidőtartamok pontos megvalósítása nem történik meg, mivel a DNN modellek saját időzítési adatokat generálnak, de a bemeneten jelöljük a hangkódnál a fonológiai hanghosszúsági kategóriát, azaz hogy rövid vagy hosszú hangról van-e szó.

A hullámforma előállítása A DNN modellek előállítása történhet tisztán CPU vagy CPU+GPU-s környezetben. A modellek futtatása jelentősen gyorsabb GPU-s környezetben. A Tacotron2 és WaveGlow modellek csak CPU (azaz GPU rendelkezésre állása nélkül) esetén lassúak, a jelentős konverziós idő miatt online, valós idejű dialógus rendszerben nem használhatók. GPU-s gyorsítás esetén a valós idő elérhető, amely már megfelelő erre a célra. A FastPitch és HiFi-GAN modellek nagyságrenddel gyorsabb futást tesznek lehetővé, szerver környezetben több CPU használatával a valós idejű működés elérhető, GPU-s gyorsítással pedig több csatorna párhuzamos kiszolgálása is megvalósítható egyetlen gépen.

A hullámformát a bemutatott modelleknél két lépésben állítja elő a rendszer. Az első lépésben a mel spektrogram reprezentációt állítja elő a Tacotron2 vagy a FastPitch modell. Ezek kompatibilisek egymással, ugyanolyan paraméterezéssel készültek, a két modell kimenete felcserélhető. Ezt a mel spektrogram reprezentációt utána WaveGlow vagy HiFi-GAN modellek segítségével alakítja át a rendszer hullámformává. Az így elkészült hullámformán további feldolgozás már nem történik, az elkészült beszédjel a csatoló HTTP felületen keresztül jut el a dialógus-vezérlőhöz, majd a későbbiek során a megfelelő időben lejátszódik.

2.4. Előzetesen elvárt hangminőség

A különböző rendszerek angol nyelven már más szerzők által implementálásra kerültek (Shen és mtsai, 2018; Łańcucki, 2021; Prenger és mtsai, 2019; Su és mtsai,

2020), és különböző összehasonlító vizsgálatokat is végeztek rajtuk. Egybeszélős modell esetén a FastPitch kismértékben, míg több beszélős esetben lényegesen jobb minőséget adott, mint a Tacotron2. A különbség abból adódott, hogy több beszélős környezetben a FastPitch azonos teljesítményt nyújtott, míg a Tacotron2 gyengébbet. Sebesség tekintetében a FastPitch kb. 60-szor gyorsabb (Łańcucki, 2021). A mel-spektrogram - hullámforma konverziós modellek esetében a beszédszintetizátor elemeit kihagyva vizsgálták a minőséget. A hullámformákat átalakították mel-spektrogrammra, majd ezt a vizsgált modellel visszakonvertálták. Az eredmények szerint a WaveGlow 0.55 ponttal gyengébben végzett, mint a HiFi-GAN v1-es verziója, miközben a HiFi-GAN kb. 7-szer gyorsabban fut. A kb. 60-szor gyorsabb HiFi-GAN v3-as verzió is jobban teljesített a WaveGlow modellnél az összehasonlításban.

Az irodalomban megtalálható összehasonlítás alapján azt várjuk, hogy a HMM rendszernél a DNN megoldások jobb minőséget adnak. A két fajta DNN megoldás közül a FastPitch+HiFi-GAN páros várhatóan jobb eredményt ad a teszteken. A beszédszintetizátorok minősége általában függ a beszélő alany hangjától. Ennek vannak objektív és szubjektív okai is. Szubjektív, hogy mennyire tartják kellemesnek, barátságosnak az adott személy hangját, objektív pedig, hogy a hangnyomás-változás fizikai folyamatai mennyire illeszkednek a gépi algoritmusokhoz. Ezek alapján nem tudunk előzetes becslést adni, hogy a két női és az egy férfi hang közül melyik teljesít jobban.

3. A szintetizált beszédminták vizsgálata

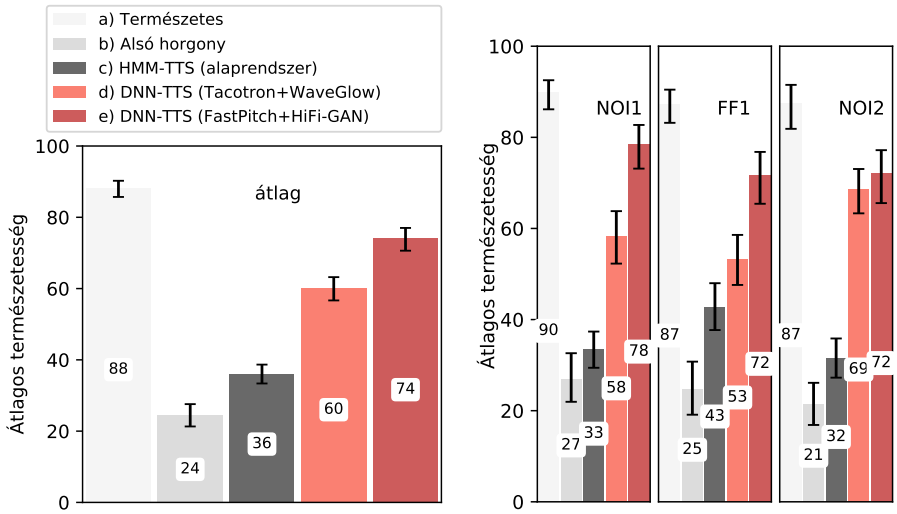
A három beszélőtől (két nő: NOI1 és NOI2 és egy férfi: FF1) mondatokat szintetizáltunk, majd ezek egy részét kiválasztottuk internetes meghallgatásos tesztekhez. A beszédszintézis kutatásának egyik célja, hogy a jövőben az emberek könnyedén használhassák az alkalmazásával készített szolgáltatásokat, és a ráépülő alkalmazásokat. Így a tesztelés folyamán a felhasználói élmény felmérése kiemelten fontos, hiszen lehet bármilyen hasznos egy alkalmazás, ha a felhasználóknak kényelmetlen, nehézséget okoz a használata, akkor nem fogják használni.

3.1. MUSHRA típusú szubjektív meghallgatásos teszt

Először úgynevezett MUSHRA-jellegű (MUltiple Stimuli with Hidden Reference and Anchor; ITU-R Recommendation BS.1534) szubjektív meghallgatásos tesztet végeztettünk el, hogy felmérjük, melyik modell hogyan teljesít, ezáltal képet kapva az eredményességükről. A MUSHRA teszt széles körben elterjedt beszédszintetizátorok minőségének összehasonlítására.

A teszt során a teszthalmazban szereplő szintetizált beszédmintákat kellett meghallgatniuk a kitöltőknek. Egy-egy oldalon mindegyik hangminta ugyanahhoz a mondathoz tartozott. Referenciaként szerepelt az eredeti (nem szintetizált) hangfájl is. A minőségi skála alsó pontjának érzékelését segítendő a tesztek mindig tartalmaztak egy rossz minőségű ún. „alsó horgony” mondatot, melyet az eredeti beszédminta torzításával állítottunk elő (MP3 kódolás, 8kbps). A kitöltők

feladata az volt, hogy minden egyes mintát osztályozzanak egy skálán aszerint, hogy mennyire hangzik természetesnek az adott beszéd (0: nagyon rossz, 100: teljesen természetes). A kitöltők nem tudták, melyik minta melyik modellhez tartozott és a minták sorrendje tesztesetenként meg is volt keverve.



1. ábra. A MUSHRA teszt eredményei: a különböző TTS modellek átlagos szubjektív értékei az egész tesztet nézve (fent) és beszélőnként (lent), a 95%-os konfidenciaintervallumokat is feltüntetve.

A MUSHRA-jellegű meghallgatásos teszt eredményét mutatja be az 1. ábra, a három beszélőre külön-külön (alul), valamint összesítve (felül). Összesen három modellt hasonlítottuk össze: az alaprendszert (HMM-TTS: Tóth, 2013; Csapó, 2013; Csapó és Németh, 2017), továbbá a DNN-TTS két változatát: Tacotron2+WaveGlow és FastPitch+HiFi-GAN, melyeket a cikk korábbi részében ismertettünk. A tesztben 10 ép hallású, magyar anyanyelvű kísérleti személy vett részt (1 nő és 9 férfi; 24–63 évesek; átlag: 38 év), 4 rendelkezett beszédtechnológiai ismeretekkel. Átlagosan 18 percig tartott a hangminták meghallgatása. A „természetes” minták érték el a legjobb eredményt, míg az „alsó horgony” a legrosszabbat – ezeknek a referenciáknak az volt a célja, hogy a tesztelők tudják mihez viszonyítani a szintetizált mondatokat. A tesztből kiderült, hogy a HMM-TTS alaprendszer viszonylag gyenge eredményt tudott elérni, átlagosan 36%-ra értékelték a tesztelők. A kutatás aktuális szakaszában javasolt DNN-TTS mind a Tacotron2+WaveGlow, mind a FastPitch+HiFi-GAN változatban jobb eredményt tudott elérni, mint az alaprendszer (Tacotron2+WaveGlow: 60%, FastPitch+HiFi-GAN: 74%). A meghallgatásos tesztben előállt rangsort Mann-Whitney-Wilcoxon teszttel is összevetettük, 95%-os konfidenciaszintet használ-

va. Ennek alapján a DNN-TTS segítségével szintetizált mondatok (mind Tacotron2, mind FastPitch) szignifikánsan jobb minőségűnek bizonyultak a horgonyhoz és az alaprendszerhez (HMM-TTS) képest, ugyanakkor szignifikánsan kevésbé voltak természetesek az eredeti referencia mondatokhoz képest.

A tendenciák hasonlóak mindhárom beszédhang esetén: HMM-TTS (alaprendszer) < DNN-TTS (Tacotron2+WaveGlow) < DNN-TTS (FastPitch+HiFi-GAN). NOI1 és FF1 esetén a FastPitch lényegesen természetesebb, mint a Tacotron2, míg a NOI2 beszélő esetén a két DNN-TTS közti különbség kevésbé jelentős (és a statisztikai teszt szerint nem különbözik szignifikánsan).

A MUSHRA teszt konklúziójaként a %-os értékekből is látszik, hogy a korábbi HMM-TTS alaprendszerhez képest a DNN-TTS eredménye lényegesen jobb, és összességében a tesztelők a legmodernebb FastPitch rendszert preferálják.

3.2. Dialógusba ágyazott szubjektív meghallgatásos teszt

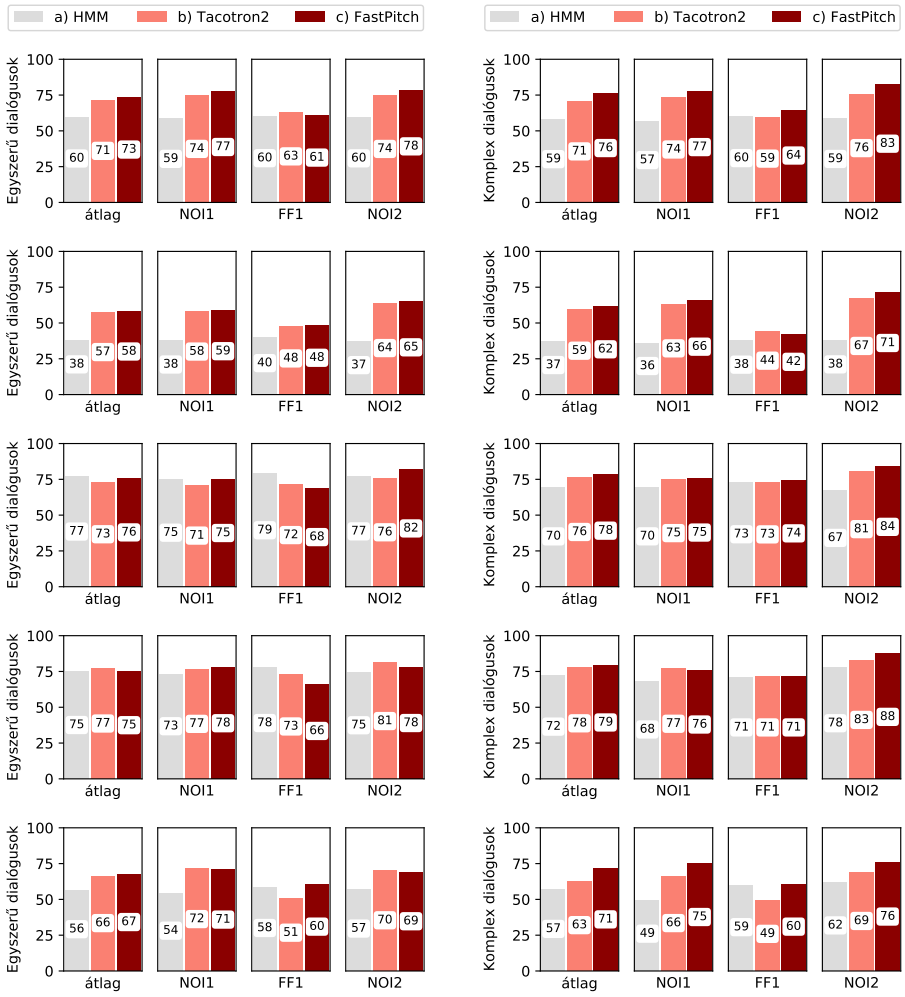
A második teszt, amelyet elvégeztünk, dialógusba ágyazva tartalmazta a szintetizált beszédmintákat. A dialógusok szöveges anyagai egy ChatBot rendszerből származtak. Most egy-egy oldalon csak egy, hosszabb hangminta szerepelt, videó formájában, mely egy elképzelt ChatBot dialógust imitált.

Összesen 126 videót generáltunk a teszthez, azaz beszélőnként (NOI1 / NOI2 / FF1) és rendszerenként (HMM-TTS / DNN-TTS-Tacotron2 / DNN-TTS-FastPitch) 14 dialógust. Ebből a 14-ből egy dialógus komplex volt, azaz több ügyfél kérdést és szintetizált mondatot tartalmazott; hosszuk kb. 1 perc volt. A többi 13 dialógus rövidebb volt, csak egy ügyfél kérdést és chatbot választ tartalmazott. Egy adott tesztelő összesen 9 dialógust értékelt egy adott beszélőtől: 3 komplexet (a 3 rendszerrel), és 6 egyszerűt (melyet a rendszer véletlenszerűen sorsolt a fenti 13-ból).

A kitöltők feladata itt az volt, hogy minden egyes mintát osztályozzanak öt kérdésre válaszolva: 1) 'Mennyire érthető a beszéd?', 2) 'Mennyire természetes a beszéd?', 3) 'Releváns-e a válasz?', 4) 'Elég részletes-e a válasz?', 5) 'Illeszkedik-e a hang a dialógushoz?' (0: negatív, 100: pozitív). A kitöltők nem tudták, melyik minta melyik modellhez tartozott és a minták sorrendje meg is volt keverve. Egy-egy kitöltő csak egy beszélő hangmintáit hallgatta (az véletlenszerűen dőlt el, hogy adott tesztelő melyik beszélőét).

A dialógusba ágyazott meghallgatásos tesztet másik tesztelői kör végezte: ebben 19 ép hallású, magyar anyanyelvű kísérleti személy vett részt (1 nő és 18 férfi; 20–63 évesek; átlag: 36 év), 6 rendelkezett beszédtechnológiai ismeretekkel. Átlagosan 9 percig tartott a dialógus videók meghallgatása és értékelése.

A dialógus teszt eredményeit az 5 kérdésre és két dialógus típusra a 2. ábra tartalmazza, melyeket kérdésenként foglalunk össze a következőkben (különkülön részábra készült minden kombinációra). Mivel egy adott tesztelő csak egy beszélőtől (NOI1 / NOI2 / FF1) hallgatott hangmintákat a teszt során, ezért a beszélők közötti értékek nem feltétlenül vethetőek össze egymással, hiszen minden tesztelő saját szubjektív skálát alkalmaz (azaz van, aki a legjobb minőséget 100%-nak állítja be, míg más csak 90%-ra értékeli a legjobb rendszert). Ugyanakkor az átlagos eredményeken érdemes a tendenciákat vizsgálni.



2. ábra. A dialógusba ágyazott meghallgatásos teszt eredményei kérdésenként és dialógus típusonként (egyszerű / komplex – bal oldali felirat szerint): átlagos eredmények (balra) és beszélőnkénti eredmények (jobbra).

Kérdés: 'Mennyire érthető a beszéd?'

Az előzetes tapasztalataink szerint mind az alaprendszerrel (HMM-TTS), mind a DNN-TTS rendszerekkel jól érthető beszédet lehet létrehozni, melyet a meghallgatásos teszt is igazolt. Az egyszerű és komplex dialógusok esetén is 55% feletti értéket kaptunk mindegyik rendszerre és beszélőre, bár a női beszélők esetén a tesztelők nagyobb különbséget éreztek a HMM-TTS alaprendszer és DNN-TTS-ek között (azaz az alaprendszert kevésbé tartották jól érthetőnek NOI1 és NOI2 esetén, míg FF1 esetén hasonló a három rendszer érthetősége).

Kérdés: 'Mennyire természetes a beszéd?'

A beszédszintetizátorok értékelésénél általában a természetesség az, amit a kutatási eredmények során vizsgálnak; illetve a korábbi MUSHRA-jellegű teszt is részben ezt vizsgálta. A HMM-TTS alaprendszer itt egyértelmű hátránnyal rendelkezik: mivel ez egyszerűbb technológia, ezért az átlagos értékelés csak 36-40% körüli. A DNN-TTS rendszerek, az általunk is elvárt és nemzetközi tapasztalatok szerint temészetesebb beszédet tudnak létrehozni; melyet a teszt igazolt: a jelen tesztben 60% körüli értéket értek el. Lényeges különbség nem látszik a Tacotron2 és a FastPitch rendszerek között sem az egyszerű, sem a komplex dialógusok esetén – bár a komplex dialógus esetén a FastPitch értékelése némileg magasabb a két női beszélő esetén. Ami egyértelműen észrevehető, hogy a férfi beszélő (FF1) esetén a DNN-TTS rendszerek természetessége kevésbé kiemelkedő a HMM-TTS alaprendszerhez képest: ugyan a minőségbeli különbség látható az ábrákon a DNN-TTS javára, de az eltérés csak 4-8% körüli.

Kérdés: 'Releváns-e a válasz?'

A tesztelők mindhárom rendszer esetén relevánsnak tartották a válaszokat. Érdekes módon a HMM-TTS alaprendszer az egyszerű dialógusok esetén kissé magasabb értékeléseket kapott, mint a DNN-TTS rendszerek (a különbség leginkább FF1 esetén látványos), de az eltérések nem jelentősek. A komplex dialógusok esetén a HMM-TTS alaprendszer alacsonyabb értékelést kapott a releváns-e kérdésre.

Kérdés: 'Elég részletes-e a válasz?'

A válasz részletessége szempontjából az egyszerű dialógusoknál nincs lényeges különbség. A komplex dialógusoknál ugyanakkor megjelenik a DNN-TTS előnye: a tesztelők NOI1 és NOI2 esetén a részletességre is magasabb értéket javasoltak, ami talán azért lehet, mert ezeknek a rendszereknek a természetessége is magasabb (lásd a 2. kérdés eredményeit a természetességgel kapcsolatban).

Kérdés: 'Illeszkedik-e a hang a dialógushoz?'

A tesztelők szerint a FastPitch rendszer illeszkedik leginkább a dialógushoz, melyet a Tacotron2 követ, majd végül a HMM-TTS alaprendszer következik – mind az egyszerű, mind a komplex dialógusok esetén; bár a különbség inkább csak a komplex esetben jelentős a FastPitch javára. A tesztelők a két női beszélő hangját tartották leginkább a dialógushoz illeszkedőnek, akit a FF1 beszélő követ. A dialógus videókon női chatbot fej szerepel, így érthető, hogy a tesztelők a férfi hangot kevésbé tartották ehhez illeszkedőnek. A két női beszélő közötti preferencia eltérés abból is fakadhat, hogy a teszt megvalósítása miatt különböző

tesztelők hallgatták a különböző beszélőktől származó hangmintákat, és az egyes tesztelők eltérő skálát alkalmaztak.

4. Összefoglalás

Jelen tanulmányban olyan DNN-TTS rendszert mutatunk be, amely hangsorozat bemenetet vár és a beszéd hullámformáját két lépésben állítja elő. Részletesen bemutattuk és összehasonlítottuk a Tacotron2+WaveGlow és FastPitch+HiFi-GAN rendszereket és komponenseiket. A magyar nyelvű adatokon végzett kísérletekben három beszélővel (két női és egy férfi) generáltunk szintezitált beszédmintákat, és és vizsgáltuk azok beszédminőségét.

Szubjektív meghallgatásos tesztjeink során a tesztalanyok a DNN-TTS beszédszintetizátorral előállított mondatokat lényegesen természetesebbnek minősítették, mint a HMM-TTS alaprendszert. A szintetizált beszédminták minősége (természetessége) ugyan nem éri el a természetes beszéd szintjét, de közel áll hozzá (Tacotron2: 58%, FastPitch: 73%, természetes: 89%). Összességében a tesztelők a FastPitch rendszert preferálták a Tacotron2-vel szemben természetesség szempontjából, és ezek az eredmények szignifikánsan különböznek.

A dialógus tesztek eredménye szerint a női beszélők preferáltak, és a DNN-TTS rendszerekkel előállított beszéd érthetőbb, természetesebb, mint a HMM-TTS alaprendszer, és tesztelők a válaszokat is relevánsabbnak és részletesebbnek érezték az alaprendszerhez képest.

Valós dialógus rendszerbe integráláskor célszerű megvizsgálni a különböző technológiákhoz szükséges hardverkövetelményeket is: a HMM-TTS CPU-n is gyorsan fut sok csatornán; a Tacotron2+WaveGlow rendszerhez GPU szükséges a valós idejű szintézishez, míg a FastPitch+HiFi-GAN modellek nagyságrenddel gyorsabb futást tesznek lehetővé (a Tacotron2+WaveGlow-hoz képest): szerver környezetben több CPU használatával a valós idejű működés elérhető, GPU-s gyorsítással pedig több csatorna párhuzamos kiszolgálása is megvalósítható.

Köszönetnyilvánítás

A kutatás az „Infokommunikáció, információtechnológiai kutatások-fejlesztések Nemzeti Laboratórium” c. projekten belül „A mesterséges intelligencia alkalmazásának kutatása” kutatási irányához kapcsolódik. Köszönjük a meghallgatásos tesztek résztvevőinek (IdomSoft Zrt. és BME TMIT kollégák) a teszt kitöltését. Köszönjük Olaszy Gábornak a cikk korábbi változatával kapcsolatos megjegyzéseket és segítséget. Csapó Tamás Gábor kutatásait az MTA Bolyai János kutatói ösztöndíja, valamint az Új Nemzeti Kiválóság Program Bolyai+ (ÚNKP-22-5-BME-316) pályázata támogatta. Zainkó Csaba kutatásait az NVIDIA Corporation GPU-val támogatta.

Hivatkozások

Csapó, T.G.: A gépi beszéd-előállítás természetességének növelése rejtett Markov-modell alapú szövegfelolvasó rendszerben [Increasing the naturalness of synthesized speech in hidden Markov-model based text-to-speech synthesis]. Phd thesis, Budapest University of Technology and Economics (2013), http://dokutar.omikk.bme.hu/collections/phd/Villamosmernoki_es_Informatikai_Kar/2014/Csapo_Tamas_Gabor/ertekezes.pdf

Csapó, T.G., Németh, G.: Folytonos paraméterű vokóder rejtett Markov-modell alapú beszédszintézisben - magyar nyelvű kísérletek 12 beszélővel. In: MSZNY 2017. pp. 308–315 (2017)

Kong, J., Kim, J., Bae, J.: HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (szerk.) Advances in Neural Information Processing Systems. vol. 33, pp. 17022–17033 (2020), https://proceedings.neurips.cc/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf

Łańcucki, A.: FastPitch: Parallel text-to-speech with pitch prediction. In: Proc. ICASSP. pp. 6588–6592. Toronto, ON, Canada (2021)

Olaszy, G., Németh, G., Olaszi, P., Kiss, G.: Profivox—A Hungarian text-to-speech system for telecommunications applications. International Journal of Speech Technology 3(3-4), 201–215 (2000), http://link.springer.com/article/10.1023/A:1026558915015

Olaszy, G.: Precíziós, párhuzamos magyar beszédadatbázis fejlesztése és szolgáltatásai [Development and services of a Hungarian precisely labeled and segmented, parallel speech database] (in Hungarian). Beszédkutatás 2013 [Speech Research 2013] pp. 261–270 (2013)

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio. CoRR abs/1609.0 (2016), http://arxiv.org/abs/1609.03499

Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A Flow-based Generative Network for Speech Synthesis. In: Proc. ICASSP. pp. 3617–3621. Brighton, UK (2019)

Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R.A., Agiomvrgiannakis, Y., Wu, Y.: Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In: Proc. ICASSP. pp. 4779–4783. Calgary, Canada (2018)

Su, J., Jin, Z., Finkelstein, A.: HiFi-GAN: High-Fidelity Denoising and Dereverberation Based on Speech Deep Features in Adversarial Networks. In: Proc. Interspeech. pp. 4506–4510. International Speech Communication Association (jun 2020), https://arxiv.org/abs/2006.05694v2

Tan, X., Qin, T., Soong, F., Liu, T.Y.: A Survey on Neural Speech Synthesis (jun 2021), https://arxiv.org/abs/2106.15561v3

Tóth, B.: Rejtett Markov-modell alapú gépi beszédkeltés. Phd thesis, BME TMIT, Hungary (2013)

Tóth, B.P., Németh, G.: Rejtett Markov-modell alapú szövegfelolvasó adaptációja félig spontán magyar beszéddel. In: MSZNY 2009. pp. 246–256. Szeged, Hungary (2009)

Tóth, B.P., Németh, G., Olaszy, G.: Beszédkorpusz tervezése magyar nyelvű, rejtett Markov-modell alapú szövegfelolvasóhoz. Beszédkutatás 2012 [Speech Research 2012] 20, 278–295 (2012)

Zainkó, C., Tóth, B.P., Németh, G.: Magyar nyelvű WaveNet kísérletek. In: MSZNY 2017. Szeged (2017)

Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In: Proc. ICASSP. pp. 7962–7966. Vancouver, Canada (2013), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6639215

Zen, H., Tokuda, K., Black, A.W.: Statistical parametric speech synthesis. Speech Communication 51(11), 1039–1064 (nov 2009), http://linkinghub.elsevier.com/retrieve/pii/S0167639309000648

Effects of emotional speech on forensic voice comparison using deep speaker embeddings

Mohammed Hamzah Abed[1,2], Dávid Sztahó[1]

[1]Department of Telecommunications and Media Informatics,Budapest University of Technology and Economics,Magyar tudósok körútja 2, 1117 Budapest, Hungary

[2]Department of Computer Science, Faculty of Computer Science and Information Technology,Al-Qadisiyah University,Al Diwaneyah, Iraq

m.abed@tmit.bme.hu ,sztaho.david@vik.bme.hu

Abstract. Emotional conditions play a significant role in forensic voice comparison and speaker verification systems. When emotion is present in speech, the verification's performance will deteriorate. In this paper, speaker verification has been investigated and analyzed in the case of emotional speech using metrics evaluating the performance of forensic voice comparison using pre-trained speaker embedding models: x-vector and ECAPA-TDNN for embedded feature extraction. This study investigates whether emotional content affects the forensic voice comparison and verification performance evaluated on a Hungarian speech dataset. The speaker verification performance has been assessed using the likelihood-ratio framework using Cllr and $Cllr_{min}$ and Equal Error Rate. The ECAPA-TDNN achieved higher performance than the x-vector. In the same emotion scenario, the best EERs were 2.6% and 7.7% for ECAPA-TDNN and x-vector. Both models are sensitive to the emotional content of the speech samples.

Keywords: Forensic voice comparison, speaker verification ; x-vector, ECAPA-TDNN, likelihood-ratio framework.

1 Introduction

Speaker recognition is usually classified into two major fields: speaker verification (automatic speaker verification, ASV) and identification (Sztahó et al., 2021). Speaker verification aims to verify if the voice belongs to the claimed identity by comparing the voice with another voice in the dataset and verifying whether the same person produces it. Unlike speaker identification concept, which aims to identify the speaker by selecting one model from a set of enrolled speaker models (Sztahó & Fejes, 2022). One of the applications of speaker verification dealing with forensic voice comparison is based on comparing an unknown criminal's voice with a well-known suspect's voice like when a DNA sequence is matched with another known DNA profile (Sztahó & Fejes, 2022). Furthermore, the decision of speaker verification in such a forensics case is very critical. It must present a high level of confidence because the error-prone behavior is critical and the error is unacceptable. Due to the behavioral and biological differences between people and the way they speak, each person's voice contains unique information. This allows people to recognize each person from his\her voice (Arya et al., 2021). The main

aim of this study is to analyze and investigate how emotional content affects forensic voice comparison and verification performance. The experiments were done on a Hungarian speech dataset.

To compare two samples of voices and judge whether they belong to the same identity, we need to consider how the voices are similarly based on the features extracted from each sample. In the last few years, deep learning has been a significant tool in speaker verification and emotional speaker recognition. Studies show that these deep-learning embeddings outperform previous i-vector based features. However, a lot of training data is needed to get a highly efficient model (Sztahó & Fejes, 2022). In this paper, we have used two speaker embedding methods commonly used in speaker verification: x-vector and ECAPA-TDNN. Pre-trained models were applied that are available in the Huggingface repository. The speaker identities were evaluated in a forensic voice comparison framework by calculating the likelihood ratio based on cosine distance between sample pairs and logistic regression models.

Recently, in forensic voice comparison, there have been studies dealing with speaker verification where emotional content is present (Scherer et al., 2000). An acoustic analysis regarding the effect of the emotional content of possible automatic speaker verification systems shows that an evaluation of training ASV material on emotional speech requires in-depth analyses of the individual differences in vocal reactivity and further exploration of the link between acoustic changes under stress or emotion and verification results. In (Rusko et al., 2018), researchers investigate the weakness of voice as a biometric model and try to improve the performance of the verification system. In addition, they used the emotional speech dataset to increase the diversity of all cases belonging to the speaker. The suggested model has been evaluated based on CRISIS dataset with six levels of emotion per speaker and used i-vector as embedded features with PLDA. In (Shahin et al., 2021), the authors proposed a hybrid deep neural network for speaker verification in an emotional case study. Four DNN models (DNN-HMM, DNN-GMM, GMM-DNN and HMM-DNN) have been used. These models were evaluated using three different speech datasets: private Arabic and two English public datasets. Their result shows the HMM-DNN outperformed all other models in an emotional and stressful environment and shows high performance in terms of equal error rate (EER), which was 7.19%. In another work (Prasetio et al., 2020), the authors proposed a speaker verification model under stress conditions by applying i-vector and investigated the effect of the emotional speaking style in speech. Emotional Variability Analysis (EVA) has been proposed which is based on i-vector technique, but it considers the emotional effect as the channel variability component. Based on the experimental result, the proposed model was outperformed the standard i-vector. Biswajit and his colleagues (Dev Sarma & Kumar Das, 2020) tried to map i-vector embeddings to an emotionally invariant space. They obtained a slight performance increase in speaker identification using the IEMOCAP dataset compared to models trained only on neutral samples. Parthasarathy and his colleagues (Parthasarathy et al., 2017) also used i-vector features to test speaker verification with expressive speech. The results show that speaker verification errors increase when the values of the emotional attributes increase, but the overall results are reliable. Pappagari and his colleagues (Pappagari et al., 2020) applied a more novel, deep learning-based embedding method, the x-vector for speaker

verification with an emotional speech. They observed that speaker verification performance is prone to changes in testing speaker emotions. They found that trials with angry utterances performed worst in all three datasets.

The present study aims to extend the speaker verification topic by applying two speaker embedding models based on deep learning in a forensic voice comparison scenario. We investigate how the ECAPA-TDNN embeddings perform with emotional speech and how they relate to the x-vector embeddings. The evaluation of the workflow model was investigated in the likelihood-ratio framework. And the performance of the models was evaluated by equal error rate (EER) of speaker verification, log-likelihood-ratio cost (Cllr) and $Cllr_{min}$ (Sztahó & Fejes, 2022). Samples were compared in multiple emotion combination cases.

The rest of this paper is organized as the following: section 2 illustrates the related work of emotional speech verification and deep learning embedding in speaker recognition. The proposed model and the methodology are described in section 3. Section 4 included the result and analysis. Finally, section 5 shows the conclusion and future scope.

2 Materials and Methods

The workflow of speaker verification based on speaker embedding vectors (x-vector and ECAPA) has been evaluated on a Hungarian dataset containing emotional and neutral speech. Figure 1 illustrates the layout of the applied process. Two methods have extracted speaker embeddings from the audio samples: x-vector and ECAPA-TDNN. Feature extraction was implemented in the SpeechBrain toolkit (Ravanelli et al., 2021), and the pre-trained models were downloaded from Huggingface[1,2]. After extracting features from speech samples, cosine similarity was calculated between feature vector pairs. The likelihood-ratio scores were obtained by feeding the cosine similarity scores to a logistic regression model trained on the ForVoice120+ dataset (Sztahó & Fejes, 2022).

[1] x-vector: https://huggingface.co/speechbrain/spkrec-xvect-voxceleb

[2] ECAPA-TDNN: https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

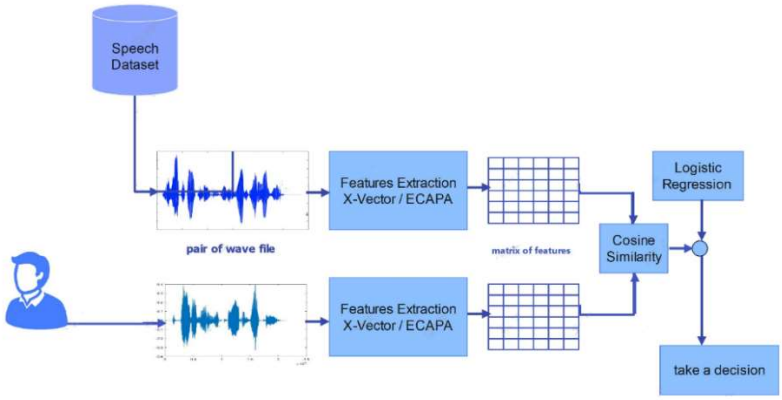


Fig. 1. Speaker verification process applied in the study

2.1 Dataset

The Hungarian Emotional Speech Database has been used in this work to evaluate the proposed models. This dataset consists of 38 volunteers (20 females and 18 males). The voices have been recorded in a quiet office room, and the recording equipment was a Sound Blaster NX 2 USB external sound card with a Monacor ECM-100 microphone. The recordings were PCM encoded with 16-bit quantization and 16 kHz sample rates. Three different sentences were recorded with each speaker in Hungarian with eight other emotions: sadness, anger, fear, excitement, disgust, surprise, joy and neutral. The linguistic content of the sentences are:

(1) "Kovács Katival szeretnék beszélni" (English: "I would like to speak with Kovács Kati.")
(2) "A falatozóban sört, bort, üdítőitalokat és finom malacsültet lehet kapni." (English: "In the snack bar you can get beer, wine, beverages and delicious pork steak.")
(3) "A jövő hétvégén megyek el." (English: "I will leave next weekend.")

2.2 Speaker embedding models

Two pre-trained models have been used in this work for embedding feature extraction: x-vector and ECAPA-TDNN.

2.2.1 The x-vector

The deep learning-based feature extraction method, x-vector was developed primarily for speaker verification (Egas-López et al., 2022; Snyder et al., 2018). It is based on a multiple-layered DNN architecture (with fully connected layers) with different temporal contexts at each layer (which they call ‘frames’). Due to the wider temporal context, the architecture is called time-delay NN (TDNN). The TDNN embedding architecture can be seen in Figure 2 and Table 1.

The first five layers operate on speech frames, with a slight temporal context centred at the current frame t. For example, the frame indexed as 3 sees a total of 15 frames, due to the temporal context of the earlier layers. After training with speaker ids as target vectors, the output of layer segment6 ('x-vector') is used as input to a classifier.

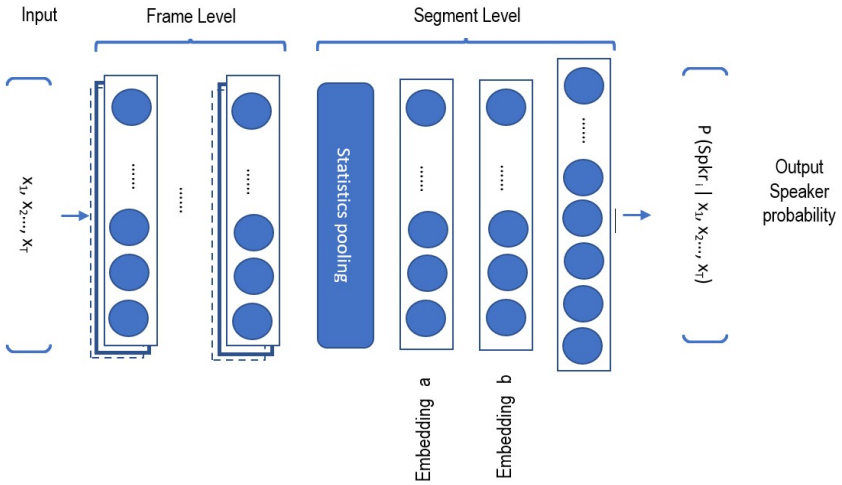


Fig. 2. The x-vector DNN embedding architecture in (Snyder et al., 2018). The two parts: frame level (with the 5 frame layers) and segment level (with segment6, segment7 and softmax).

Table 1. The x-vector embedding DNN architecture (Snyder et al., 2018)

Layer	Layer context	Total context	Input x output
Frame 1	{t-2,t+2}	5	120 x 512
Frame 2	{t-2,t,t+2}	9	1536 x 512
Frame 3	{t-3, t,t+3}	15	1536 x 512
Frame 4	{t}	15	512 x 512
Frame 5	{t}	15	512 x 1500
Stats pooling	[0,T)	T	1500 T x 3000
Segment 6	{0}	T	3000 x 512
Segment 7	{0}	T	512 x 512
softmax	{0}	T	512 x N

2.2.2 ECAPA-TDNN

The ECAPA-TDNN model is the extension of the x-vector model architecture in three ways (Desplanques et al., 2020): channel- and context-dependent statistics pooling, 1-Dimensional Squeeze-Excitation Res2Blocks (1D SE-Res2Block) and multi-layer feature aggregation and summation. The channel- and context-dependent statistics pooling enables the network to focus more on speaker characteristics that do not activate on identical or similar time instances, e.g. speaker-specific properties of vowels versus

speaker-specific properties of consonants. Using the SE-Res2Block (taken from the field of computer vision), the limited frame context of the x-vector (15) is extended to the global properties of the recording. The multi-layer feature aggregation means that not only the activation of the selected deep layer is used as a feature map (as in x-vector), but the shallower layers (here:SE-Res2Blocks) are also concatenated, because they also hold information about the speaker identity.The architecture is shown in Figure 3.

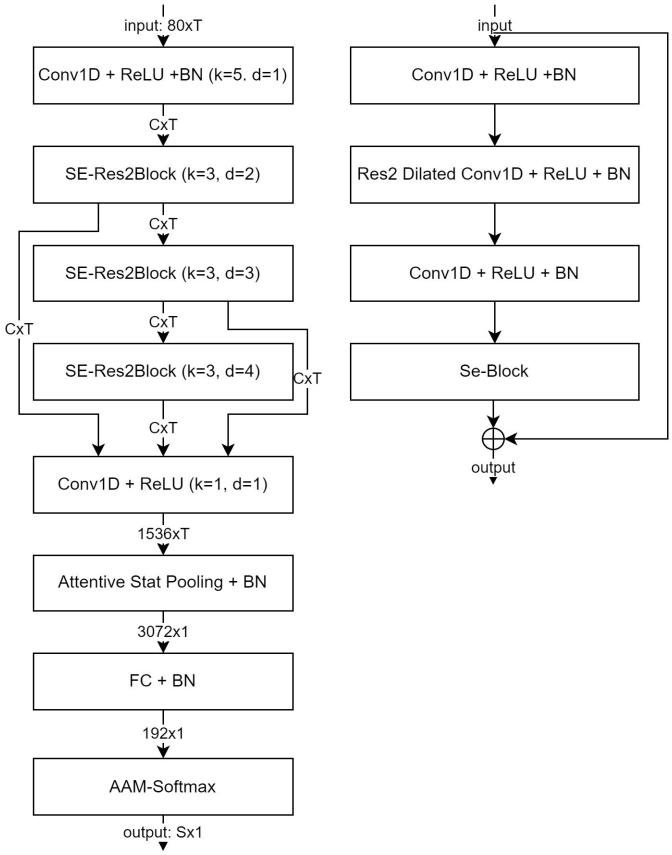


Fig. 3. The ECAPA-TDNN layer architecture and its SE-Res2Block (Desplanques et al., 2020)

2.3 Cosine similarity

Cosine similarity has been used to measure the similarity of pairs of embedded feature vectors. It is a measure by the cosine of the angle between two vectors calculated using Eq. 1 (Han et al., 2012). Figure 4 illustrates the cosine similarity mechanism between two vectors.

$$\text{Cosine similarty}(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad \textbf{(1)}$$

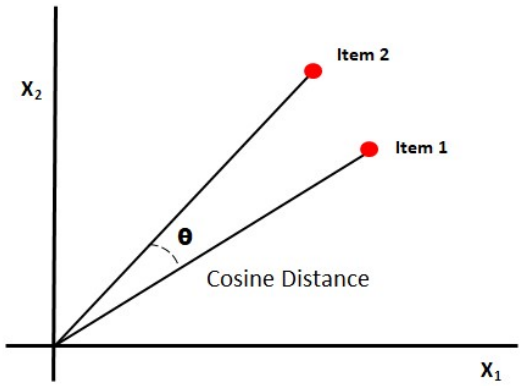


Fig. 4. Cosine similarity between pair of vectors (Han et al., 2012)

2.4 LR score calculation

Logistic regression has been used (implemented based on the python sklearn package) to calculate LR scores. All possible sample pairings (trials) were constructed with the target/non-target indicator (same speaker or different speaker). Each trial contains a speaker pair, the suspect and the offender and the indication if the speakers are the same or not. The probability of the same speaker decision was computed based on the logistic regression model using Eq. 2, where E is the evidence, H_{so} is the hypothesis of same-origin speakers and H_{do} is the hypothesis of different-origin speakers. The probability of different speaker origins can be calculated using Eq. 3. The applied logistic regression models (two separate models for x-vector and ECAPA-TDNN) were trained on the ForVoice120+ dataset using 2-10 second long speech samples (Sztahó & Fejes, 2022). Figure 5 illustrates an example of a trained logistic regression model. The distribution of the same and different origin vector pairs is shown in blue and yellow, respectively.

$$LR = \frac{P(E|H_{so})}{P(E|H_{do})} \quad \textbf{(2)}$$

$$P(E|H_{do}) = 1 - P(E|H_{so}) \quad \textbf{(3)}$$

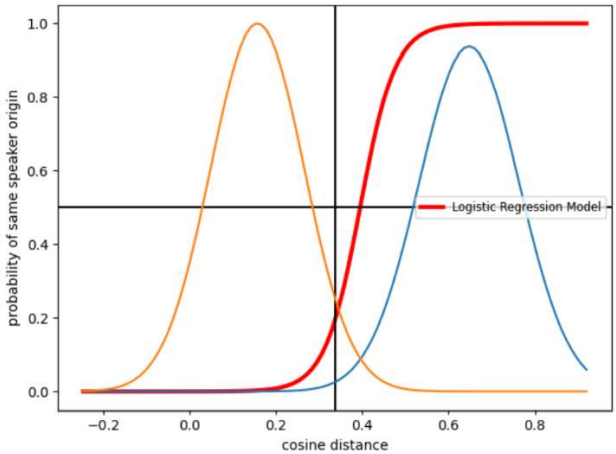


Fig. 5. Trained Logistic Regression model example. Blue and yellow lines show the distributions of cosine distances of embedding vector pairs of the same and different speaker origins, respectively. (Sztahó & Fejes, 2022)

2.5 Evaluation metric

The output of the two pre-trained models has been evaluated by using an Equal Error Rate (EER) and log-likelihood ratio cost (Eq. 4) between each sample pair. In speaker verification, the EER's level is where the false acceptance and rejection rates are equal.

$$Cllr = \frac{1}{2}\left(\frac{1}{N_{so}}\sum_{i=1}^{N_{so}}\left(1+\frac{1}{LR_{so_i}}\right)+\frac{1}{N_{do}}\sum_{j=1}^{N_{do}}\left(1+LR_{do_j}\right)\right) \quad \textbf{(4)}$$

where N_{so} and N_{do} are the number of same-origin and different-origin comparisons, LR_{so} and LR_{do} are the likelihood ratios derived from same-origin and different-origin comparisons.

Cllr is a function used to measure the balance of LR scores of same and different origin comparisons (Brümmer & du Preez, 2006). Ideal same-origin and different-origin comparisons have logLR>0 and logLR<0, respectively. Besides Cllr, the minimum Cllr value is also reported, which is the generalization of the original cost function and produces application-independent Cllr values.

Tippet plots, commonly used as a visualization in speaker verification, are used to display the proportion of correctly identified same and different speaker origin pairs.

3 Results

In this section, we evaluate the effectiveness of the forensic voice comparison and speaker verification system if emotional speech is present and how it affects performance. We investigated the models in multiple scenarios, considering how the emotions

are paired for the known and unknown speakers. Table 2 shows the results if the same or different emotions are used in the trials compared to the case when all samples are used together. The first row of the table can be considered as a baseline because all samples were used without any filtering. Higher performance was found using the same emotion (0.026 and 0.046 EER for ECAPA-TDNN with the same and different emotions, respectively). Both models are affected by emotional content. ECAPA-TDNN outperforms the x-vector, as was expected. Tippet plots for the two scenarios for ECAPA-TDNN are shown in Figure 6.

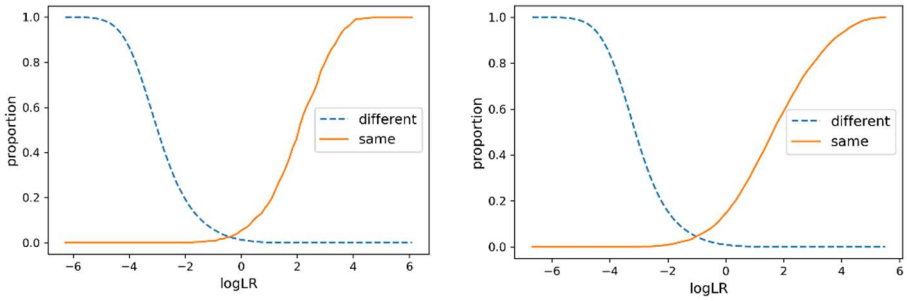


Fig. 6. Tippet plots for trials with same (left) and different (right) emotions for ECAPA-TDNN embeddings.

Table 2. Performance metrics of x-vector and ECAPA-TDNN when same and different emotions are used in the trials

Scenario	x-vector			ECAPA-TDNN		
	Cllr	$Cllr_{min}$	EER%	Cllr	$Cllr_{min}$	EER%
All samples	0.634	0.373	10.9	0.274	0.161	4.5
Same emotions	**0.387**	**0.270**	**7.7**	**0.122**	**0.092**	**2.6**
Different emotions	0.658	0.377	11.1	0.288	0.163	4.6

Table 3. Performance metrics of x-vector and ECAPA-TDNN when neutral samples were used for the suspect (first sample of the trials).

Emotion in offender samples	x-vector			ECAPA-TDNN		
	Cllr	$Cllr_{min}$	EER%	Cllr	$Cllr_{min}$	EER%
Neutral	**0.402**	**0.263**	**7.8**	**0.117**	**0.080**	**2.7**
All emotions	0.707	0.396	11.6	0.291	0.164	4.7
Anger	1.233	0.447	15.1	0.500	0.201	5.2
Sadness	0.536	0.324	10.8	0.149	0.083	3.1
Joy	0.632	0.355	11.5	0.314	0.159	4.6
Fear	0.703	0.362	10.8	0.369	0.174	5.4
Excitement	0.851	0.399	12.8	0.330	0.134	4.8
Disgust	0.713	0.333	12.9	0.311	0.120	4.3
Surprise	0.650	0.369	11.8	0.286	0.148	4.7

Table 3 shows the results when neutral samples were used as the first sample in the trials. These cases describe how a given emotion (and all emotions at once) affect the ASV performance if only neutral samples are recorded from the suspect. The results show that the ECAPA-TDNN outperformed the x-vector in this case also. The first row can be considered as a baseline because all samples were used without emotional content. Higher performance was achieved in the case of neutral vs. neutral as was expected. EERs were 2.7% and 7.8 % for ECAPA-TDNN and x-vector, respectively. In the case when emotional sentences were used as the second part of the trials, the EERs were slightly worse in each case.

Table 4. Performance metrics of x-vector and ECAPA-TDNN when all emotions samples were used for the suspect (first sample of the trials).

Emotion in offender samples	x-vector			ECAPA-TDNN		
	Cllr	$Cllr_{min}$	EER%	Cllr	$Cllr_{min}$	EER%
Anger	0.735	0.383	11.7	9.887	0.780	4.5
Sadness	**0.527**	0.345	**9.9**	**9.831**	**0.698**	**3.3**
Joy	0.645	0.352	10.7	9.939	0.834	4.9
Fear	0.633	0.379	11.3	9.883	0.739	5.4
Excitement	0.669	0.361	10.6	9.847	0.805	5.1
Disgust	0.596	0.346	10.4	9.943	0.796	3.6
Surprise	0.563	**0.340**	10.4	9.877	0.764	4.4

Table 4 shows the results when all emotional samples were used as the first sample in the trials. These cases describe how a given emotion and all emotions affect the ASV performance if all emotions samples are recorded from the suspect. The results show that the ECAPA-TDNN outperformed the x-vector in this case also. Higher performance was achieved in the case of sadness vs. all emotions for both models. EERs were 3.3% and 9.9 % for ECAPA-TDNN and x-vector, respectively.

4 Discussion and Conclusion

Based on the metrics obtained, we can conclude that the emotional content affects the pre-trained speaker embedding models' performance. Both ECAPA-TDNN and x-vector performed worse when the trials were composed of different emotional samples. A 2% decrease was found if different emotions were present compared to the case when the same emotions were applied. Considering the same emotional samples in the trials, the separate emotion did not have a large effect compared to the case when only neutral samples were used (2.6% and 2.7% EER for all emotions and neutral, respectively). Inspecting the separate emotions when the first sample in the trial was always neutral, the best results were obtained in the case of comparing them to neutral sentences (7.8% and 2.7% EER for x-vector and ECAPA-TDNN, respectively). As was expected, comparing neutral samples to emotional samples, worse performances were achieved. Table 3 shows that only sadness is close to the neutral case. There is no emotion that can be

said to have the highest effect on speaker verification. All emotions deteriorate the metrics. This implies that if the emotional content on the recording is not neutral, but is the same in the trial, the performance deterioration is not present.

In this work, the effects of emotions were investigated in a forensic voice comparison setting using deep speaker embeddings and a Hungarian dataset. The main goal of this study was to evaluate whether emotional characteristics affect ASV performance. Two pre-trained deep learning models have been used for feature extraction (x-vector and ECAPA-TDNN). The similarity of the embedding vectors was measured by cosine similarity. The performances were evaluated in the likelihood-ratio framework by calculating the LR and logLR based on the cosine similarities. It can be stated that for a public service expert, emotional content can be a significant factor during speaker verification. In future work, more emotion-robust models can be built, trained or fine-tuned to make ASV more reliable in such cases.

Acknowledgement

The work was funded by project no. FK128615, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK_18 funding scheme.

References

Arya, R., Pandey, D., Kalia, A., Zachariah, B. J., Sandhu, I., & Abrol, D. (2021). Speech based Emotion Recognition using Machine Learning. *2021 IEEE Mysore Sub Section International Conference, MysuruCon 2021*, 613–617. https://doi.org/10.1109/MysuruCon52639.2021.9641642

Brümmer, N., & du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech and Language*, *20*(2-3 SPEC. ISS.), 230–275. https://doi.org/10.1016/j.csl.2005.08.001

Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). *ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification*. https://doi.org/10.21437/Interspeech.2020-2650

Dev Sarma, B., & Kumar Das, R. (2020). Emotion Invariant Speaker Embeddings for Speaker Identification with Emotional Speech; In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

Egas-López, J. V., Kiss, G., Sztahó, D., & Gosztolya, G. (2022). AUTOMATIC ASSESSMENT OF THE DEGREE OF CLINICAL DEPRESSION FROM SPEECH USING X-VECTORS. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2022-May*, 8502–8506. https://doi.org/10.1109/ICASSP43922.2022.9746068

Han, J., Kamber, M., & Pei, J. (2012). 2 - Getting to Know Your Data., *Data Mining (Third Edition)* (Third Edition, pp. 39–82). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00002-2

Pappagari, R., Wang, T., Villalba, J., Chen, N., & Dehak, N. (2020). X-Vectors Meet Emotions: A Study On Dependencies Between Emotion and Speaker Recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7169–7173. https://doi.org/10.1109/ICASSP40776.2020.9054317

Parthasarathy, S., Zhang, C., Hansen, J. H. L., & Busso, C. (2017). A study of speaker verification performance with expressive speech. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5540–5544. https://doi.org/10.1109/ICASSP.2017.7953216

Prasetio, B. H., Tamura, H., & Tanno, K. (2020). Emotional variability analysis based I-vector for speaker verification in under-stress conditions. *Electronics (Switzerland)*, *9*(9), 1–15. https://doi.org/10.3390/electronics9091420

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., … Bengio, Y. (2021). *SpeechBrain: A General-Purpose Speech Toolkit*. http://arxiv.org/abs/2106.04624

Rusko, M., Trnka, M., Darjaa, S., Stelkens-Kobsch, T., & Finke, M. (2018). *WEAKNESSES OF VOICE BIOMETRICS-SENSITIVITY OF SPEAKER VERIFICATION TO EMOTIONAL AROUSAL*.

Scherer, K. R., Johnstone, T., Klasmeyer, G., & Bänziger, T. (2000). *CAN AUTOMATIC SPEAKER VERIFICATION BE IMPROVED BY TRAINING THE ALGORITHMS ON EMOTIONAL SPEECH?* http://www.iscaaspeech.org/archive

Shahin, I., Nassif, A. B., Nemmour, N., Elnagar, A., Alhudhaif, A., & Polat, K. (2021). Novel hybrid DNN approaches for speaker verification in emotional and stressful talking environments. *Neural Computing and Applications*, *33*(23), 16033–16055. https://doi.org/10.1007/s00521-021-06226-w

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2018-April*, 5329–5333. https://doi.org/10.1109/ICASSP.2018.8461375

Sztahó, D., & Fejes, A. (2022). *Effects of language mismatch in automatic forensic voice comparison using deep learning embeddings*.

Sztahó, D., Szaszák, G., & Beke, A. (2021). Deep learning methods in speaker recognition: A review. *Periodica Polytechnica Electrical Engineering and Computer Science*, *65*(4), 310–328. https://doi.org/10.3311/PPee.17

Cross-lingual dysphonic speech detection using pre-trained speaker embeddings

Aziz Dosti Ali Hama Salih, Dávid Sztahó

Budapest University of Technology and Economics,
Magyar tudósok körútja 2., 1117 Budapest, Hungary
azizd@edu.bme.hu, sztaho.david@vik.bme.hu

Abstract: In this study, cross-lingual binary classification and severity estimation of dysphonic speech have been carried out. Hand-crafted acoustic feature extraction is replaced by the speaker embedding techniques used in the speaker verification. Two state of art deep learning methods for speaker verification have been used: the X-vector and ECAPA-TDNN. Embeddings are extracted from speech samples in Hungarian and Dutch languages and used to train Support Vector Machine (SVM) and Support Vector Regressor (SVR) for binary classification and severity estimation, in a cross-language manner. Our results were competitive with manual feature engineering, when the models were trained on Hungarian samples and evaluated on Dutch samples in the binary classification of dysphonic speech and outperformed in estimating the severity level of dysphonic speech. Moreover, our model achieved 0.769 and 0.771 in Spearman and Pearson correlations. Also, our results in both classification and regression were superior compared to manual feature extraction technique when models were trained on Dutch samples and evaluated on Hungarian samples with only a limited number of samples are available for training. An accuracy of 86.8% was reached with features extracted from embedding methods, while the maximum accuracy using hand-crafted acoustic features was 66.8%. Overall results show that Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network (ECAPA-TDNN) performs better than the former X-vector in both tasks.

Keywords: dysphonic, cross-lingual, speaker verification, embedding, X-vector, ECAPA-TDNN

1 Introduction

Biomarkers are implications that indicate a medical condition observed in the patient. They contain a wide range of medical marks that can be accurately measured. They refer to any sign or indication which may lead to predicting a disease, starting from a basic blood test, and going to more complicated tests performed in a specialized laboratory (Strimbu & Tavel, 2010).

Even though the human voice is the main source for communication and social interactions among individuals, it also carries further information about the identity and health status of a person (Grossmann et al., 2013). Speech is considered as a biomarker due to its ability to make health specialists diagnose diseases based on a patient's

speech. Many diseases affect the speech-production organs in the human body, making them face difficulties in producing normal speech (Lin et al., 2020). Dysphonia, also called hoarseness, is a condition in which the produced voice has alterations in the quality, pitch and loudness. It affects specific groups of people, mostly elderly ones, teachers and individuals with extensive use of vocal voices. Nearly one out of three people at some time in their life will be diagnosed as having dysphonia (Schwartz et al., 2009; Stemple et al., 2018; Van Houtte et al., 2011). A person diagnosed with dysphonia requires regular consultations with clinicians to assess the severity of their condition. The implementation of these measures is expected to result in significant costs for both work environments and healthcare systems due to the disruption caused by the illness and the need for medical care for affected individuals. (Cohen et al., 2012; Schwartz et al., 2009).

Therefore, using artificial intelligence to analyse human speech to detect disorders has been extensively researched by academia and clinicians. Voice samples from both healthy and disordered persons were converted to feature vector representations using a digital signal processing technique (Hegde et al., 2019).

Various research has been conducted using extracted features from speech samples to train various machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Random Forest, Decision Tree and many others to distinguish between healthy and dysphonic speech (Hadjaidji et al., 2021; Syed et al., 2021; Wu et al., 2017).

The vast development of deep learning techniques during the last few decades led to advancement in many research areas including computer vision, natural language processing and speech and speaker recognition. Speaker embedding methods originally used for speaker verification and identification can capture speaker-related characteristics which are used to identify speakers based on their speech signals (Bimbot et al., 2004; Togneri & Pullella, 2011).

Some researchers adopted speaker embedding techniques for classifying disordered speech from a normal one. The objective of these methods is that they don't require hand-crafted feature extraction and engineering (Egas-López et al., 2022; Scheuerer et al., 2021).

In this research, we adapted two state of art speaker embedding techniques, the X-vector and Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network (ECAPA-TDNN) to extract features from speech samples in different languages and use them for classifying dysphonia. SVM and SVR algorithms are trained on features extracted from samples from a Hungarian dataset and tested on features extracted from a Dutch dataset and vice versa. The result shows that the embedding techniques are suitable for cross-language detection of dysphonic speech.

In section 2, we will present related works using feature engineering and speaker embedding techniques. A detailed description of the used dataset and methods will be examined in section three. Section 4 will illustrate the obtained results from classification and regression. Lastly, the discussion and conclusion will be included in 5 and 6 sections, respectively.

2 Previous related work

Machine learning models have shown great advancements in many medical areas such as cancer and tumour detection. Analysing speech to diagnose disease has been an active research area; there have been enormous machine learning approaches to classify dysphonic speech from a healthy control.
Al-Dhief et al. extract features from Mel-Frequency Cepstral Coefficients (MFCC) in speech samples that were found to be effective in pathological voice analysis. Features extracted using vowel /a/ from the German dataset Saarbrucken Voice Database (SVD). They achieved an accuracy of 85% using Online Sequential Extreme Learning Machine (OSELM) algorithm with 200 hidden units, an improved version of the ELM algorithm (AL-Dhief et al., 2020). Their work (Dankovičová et al., 2018) presents results of identifying dysphonic samples from healthy control speech using Random Forest, SVM and K-nearest neighbours classifiers. A total of 1560 features were extracted from the vowels /a/, /i/ and /u/, 520 features for each vowel, and dimensionality reduction of features was performed using Principal Component Analysis (PCA). The highest achieved accuracy was by Support Vector Machine (SVM) classifier on the male samples, and the obtained accuracy was 91%. In another study (Awan & Roy, 2006), predicting the severity of dysphonic speech was researched in which, time and spectral-based features derived from sustained vowels were considered with stepwise multiple regression. The reported results were 0.880 and 0.775 for mean R and mean R2, respectively. The study shows that the four-variable model included time and spectral-based acoustic measures were able to strongly predict perceived severity.

Several studies have been conducted by adopting deep learning methods for speaker embedding for identifying disordered speech. These approaches do not require handcrafted feature extraction, as they are able to include speaker characteristic features in the embedding. In (Scheuerer et al., 2021) implementation of two models of X-vectors based on Mel-frequency cepstral coefficient (MFCC) and gammatone frequency cepstral coefficients (GFCC) carried out for both regression and binary classification on SVD dataset. According to their findings, the GFCC-based multi-layer perceptron was the best, reaching 0.8810 and 0.8810 ROG AUC scores for both regression and classification, respectively. Pre-trained i-vector and X-vector are used for classifying Parkinson's disease and obstructive sleep apnnea, and they perform better than handcrafted feature extractions. Moreover, X-vector performs better when there is a domain mismatch between the train and test speech samples (Botelho et al., 2020).

Relating to cross-lingual voice disorder detection, some research has been carried out. Cross-lingual detection of voice disorders was implemented using speech samples from Spanish, Czech and Dutch. Different kinds of training and testing cases were performed using these languages. According to their experiments, the highest accuracies achieved in the test set were nearly 70%, and 60% on Czech and German samples, respectively, when the algorithm was trained on Spanish samples. They also examined the improvement of accuracies by moving speech samples from the target database to the training set. They reported nearly 30% improvement in German samples by adding only 20% of German samples to the training set (Orozco-Arroyave et al., 2016).

In (Shinohara et al., 2017), monolingual evaluation of the pitch-related features has been performed in German, Spanish and Czech languages using normal speech utterances to identify voice disorders.

Results achieved in this study are compared to previous work that has been done by (Sztahó et al., 2022), performing a cross-lingual evaluation of classifying and estimating the severity level of dysphonic speech in Hungarian and Dutch languages using hand-crafted acoustic features. The results in that study show training the machine learning algorithms on features extracted from Hungarian utterances and testing on Dutch samples is possible. The accuracy of 88% in test samples was achieved with acoustic features extracted from entire utterances with phoneme level features of \E\. Results of severity estimation were 0.72 and 0.79 for Pearson correlation and RMSE, respectively.

3 Methods

3.1 Description of Databases

Hungarian and Dutch samples from two dysphonic speech datasets were used for the experiment. Speech samples were collected from patients in each language, also healthy control speech was included in the dataset. All speakers read a short passage "The North Wind and The Sun" in both languages. Patients in the Hungarian samples were all native speakers, the recordings were done at the Head and Neck surgery department at National Institute of Oncology, and all patients agreed to record their voices for the experiments. A total of 179 recordings were used for the Hungarian patients diagnosed with dysphonia (81 females and 98 males). Alongside these numbers, also 179 healthy control speech samples were included in the Hungarian dataset. The severity level of the dysphonic patients is determined by RBH scale which stands for Roughness, Breathiness and Hoarseness (Schönweiler et al., 2000) with a number between 0 to 3, with 0 specified for the healthy control samples. For Hungarian samples, H is selected as the severity level ranges from 0 to 3, 0 indicates no hoarseness (healthy speaker).

For the Dutch dataset, 30 samples from dysphonic patients have been recorded reading the same "The North Wind and The Sun" passage in Dutch. The recordings were organized at the university hospital of KU Leuven, Belgium. Severities were measured by GRBAS (Grade – overall judgement of hoarseness, Roughness, Breathiness, Asthenia, and Strain), with values ranging from 0 which indicates no hoarseness to 3 means severe dysphonia (Wood et al., 2014). 30 samples from normal-speaking persons were included in the database with the same age distribution as dysphonic samples. Distribution of the severity is shown in Table 1.

Table 1. Severity distribution of samples in datasets

Datasets \ Severity level	0	1	2	3
Hungarian	179	58	63	58
Dutch	30	10	17	3

3.2 Embedding techniques

In this study, two state of the art speaker embedding models have been adopted for the task of binary classification of dysphonic speech and estimating severity level. The X-vector model, which is based on deep neural network architecture, primarily used for speaker verification, has been used for extracting the embedding of the speech samples (Snyder et al., 2018). The architecture is based on the work in (Snyder et al., 2017) with adding data augmentation techniques. The first five hidden layers operate at the frame level using a time-delay neural network. For each time frame ***t*** small temporal context before and after is added in the first three layers. The remaining two layers also work at the segment level but without temporal context, in total frame-level part of the model has a temporal context of 16 frames. The mean and standard deviation of the output of the last layer in the frame level is calculated by the statistic pooling layer. The output of the statistics pooling layer will be used as an input for two hidden layers of size 512 and 300 dimensions. The embedding of 512 dimensions can be extracted for both dysphonic and normal speech from the layer after statistics pooling.

ECAPA-TDNN is based on the X-vector model, extending it mainly in three parts: channel and context dependent statistics pooling, which extends temporal attention to channel attention (Desplanques et al., 2020). This enables the network to focus more on speaker properties and not activate on identical or similar time instances. 1-Dimension Squeeze and Excitation proposed for scaling the frame level features that were limited to 15 frames in the original X-vector to give global properties of the recording. Squeeze and Excitation has been used for computer vision task for modelling global channel interdependencies, also residual block has been used for concatenating 1-Dimentaionl SE to the X-vector model for the sake of keeping the number of parameters relatively near to the original X-vector.

We have downloaded the two pre-trained models using Hugging Face repository[1][2] and SpeechBrain pretrained class. SpeechBrain is an open-source toolkit based on PyTorch. It supports many speech processing tasks such as speech recognition, speaker verification, source separation and many others. The two pretrained models can be accessed and downloaded in the mentioned repository (Ravanelli et al, 2021).

3.3 Classification and regression

Binary classification of dysphonic vs healthy samples has been carried out using SVM (Support Vector Machine) (Cortes et al., 1995) with both linear and rbf kernels. It shows good performance compared to other algorithms such as Decision Tree and K-nearest neighbour with good generalization ability (Dankovičová et al., 2018). Due to the size of datasets and nature of Deep Neural Network (DNN) which requires considerably larger datasets to be able to make a good generalization, using DNN might not be a good for the problem. Classification is done in a cross-lingual approach; first, models were trained on Hungarian samples and tested on Dutch samples, the other direction

[1] https://huggingface.co/speechbrain/spkrec-xvect-voxceleb

[2] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

was also performed. For both scenarios, hyperparameter tuning was performed for parameters cost and gamma using grid search method with 10-fold cross-validation on the training language. Also, to provide a more accurate assessment of performance, the results of this study were obtained through the implementation of 10-fold cross-validation. This approach allows for a more realistic evaluation of the model's ability to generalize, particularly when the number of training samples is limited.

Estimating the severity level of dysphonic patients has been carried out using Support Vector Regressor (SVR) in both cross-language directions, linear and rbf kernel have been used in the experiments. Also, using the same techniques as in classification phase hyperparameter tuning of both cost and gamma has been performed with 10-fold cross-validation for the training language.

For both classification and severity estimation output features from X-vector and ECAPA were normalized using L2 norm.

3.4 Evaluation metrics

For evaluating binary classification model, we use accuracy, sensitivity, specificity, and F1-score in addition to area under the curve (AUC). Performance of severity estimation of dysphonic patients has been conducted using Root Mean Square Error (RMSE), Pearson correlation and Spearman correlation indicating the linear relationship between actual and predicted severity scores.

4 Results

Binary classification and severity level estimation have been carried out using X-vector and ECAPA-TDNN embeddings extracted as features from speech for both Hungarian and Dutch datasets. The experiments were carried out in cross-lingual nature, where algorithms were trained with speech features from one language and tested using features from the other language.

4.1 Binary classification

Results from binary classification of dysphonic and normal speech are shown in Tables 2 and 3 using embeddings extracted from both models. In the direction where the algorithm trained on Hungarian samples and tested on Dutch utterances. Overall, ECAPA-TDNN model consistently outperformed the X-vector model, achieving an approximately 10% improvement in both accuracy and AUC, particularly when using the rbf kernel. In X-vector scenario linear kernel performs slightly better than rbf kernel in accuracy and AUC. Comparing to the results previously published by (Sztahó et al., 2022); in Hungarian to Dutch direction, it can be concluded that embedding techniques especially ECAPA-DTNN is competitive to the manual feature extraction techniques.

Table 2: Binary classifications results from embedding (Train= Hungarian, Test=Dutch).

	Model	Accuracy %	Specificity %	Sensitivity %	F1-score %	AUC %
X-vector Embedding	SVM-Linear	78.3	74.2	84.0	80.0	79.1
	SVM-rbf	75.0	74.1	75.8	75.4	75.0
ECAPA Embedding	SVM-Linear	83.3	81.2	85.7	83.8	83.4
	SVM-rbf	**85.0**	**92.0**	80.0	**83.6**	**86.0**

In Dutch to Hungarian direction, compared with results from manual feature extraction in binary classification, results reported in the second part of Table 4. It's clear that both embedding methods perform much better than manual feature extraction despite a limited number of speech samples, the best accuracy achieved in manual feature extraction was 66.8% with vowel \E\ included, while our results using embedding methods achieved 86.8% accuracy in case of using rbf kernel which is 20% increase compared to knowledge-based feature engineering.

Table 3: Binary classifications results using embedding (Train= Dutch, Test= Hungarian).

	Model	Accuracy %	Specificity %	Sensitivity %	F1-score %	AUC %
X-vector Embedding	SVM-Linear	75.13	80.0	71.6	72.9	75.8
	SVM-rbf	74.8	79.8	71.2	72.5	75.8
ECAPA Embedding	SVM-Linear	82.6	79.3	86.7	83.5	83.0
	SVM-rbf	**86.8**	**84.3**	**89.7**	**87.3**	**83.0**

Table 4: Binary classifications results from manual feature selection, taken from (Sztahó et al., 2022).

Features		Accuracy %	Specificity %	Sensitivity %	F1-score %	AUC %
Train: Hun	With \E\	**86.2**	**86.7**	**85.7**	**85.7**	**95.5**
Test: Du	Without \E\	81.4	80.0	82.8	81.4	91.0
Train: Du	With \E\	51.6	99.4	2.60	5.00	64.6
Test: Hun	Without \E\	**66.8**	61.9	71.8	68.1	74.3

4.2 Severity estimation

Similarly, predicting the severity level of dysphonic speech has been performed in a multi-lingual fashion. The model was trained using extracted features from Hungarian speech, and tested on Dutch samples, the other way round also performed. The performance of the model is measured using Spearman correlation, Pearson correlation and RMSE. The best results in (Sztahó et al., 2022) by manually extracted features were bolded and underlined in Table 5. The last part of Table 5 refers to the results achieved in Dutch to Hungarian direction using knowledge-based feature extractions, the performance of the model is worse due to the limited number of speech samples in the Dutch database, which made the model unable to make a good generalization.

Table 5: Severity approximation results from manual feature selection, taken from (Sztahó et al., 2022).

	Features	Spearman	Pearson	RMSE
Train = Hu	With \E\	**0.740**	**0.720**	**0.790**
Test = Du	Without \E\	0.660	0.650	0.880
Train = Du	With \E\	0.354	0.382	1.055
Test= Hun	Without \E\	**0.535**	**0.590**	**0.960**

As can be seen from Table 6, our model using embedding features extracted using both speaker verification models outperform hand-crafted feature selection for predicting severity level when we trained the model on Hungarian samples and tested it on Dutch speech, which shows better generalization in all three-evaluation metrics. The best performance was achieved using ECAPA-TDNN model with rbf kernel. Figure 1 shows the scatter plots of original and the estimated H scores for the embedding features, colours indicate the severity scores.

Table 6: Severity approximation results from embedding (Train=Hungarian, Test = Dutch)

	Model	Spearman	Pearson	RMSE
X-vector Embedding	SVR-Linear	0.606	0.638	0.823
	SVR-rbf	0.644	0.677	0.760
ECAPA Embedding	SVR-Linear	0.768	0.770	0.674
	SVR-rbf	**0.769**	**0.771**	**0.650**

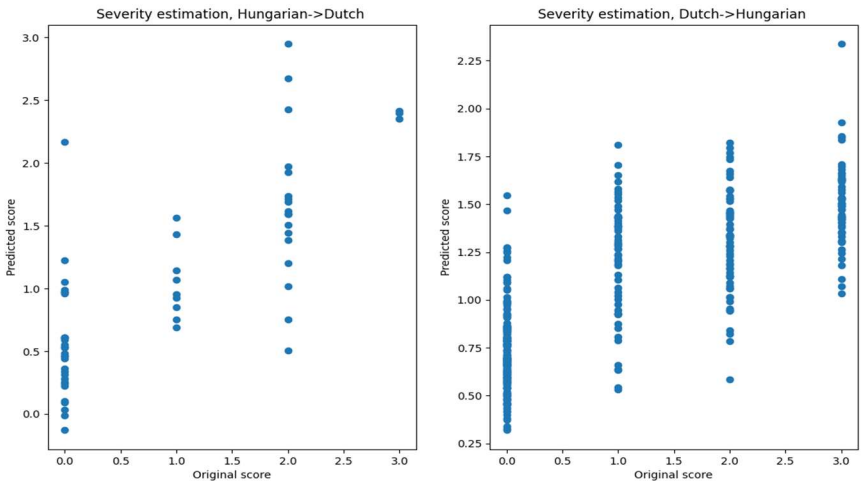


Figure 1: Scatter plot of original and the estimated H scores for both cross-language directions.

Our results from Dutch to Hungarian direction of predicting hoarseness levels is reported in Table 7. Compared to the result from handcrafted features included in Table 5, using features extracted from speaker embedding models performs much better. The capability of ECAPA-TDNN can be seen in Table 7 and scatter plot in Figure 1. Left side of the scatter plot shows severity prediction from Hungarian to Dutch direction. It's clear from the plot the algorithm made better generalizations compared to the right

side of the plot. for estimating the severity level of dysphonic patients in cross-lingual approaches that were not possible using hand-crafted acoustic feature extractions because of the weak performance in predicting severity level.

Table 7: Severity approximation results from embedding (Train=Dutch, Test = Hungarian)

	Model	Spears	Pearson	RMSE
X-vector Embedding	SVR-Linear	0.716	0.662	0.983
	SVR-rbf	0.667	0.599	0.964
ECAPA Embedding	SVR-Linear	0.746	0.730	**0.857**
	SVR-rbf	**0.783**	**0.771**	0.881

5 Discussion

The findings from this experiment indicate that binary classification of normal and dysphonic speech in a cross-lingual manner is possible using deep learning embedding techniques. For the Hungarian to Dutch direction, it can be noted that our results are competitive with manually extracted acoustic features. The best accuracy achieved using manual feature extraction was 86%, in contrast using speaker embedding we achieved 85% accuracy with ECAPA-TDNN.

The biggest difference between our model and model based on knowledge-based features was in Dutch to Hungarian direction. Because of the limited number of samples in Dutch language, training the model with acoustic features and testing on Hungarian samples were not reasonably possible. The best accuracy achieved was 66.8%, while using features extracted from both embedding models, we were able to achieve 75% and 86% accuracy using X-vector and ECAPA-TDNN, respectively.

Embedding techniques adapted from speaker recognition have shown a good result in estimating the severity of dysphonic speech. The two models outperform traditional hand-crafted acoustic features in a cross-language setup (Hungarian to Dutch and Dutch to Hungarian). The severity estimation was performed using Support Vector Regressor (SVR) with both linear and rbf kernels. Nearly 15% difference can be observed in RMSE metrics in Hungarian to Dutch direction (knowledge-based achieved 0.79 while our model achieved 0.65 with ECAPA-TDNN embedding). The highest Pearson correlation achieved was 0.771 in both directions of the model using rbf kernel. Embedding methods achieved 0.796 and 0.783 in Spearman correlation for both Hungarian to Dutch and Dutch to Hungarian direction, respectively. Which is superior compared to manual feature extraction. Due to the size of datasets and nature of Deep Neural Network (DNN) which requires considerably larger dataset to be able to make a good generalization, using DNN might not be a desirable choice for the problem.

The overall observation of this study is that using two state of art speaker embedding extraction methods can be a good replacement for classical hand-crafted acoustic feature extractions for binary classification as well as predicting the severity level of dysphonic speech in a cross-lingual way. The results show that cross-lingual dysphonic speech detection might be possible using deep learning embedding adapted from speaker verification. The models can extract language-independent features from both datasets, and they can be used for the above-mentioned tasks. The main advantage of these embedding models is that they are fast; since they have been trained before, much computation is not required. Another benefit, they don't require hand-crafted feature selection and engineering which is a very problematic and time-consuming task.

Another observation of this study is that ECAPA-TDNN shows better performance compared to the former X-vector in both binary classification and severity level estimation. This is expected because it is an extension of the X-vector, which is the result of the improvements done in former model in three different ways mentioned in section 3.2.

6 Conclusion

In this paper binary classification and severity level estimation of dysphonic speech have been performed in a cross-lingual fashion adapting different speaker verification speaker embedding methods. Two state-of-the-art pre-trained embedding models have been used, which were trained using out-of-domain speech. The embeddings of speech samples in both datasets were extracted using X-vector and ECAPA-TDNN and are used with SVM and SVR for classification and regression. The results proved that these pre-trained deep learning models can be used as a replacement for acoustic feature extraction methods. Models trained on embedding features that are results of pre-trained speaker verification model achieved 85% and 86.6% in binary classification in Hungarian to Dutch, and Dutch to Hungarian, respectively. Compared to manual feature extraction, we can see that our results are competitive in Hungarian to Dutch direction and outperformed in Dutch to Hungarian. This is an indication that we can use these embedding techniques with a small number of samples and still get a good result. Results were superior compared to ones we got from hard-crafted feature engineering. Moreover, the embeddings show a good result in estimating the severity level of dysphonic speech, which leads us to the conclusion that cross-lingual detection is possible with acceptable results.

Acknowledgement

The work was funded by project no. FK128615, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK_18 funding scheme. Also, the work was funded by Project no. K128568 with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K_18 funding scheme.

References

AL-Dhief, F. T., Latiff, N. M. a. A., Malik, N. N. N. A., Sabri, N., Baki, M. M., Albadr, M. A. A., Abbas, A. F., Hussein, Y. M., & Mohammed, M. A. (2020). Voice pathology detection using machine learning technique. 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT),

Awan, S. N., & Roy, N. (2006). Toward the development of an objective index of dysphonia severity: a four-factor acoustic model. Clinical linguistics & phonetics, 20(1), 35-49.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., & Reynolds, D. A. (2004). A tutorial on text-independent speaker verification. EURASIP Journal on Advances in Signal Processing, 2004(4), 1-22.

Botelho, C., Teixeira, F., Rolland, T., Abad, A., & Trancoso, I. (2020). Pathological speech detection using x-vector embeddings. arXiv preprint arXiv:2003.00864.

Cohen, S. M., Kim, J., Roy, N., Asche, C., & Courey, M. (2012). Prevalence and causes of dysphonia in a large treatment-seeking population. The Laryngoscope, 122(2), 343-348.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Dankovičová, Z., Sovák, D., Drotár, P., & Vokorokos, L. (2018). Machine learning approach to dysphonia detection. Applied Sciences, 8(10), 1927.

Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. arXiv preprint arXiv:2005.07143.

Egas-López, J. V., Kiss, G., Sztahó, D., & Gosztolya, G. (2022). Automatic Assessment of the Degree of Clinical Depression from Speech Using X-Vectors. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),

Grossmann, T., Vaish, A., Franz, J., Schroeder, R., Stoneking, M., & Friederici, A. D. (2013). Emotional Voice Processing: Investigating the Role of Genetic Variation in the Serotonin Transporter across Development. PLoS ONE, 8(7), e68377. https://doi.org/10.1371/journal.pone.0068377

Hadjaidji, E., Korba, M. C. A., & Khelil, K. (2021). Spasmodic Dysphonia Detection Using Machine Learning Classifiers. 2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI),

Hegde, S., Shetty, S., Rai, S., & Dodderi, T. (2019). A survey on machine learning approaches for automatic detection of voice disorders. Journal of Voice, 33(6), 947. e911-947. e933.

Lin, H., Karjadi, C., Ang, T. F., Prajakta, J., McManus, C., Alhanai, T. W., Glass, J., & Au, R. (2020). Identification of digital voice biomarkers for cognitive health. Exploration of medicine, 1, 406.

Orozco-Arroyave, J. R., Hönig, F., Arias-Londoño, J., Vargas-Bonilla, J., Daqrouq, K., Skodda, S., Rusz, J., & Nöth, E. (2016). Automatic detection of Parkinson's disease in running speech spoken in three different languages. The Journal of the Acoustical Society of America, 139(1), 481-500.

Scheuerer, R., Haderlein, T., Nöth, E., & Bocklet, T. (2021). Applying X-Vectors on Pathological Speech After Larynx Removal. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU),

Schönweiler, R., Hess, M., Wübbelt, P., & Ptok, M. (2000). Novel approach to acoustical voice analysis using artificial neural networks. Journal of the Association for Research in Otolaryngology, 1(4), 270-282.

Schwartz, S. R., Cohen, S. M., Dailey, S. H., Rosenfeld, R. M., Deutsch, E. S., Gillespie, M. B., Granieri, E., Hapner, E. R., Kimball, C. E., & Krouse, H. J. (2009). Clinical practice guideline: hoarseness (dysphonia). Otolaryngology–Head and Neck Surgery, 141(1_suppl), 1-31.

Shinohara, S., Omiya, Y., Nakamura, M., Hagiwara, N., Higuchi, M., Mitsuyoshi, S., & Tokuno, S. (2017). Multilingual evaluation of voice disability index using pitch rate. Adv. Sci. Technol. Eng. Syst. J, 2, 765-772.

Snyder, D., Garcia-Romero, D., Povey, D., & Khudanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. Interspeech,

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Stemple, J. C., Roy, N., & Klaben, B. K. (2018). Clinical voice pathology: Theory and management. Plural Publishing.

Strimbu, K., & Tavel, J. A. (2010). What are biomarkers? Current Opinion in HIV and AIDS, 5(6), 463-466. https://doi.org/10.1097/coh.0b013e32833ed177

Syed, S., Rashid, M., Hussain, S., Imtiaz, A., Abid, H., & Zahid, H. (2021). Inter classifier comparison to detect voice pathologies. Mathematical Biosciences and Engineering, 18(3), 2258-2273.

Togneri, R., & Pullella, D. (2011). An overview of speaker identification: Accuracy and robustness issues. IEEE circuits and systems magazine, 11(2), 23-61.

Van Houtte, E., Van Lierde, K., & Claeys, S. (2011). Pathophysiology and treatment of muscle tension dysphonia: a review of the current knowledge. Journal of Voice, 25(2), 202-207.

Wood, J. M., Athanasiadis, T., & Allen, J. (2014). Laryngitis. Bmj, 349, g5827. https://doi.org/10.1136/bmj.g5827

Wu, Y., Chen, P., Yao, Y., Ye, X., Xiao, Y., Liao, L., Wu, M., & Chen, J. (2017). Dysphonic voice pattern analysis of patients in Parkinson's disease using minimum interclass probability risk feature selection and bagging ensemble learning methods. Computational and mathematical methods in medicine, 2017.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., ... & Bengio, Y. (2021). SpeechBrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

Sztahó, D., Tulics, M. G., Qi, J., & Vicsi, K. (2022). Cross-lingual detection of dysphonic speech for Dutch and Hungarian datasets. Proceedings Biosignals 2022.

Korpuszok, adatbázisok

HunEmPoli: magyar nyelvű, részletesen annotált emóciókorpusz

Ring Orsolya[1], Vincze Veronika[2], Guba Csenge[1,3], Üveges István[1]

[1]Társadalomtudományi Kutatóközpont, Politikatudományi Intézet
Budapest, Tóth Kálmán utca 4.
[2]ELKH-SZTE Mesterséges Intelligencia Kutatócsoport
Szeged, Tisza Lajos körút 103.
[3] SZTE Nyelvtudományi Doktori Iskola
Szeged, Egyetem utca 2.
{ring.orsolya, uveges.istvan}@tk.hu
vinczev@inf.u-szeged.hu
csenge.guba@gmail.com

Kivonat Cikkünkben egy részletesen annotált, emócióelemzésre használható korpuszt mutatunk be, amely a projekthez kidolgozott emóciókategóriarendszer szerint, tagmondat szinten került annotálásra, alapját pedig 1008 db az Országgyűlés ülésein 2014 és 2018 között elhangzott napirend előtti felszólalás szabadon elérhető szövege jelentette, összesen 764008 token terjedelemben. Munkánkban részletesen ismertetjük az alkalmazott kategóriákat, az annotás menetét, közöljük az alapvető korpuszstatisztikai és minőségbiztosítási adatokat, valamint példát mutatunk arra, hogyan használható a léterehozott korpusz gépi érzelem- vagy szentimentazonosításra.
Kulcsszavak: Argumentumszintű szentimentelemzés, emócióanalízis, ABSA, politikai kommunikáció, parlamenti beszéd

1. Bevezetés

Bár a politikai kommunikációnak korpuszokon és NLP-módszereken alapuló kutatása párhuzamosan a szövegek online elérhetőségének növekedésével egyre nagyobb jelentőségű, a megjelent tanulmányok legtöbbször a politikusok médiában és közösségi médiában elhangzó megnyilatkozásait elemzik (Gollust és mtsai, 2020; Mariani és mtsai, 2020; Aparicio és mtsai, 2021; Wang és mtsai, 2021; Rufai és Bunce, 2020).

A parlamenti viták leiratai lényegében a beszélt nyelv ellenőrzött és szabályozott körülmények között készült átiratai, melyek szabadon elérhetőek, mivel az információszabadságról szóló törvény alapján nem vonatkoznak rájuk a szerzői jogi vagy a személyes adatok védelmére vonatkozó jogszabályok. Éppen ezért az utóbbi években több nemzetközi projekt keretében készült és készül jelenleg is korpusz parlamenti felszólalásokból[1]. Hiszen a parlamenti viták jegyzőkönyvei

[1] Ilyen például a CLARIN https://www.clarin.eu/, a Comparative Agendas https://www.comparativeagendas.net/ vagy az OPTED https://opted.eu/ Projekt

egyedi tartalmuk, szerkezetük és nyelvezetük miatt fontos forrásai a társadalomtudományi és nyelvészeti kutatásoknak.

A parlamentek a politikai kommunikáció fontos helyszínei, ahol a választott képviselők megvitatják a benyújtott törvényjavaslatokat és más országos jelentőséggel bíró ügyeket. Az itt elhangzó beszédek általában előre megtervezett beszédaktusok, amelyek által a képviselők kiemelt célja nem pusztán tájékoztatás, hanem a hallgatóság meggyőzése és támogatásuk megszerzése is. Ezen beszédek gépi érzelemelemzése nagy kihívást jelent, és bár az elmúlt években több erőforrás is elérhetővé vált, ezek nagy része angolul áll rendelkezésre és alkalmazásuk jelentős munkával járó adaptációt igényel, különösen az olyan morfológiailag gazdag nyelvek esetén, mint a magyar (Jang és Shin, 2010; Mladenović és mtsai, 2016). A magyar nyelvű szövegek hatékony elemzéséhez így felmerült az igény egy mélyen annotált emóciókorpusz létrehozására, amely a későbbiekben jól alkalmazható majd különböző gépi tanítási feladatok végrehajtásánál, nyelvi modellek tanításánál. Így egy ilyen korpusz egyszerre szolgáhatja a politikatudományi mellett a számítógépes nyelvészeti- és mesterséges intelligencia-kutatásokat is.

2. Kapcsolódó irodalom

2.1. Politikai kommunikáció

A politikai kommunikáció az érzelmi hatások kiváltásával és a különböző eszmék terjesztésével a politikai cselekvést ösztönzi. A közelmúltban lezajott technikai és társadalmi változások következtében jelentősen megszaporodott a kommunikációban résztvevők, valamint a rendelkezésre álló kommunikációs csatornák száma, mindez pedig befolyásolta a politikai kommunikáció jellegét és intenzitását is. A politikai szereplők szerepük felértékelődésére a kommunikáció professzionalizálásával reagálnak. A politikai kommunikáció egyik kiemelt jelentőségű helyszíne a parlament, ahol a választott képviselők a benyújtott törvényjavaslatok és más nemzeti jelentőségű ügyek megvitatása közben igyekeznek olyan módon artikulálni véleményüket, hogy az minél szélesebb közönséget érjen el. A parlamenti viták során különféle témák merülnek fel, érvek és ellenérvek ütköznek és ezeken keresztül alakul ki egy politikai napirend, amely aztán tematizálja a nyilvános vitákat (Bene és Nábelek, 2019). Az érzelmek politikai kommunikációban való kifejezésének kutatása az utóbbi években egyre nagyobb hangsúlyt kapott mind a nemzetközi, mind a hazai társadalomtudományi kutatásokban (Szabó, 2020; Crigler és Just, 2012; Wagner és Morisi, 2019; Settle, 2020; Richards, 2004; Haselmayer és Jenny, 2017), de a politikai és különösen a parlamenti beszédek érzelmi töltésének NLP-eszközökkel történő elemzése viszonylag új és kevéssé kiforrott (Gold és mtsai, 2018; Jafarian és mtsai, 2021), különösen a magyar nyelvre.

2.2. Emócióelemzés

A szentiment- vagy emócióelemzés célja az egyes szövegek tartalmából kinyerni azokat az információkat, amelyek értékelést fejeznek ki. Az elemzés különböző szinteken végezhető részint attól függően, hogy mi az elemzés alapegysége

(például szöveg, mondat vagy tagmondat) és meghatározzuk-e azt, hogy az érzelem mire irányul, vagy mi váltja ki azt. A szakirodalom különbséget tesz a pozitív-negatív-semleges skálán mozgó szentiment és a több kategóriával dolgozó érzelemelemzés között, melyek közül az utóbbi sokkal több információt nyújt az adott egység érzelmi töltetéről (Marcus, 2000). Bár a két fogalmat sokszor egymás szinonimájaként is használják, jelen dolgozatban mi a fenti különbségtételt alkalmazzuk a szentiment-, illetve emócióelemzés kifejezések használatánál.

A legismertebb érzelemkategória-rendszerek valószínűleg Ekman (Ekman és Wallace V. Friesen, 1982) és Plutchik (Plutchik, 1964, 1980a, 1982) nevéhez fűződnek. Ekman, aki az emberi arckifejezések kultúrákon átívelő egységes jellegét tanulmányozta, 6 (harag, undor, félelem, boldogság, szomorúság és meglepetés), Plutchik pedig 8 kategóriát határozott meg az érzelmek osztályozására (várakozás, meglepetés, öröm, bánat/szomorúság, elfogadás, undor, harag, félelem).

A pszichológiai szakirodalomban az érzelem fogalma számos kérdést vet fel, és nincsenek formális kritériumok arra vonatkozóan, hogy mit tekintünk érzelemnek és mit nem (Chapman és Nakamura, 1998; Cabanac, 2002; Griffiths, 2008). Az érzelmek empirikus elemzése ugyanis rámutatott azok összetettségére (Lakoff és Kovecses, 1987), illetve a különböző érzelmek jelentésének összefonódására a helyi kultúra sajátosságaival (Wierzbicka, 1999), valamint a speciális érzelmi kifejezések eltéréseire a különböző nyelveken (Russell, 2003). Az érzelmek komplexitása pedig nagyban nehezíti azok NLP-módszerekkel történő elemzését.

Az érzelemelemzés másik alapvető kérdése, hogy pontosan mi hordoz érzelmeket a tartalom szintjén. Ezért például több kutatásban próbálkoztak azzal (Feng et al., 2013; Loukachevitch és Levchik, 2016), hogy olyan érzelemszótárakat készítsenek, amelyek nem csak explicit, hanem konnotatív érzelemértékkel is bírnak (Feng és mtsai, 2013; Loukachevitch és Levchik, 2016). Például a kitüntetés vagy az előléptetés pozitív konnotációval bírnak, míg a munkanélküliség és a terrorizmus negatívak. Ahogy Loukachevitch és Levchik megállapította, „a konnotációs értékkel bíró, nem érzelmi töltetű szavak általában a társadalmi élet negatív vagy pozitív jelenségeiről (tényekről) közvetítenek információt" (Loukachevitch és Levchik, 2016), így ezen konnotatív szemantikai tartalmak automatikus elemzése jelentős kihívást jelent.

2.3. Magyar nyelvű szentiment- és emóciókorpuszok

A most bemutatott emóciókorpusz kialakításával a hazai szentiment- és emócióelemzési feldatokra alkalmas korpuszok, valamint a kapcsolódó kutatások két alapvető hiányosságára reflektáltunk. Az első ezek közül az eddig vizsgált doméneket illeti, tekintettel arra, hogy magyar nyelvű politikai szövegek gépi tanulással történő érzelemelemelemzésére korábban nem létezett megfelelő erőforrás, a második pedig abban keresendő, hogy az elmúlt évtizedben mindössze két olyan nyilvánosan elérhető magyar korpusz készült, amelyek szentimentelemzési feladatokra lettek annotálva, míg emócióelemzésre egy sem.

Az eddig rendelkezésre álló korpuszok közül az Opinhubank (Miháltz, 2013) megközelítőleg tízezer mondatos állománya mondatszinten, pozitív-negatív-semleges szentimentek annotációját tartalmazza, ezenfelül jelöli az adott vélemények

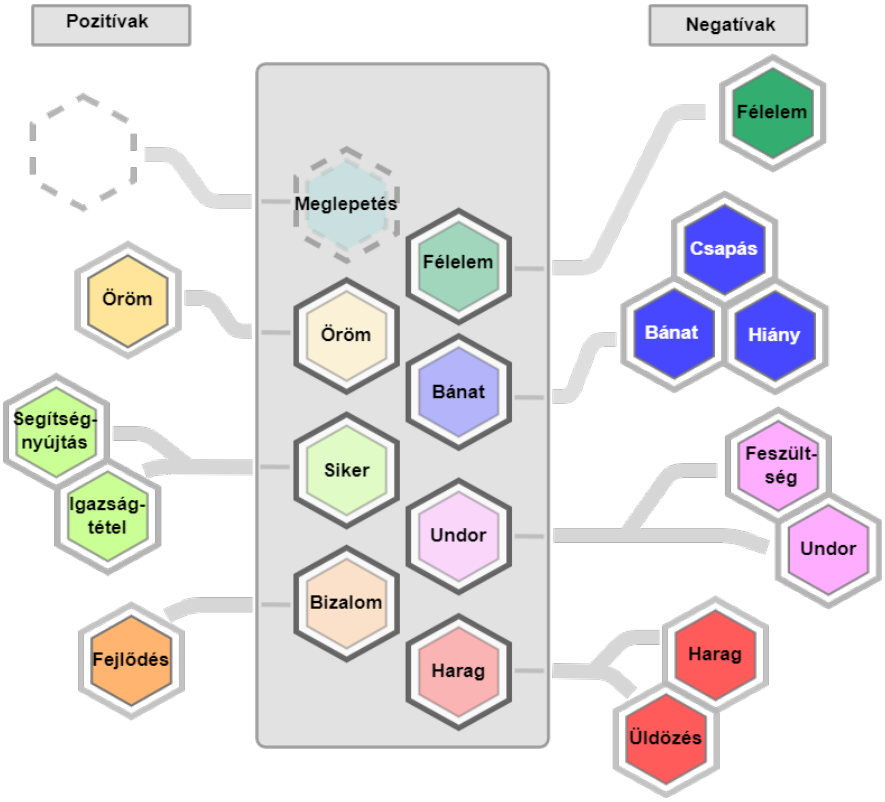
„célpontját” is a mondatokban megjelenő névelemek esetében. A forrásszövegeket ez esetben online hírforrások (híroldalak, blogok) jelentették. A HuSent korpusz (Szabó és mtsai, 2016) nagyjából 17 ezer mondatos állománya ezzel szemben online forrásból származó termékvéleményeket tartalmaz. A kapcsolódó annotáció egyebek mellett a (pozitív/negatív) szentimentértéket kifejező kifejezést és az érzelem célpontját tartalmazza.

3. A HunEmPoli korpusz létrehozása

A korpusz építéséhez a 2014-2018 közti parlamenti ciklusban elhangzó napirend előtti felszólalásokat használtuk fel. A vizsgálatunk alapjául szolgáló metaadatokkal ellátott adatbázis a Társadalomtudományi Kutatóközpont Politikatudományi Intézetének Hungarian Comparative Agendas Projektje keretében készült, kutatási célokra, regisztráció ellenében szabadon hozzáférhető. A napirend előtti felszólalások az országgyűlés plenáris ülésén hangoznak el országos jelentőségű, halaszthatatlan és rendkívüli ügyekben 2-5 perces időterjedelemben. A korpusz eltér a magyar nyelvre elérhető, beszélt nyelvi korpuszoktól, melyek spontán beszédet tartalmaznak és/vagy informális jellegűek (pl. Szabó és mtsai (2021), Vincze és mtsai (2021)), hiszen a műfajból adódóan beszélt nyelvi adatokról van szó, de ezek stílusa hivatalos, formális, a képviselők általában előre megtervezték/megírták beszédüket. A szövegek szinte minden esetben kizárólag magázó formulákat használnak, ugyanakkor a leiratok nem tartalmazzák az élőbeszédre jellemző hezitációkat, szüneteket, téves szókezdeteket. Sajátosságként ugyanakkor megemlítjük, hogy a beszédek közbeni bekiabálások is szerepelnek a leiratokban, ezeket azonban a későbbi annotáció során nem vettük figyelembe, hiszen az azokban található érzelmek nem az adott felszólalóhoz köthetők.

3.1. Az emóciókategóriák

Kutatásunk során létrehoztunk egy induktív emóció-kategóriarendszert, melyeknek kategóriái megfeleltethetőek Plutchik (1980b) emóció-kategóriarendszerének, amely nyolc osztályt különböztet meg (egy kivétellel, mivel szövegeinkben *meglepetés* nem volt azonosítható, így az induktív kategóriarendszerből hiányzik) és átválthatóak a szentimentelemzésben használatos pozitív-negatív kategóriákra is. Erre a kibővített rendszerre azért volt szükség, mert korábbi tapasztataink szerint a politikai szövegekben található mondatok egyébként nem, vagy csak rendkívül alacsony annotátori egyetértés mellett voltak besorolhatóak az emócióelemzés során egyik leggyakrabban használt, Plutchik-féle kategóriarendszerbe, míg a bővített rendszer segítségével a korpusz magas kódolók közötti egyetértéssel volt annotálható. A végső annotációs útmutatóban összesen 12 ún. emóció-topikot (későbbiekben ET) határoztunk meg, melyek mindegyikéhez legalább három hívószót vagy kifejezést adtunk meg, hogy megkönnyítsük az annotátorok munkáját.



1. ábra: Plutchik kategóriái és a belőlük képzett emóciókategóriák a HunEmPoli korpuszban

Kapcsolódó fogalmak	**Emóciótopik**	**Plutchik szerint**	**Szentiment**
fenyegetettség, fenyegetés, félelem, rettegés, aggodalom	**Félelem**	Félelem	Negatív
hiány, nélkülözés, nyomor, nyomorúság, nincstelenség, gyötrelem, szenvedés, kín, sikertelenség, kudarc, negatív irányú változás	**Hiány, nélkülözés**	Bánat	
szomorúság, bánat, elkeseredettség, reménytelenség, mélabú	**Szomorúság, bánat**		
szerencsétlenség, katasztrófa, csapás	**Szerencsétlenség, csapás**		
terror, merénylet, üldözés, kegyetlenség, kegyetlenkedés, szervezett bűnözés, vandalizmus, rongálás, szándékos károkozás, bűntett, erőszak	**Terror, bűnözés**	Harag	
düh, harag, gyűlölet	**Harag**		
feszültség, zavar, érdekellentét, revans, megbüntetés	**Feszültség, zavar**	Undor	
lenézés, undor, irtózás, megvetés, gúny	**Lenézés, undor**		
javulás, közeledés, békülés, „relief”, enyhülés, fejlesztés, fejlődés, siker, pozitív irányú változás	**Javulás, fejlődés**	Siker	Pozitív
öröm, élvezet, vidámság, derű, szeretet, elfogadás, tolerancia	**Öröm**	Öröm	
segítségnyújtás, mentés, segély, gyógyítás, ellátás, kiszabadítás	**Segítségnyújtás**	Bizalom	
igazságtétel, nyomozás, törvényhozás, hivatalos szervek fellépése	**Igazságtétel, nyomozás**		

1. táblázat. Az emóciótopikok megfeleltetése a Plutchik-féle rendszer kategóriáinak és a pozitív/negatív szentimentértékeknek.

3.2. Annotálás

Az annotálást összesen hat annotátor végezte, akik részben nyelvészetet, részben politikatudományt tanuló MA- vagy PhD-hallgatók voltak. Az annotálás megkezdése előtt részletes annotálási útmutatót, valamint betanítást kaptak, majd próbaannotációkat végeztek. Az annotálást a tagtog nevű online is elérhető eszköz segítségével végeztük[2], melynek grafikus felülete megkönnyítette az annotátorok munkáját. Az annotátorok minden esetben egy-egy teljes napirend előtti

[2] https://www.tagtog.com/

felszólalást kaptak meg. Egy szöveget mindig csak egy annotátor annotált, de a korpusz 25 százalékát minőségbiztosítási célból ketten-ketten párban is feldolgozták. A minőségbiztosított fájlok esetén arra is ügyeltünk, hogy a teljes feladat során az annotátorpárok változzanak.

Az annotálást tagmondat szinten végeztük, azaz a tagmondatok szintjén volt szükséges meghatározni az ET-t, míg a tagmondatok közötti esetleges kötőszavakat nem kellett belevenni az annotációba. Több tagmondatot folyamatosan, együtt csak akkor lehetett annotálni, ha ugyanaz az ET folytatódott több tagmondaton keresztül (ebben az esetben értelemszerűen a kötőszavak is bekerültek az annotációba). Az annotálás során az annotátornak a következőket kellett megtalálnia és bejelölnie a szövegekben az ET-t tartalmazó tagmondatokon kívül : (1) az ET-t kifejező részt, azaz a kulcskifejezést, ami lehetett egy szó, de akár többszavas kifejezés is. Egy ET-n belül több kulcskifejezés is jelölhető volt. (2) az ún. argumentumot, tehát azt a dolgot, fogalmat (vagy akár annak részét vagy sajátságát), ami az azonosított ET-t kiváltja, vagy amire az ET irányul. Fontos, hogy az argumentum: (1) Nem keverendő össze az érzelem átélőjével, illetve az események elszenvedőjével. (2) Az argumentum egy vagy több szóból is állhat. Az annotáció során az argumentum jelölése csak akkor volt szükséges és lehetséges, ha az egyértelműen, szavak szintjén is megfogható volt. Rejtett névmások esetén nem jelöltük azokat. Abban az esetben, ha az argumentum egy névmás volt az adott egységben, azt ugyanúgy jelöltük, mintha tartalmas szó szerepelne a tagmondatban. Mivel az argumentum az ET-vel jelölt mondatrészen kívül is állhatott, így az annotáció során minden esetben összekötöttük az ET-t hordozó mondatrészt és a vele kapcsolatban álló argumentumot.

Az annotátorok első feladatként tehát meghatározták, hogy mely tagmondatok hordozzák a 12 ET valamelyikét, majd kiválasztották a felkínált lehetőségek közül az ennek megfelelő annotációs taget. Ezután pedig tetszőleges sorrendben bejelölték az adott ET-vel jelölt mondatrészhez tartozó kulcskifejezéseket és argumentumokat, illetve az argumentumok esetében összekötötték azt a hozzá tartozó mondatrésszel.

Az annotálás során felmerült az a probléma, hogy esetenként egy-egy (tag)-mondatra több ET is illett. Pl.: *...ahol a multinacionális kereskedelmi cégek kezdeményezésével szemben az alkotmánybírósággal összefogtak a baloldali szakszervezetek.* Erre a mondatra a javulás, fejlődés ET, illetve a feszültség, zavar ET is kiosztható lett volna. Ilyen esetekben az annotátorok azt az utasítást kapták, hogy azt az egy ET-t jelöljék, amelyet szerintük a kontextus jobban indokol.

Továbbá problémát okozott még bizonyos esetekben a kulcskifejezés hiánya: *ténylegesen Matolcsy György hozott döntést arról, hogy mikor mennyi pénzt kötnek le unokatestvérének bankjánál, mikor mennyi állampapírt vásárolnak unokatestvére bankjától*, vagy az olyan kulcskifejezések jelölése, ahol a kifejezés tagjai a mondaton belül távol állnak egymástól: *...hogy a háromgyermekes családoknak körülbelül csak a 8 százaléka tudja majd igénybe venni teljes mértékben a családi adókedvezményt..* Az első esetben egyáltalán nem jelöltünk kulcskifejezést, a második példánál viszont a kulcskifejezésbe amúgy nem tartozó szavak is belekerültek a kijelölt szakaszba. Speciális esetként kezeltük az idiomatikus

kifejezéseket az argumentum jelölésének szempontjából. Az idióma alanyát és tárgyát - amennyiben azt a beszélő sem szó szerint értette - nem jelöltük külön argumentumnak, de az esetek többségében a kulcskifejezés részét képezte, pl.: *Nyilván sokan ismerik azt a közmondást, hogy más tollával nem illik ékeskedni.*

Emóciótopik	**Példa**
Félelem	*...és ez veszélyt jelent Magyarország biztonságára.*
Hiány, nélkülözés	*...miközben az embereket érintő adóterhek abban az évben is meg az azt követő évben is durván növekedtek.*
Szomorúság, bánat	*Sokukat elvitték málenkij robotra a Gulagra, nagyon sokan nem jöttek vissza közülük.*
Szerencsétlenség, csapás	*...mint a bányaszerencsétlenségben rekedt chilei bányászokat.*
Terror, bűnözés	*...leszámítva azt a szűk 10 százalékot, akik az állam vagyonát szétlopják, urizálnak, több mint 90 százalék rovására.*
Harag	*A tehetetlen dühöt érzem a kormány elhibázott bérpolitikájával és a társadalompolitikájával szemben.*
Feszültség, zavar	*Ők azok, akik fizetnek azért, hogy Trump mielőbb megbukjon.*
Lenézés, undor	*...majd fülét-farkát behúzva szépen hazaballagott, és azután az a bizonyos bankadó az ilyen iciri-piciri lett.*
Javulás, fejlődés	*...hogy megváltozik az életük; a munka tartást és méltóságot ad az embernek, javítja az önbecsülését, és nem utolsósorban pénzt hoz a házhoz.*
Öröm	*És nagyon örülünk ennek az eredménynek.*
Segítségnyújtás	*...egyrészt az üldözött, bajba jutott emberen segíteni kell...*
Igazságtétel, nyomozás	*...és ahol már az Európai Unió vizsgálóbiztosai is vizsgálódnak.*

2. táblázat. Példamondatok az egyes emóciókategóriák szerint.

3.3. Minőségbiztosítás

A korpusz minőségének biztosítása érdekében időről időre kiszámoltuk az annotátorok közti egyetértést (Cohen's Kappa). Mivel az annotátorok tagmondatok szintjén jelölték az emóciótopikokat, tokenszintű kiértékelést végeztünk, hiszen nem szerettük volna, ha az esetleges kisebb tévesztések (pl. írásjelek vagy kötőszavak jelölésében adódó eltérések) torzítják az eredményeket.

A minőségbiztosítás eredményei átlagosan 0,41 (Kappa) egyetértést mutattak, ami mérsékelt egyetértést jelent, ezért második körben egy annotátorpárral (akik az annotálás során minden összehasonlításban a legjobb eredményt produkálták) a teljes korpusz javítását elvégeztettük. Ennek eredményeként a javított korpusz átlagos egyetértése 0,7574 (Kappa) már kategóriánként erős egyetértést mutat, lásd a 3. táblázat adatai.[3] Ezen felül, az argumentumok esetében a mért

[3] A minőségbiztosításra véletlenszerűen kiválasztott fájlokban a 9-es és a 10-es ET nem fordult elő. Ennek oka az, hogy ezek a teljes korpuszban csak igen alacsony elemszámmal szerepelnek.

egyetértés 0,8947 (Kappa), míg a kulcskifejezések esetében 0,9101 (Kappa) szerint alakult, amely már a majdnem teljes egyetértést jelent.

ET	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**11**	**12**
Kappa	0,564	0,885	0,971	0,930	0,615	0,846	0,5	1	0,264	1

3. táblázat. Emóciótopikok esetében mért egyetértés - κ (1: félelem, 2: hiány, nélkülözés, 3: terror, bűnözés, 4: javulás, fejlődés, 5: feszültség, zavar, 6: lenézés, undor, 7: szomorúság, bánat 8: öröm, 11: segítségnyújtás, 12: igazságtétel, nyomozás).

3.4. Korpuszadatok

Az elkészült korpusz a korábban említetteknek megfelelően 1008 napirend előtti felszólalást tartalmaz, amelyek mindösszesen 764008 tokenből, illetve 36475 mondatból állnak. Ezek pártok szerinti megoszlását részletesen a 4. táblázat ismerteti. Jól látszik, hogy a parlamenti összetételnek megfelelően, a Fidesz képviselői szólaltak fel a legtöbbször. Ugyanakkor az is megfigyelhető, hogy a független képviselők, valamint a Jobbik felszólalásaiban szerepel átlagosan a legtöbb szó, azaz az ő felszólalásaik voltak a leghosszabbak. Ezzel ellentétben, a KDNP politikusai viszonylag röviden beszéltek.

Párt	**Token**	**Mondat**	**Token** (%)	**Mondat** (%)	**Beszéd**	**Token/ beszéd**	**Mondat/ beszéd**
LMP	156504	7430	20,48	20,37	197	794,44	37,72
KDNP	120844	6173	15,82	16,92	192	629,40	32,15
MSZP	142538	7222	18,66	19,80	193	738,54	37,42
Jobbik	160461	6900	21,00	18,92	197	814,52	35,03
Fidesz	176325	8415	23,08	23,07	221	797,85	38,08
Független	7336	335	0,96	0,92	8	917,00	41,88

4. táblázat. A korpusz alapvető statisztikai adatai.

Az 5. táblázat szemlélteti az egyes emóciótopikok pártonkénti megoszlását. Összességében a javulás számít a leggyakoribb emóciótopiknak, ez különösen a kormányoldali kommunikációban – azaz a Fidesz és a KDNP felszólalásaiban – nyilvánul meg. A második és harmadik leggyakoribb kategória, az érdekellentét és a romlás főként az ellenzéknél (LMP, MSZP, Jobbik) domináns. A 2. ábrán megfigyelhető az is, hogy e három kategória együttesen mintegy 75%-át teszik ki a korpuszban található emóciótopikoknak, azaz a többi 9 emóciótopik előfordulási gyakorisága jóval kisebb, mint e háromé (ezért is fontos, hogy az erre a három ET-re mért egyetértés magas, (0,81), azaz ezeket közel teljes egyetértéssel sikerült annotálni. E jelenség magyarázata valószínűleg a parlamenti beszédek

sajátságában rejlik: a napirend előtti felszólalások nagy hányada szól arról, hogy egy adott jelenség vagy problémakör, esetleg annak kormányzati kezelése javult-e, esetleg romlott az utóbbi időben. Ugyanakkor az érdekellentét erős jelenléte is indokolható a parlamenti vitakultúrával: a kormánypártok, illetve az ellenzéki pártok gyakran állítják szembe saját tevékenységüket a másik oldaléval, így az érdekellentétre utaló nyelvi kifejezések száma is magas a korpuszban.

	LMP	**KDNP**	**MSZP**	**Jobbik**	**Fidesz**	**Független**	**Összesen**
Félelem	133 21,49%	96 15,51%	87 14,05%	133 21,49%	161 26,01%	9 1,45%	619
Hiány, nélkülözés	2702 29,59%	880 9,64%	2370 25,95%	1968 21,55%	1156 12,66%	56 0,61%	9132
Terror, bűnözés	265 16,76%	172 10,88%	284 17,96%	452 28,59%	406 25,68%	2 0,13%	1581
Javulás, fejlődés	2031 16,31%	2694 21,63%	1923 15,44%	2189 17,58%	3454 27,73%	164 1,32%	12455
Feszültség, zavar	2202 23,45%	953 10,15%	1974 21,02%	2271 24,19%	1977 21,06%	12 0,13%	9389
Lenézés, undor	1044 26,81%	298 7,65%	988 25,37%	989 25,40%	575 14,77%	0 0,00%	3894
Bánat, szomorúság	64 1,42%	84 29,89%	33 11,74%	43 15,30%	56 19,93%	1 0,36%	281
Öröm	47 5,43%	514 59,42%	69 7,98%	52 6,01%	168 19,42%	15 1,73%	865
Harag	33 19,64%	30 17,86%	36 21,43%	49 29,17%	20 11,90%	0 0,00%	168
Szerencsétlenség	14 19,44%	4 5,56%	5 6,94%	0 0,00%	37 51,39%	12 16,67%	72
Segítségnyújtás	43 10,46%	165 40,15%	82 19,95%	32 7,79%	84 20,44%	5 1,22%	411
Igazságtétel	138 14,18%	82 8,43%	170 17,47%	250 25,69%	332 34,12%	1 0,10%	973

5. táblázat. Az emóciótopikok megoszlása pártonként.

A 6. táblázatban láthatjuk a Plutchik-féle emócióknak megfelelő kategóriákat (az eredeti annotációt leképezve ezekre), valamint a pozitív, illetve negatív érzelmi töltetű megnyilatkozások számát, szintén az eredeti annotációból leképezve. Észrevehetjük, hogy míg az ellenzéki pártoknál a negatív töltetű megnyilvánulások vannak egyértelmű fölényben, addig a kormánypártoknál más a helyzet: a Fidesznél ugyan a negatív megnyilatkozások száma magasabb, mint a pozitívaké, ám nem olyan nagy az eltérés köztük, mint az ellenzéki pártok esetében. A KDNP képviselőinek beszédeiben pedig a pozitív megnyilvánulások száma meg is haladja a negatívok számát.

	LMP	KDNP	MSZP	Jobbik	Fidesz
Félelem	133	96	87	133	161
Bánat	2780	968	2408	2011	1249
Harag	298	202	320	501	426
Undor	3246	1251	2962	3260	2552
Siker	2031	2694	1923	2189	3454
Öröm	47	514	69	52	168
Bizalom	181	247	252	282	416
Pozitív	2259	3455	2244	2523	4038
Negatív	6457	2517	5777	5905	4388

6. táblázat. Plutchik-féle emóciók és szentiment a korpuszban pártonként.

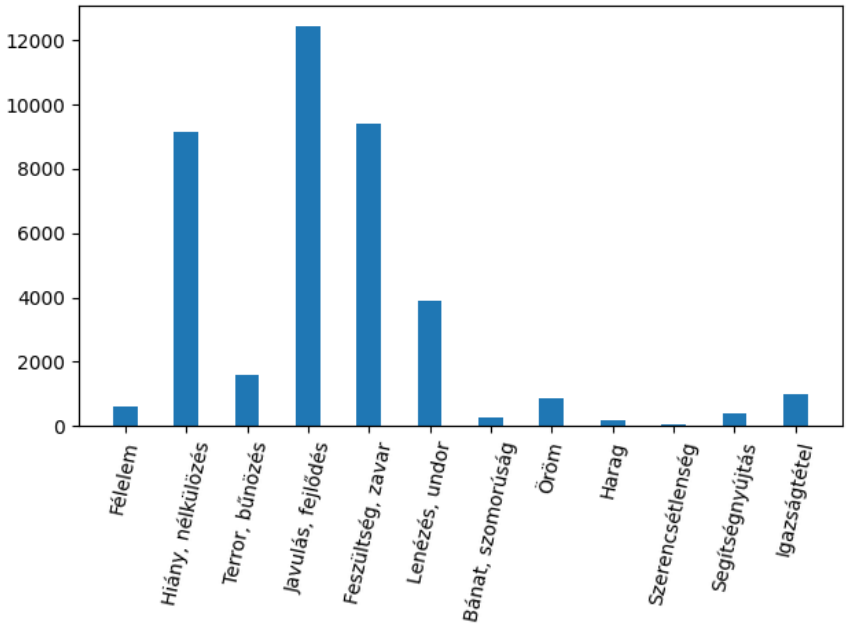
4. A korpusz felhasználhatósága

Ahogyan az 1. ábra is szemlélteti, az alkalmazott kategóriarendszer egyik nagy előnye, hogy „visszafelé kompatibilis" a szentiment- és emócióelemzéshez tradicionálisan alkalmazott, kevesebb osztállyal operáló annotálási rendszerekkel. Ez a gyakorlatban azt jelenti, hogy a választott kutatási kérdésnek megfelelően kiválasztható olyan átváltás az emóciótopikok és más rendszerek kategóriái között, amely után a korpusz tanítóadatként tud funkcionálni változatos problémák megoldásához.

Ilyen (magától értetődő) felhasználási lehetőség érzelemelemzésre alkalmas gépi tanulási modellek létrehozása, amelyre jó példa lehet Üveges és mtsai (2022), amelynek során a szerzők a korpusz egy korábbi változatának felhasználásával kíséreltek meg gépi tanítást a megadott 12 emóciótopikra. Tekintettel arra, hogy a kísérlet lefolytatásakor a korpusz minőségbiztosítása még nem volt befejezettnek tekinthető, az akkor felhasznált adatok csak mintegy a teljes szövegállomány kétharmadát tették ki (608 beszéd mondatai az 1008-ból).

A kísérlet során a huBERT modell (Nemeskey, 2020) finomhangolása történt meg a rendelkezésre álló tanítóadatok alapján aspektusszintű szentimentelemzési (ABSA) feladatra. A gépi tanult modell feladata tehát az volt, hogy a szövegegységben megjelenő érzelem-argumentum párokat kategorizálja. A kapott eredmények alapján az olyan kategóriák esetében, ahol rendelkezésre állt elégséges mennyiségű tanítóadat, a hangolt modell 0,6 - 0,8 körüli[4] F1-értékekkel teljesített. Elégséges mennyiségű adat alatt ez esetben tipikusan néhány ezer annotált mondat - aspektus pár volt értendő egy-egy emóciótopik esetében. Tekintettel az egyes kategóriák erősen kiegyensúlyozatlan eloszlására, ez mindösszesen 4 kategória esetében teljesült a korábbi, megszorítottabb részkorpuszon (Üveges és mtsai, 2022). Ezek az arányok a végleges korpuszban is hasonlóan kiegyensúlyozatlanok maradtak (vö. 5. táblázat). Itt megjegyzendő, hogy pusztán az emóciótopikok felismerése feltehetőleg ennél hatékonyabb lett volna, hiszen az emóciótopik + aspektus azonosítás alapvetően összetettebb feladatnak számít.

[4] Például a 'javulás, fejlődés' kategória esetében az első epoch alkalmával 0,86, a 'szomorúság, bánat' kategória esetén 0,7.



2. ábra: Az egyes kategóriák eloszlása a korpuszban.

Ezen felül természetesen számos további alkalmazás elképzelhető például a jelen korpuszon tanított modellek továbbhasználatával más politikai szövegek elemzése során.

5. Összegzés

Ebben a munkában bemutattuk a HunEmPolit, az első magyar, részletesen annotált, emócióelemzésre használható korpuszt. A korpusz 1008 db parlamenti napirend előtti felszólalást tartalmaz, összesen 764008 token terjedelemben, melyben összesen 39840 emóciót, 61890 kulcskifejezést és 126023 argumentumot azonosítottunk. Ismertettük az alkalmazott kategóriákat, az annotáció és a minőségbiztosítás menetét és alapelveit, emellett alapvető adatokat közöltünk az adatbázisról, valamint bemutattuk, hogyan használható érzelem- vagy szentimentazonosításra is. A korpusz kutatási célokra szabadon elérhető az alábbi linken: https://github.com/poltextlab/HunEmPoli_corpus.

A továbbiakban a korpusz adatainak társadalomtudományi szempontú elemzése, illetve újabb szövegek gépi emócióklasszifikálása mellett terveink között szerepel különféle gépi tanulási kísérletek elvégzése, valamint a huBERT-modell fimomhangolása emócióelemzési feladatokra.

Köszönetnyilvánítás

A publikációban szereplő kutatást a Társadalomtudományi Kutatóközpont és az ELKH-SZTE Mesterséges Intelligencia Kutatócsoport az Európai Unió támogatásával valósította meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében. Köszönjük Szabó Martina Katalin segítségét az induktív kategóriarendszer és az annotációs elvek kidolgozásában, valamit a projektben résztvevő annotátorok munkáját is.

Hivatkozások

Aparicio, J.T., de Sequeira, J.S., Costa, C.J.: Emotion analysis of Portuguese Political Parties Communication over the covid-19 Pandemic. In: 2021 16th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6. IEEE (2021)

Bene, M., Nábelek, F.: A politikai kommunikáció története a külföldi szakirodalomban . In: Kiss, B. (szerk.) A szavakon túl. Politikai kommunikáció Magyarországon, 1990-2015, pp. 11–29. L'Harmattan Kiadó (2019)

Cabanac, M.: What is emotion? Behavioural processes 60(2), 69–83 (2002)

Chapman, C., Nakamura, Y.: A bottom up view of emotion. In: ASSC Seminar http://server. phil. vt. edu/assc/watt/chapman1. html (1998)

Crigler, A.N., Just, M.R.: Measuring affect, emotion and mood in political communication. The Sage handbook of political communication pp. 211–224 (2012)

Ekman, P., Wallace V. Friesen, P.E.: What emotion categories or dimensions can observers judge from facial behavior In: Ekman, P. (szerk.) Emotion in the Human Face. 2nd ed., p. 39–55. Cambridge University Press, Cambridge, UK (1982)

Feng, S., Kang, J.S., Kuznetsova, P., Choi, Y.: Connotation lexicon: A dash of sentiment beneath the surface meaning. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1774–1784 (2013)

Gold, D., Bexte, M., Zesch, T.: Corpus of Aspect-based Sentiment in Political Debates. In: KONVENS (2018)

Gollust, S.E., Nagler, R.H., Fowler, E.F.: The emergence of COVID-19 in the US: a public health and political communication crisis. Journal of health politics, policy and law 45(6), 967–981 (2020)

Griffiths, P.E.: What emotions really are. In: What Emotions Really Are. University of Chicago Press (2008)

Haselmayer, M., Jenny, M.: Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. Quality & quantity 51(6), 2623–2646 (2017)

Jafarian, H., Taghavi, A.H., Javaheri, A., Rawassizadeh, R.: Exploiting BERT to improve aspect-based sentiment analysis performance on Persian language. In: 2021 7th International Conference on Web Research (ICWR). pp. 5–8. IEEE (2021)

Jang, H., Shin, H.: Language-specific sentiment analysis in morphologically rich languages. In: Coling 2010: Posters. pp. 498–506 (2010)

Lakoff, G., Kovecses, Z.: The cognitive model of anger inherent. American English-In Cultural Models_in_Language and Thought-Dorothy Holland and Naomi Quinn. eds pp. 195–221 (1987)

Loukachevitch, N., Levchik, A.: Creating a general Russian sentiment lexicon. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 1171–1176 (2016)

Marcus, G.E.: Emotions in politics. Annual review of political science 3(1), 221–250 (2000)

Mariani, L.A., Gagete-Miranda, J., Retti, P.: Words can hurt: How political communication can change the pace of an epidemic. Covid Economics 12, 104–137 (2020)

Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013)

Mladenović, M., Mitrović, J., Krstev, C., Vitas, D.: Hybrid sentiment analysis framework for a morphologically rich language. Journal of Intelligent Information Systems 46(3), 599–620 (2016)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)

Plutchik, R.: The emotions: Facts, theories and a new model. American Journal of Psychology 77, 518 (1964)

Plutchik, R.: Emotion, a psychoevolutionary synthesis. Harper and Row, New York (etc.) (1980a)

Plutchik, R.: A general psychoevolutionary theory of emotion. In: Theories of emotion, pp. 3–33. Elsevier (1980b)

Plutchik, R.: A psychoevolutionary theory of emotions. Social Science Information/sur les sciences sociales 21 (4-5), 529–553 (1982)

Richards, B.: The emotional deficit in political communication. Political Communication, 21(3), 339–352 (2004)

Rufai, S.R., Bunce, C.: World leaders' usage of Twitter in response to the COVID-19 pandemic: a content analysis. Journal of public health 42(3), 510–516 (2020)

Russell, J.A.: Core affect and the psychological construction of emotion. Psychological review 110(1), 145 (2003)

Settle, J.: Moving beyond sentiment analysis: Social media and emotions in political communication. The Oxford Handbook of Networked Communication. Oxford: Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190460518.013 20 (2020)

Szabó, G.: Emotional Communication and Participation in Politics. Intersections. East European Journal of Society and Politics 6(2) (2020)

Szabó, M.K., Vincze, V., Ring, O., Üveges, I., Vit, E., Samu, F., Gulyás, A., Galántai, J., Szvetelszky, Zs., Bodor-Eranus, E.H., Takács, K.: StaffTalk: magyar nyelvű spontán beszélgetések korpusza. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2021)

Szabó, M.K., Vincze, V., Simkó, K.I., Varga, V., Hangya, V.: A Hungarian Sentiment Corpus Manually Annotated at Aspect Level. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 2873–2878. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://aclanthology.org/L16-1459

Üveges, I., Vincze, V., Ring, O., Guba, Cs.: Aspect-based emotion analysis of Hungarian parliamentary speeches. In: Ines, R., Gabriella, L., Christopher, K., Simone, P. (szerk.) Proceedings of the 2nd Workshop on Computational Linguistics for Political Text Analysis (CPSS-2022) Potsdam, Germany. University of Mannheim, University of Stuttgart (2022)

Vincze, V., Üveges, I., Szabó, M.K., Takács, K.: A magyar beszélt és írott nyelv különböző korpuszainak morfológiai és szófaji vizsgálata. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2021)

Wagner, M., Morisi, D.: Anxiety, fear, and political decision making. In: Oxford research encyclopedia of politics. Oxford University Press (2019)

Wang, Y., Croucher, S.M., Pearson, E.: National Leaders' Usage of Twitter in Response to COVID-19: A Sentiment Analysis. Frontiers in Communication 6, 732399 (2021)

Wierzbicka, A.: Emotions across languages and cultures: Diversity and universals. Cambridge university press (1999)

MILQA kérdés-válasz benchmark adatbázis

Novák Attila[1,2], Novák Borbála[1,2]

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

Kivonat Jelen cikkben bemutatott kutatás során új kérdés-válasz benchmark adatbázist hoztunk létre magyar nyelvre. A MILQA[1] adatbázis létrehozásakor nagyrészt az angol nyelvű SQuAD 2.0 elveit követtük, azonban néhány a SQuAD-nál újabb keletű angol nyelvű kérdés-válasz adatbázishoz hasonlóan bevezettünk számos a SQuAD-on túlmutató újítást: többek között igen/nem válaszokkal megjelölt eldöntendő kérdéseket, több szakaszból álló listaszerű válaszokat, hosszú válaszokat, számolást igénylő és egyéb olyan kérdéseket, ahol nem másolható ki egyszerűen a válasz a szövegből.[2]
Kulcsszavak: kérdés-válasz rendszerek, benchmark adatbázis

1. Bevezetés

A számítógépet használó laikus felhasználók részéről természetes igényként jelenik meg, hogy természetes nyelven tehessenek fel kérdéseket a rendszernek, és arra értelmes és releváns természetes nyelvű válaszokat kapjanak. A mesterséges (gépi) intelligencia Alan Turing általi korai definíciója (a később Turing-teszt néven ismertté vált „imitációs játék”)[3] is ilyen jellegű interakció köré épül: javaslata szerint intelligensnek az embertől megkülönböztethetetlen módon társalgó gépet tekinthetjük. A neurális nyelvmodellek térhódításával ez az igény a korábbiaknál sokkal inkább kielégíthetővé vált. A gépi szövegértés tesztelésének egy módja (hasonlóképpen ahhoz, ahogy az emberi szövegértést tesztelni szokták), hogy a szöveg alapján a szöveghez kapcsolódó kérdésekre kell válaszolni, és a válaszok helyességét ellenőrizzük. Angol nyelvre viszonylag jól működő kérdés-válasz rendszerek léteznek, amelyeknek alapját az alapkutatás keretében (a legtöbb esetben azonban céges kutatólaborok által finanszírozott módon) létrehozott adatbázisok

[1] A korpusz nevét a lila tehenes csoki mellett a projektet támogató Mesterséges Intelligencia Nemzeti Labor kezdőbetűi ihlették.

[2] Az adatbázist a tesztkorpusz ellenőrzését követően publikáljuk: `https://github.com/ppke-nlpg/MILQA`.

[3] Az eredeti imitációs játékban két különböző nemű szereplő feladata az volt hogy a valódi nemüktől függetlenül adott nemű személynek adják ki magukat, és a harmadik szereplőnek ki kellett találnia pusztán a megnyilatkozások tartalma alapján, hogy melyikük az, aki csal.

képezik. Jelen kutatásban olyan magyar nyelvű kérdés-válasz adatbázis létrehozása volt a célunk, amely lehetővé teszi magyar nyelvű kérdés-válasz rendszerek betanítását, illetve teljesítményük gépi kiértékelését.

2. Előzmények

A korai kérdés-válasz adatbázisok vagy nagyon kis méretűek voltak, vagy nem tartalmaztak grammatikailag kérdő mondat formájú kérdéseket, hanem ún. cloze-típusú „kérdésekből" álltak: ezek olyan kijelentő mondatok, amelyeknek egy része ki van maszkolva, és azt kell kiegészíteni a szöveg alapján. Az utóbbi erőforrásokat géppel generálták, így könnyen létrehozhatóak voltak, azonban a maszkolt részt tartalmazó mondatok egyáltalán nem hasonlítanak a valódi kérdésekhez. Az egyik legfontosabb mérföldkő a kérdés-válasz tanító-adatbázisok sorában a Stanfordi Egyetemen létrehozott angol nyelvű SQuAD adatbázis (Stanford Question Answering Dataset) volt (Rajpurkar és mtsai, 2016). Ez egy a korábbiaknál jóval nagyobb méretű, első verziójában több mint 108000 kérdés-válasz párt tartalmazó adatbázis, amelyet később a második verzióban az adott szövegrészlet alapján nem megválaszolható kérdésekkel egészítettek ki (151000 kérdés, (Rajpurkar és mtsai, 2018)). A nyilvánosan elérhető tanító és a hangolókészlet 143000 (93000 megválaszolható és 50000 nem megválaszolható) kérdést tartalmaz. A méretén túl azért tekinthető mérföldkőnek ez az erőforrás, mert egyrészt ténylegesen jólformált kérdéseket tartalmazott a korábbi ún. cloze-típusú kérdéseket tartalmazó erőforrásokkal ellentétben (pl. CNN/Daily Mail (Hermann és mtsai, 2015)), illetve nem feleletválasztós volt (mint pl. az MCTest (Richardson és mtsai, 2013) vagy a WikiQA (Yang és mtsai, 2015)). Valamint mert óriási lökést adott a kérdés-válasz rendszerek fejlesztésének.

A kérdés-válasz adatbázisok, illetve rendszerek körében megkülönböztethetünk extraktív, illetve absztraktív/generatív megközelítést. Az előző esetében a választ egyszerűen a szövegből szó szerint kiemelt rész adja (mintha szövegkiemelővel dolgoznánk, ilyen a SQuAD), az utóbbi esetében pedig ténylegesen jólformált emberi nyelven van a válasz megfogalmazva (pl. MS MARCO (Nguyen és mtsai, 2016), NarrativeQA (Kočiský és mtsai, 2018)). Ezen túl a kérdés-válasz adatbázisok egy része olyan kérdéseket tartalmaz, amelyeknek megválaszolása többlépéses következtetési láncok végrehajtását követeli meg (multi-hop/többlépcsős kérdés-válasz feladatok). Ez nemcsak a mögöttes logikai levezetés nagyobb komplexitását jelenti, hanem ez a feladattípus így túlléphet az egyedi dokumentumok, illetve szövegrészletek szintjén is, amennyiben az adott kérdés csak több különböző dokumentumban vagy szövegrészben szereplő információk kombinálásával válaszolható meg (pl. HotpotQA (Yang és mtsai, 2018), NarrativeQA (Kočiský és mtsai, 2018)).

A többlépcsős kérdés-válasz feladatok és a SQuAD esetében is az adott szövegek alapján az annotátorok feladata volt a kérdések megfogalmazása. A nagy keresőmotorokat üzemeltető cégek ezzel szemben olyan erőforrásokat hoztak létre, amelyekben a keresőmotorba beírt gyakori kérdésekből kiindulva gyűjtöttek össze hozzájuk tartozó releváns dokumentumokat, és ezeket felhasználva jelölték

ki vagy fogalmazták meg az annotátorok a válaszokat. Az előbbi extraktív típusú adathalmaz a Google keresőbe beírt kérdéseken alapuló Natural Questions (NQ, Kwiatkowski és mtsai (2019)), ahol a kontextusként használt dokumentumok a SQuAD-hoz hasonlóan Wikipedia szócikkek voltak. Az utóbbi absztraktív/generatív típusba tartozik a Microsoft Bing-kérdéseken alapuló MS MARCO QnA adathalmaza (Nguyen és mtsai, 2016). Hasonlóképpen kereséssel jöttek létre a szintén eleve létező kvíz- és műveltségi kérdéshalmazokon alapuló erőforrások is (pl. TriviaQA (Joshi és mtsai, 2017)).

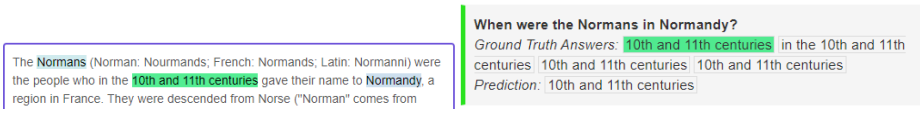
A SQuAD népszerűségének egyik forrása talán az, hogy egy olyan viszonylag leegyszerűsített modellt feltételez, amely szerint minden megválaszolható kérdésre egyetlen összefüggő szövegrész kijelölhető válaszként, és ez a SQuAD-alapú rendszerek implementációját jelentősen egyszerűsíti. Ez a megszorítás többékevésbé jól megvalósítható, ha az annotátorok eleve azt a feladatot kapják, hogy így megválaszolható kérdéseket tegyenek fel. A valóságban előforduló kérdések nem elhanyagolható része esetében azonban a válasz valamilyen lista, aminek az elemei nem feltétlenül összefüggő részt foglalnak el a szövegben. Ilyenkor az összes releváns részt magában foglaló összefüggő szakasz jelentős mennyiségű a válasz szempontjából irreleváns szövegrészt tartalmazhat. Ezért például a valóságos kérdéseken alapuló NQ adathalmazban nem korlátozták a válasz lehetőségét egyetlen összefüggő szakaszra, hanem ilyenek listája lehet a válasz (a Natural Questionsben a kérdések 6,9%-ánál nem egyelemű lista a válasz). A SQuAD esetében a kérdések kontextusa (a szövegrész, amiben a kérdésre a választ meg kell találni) viszonylag korlátozott hosszúságú: 150 és 4000 karakter közötti, átlagosan 740 karakter, ami szintén behatárolja a feladat komplexitását.

Az NQ-hoz hasonló ténylegesen feltett kérdésekből kiinduló adathalmazokban természetes módon előfordulnak eldöntendő kérdések is (Natural Questions: 2,5%). Ezekre önmagában tipikusan nem válasz a szöveg egy kijelölt részlete, hanem valamely releváns szövegrészből következik egy (bizonyos esetekben nem egyértelmű, csak valószínűsíthető) igen/nem válasz. Vannak kifejezetten csak eldöntendő kérdéseket tartalmazó adathalmazok is (pl. a szintén a Wikipedián alapuló BoolQ (Clark és mtsai, 2019), az Amazon termékértékeléssel kapcsolatos szövegeire épülő AmazonYesNo (Dzendzik és mtsai, 2019) vagy a orvosbiológiai cikkabsztraktokon alapuló PubMedQA (Jin és mtsai, 2019)). A BoolQ és az AmazonYesNo ugyanakkor jelentős átfedést mutat a NaturalQuestions, illetve az AmazonQA (Gupta és mtsai, 2019) adatbázisok eldöntendő kérdéseivel (az Amazonos erőforrások esetében lényegében részhalmaz viszony áll fenn).

Az NQ-hoz hasonlóan „természetes eredetű” orvosbiológiai témájú kérdéshalmazokban gyakran az NQ adatbázissal kapcsolatban korábban említettnél is jóval magasabb arányú a „nem SQuAD-kompatibilis” kérdések aránya. A ténylegesen praktizáló orvosok napi szakmai tevékenysége során megfogalmazott kérdéseket tartalmazó Clinical Questions Collection (CQC) adathalmaz (D'Alessandro és mtsai, 2004; Ely és mtsai, 1997, 1999) és a PubMed felhasználók által egy nap alatt megfogalmazott kérdésekből összeállított PubMed Query Log Dataset (Herskovic és mtsai, 2007) esetében például az eldöntendő kérdések aránya

28,1%-nak, a lista típusú válasszal rendelkezőké pedig 21,9%-nak adódik (Yoon és mtsai, 2022).

A lista típusú válaszok és az eldöntendő kérdések (valamint a korábban említett többlépcsős következtetést igénylő válaszok) hiánya miatt a SQuAD-típusú adatbázisokon betanítható egyetlen szövegszakaszt válaszként visszaadó extraktív rendszerek viszonylag hamar megbuknának a korábban említett Turing-teszten még abban az esetben is, ha egyébként tökéletes teljesítményt nyújtanának a maguk műfajában, mert ilyen típusú kérdésekre nem tudnának meggyőzően válaszolni. Ugyanakkor további probléma adódik abból a tényből, hogy egy adott szöveggel kapcsolatban az igazán érdekes kérdések nem azokat a szavakat használják, amelyek az eredeti szövegkörnyezetben szerepeltek (hiszen ellenkező esetben gyakran egyszerű lexikai kereséssel meg lehetne találni a választ). A SQuAD összeállítása során is törekedtek arra, hogy az annotátorok a kérdések megfogalmazásakor parafrazeálják a kérdés szövegkörnyezethez horgonyozó részét, és ne egyszerűen lemásolják azt. Ez önmagában nem feltétlenül jelent komoly problémát a kurrens előtanított nyelvmodelleken alapuló neurális modellek számára, hiszen ezek általában megfelelően absztrakt belső szemantikai reprezentációkkal bírnak ahhoz, hogy a parafrázisok ne zavarják meg őket (vagy legalábbis sokkal kevésbé, mint egy lexikai mintaillesztésen alapuló rendszert). A probléma azonban abból adódik, hogy ha a kérdésben például az eredeti kontextusban szereplőtől eltérő igét használunk, akkor sok esetben ez más vonzatkerettel jár, ami azt jelenti, hogy sokszor más esetben kellene szerepelnie a válaszban az adott kifejezésnek, mint amilyen formában az eredeti szövegben állt. A SQuAD esetében erre a problémára az volt a megoldás, hogy a prepozíciókat általában nem tették a válasz részévé, hanem csak a minimálisan szükséges lexikális tartalmat (erre utasították az annotátorokat, l. 1. ábra).



1. ábra. A SQuAD-ban az egyébként lényeges prepozíciók általában nem képezik részét a válaszoknak

Ez ugyan a lényegében izoláló nyelvnek tekinthető angol esetében az esetek nagy részében megoldja a fenti problémát, de azon az áron, hogy a rendszer többnyire „foghegyről” válaszol, és nem a pragmatikailag megfelelő alakban adja vissza/jeleníti meg a választ. Az emberek a megfelelő prepozíciót is beillesztenék a válaszba, tehát ezen a ponton is „kilóg a lóláb” a Turing-teszt szempontjából. Morfológiailag komplex (pl. agglutináló) nyelvek esetében pedig ez a megoldás nyilvánvalóan nem működik, ilyen esetekben egy SQuAD-típusú extraktív rendszer mindenképpen pragmatikailag inadekvát választ fog adni, mert az eredetileg a szövegben szereplő esetraggal, esetleg valamilyen képzős formában adja vissza a választ. Ugyanakkor ez a szövegkiemelős megjelenítés esetén mégsem jelent igazi

problémát, hiszen ebben az esetben a felhasználó nem úgy értékeli, hogy a gép „furcsán válaszolt”, hanem hogy „jól kiemelte a választ a szövegből”. Amennyiben ugyanakkor a válasz valóban válaszként jelenik meg, akkor mindenképpen célszerű továbblépni valamilyen generatív/absztraktív megoldás irányába.

Van néhány további kérdéstípus: a megszámlálást, valamilyen aritmetikai műveletek végrehajtását, illetve összehasonlítást igénylő kérdés-válasz párok *(hány, mennyi, melyik a leg... stb.),* amelyek már a minimális iskolázottságú embereknek sem jelentenek problémát, de a modelleket speciálisan fel kell készíteni az ilyen jellegű feladatok végrehajtására annak érdekében, hogy a gép ne bukjon el csúfosan. Elsősorban ilyen kérdésekre fókuszál a DROP (Dua és mtsai, 2019) kérdés-válasz adatbázis.

Egyes erőforrások valamilyen dialógus kontextusában közelítik meg a kérdés-megválaszolás problémáját. A kérdések sokszor többértelműek vagy hiányosak, és kiegészítő információra van szükség a megválaszolásukhoz. Ilyen szituációk modellezését tűzik ki célul az olyan adathalmazok, mint pl. a ShARC (Saeidi és mtsai, 2018).

2.1. Nem angol nyelvű erőforrások

A korábban említett minden kérdés-válasz adatbázis (illetve ezek mellett még számtalan másik) angol nyelvű. Ugyanakkor számos más nyelvre is adaptálták a bemutatott módszereket, illetve többnyelvű kérdés-válasz adatbázisok is születtek.

Viszonylag sok és nagy méretű kínai nyelvű adathalmaz született. A legismertebb a DuReader (He és mtsai, 2018). Ez és számos későbbi hozzá kapcsolódó erőforrás az NQ-hoz és az MS MARCO-hoz hasonlóan a kínai Baidu keresőmotor-kérdéseken és a magyar viszonylatban talán leginkább a gyakorikerdesek.hu-hoz hasonlítható, de annál sokkal kiterjedtebb, és hivatalos entitások, például cégek által szolgáltatott tartalmakat is tartalmazó Baidu Zhidaón alapuló kérdés-válasz adatbázisa.

A SQuAD módszertana alapján készült francia (FQuAD 2.0, (Heinrich és mtsai, 2022), majdnem 80000 kérdés), koreai (KorQuAD 2.0 (Youngmin Kim, 2020) 100000 kérdés), orosz (SberQuAD (Efimov és mtsai, 2020)) és német (GermanQuAD (Möller és mtsai, 2021) kb. 14000 kérdés) erőforrás is. Az XQuAD (Artetxe és mtsai, 2019) a SQuAD 1.1 hangolóhalmazából (dev. set) 240 bekezdéshez kapcsolódó 1190 kérdés-válasz pár professzionális fordítók által készített fordítását tartalmazza 10 nyelven.

Az angol mellett hat másik nyelvet lefedő MLQA benchmark adatbázis (Lewis és mtsai (2020); angolra 12 ezer körüli, a többi nyelvre 5-6 ezer kérdés-válasz pár) kváziekvivalens Wikipédia-mondatok köré épül, amelyekhez a kérdéseket fordítók fordították angolról. A SQuAD-ot számos nyelvre (pl. koreai, hindi, japán, spanyol, cseh, francia, illetve az MLQA adathalmazban szereplő nyelvekre) gépifordították.

11 tipológiai szempontból változatos nyelvet fed le a TyDi QA adathalmaz (Clark és mtsai (2020); összesen 200 ezer kérdés-válasz pár), amely szintén a Wikipédián alapul. A kérdéseket csak a szócikkek bevezető szakasza ismeretében

fogalmazták meg, de bármit lehetett a témával kapcsolatban kérdezni. Így TyDi QA-ben megfogalmazott kérdések nagyobb részére nincs válasz, ahol viszont van, ott a módszer jól garantálja, hogy a kérdés másképp van megfogalmazva, mint a válasz.

2.2. Az első magyar extraktív kérdés-válasz-adatbázis

A SQuAD mintájára készült az Arcanum Adatbázis Kiadó fejlesztésében az AQuAD erőforrás, amelyben elsősorban a SQuAD-ban szereplő szócikkek egy részének magyar megfelelőjéhez írtak kérdéseket, illetve ezt kiegészítették 28 – harmadrészt magyar tematikájú – további magyar Wikipédia szócikkel (összesen 308 szócikket dolgoztak fel). A módszertanból kifolyólag az erőforrásban szereplő szövegek egy része fordítás. A szócikkekből csak a legalább ezer karakter hosszú szakaszokat tartották meg (a szócikket a belső címek osztják szakaszokra). Az eredeti SQuAD-dal ellentétben az egységek itt nem bekezdések, hanem teljes szakaszok, ebből kifolyólag az annotált egységek hosszának eloszlása más, mint a SQuAD-ban (2. ábra). Az annotációt három annotátor végezte. Az AQuAD-ban 8463 megválaszolható kérdés szerepel. Emellett az eredeti adatbázis 8368 nem megválaszolható kérdést is tartalmaz, amelyeknek többségét úgy hozták létre, hogy a megválaszolható kérdésekben valamelyik kifejezést lecserélték egy másikra, amely az esetek többségében egyébként a szövegben nem szerepel. Így a nem megválaszolható kérdések viszonylag könnyen felismerhetőek. Az AQuAD eredetileg az Arcanum belső adatbázisa volt, nem publikálták és publikáció sem készült róla. A jelen munkát magába foglaló projekt keretében azonban a megválaszolható kérdéseket tartalmazó részét a konzorcium megvásárolta, és így ez az erőforrás is szélesebb körben hozzáférhetővé válik a kutatás számára.

2.3. Egyéb magyar nyelvű előzmények

Tágabb értelemben az itt bemutatott kutatás előzményének tekinthető egyrészt a HuLU benchmark (Ligeti-Nagy és mtsai, 2022), amely jelenleg 5 az angol GLUE, illetve SuperGLUE benchmarkok által ihletett vagy azokból fordított magyar nyelvű gépi szövegértési feladatot tartalmaz, másrészt a Novák és mtsai (2019) cikkben bemutatott magyar nyelvű kérdezőrendszer.

3. A MILQA magyar kérdés-válasz adatbázis

Jelen cikkben bemutatott kutatás során egy újabb extraktív kérdés-válasz benchmark adatbázist hoztunk létre magyar nyelvre. Az adatbázis létrehozásakor nagyrészt a SQuAD 2.0 elveit követtük, azonban hasonlóan néhány a 2. szakaszban említett a SQuAD-nál újabb keletű angol nyelvű kérdés-válasz adatbázishoz (Natural Questions, MS MARCO, DROP) bevezettünk számos olyan a SQuAD-hoz képest új kérdés-válasz típust, amely nehezebb, de realisztikusabb feladatokat tartalmaz.

A SQuAD 2.0 nyomán az alábbiak jellemzik a korpuszt:

- jó minőségű Wikipedia szócikkek szolgálnak a kérdések kontextusaként,
- tényszerű (nem vélemény jellegű) kérdések szerepelnek benne,
- olyan kérdéseket is tartalmaz, amelyekre nincs válasz az adott szövegben,
- az eredeti szövegben jelöltük az adott kérdésre vonatkozó lehető legrövidebb választ (ha van),
- a kérdések megfogalmazásakor parafrazeáltuk az eredeti szöveget, így lexikális kereséssel a legtöbb esetben nem található meg a válasz,
- a kérdések nem csak az adott szöveg kontextusában értelmezhetőek, hanem önállóan is megállják a helyüket (pl. nem tartalmaznak nem kötött névmásokat).

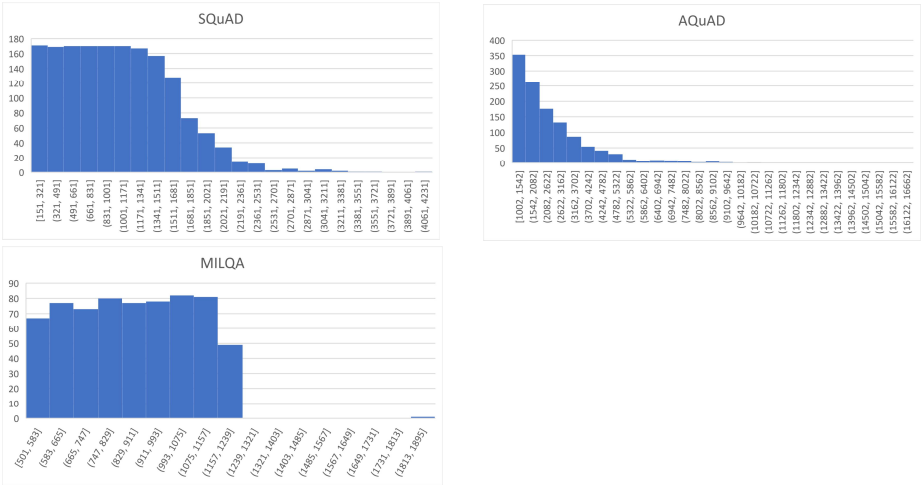
A SQuAD-hoz képest az alábbi újításokat vezettük be. A speciális kérdéstípusok az adatbázisban meg vannak jelölve.

- Egynél több rövid válasz is lehet az adott szövegben az adott kérdésre (a megválaszolt kérdések kb. 8,5%-a).
- A rövid válasz mellett hosszú választ is adtunk, amely a kérdés megválaszolásához lényeges összes körülményt magában foglalja (min. 1 tagmondat, sokszor több mondat).
- Tartalmaz eldöntendő kérdéseket (kb. 9%). Itt a lényeges körülményeket tartalmazó hosszú válasz mellett igen/nem válasz is meg van adva (ha van egyértelmű válasz).
- A „nincs válasz”-kérdések (a kérdések kb. 28,3%-a) az adott témához kapcsolódó releváns kérdések, nem „van válasz”-típusú kérdésekből helyettesítéssel generált kérdések.
- Csak megszámlálás vagy aritmetikai műveletek elvégzése után megválaszolható kérdések is vannak (hasonlóan a DROP adatbázishoz, ezek nehéz feladatot jelentenek a jelenlegi modellek számára). A számítások felsorolt elemek összeszámolására, dátumok, időtartamok és egyéb mennyiségek egyszerű aritmetikai műveletekkel való kiszámítására vonatkoznak.
- A „nincs válasz”-kérdések egy része „beugratós” kérdés, ahol az anyanyelvi beszélők nagy része kiolvasna egy választ a szövegből gyakran téves default feltételezések alapján. Ezeket az eseteket külön megjelöltük, a feltételezhető választ is megadva.
- Ha a szövegben szereplő kifejezés formailag nem felel meg annak, ahogy az adott kérdésre válaszolni kellene (pl. nem megfelelő az esetrag), az annotátorok megadták a válasznak a kérdés kontextusában megfelelő formáját.

3.1. A korpusz létrehozása

Az adatbázis alapját képező adathalmaz létrehozásához a magyar Wikipédiából kiválasztottuk a kiemeltnek, illetve jónak minősített szócikkeket, és a hozzájuk tartozó 2016 eleje és 2021 vége közötti megtekintési statisztika alapján sorba rendeztük őket. Ezek közül az AQuAD-ban nem szereplő szócikkeket vettük be a listába, amelyből az annotátorok választották ki az annotálandó szócikkeket. Minden szócikk első szakaszát, és ezen kívül maximum 10 legalább 500

karakteres véletlenszerűen kiválasztott szakaszt használtunk fel. A SQuAD-hoz hasonlóan bekezdések voltak az egységek, de az 500 karakternél rövidebb bekezdéseket összevontuk, és a kb. 1200 karakternél hosszabbakat kihagytuk (ekkora szövegrészeket lehetett jól áttekinthetően megjeleníteni az annotálófelületen). A szöveghosszak eloszlását l. a 2 ábrán.

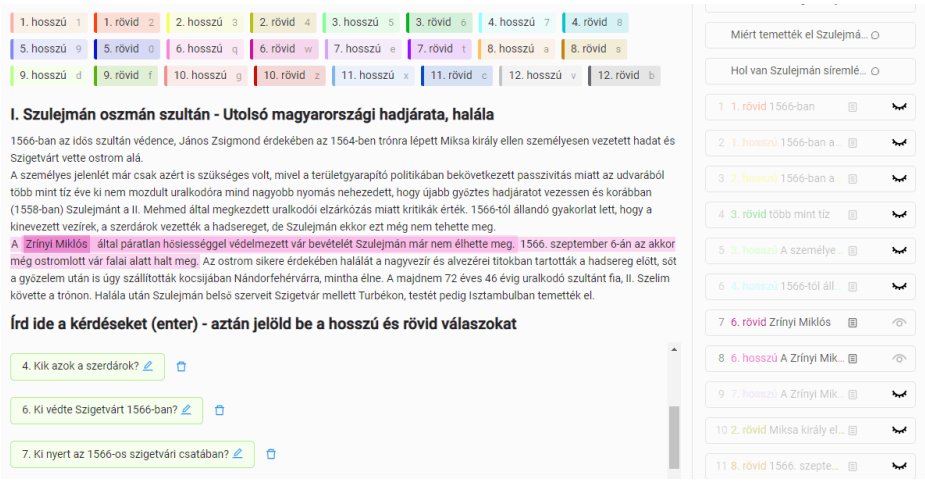


2. ábra. A szöveghosszak eloszlása a SQuAD, AQuAD és MILQA kérdés-válasz korpuszokban.

Az annotálófelületet a Label Studio nyílt forráskódú webes annotálóplatform 1.4 verziójának testreszabásával hoztuk létre. Viszonylag komplex feladatot jelentett a felület alkalmassá tétele arra, hogy mind kérdések feltevését, mind az azokhoz tartozó válaszok bejelölését, mind a speciális kérdés- és választípusok megjelölését áttekinthető módon el lehessen végezni, de sikerült az annotátorok számára viszonylag jól használható felhasználói felületet és workflow-t összeállítani (3. ábra). Az annotálórendszer az előrehaladásról folyamatos statisztikát biztosított az annotátorok számára, és az adott annotált szövegrészből generált kérdés-válasz lista formájában is ellenőrizhették, hogy a válaszok bejelölése valóban a szándékuknak megfelelően sikerült-e. Az annotációt öt annotátor készítette. A később újraszerkesztett problémásabb esetektől eltekintve a Label Studio által tárolt szerkesztési idő alapján jól becsülhető a munka időigénye: kérdésenként nagyjából 85 másodpercet vett igénybe a kérdések megfogalmazása és a válaszok bejelölése.

A korpusz egy 36 szócikkből álló 2391 kérdést (köztük 1751 megválaszolható kérdést) tartalmazó részét (nagyjából a korpusz 10%-át) teszt/hangolókészlet számára leválasztottuk, és ehhez a részhez két független annotáció készült. A kérdésírást nem igénylő annotációs munka gyorsabban haladt: átlagosan 46 má-

sodpercet vett igénybe a hosszú és a rövid válaszok bejelölése. A tesztkorpusz ellenőrzése jelenleg folyamatban van.



3. ábra. A korpusz annotálásához használt webes felület

3.2. A korpusz jellemzői

Az adatbázis összesen 23700 (17000 megválaszolható és 6700 nem megválaszolható) kérdést tartalmaz. 142 Wikipédia-szócikkhez születtek kérdések. Az 1. táblázatban összefoglaltuk a speciális kérdés-, illetve választípusok előfordulását a korpuszban.

9,20% eldöntendő kérdés. Ezeknek 7,65%-a esetében nincs egyértelmű igen/nem válasz, hanem a szövegből az derül ki, hogy az adott kérdéssel kapcsolatban megoszlanak a vélemények, vegyes az eredmény, vagy bizonytalanság van. Ugyanakkor ez nem az egyértelmű *Nincs válasz* esete, amikor a szöveg egyáltalán nem válaszol a kérdésre, hanem a szövegből konkrétan az derül ki, hogy a világ az adott kérdés szempontjából nem fehér-fekete. Az eldöntendő kérdések esetében a korpuszban szereplő annotáció azért lényeges, mert a megjelölt szövegrészekből következik a válasz. Az eldöntendő kérdés, mint kérdéstípus önmagában egyébként nem SQuAD-inkompatibilis: az eredeti SQuAD-ban vannak eldöntendő kérdések, amelyeket úgy fogalmaztak meg, hogy szép extraktív választ lehessen rájuk adni. A MILQA-ban újdonság, hogy explicit jelölés szerepel benne a kérdéstípusra, és a válasz igen/nem voltára. A „nincs válasz” típusú kérdéseknek is mintegy 9%-a eldöntendő kérdés.

Érdekes, hogy miután az annotációs környezet lehetővé tette, hogy a SQuAD-ban szokásos megszorításoktól (hogy csak egy összefüggő extraktív válasz lehet) mentesen dolgozzanak az annotátorok, azoknak a kérdéseknek, amelyekre van

Típus	eset	arány
Van válasz	**16992**	71,67%
. Eldöntendő	1621	9,20%
. . Igen	859	52,99%
. . Nem	638	39,36%
. . Homályos	124	7,65%
. Nem extraktív a válasz	4452	26,20%
. Aritmetika	427	2,51%
. Lista	1455	8,56%
. Nem SQuAD-komp.	5203	30,62%
Nincs válasz	**6716**	28,33%
. Trükkös nincs válasz	629	9,37%
Összesen	**23708**	100

1. táblázat. A kérdés-/választípusok megoszlása a MILQA korpuszban. A beljebb kezdve szedett altípusok esetében az arány oszlop az adott főtípuson belüli arányt adja meg.

válasz a szövegben, több mint 30%-a nem SQuAD-kompatibilisnak adódott (ha az eldöntendő kérdéseket nem soroljuk ide).

A (megválaszolt) kérdések 26,2%-a nem extraktív: az adott kérdésre más lenne a válasz természetes formája, mint ami a szövegben szerepel (pl. más esetben kellene állnia az adott kifejezésnek ahhoz, hogy adekvát válasz legyen a kérdésre). 2,51%-uk megválaszolásához számolni kell (a DROP adatbázisban szereplőkhöz hasonló jellegű kérdések; értelemszerűen ezeknél sem lehet szövegből kimásolni a választ, tehát ezek benne vannak az előbbi 26,2%-ban). 8,56% esetében pedig a SQuAD „egyetlen összefüggő szakasz a válasz” feltételezése nem teljesül (ezen belül persze szintén lehetnek még ragozási problémák is).

A „nincs válasz” kérdések 9,37%-a trükkös. Ezeknél valamilyen ökölszabály jellegű feltételezések mentén hajlamosak lehetünk levezetni egy választ (akár még számolást is végezve), aminek eredményéről viszont könnyen bebizonyosodhatna, hogy téves. Például a Normandiai partraszállásról szóló szócikk egy adott bekezdésében abból, hogy a partraszállásban résztvevő flottaegységeknek három parancsnoka volt, arra következtethetnénk, hogy három flottaegység volt (valójában azonban csak kettő, és volt egy főparancsnok).

A 2. táblázatban a leggyakoribb száz kérdőszavas kifejezést és ezek eloszlását mutatjuk be. A leggyakoribb kérdések időpontokra, okokra, mennyiségekre, helyekre és az alanyra kérdeznek rá.

4. Összefoglalás

Cikkünkben egy új magyar nyelvű kérdés-válasz benchmark adatbázist mutattunk be. A MILQA több szempontból túlmutat a SQuAD-ból ismert egy rövid összefüggő extraktív választ tartalmazó kérdésekre szorítkozó adatbázisokon. Igen/nem válaszokkal megjelölt eldöntendő kérdéseket, több szakaszból álló

mikor	9,78%	hová	0,37%	milyen szerepet	0,18%	mikorra	0,10%	milyen néven	0,08%
miért	7,81%	mi történt	0,36%	milyen a ...	0,18%	milyen viszonyban	0,10%	mit csinált	0,08%
mi	7,46%	kit	0,35%	mi az oka	0,17%	melyek voltak	0,10%	mit tett	0,08%
hány	7,17%	miben	0,35%	mennyire	0,17%	mikortól	0,10%	melyik	0,08%
ki	6,92%	hányadik	0,33%	milyen anyagból	0,17%	hogy nevezték	0,09%	melyik csapat	0,08%
hol	5,79%	melyik országban	0,33%	milyen volt	0,17%	hogyan lehet	0,09%	mennyibe	0,08%
mit	3,16%	melyik a ...	0,32%	mennyi idő	0,16%	kiknek	0,09%	hogyan kell	0,07%
kik	1,77%	melyik városban	0,31%	melyek a ...	0,15%	milyen stílusban	0,09%	melyik az ...	0,07%
honnan	0,94%	hányan	0,28%	kitől	0,14%	melyik magyar	0,09%	melyik napon	0,07%
mivel	0,85%	meddig	0,28%	kivel	0,14%	mennyit	0,09%	milyen célból	0,07%
melyik évben	0,76%	mekkora a ...	0,27%	milyen magas	0,14%	milyen módon	0,09%	hogy mondják	0,07%
kinek	0,68%	milyen színű	0,24%	hogy néz ki	0,13%	milyen típusú	0,09%	melyik cég	0,07%
hogy hívják	0,64%	mennyi ideig	0,22%	hogyan halt meg	0,12%	hova	0,08%	melyik város	0,07%
mire	0,62%	melyik volt	0,22%	mennyivel	0,12%	melyik ország	0,08%	melyik évszázadban	0,07%
minek	0,62%	milyen nemzetiségű	0,22%	milyen nyelven	0,11%	mi volt az oka	0,08%	milyen gyakran	0,07%
miből	0,60%	mi történik	0,21%	milyen állatok	0,11%	milyen funkciót	0,08%	melyik folyó	0,06%
mik	0,58%	mekkora volt	0,20%	milyen pozíciót	0,11%	milyen messze	0,08%	mennyi pénzt	0,06%
hogy hívták	0,56%	hányszor	0,19%	milyen stílusú	0,11%	milyen vallású	0,08%	mennyi volt	0,06%
miről	0,56%	milyen hosszú	0,19%	kiket	0,10%	milyen	0,08%	merre	0,06%
mióta	0,38%	mitől	0,19%	kiről	0,10%	milyen alakú	0,08%	milyen az ...	0,06%

2. táblázat. A leggyakoribb száz kérdőszavas kérdéstípus a MILQA korpuszban (az összes kérdés arányában).

listaszerű válaszokat, hosszú válaszokat, számolást igénylő és egyéb olyan kérdéseket is tartalmaz, ahol nem másolható ki egyszerűen a válasz a szövegből. Az adatbázist a tesztkorpusz ellenőrzése után közzétesszük.

Köszönetnyilvánítás

Jelen kutatás a Nemzeti Kutatási Fejlesztési és Innovációs Alap által biztosított támogatással az FK 125217 és a PD 125216 számú projekt keretében az FK 17 és a PD 17 pályázati program valamint az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatásával a Mesterséges Intelligencia Nemzeti Laboratórium keretében valósult meg.

Hivatkozások

Artetxe, M., Ruder, S., Yogatama, D.: On the cross-lingual transferability of monolingual representations. CoRR abs/1910.11856 (2019)

Clark, C., Lee, K., Chang, M.W., Kwiatkowski, T., Collins, M., Toutanova, K.: BoolQ: Exploring the surprising difficulty of natural yes/no questions. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 2924–2936. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), `https://aclanthology.org/N19-1300`

Clark, J.H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., Palomaki, J.: TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics 8, 454–470 (2020), `https://aclanthology.org/2020.tacl-1.30`

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 2368–2378. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1246

Dzendzik, D., Vogel, C., Foster, J.: Is it dish washer safe? Automatically answering „yes/no" questions using customer reviews. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop. pp. 1–6. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-3001

D'Alessandro, D.M., Kreiter, C.D., Peterson, M.W.: An evaluation of information-seeking behaviors of general pediatricians. Pediatrics 113(1), 64–69 (2004)

Efimov, P., Chertok, A., Boytsov, L., Braslavski, P.: SberQuAD – Russian reading comprehension dataset: Description and analysis. In: Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., Ferro, N. (szerk.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. pp. 3–15. Springer International Publishing, Cham (2020)

Ely, J.W., Osheroff, J.A., Ebell, M.H., Bergus, G.R., Levy, B.T., Chambliss, M.L., Evans, E.R.: Analysis of questions asked by family doctors regarding patient care. Bmj 319(7206), 358–361 (1999)

Ely, J.W., Osheroff, J.A., Ferguson, K.J., Chambliss, M.L., Vinson, D.C., Moore, J.L.: Lifelong self-directed learning using a computer database of clinical questions. Journal of family practice 45(5), 382–390 (1997)

Gupta, M., Kulkarni, N., Chanda, R., Rayasam, A., Lipton, Z.C.: AmazonQA: A review-based question answering task. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 4996–5002. International Joint Conferences on Artificial Intelligence Organization (7 2019), https://doi.org/10.24963/ijcai.2019/694

He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., Liu, Y., Wang, Y., Wu, H., She, Q., Liu, X., Wu, T., Wang, H.: DuReader: a Chinese machine reading comprehension dataset from real-world applications. In: Proceedings of the Workshop on Machine Reading for Question Answering. pp. 37–46. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/W18-2605

Heinrich, Q., Viaud, G., Belblidia, W.: FQuAD2.0: French question answering and learning when you don't know. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 2205–2214. European Language Resources Association, Marseille, France (Jun 2022), https://aclanthology.org/2022.lrec-1.237

Hermann, K.M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Cortes, C.,

Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (szerk.) Advances in Neural Information Processing Systems. pp. 1693–1701 (2015), http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend

Herskovic, J.R., Tanaka, L.Y., Hersh, W., Bernstam, E.V.: A day in the life of PubMed: analysis of a typical day's query log. Journal of the American Medical Informatics Association 14(2), 212–220 (2007)

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., Lu, X.: PubMedQA: A dataset for biomedical research question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2567–2577. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1259

Joshi, M., Choi, E., Weld, D., Zettlemoyer, L.: TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1601–1611. Association for Computational Linguistics, Vancouver, Canada (Jul 2017), https://aclanthology.org/P17-1147

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K.M., Melis, G., Grefenstette, E.: The NarrativeQA reading comprehension challenge. Transactions of the Association for Computational Linguistics 6, 317–328 (2018), https://aclanthology.org/Q18-1023

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.W., Dai, A.M., Uszkoreit, J., Le, Q., Petrov, S.: Natural Questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics 7, 452–466 (2019), https://aclanthology.org/Q19-1026

Lewis, P., Oguz, B., Rinott, R., Riedel, S., Schwenk, H.: MLQA: Evaluating cross-lingual extractive question answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7315–7330. Association for Computational Linguistics, Online (Jul 2020), https://aclanthology.org/2020.acl-main.653

Ligeti-Nagy, N., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Laki, L.J., Vadász, N., Yang, Z.G., Váradi, T.: Hulu: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 431–446. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Möller, T., Risch, J., Pietsch, M.: GermanQuAD and GermanDPR: Improving non-English question answering and passage retrieval. In: Proceedings of the 3rd Workshop on Machine Reading for Question Answering. pp. 42–50. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021), https://aclanthology.org/2021.mrqa-1.4

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G. (szerk.) Proceedings of the Workshop on Cognitive Computation: Integrating neural

and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016. CEUR Workshop Proceedings, vol. 1773. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

Novák, A., Laki, L.J., Novák, B., Dömötör, A., Ligeti-Nagy, N., Kalivoda, Á.: Egy magyar nyelvű kérdezőrendszer. In: XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 83–95. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2019)

Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for SQuAD. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 784–789. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/P18-2124

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016), https://aclanthology.org/D16-1264

Richardson, M., Burges, C.J., Renshaw, E.: MCTest: A challenge dataset for the open-domain machine comprehension of text. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 193–203. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), https://aclanthology.org/D13-1020

Saeidi, M., Bartolo, M., Lewis, P., Singh, S., Rocktäschel, T., Sheldon, M., Bouchard, G., Riedel, S.: Interpretation of natural language rules in conversational machine reading. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2087–2097. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1233

Yang, Y., Yih, W.t., Meek, C.: WikiQA: A challenge dataset for open-domain question answering. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2013–2018. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), https://aclanthology.org/D15-1237

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., Manning, C.D.: HotpotQA: A dataset for diverse, explainable multi-hop question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2369–2380. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1259

Yoon, W., Jackson, R., Lagerberg, A., Kang, J.: Sequence tagging for biomedical extractive question answering. Bioinformatics 38(15), 3794–3801 (06 2022), https://doi.org/10.1093/bioinformatics/btac397

Youngmin Kim, S.L.L.P.K.: KorQuAD 2.0: Korean QA Dataset for Web Document Machine Comprehension. Journal of KIISE 47, 577–586 (2020), http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09353166

Hát te mekkorát nőttél! – A HuLU első életéve új adatbázisokkal és webszolgáltatással

Ligeti-Nagy Noémi[1], Héja Enikő[1], Laki László János[1], Takács Dávid[2], Yang Zijian Győző[1], Váradi Tamás[1]

[1]Nyelvtudományi kutatóközpont
1068 Budapest, Benczúr u. 33.
{vezeteknev.keresztnev}@nytud.hu
[2]takdavid@nytud.hu

Kivonat Az egyre nagyobb számban megjelenő neurális nyelvmodellek jelentőségének felbecsüléséhez elengedhetetlen valamilyen eszköz, amellyel ezek mérhetők és összehasonlíthatók. A nyelvtechnológiában ez a mérőeszköz a *benchmark* adatbázis, amelyből számtalan elérhető már az angol nyelvre, és több más nyelvre is. A magyarra 2022-ben jelent meg a HuLU *benchmark* adatbázis-készlet (Ligeti-Nagy és mtsai, 2022), melynek célja, hogy egyre bővülő gyűjteményként a nyelvmodellek magyarnyelvtudásának mérésére szolgáljon. Jelen cikkünkben bemutatunk két olyan új adatbázist, amelyekkel a HuLU-t bővítettük: egyrészt a Recognizing Textual Entailment (RTE) *shared taskok* anyagából válogatott RTE korpusz magyar fordítását, másrészt a CommitmentBank mintájára létrehozott magyar HuCommitmentBank korpuszt. Ezen kívül bemutatjuk a HuLU használatát megkönnyítő és kiteljesítő webszolgáltatást, amely lehetővé teszi a modellek eredményeinek gyors kiértékelését, és a feltöltött eredmények látványos összehasonlítását. Ezt kihasználva tanulmányunkban bemutatjuk néhány magyarul is tudó nyelvmodellnek a HuLU korpuszain elért eredményeit.
Kulcsszavak: *benchmarking*, NLI, nyelvmodellek, kiértékelés

1. Bevezetés

Az elmúlt évtized a neurális nyelvmodellek térnyeréséről szólt a nyelvtechnológiában: évről évre egyre több és egyre nagyobb architektúra kerül bemutatásra, melyeknek a teljesítményét a szintén egyre szélesebb körben elérhető, és egyre változatosabb *benchmark* adatbázisokon hasonlítják össze. Az angol GLUE és SuperGLUE *benchmarkokat* (Wang és mtsai, 2018, 2020) hamarosan követte a francia (FLUE, Le és mtsai, 2020), a spanyol (GLUES, Cañete és mtsai, 2020), vagy az orosz (Russian SuperGLUE, Shavrina és mtsai, 2020) megfelelőjük, illetve az XGLUE, amely többnyelvű modellek kiértékelésére fókuszál (Liang és mtsai, 2020). Mindeközben természetesen fokozatosan fény derült ezeknek a statikus *benchmarkoknak* a hiányosságaira is: hamar devalválódnak, a modellek hamar túltanulnak rajtuk, gyakran hibás annotálást tartalmaznak, és az értékelési metrikák sok esetben tökéletlenek, nem egyértelműek. A Dynabench (Kiela

és mtsai, 2021) célja, hogy dinamikusan, a modellek eredményeit és az emberi munkát fokozatosan beépítve hozzon létre *benchmark* adatbázisokat, amelyekkel a statikus elődök hibái kiküszöbölhetők.

Kis késéssel ugyan, de megkezdődött a jelentősebb neurális architektúrák magyar korpuszokon történő előtanítása is (Nemeskey, 2021; Feldmann és mtsai, 2021; Yang, 2022). A jövőben várhatóan még több, magyarra tanított modell jelenik majd meg, melyeknek a nyelvértését ugyanúgy szükséges mérni, összehasonlítani. Ezért időszerűvé vált egy magyar korpuszgyűjtemény, amely megfelel a *benchmarking* céljaira. Ebből a megfontolásból jött létre a HuLU (Ligeti-Nagy és mtsai, 2022), amelynek a kiindulópontját a *multi-task* jellegű GLUE, és a SuperGLUE jelentette. Mivel magyarra nem álltak rendelkezésre olyan adatbázisok, amelyeket be lehet emelni egy *benchmark-gyűjteménybe*, a HuLU összeállításához több új, specifikus, egy adott feladatra koncentráló korpuszt készítettünk.

Jelen cikkben a HuLU részeként prezentálunk két új adatbázist, illetve bemutatjuk a HuLUhoz készített webszolgáltatást, amellyel könnyen kiértékelhető és összehasonlítható a különböző, magyarul (is) tudó nyelvmodellek teljesítménye.

2. Előzmények

A HuLU 2022-ben bemutatott állapotában négy adatbázisból állt (Ligeti-Nagy és mtsai, 2022). A HuCOLA, az elfogadhatósági ítéletek korpusza 9 944 példát tartalmaz, melyeket manuálisan gyűjtöttünk magyar nyelvészeti szakirodalmi munkákból, majd 4-4 annotátor annotálása után a többségi döntés alapján címkéztek.[1].

A HuCoPA ok-okozati összefüggések feltárását célozza. Az angolból fordított korpuszban 1000 példa található, amelyekben egy premisszához két alternatíva közül kell kiválasztani azt, amelyik valószínűbben ok-okozati viszonyban áll a premisszával. A kérdések egy részében az okot, egy másik részében az okozatot kell az alternatívák közül kiválasztani. A CoPA 1000 kérdésének gépi fordítóval történő lefordítása után annotátorok ellenőrizték és javították a fordítást, a fluenciára törekedve. Egy-egy annotátor megjelölte a helyes választ a kérdésre. Ha az annotátor döntése eltért az eredeti címkétől, akkor kézzel lett ellenőrizve és javítva az adott példa.[2]

A HuSST szentimentkorpusz az ún. SST-5 adathalmaz fordításával és újraannotálásával készült. 11 683 mondatot tartalmaz, melyek negatív-semleges-pozitív címkékkel vannak ellátva.[3].

A magyar Winograd-sémák a referencia-feloldás feladatát célozzák. A Winograd-séma olyan mondatpár, amely csak egy vagy két szóban különbözik egymástól, és olyan kétértelműséget tartalmaz, amely a két mondatban ellentétesen oldódik fel, és amelynek feloldásához világismeret szükséges (Levesque és mtsai,

[1] `https://github.com/nytud/HuCOLA`
[2] `https://github.com/nytud/HuCoPA`
[3] `https://github.com/nytud/HuSST`

2012). Az eredeti adathalmazt 150 mondatpár alkotta, melyekből a fordítás és a validálás után 122 magyar séma lett (Vadász és Ligeti-Nagy, 2022).[4]

3. Új adatbázisok

A HuLU céljaink szerint egy folyamatosan bővülő korpuszkészlet, amelybe újabb és újabb, *benchmarkolás* céljára megfelelő korpuszok kerülhetnek be. Első, nyilvánosságra hozott formájában négy elkészült, annotált, publikált adatbázist tartalmazott (HuCoLA, HuCOPA, HuSST és a Winograd-sémák).

Mind a GLUE, mind a SuperGLUE jelentős részét képezik a *natural language inference* (NLI) feladatot képviselő adatbázisok. Az NLI-ben a feladat annak meghatározása, hogy egy adott premissza alapján a hipotézis igaz (azaz következik a premisszából), hamis (azaz ellentmond a kettő egymásnak), vagy pedig semleges-e (a premissza alapján nem dönthető el, hogy a hipotézis igaz-e). A két legjelentősebb és legtöbbet idézett NLI adatbázis az SNLI (Bowman és mtsai, 2015) és a MultiNLI (Williams és mtsai, 2018). Az SNLI 550 000 példát tartalmaz, a MultiNLI pedig 433 000 példát, 10 különböző domainből. A MultiNLI a GLUE része. Ezen kívül a nagy *benchmark* gyűjteményekben következtetési feladatként fogalmaznak meg több, eredetileg más formájú feladatot is: így a GLUE-ban következtetési feladatként jelenik meg a SQuAD (Rajpurkar és mtsai, 2016) valamint a Winográd-sémák (Levesque és mtsai, 2012), a SuperGLUE-ban pedig a CommitmentBank (de Marneffe és mtsai, 2019) is. A HuLU első formájában igazi következtetési feladatot nem tartalmazott. Ezt egyrészt a Winograd-sémák NLI formátumra hozásával, másrészt két új adatbázis elkészítésével orvosoljuk.

3.1. HuWNLI

A HuLU jelenleg a Winograd-sémákból konvertált HuWNLI adatbázist tartalmazza koreferencia-feloldást tesztelő korpuszként, melyben az anaforafeloldás következtetési feladatként fogalmazódik meg. A HuLU részét képező NLI-formátumot úgy hoztuk létre, hogy a kétértelmű névmást a sémákban szereplő minden lehetséges referenssel helyettesítettük (a módszert a GLUE-t bemutató tanulmány írja le, ld. Wang és mtsai, 2018). A sémákból származtatott mondatpárok halmazát kibővítettük azoknak a mondatpároknak a fordításával, amelyek – a Winograd-sémák mondataival együtt – a GLUE WNLI-adatbázisát alkotják. A korpuszt három részre osztva adjuk közre (zárójelben az adott halmaz elemszáma): tanítóhalmaz (562), validáló halmaz (59) és teszthalmaz (134). A felosztások követik a GLUE WNLI felosztását, de kevesebb példát tartalmaznak, mivel több mondatpárt el kellett vetnünk, mert magyarra nem voltak lefordíthatók. A teszthalmaz mondatpárjai mind a GLUE WNLI teszthalmazának lefordított példái.

[4] `https://github.com/nytud/HuWSC`

3.2. HuRTE

A GLUE-ba beválogatták a Recognizing Textual Entailment (RTE) kihívás adathalmazainak egy részét: az RTE1-ből (Dagan és mtsai, 2006), az RTE2-ből (Bar-Haim és mtsai, 2006), az RTE3-ból (Giampiccolo és mtsai, 2007) és az RTE5-ből (Bentivogli és mtsai, 2009) gyűjtöttek hírszövegekből és Wikipedia szócikkekből származó példákat. Ezekben egy (néha többmondatos) premisszáról és egy egymondatos hipotézisről kell eldönteni, hogy ez utóbbi következik-e az elsőből vagy sem. A feladat bináris címkézés, így az eredetileg hármas osztályozású példáknál a *semleges* és *ellentmondás* címkéket összevonták a konzisztencia érdekében. Az RTE adathalmaz a SuperGLUE-nak is része maradt.

Az RTE adathalmazok GLUE-ba beválogatott részét a HuLU többi korpuszának létrehozásához is használt gépi fordítóval fordíttattuk magyarra.[5] Ezt követően az így előállt 5 797 példát (amely nagyjából 18 000 mondatot jelent) annotátorok ellenőrizték és javították, a fluenciára törekedve.[6] Mivel az eredeti benchmarkban a teszthalmazhoz nem voltak megadva a helyes címkék, ezek fordításait címkéztetnünk kellett. Így az 5 797 példából 3000-et (a teszthalmaz példáit) 3-3 annotátor címkézett. Az egyetértés a 3000 példán 0.61 (Fleiss's κ). A HuRTE teszthalmazába csak azokat az eseteket válogattuk be, ahol teljes volt az egyetértés az annotátorok között (2123 példa).[7] Ezzel a lépéssel a tesztanyag tisztaságát kívánjuk biztosítani, hogy olyan példákon mérjük a nyelvmodellek tudását, amelyeknél nagyobb bizonyossággal tudjuk a válaszát helyesnek vagy helytelennek ítélni. Bár a nyelvmodellek kiértékelésének módszertanáról jelen tanulmány keretein belül nincs lehetőségünk értekezni, a kérdés komplexitásáról, a következtetéses feladatok nehézségéről és a kiértékelés lehetséges újfajta módszereiről lásd pl. Baan és mtsai (2022); Plank (2022).

Az eredeti címkével együtt elérhető angol példákat pedig a fordítás és fluenciaellenőrzés után egy-egy annotátor címkézte. Ez lehetővé tette, hogy kiderüljön, ha a fordítás során elveszett a mondatpárok közti eredeti következtetési viszony. Ebből a 2 797 példából az eredeti tanítóhalmaz 357, illetve a validációs halmaz 35 példájánál ütközik egymással az eredeti, angol adatbázisból nyert címke és a fordítás után, magyar anyanyelvű annotátortól nyert címke. Ezt a 392 példát egyelőre nem szerepeltetjük a HuRTE adatai között. Így végül a HuRTE korpuszt tanító-, validációs és teszthalmazra bontva, 2131-242-2123 példával adjuk

[5] A gépi fordításhoz az OPUS (Tiedemann, 2012) korpusztárból magunk építettünk egy angol-magyar párhuzamos korpuszt. A felhasznált alkorpuszok a következők: ParaCrawl, OpenSubtitles, Tatoeba, WikiMatrix, EUbookshop, PHP manual, TED2020, KDEdoc, KDE4. A párhuzamos korpuszból a Marian NMT (Junczys-Dowmunt és mtsai, 2018) nevű keretrendszerrel építettünk egy *transformer encoder-decoder* architektúrájú neurális fordítórendszert. A betanított modell paraméterei: 6 réteg enkóder és 6 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; előre csatolt háló méret: 4096.

[6] 14 annotátor végezte a fordítás-, 7 a fluenciaellenőrzést. 13 annotátor címkézte a teszthalmaz példáit, 4 pedig az eredeti címkével is rendelkező példákat.

[7] Bentivogli és mtsai (2009) szintén ezt a módszert alkalmazták az RTE 5 adatbázis összeállításánál: csak azokat a példákat tartották meg, amelyeknél háromból három annotátor egyetértett a címkén.

közre.[8] [9] Az 1. példa a HuRTE tanítóanyagának egyik példáját mutatja. Ezen is látszik, hogy az RTE (és elsősorban az RTE5) anyaga nem egy-egy mondatos premisszákkal és hipotézisekkel operál, hanem sokszor kettő, vagy akár három mondatos premisszákat gyűjtöttek az adatbázis összeállítói.

(1) *premise*: A veszettség vírusa megfertőzi a központi idegrendszert, agyvelőbántalmat és végül halált okoz. A veszettség korai tünetei az embernél nem jellegzetesek: láz, fejfájás és általános rossz közérzet.
hyphothesis: A veszettség az emberre nézve halálos kimenetelű.
label: *entailment*

A HuRTE kiértékeléséhez választott metrikánk, tekintettel a kiegyensúlyozatlan teszthalmazra, az MCC (Matthews, 1975).

3.3. HuCommitmentBank

A CommitmentBank (CB, de Marneffe és mtsai, 2019) olyan rövid szövegrészletekből áll, amelyekben legalább az egyik mondat tartalmaz egy alárendelő mellékmondatot. Mindegyik mellékmondat meg van címkézve azzal, hogy a szöveg írója milyen mértékben elkötelezett a mellékmondat igazsága mellett (2. példa). A SuperGLUE-ban a feladatot egy hármas osztályozású következtetési feladattá alakították: a premissza a teljes szövegrészlet, a hipotézis pedig a beágyazott tagmondat. Csak azt a részét használták a korpusznak, ahol 80% fölötti volt a mért IAA. Lényeges tulajdonsága a korpuszban szereplő példáknak, hogy minden vizsgált beágyazott mondat egy logikai következtetést semlegesítő operátor alá tartozik szintaktikailag. Logikai következtetést semlegesítő operátornak számít a kérdés, a tagadás, a feltételes mód és a modálissal való bővítés.

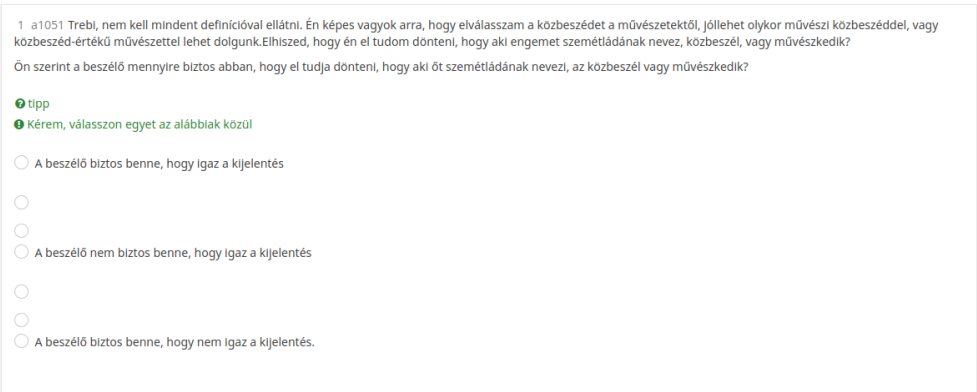
(2) A kontextus: *Polly had to think quickly.* 'Pollynak gyorsan kellett döntenie.'
A vizsgált mondat: *They were still close enough to shore for him to return her to the police if she admitted she was not an experienced ocean sailor.* 'Még elég közel voltak a parthoz ahhoz, hogy a férfi mégis átadja a rendőrségnek, ha bevallja, hogy nem egy tapasztalt tengerész.'
Az alárendelt tagmondat: Polly was not an experienced ocean sailor. 'Polly nem egy tapasztalt tengerész.'
A címke: 2 (−3-tól 3-ig terjedő skálán, ahol a 3 azt jelenti, hogy a beszélő egészen biztos az alárendelt tagmondat igazságában.)

A magyar nyelvű korpusz előállításához az angol példák 30 gyakori mátrixigéjének / -kifejezésének magyar megfelelőivel kerestünk mondatokat a korpuszban. A szövegek forrásai egyrészt az MNSZ2 (Oravecz és mtsai, 2014) beszéltnyelvi alkorpusza, másrészt a szépirodalmi alkorpusz néhány szövege, és online

[8] A teszthalmaznak a tanítóanyaghoz viszonyítva meglepően nagy aránya a GLUE és a SuperGLUE több adatbázisánál megfigyelhető. Mi egyelőre ezeknek a mintáját követve választjuk meg a tanító-, validációs és teszthalmaz méretét.

[9] A HuRTE korpusz elérhető a githubon: `https://github.com/nytud/HuRTE`.

fórumok hozzászólásainak szövegei. Az így kinyert példákon először ellenőrizni kellett, hogy található-e bennük következtetést semlegesítő operátor. 4 annotátor összesen 1100 valid szövegrészletet gyűjtött össze. A gyűjtött szövegeket az annotátorok egymás között is validálták. Az 1100 példát végül 5-5 annotátor címkézte egy 7 fokú likert-skálán (−3 és 3 között, ahol a 0 jelentette azt, hogy a beszélő nem tudja, hogy az alárendelt tagmondat igaz-e vagy hamis). Az annotálás a LimeSurvey felület segítségével történt. Összesen 9, magyar anyanyelvű annotátor dolgozott a korpuszon. Óránként 50 példa elolvasása, megértése és megítélése volt a feladatuk. Munkájuk megkönnyítése érdekében mind az 1100 példát elláttuk egy kérdéssel is, amelynek megválaszolása segíti a döntésüket. A magyar nyelvű CB előállításának részletesebb bemutatását ld. Hatvani (2022). Egy példa az annotálási felületről az 1. ábrán látható.



1. ábra: Példa a magyar nyelvű CommitmentBank előállításához használt annotálási felületről. Forrás: Hatvani (2022)

A korpusz példái a következőképpen épülnek fel: a *kontextus* a célmondat előtti 1-2 mondatot jelenti; a *célmondat* az a mondat, amely tartalmazza a következtetést törlő operátort és az alárendelt tagmondatot; minden példához külön feltüntetjük az *igét*, illetve az *alárendelt tagmondatot*; a *marker* az adott példában található következtetést semlegesítő operátor típusa. A korpuszban összesen 10 féle marker figyelhető meg: modális, feltételes, kérdés, tagadás, modális tagadás, modális kérdés, modális feltételes, feltételes kérdés, retorikai kérdés, tagadott kérdés. A 3. alatt a magyar korpusz egy példáját láthatjuk.

(3) Kontextus: *Vivien:- Tudja mit csinálnék? Bekopognék, megkérdezem elnézést van a háznak biztosítása? Igen van. Akkor itt estem fenékre. De azt hogyan bizonyítom? Mert ugye nagyon elmésen ma már egy kátyút fotóval kell bizonyítani ami szerintem egy mulatságos dolog. A fenékre esést hogyan*

lehet?
Célmondat: *Hogyan bizonyítom hogy itt történt a baleset?*
Mátrixige: *bizonyít*
Marker: kérdés
Alárendelt tagmondat: *Itt történt a baleset.*
Címke: -2

A SuperGLUE-ba a CB-nek egy olyan részhalmazát válogatták csak be, ahol az annotátorok közti egyetértés legalább 80% volt.[10] Így az eredeti 1200 példából egy 250 példát tartalmazó tanító-, egy 57 példás validáló-, és egy 250 példás teszthalmazt választottak. A feladatot hármas osztályozású következtetési feladatként fogalmazták meg, ld. a 2. példából levezetett 4. példa.

(4) premissza: *Polly had to think quickly. They were still close enough to shore for him to return her to the police if she admitted she was not an experienced ocean sailor.* 'Pollynak gyorsan kellett döntenie. Még elég közel voltak a parthoz ahhoz, hogy a férfi mégis átadja a rendőrségnek, ha bevallja, hogy nem egy tapasztalt tengerész.'
hipotézis: *Polly was not an experienced ocean sailor.* 'Polly nem egy tapasztalt tengerész.'
címke: entailment

A SuperGLUE-hoz hasonlóan a HuLU-ban is következtetési feladatként szerepel ez a korpusz, de a 7 fokú címkét hármas osztályozásra cseréltük: az eredeti címkék -1, 0 és 1 értékei a *semleges*, a -3 és -2 az *ellentmondás*, a 2 és 3 pedig a *következik* kategóriákba lettek tömörítve (5).

(5) premissza: *Vivien:- Tudja mit csinálnék? Bekopognék, megkérdezem elnézést van a háznak biztosítása? Igen van. Akkor itt estem fenékre. De azt hogyan bizonyítom? Mert ugye nagyon elmésen ma már egy kátyút fotóval kell bizonyítani ami szerintem egy mulatságos dolog. A fenékre esést hogyan lehet? Hogyan bizonyítom hogy itt történt a baleset?*[11]
hipotézis: *A beszélő szerint itt történt a baleset.* [12]
címke: *ellentmondás*

A korpuszon az IAÁ-t többféleképpen mértük (a Likert-skálára intervallum skálaként tekintve, ennek vitatottságáról ld. pl. Wu és Leung, 2017). A Krippendorf's α 60,5%. A szórás (SD) $1,01$. Csak azokat a példákat vettük fel a HuLU-ba, amelyeknél $SD < 1$. Így végül 604 példából egy 250-250 példás tanító, illetve

[10] Az nem derül ki, hogy hogyan mérték az egyetértést a korpuszon.

[11] A szöveget módosítások nélkül emeltük át az MNSZ2-ből, a helyesírási hibák, elütések javítása nélkül.

[12] Mivel eredetileg is a beszélőnek a tagmondat igazságértékével kapcsolatos meggyőződését (*commitmentjét*) vizsgálta a korpusz, nem értünk egyet azzal, hogy a SuperGLUE-ban a megfogalmazott hipotézis a tagmondat tartalma maga. Az eredeti annotáció hipotézisbe forgatva helyesebb úgy, ha a beszélő véleményeként fogalmazzuk meg.

teszthalmazt, és egy 103 példát tartalmazó validációs halmazt hoztunk létre.[13] A teszthalmaz példáinál $SD < 0,5$.

A kiértékelés metrikája, a teszthalmazban lévő címkék arányának kiegyensúlyozatlanságát is figyelembe véve, a *súlyozott F1-mérték*.

4. A webszolgáltatás bemutatása

Az új adatbázisokon túl a HuLU-t egy webszolgáltatással is kiegészítettük, amely a hulu.nytud.hu címen elérhető, regisztráció után minden funkciója bárki számára szabadon használható. A webszolgáltatás lehetővé teszi a nyelvmodellek egységes módon történő gyors és kényelmes kiértékelését, valamint az eredmények közzétételét. Így az egyes modellek teljesítményét mindenki egy kattintással megtekintheti.

A magyarul és angolul is elérhető webszolgáltatás megjelenésében és funkcióiban is követi a GLUE és SuperGLUE felületeket. Felépítésevel a teljes modellfejlesztési folyamat során végig támogatja a felhasználót, a bevezető magyarázattal kezdve a publikációk elérésén át a *benchmark* feladatok megértéséig és a gyarkorlati teendők megismeréséig.

Az oldalról közvetlenül letölthetők a *benchmark* korpuszok (jelenleg: HuCOLA, HuCommitmentBank, HuCoPA, HuRTE, HuSST, HuWNLI), amelyek a szokásos tanító-, validációs- és teszthalmazokra vannak bontva. A tesztanyag címkéit nem adjuk közre, azok csak a HuLU szerverén futó kiértékelő modul számára hozzáférhetők.

A specifikált formátumban megcímkézett teszthalmaz feltöltése után a webszolgáltatás az eredményeket az alábbi metrikák szerint értékeli ki:

- HuCOLA, HuCoPA, HuRTE: Matthew's Correlation Coefficient (MCC). Bár a GLUE/SuperGLUE az utóbbi kettő esetében abszolút pontosságot (*accuracy*) számol, mi a címkék tesztanyagbeli eloszlásának egyenlőtlensége miatt az MCC mellett döntöttünk ezeknél is.
- HuSST, HuWNLI: abszolút pontosság (*accuracy*) a GLUE/SuperGLUE mintájára.
- HuCommitmentBank: súlyozott F1-mérték, amely kiválóan alkalmas az eltérő méretű osztályok kezelésére többosztályos klasszifikáció esetében.

A kiértékelés eredményét visszajelezzük a feltöltő felhasználónak, de azok nem válnak automatikusan nyilvánossá a weboldalon, ahhoz az NYTK munkatársainak jóváhagyása szükséges.

A már nyilvánossá tett eredmények többféleképpen jelennek meg: a modellek teljesítménye időrend szerint grafikonon és teljesítmény szerint rendezett táblázatos formában is megtekinthető.

A HuLU webszolgáltatás lehetővé teszi a modellek egy-egy feladaton történő kiértékelését tetszőleges időpontban és sorrendben, míg a GLUE és SuperGLUE oldalakon a modelleket csak az összes feladaton egyszerre értékelhetjük ki.

[13] A HuCommitmentBank korpusz elérhető a githubon: `https://github.com/nytud/HuCommitmentBank`.

4.1. A webszolgáltatás implementációja

A hulu.nytud.hu weboldal a NYTK szerverein fut Docker konténerekben, Portainer platformon. A kiértékelő modul és a webes backend is Pythonban lett implementálva, Django keretrendszer felett. Az eredmények tetszetős megjelenítéséhez a Streamlit[14] rendszert használtuk, amely önmagában is alkalmas dinamikus adatok publikációjára, de, mint esetünkben, weboldalba is beágyazható.

A kiértékelésre feltölendő tesztanyagok elvárt formátuma megegyezik a letölthető tanító- és tesztanyag formátumával. Mindegyiket *json* formátumban kell feltölteni, szerepelnie kell benne az aktuális entitás azonosítójának, valamint a tesztanyaggal megegyező nevű eredménymezőnek. A kiértékelő program minden feltöltésnél ellenőrzi az elvárt formátumot, valamint ellenőrzi, hogy minden elemre érkezett-e predikció. A teljes rendszerhez hasonlóan a kiértékelő rendszert is modulárisan építettük fel, hogy könnyen tudjunk integrálni új metrikákat, valamint felvenni új *benchmark* teszteket a rendszerbe.

5. Eredmények

A HuLU 5 korpuszán kiértékeltük néhány magyarul (is) tudó nyelvmodell teljesítményét (1. táblázat). Az eredmények alapján a legnehezebb feladatnak a következtetési feladat, a HuRTE bizonyul, amelynél a legjobb eredményt a PULI BERT-Large (Yang és mtsai, 2023) érte el (MCC: 51,7; viszonyításképpen a BERT Devlin és mtsai, 2019 eredménye a GLUE-ban az RTE adathalmazon 70,1% abszolút pontosság). Jobban teljesítettek a modellek a HuCOLA és a HuSST feladatokon (mindkettőn a PULI BERT-Large a legeredményesebb, a HuCOLA MCC: 71,1, illetve abszolút pontosság a HuSST-n: 79,9%). Az eredményekből látható, hogy a huBERT (Nemeskey, 2021) alig marad el a PULI BERT-Large-tól, sőt, a HuCoPA feladaton, amely szintén nagyon nehéznek bizonyul a modellek számára, jobban is teljesít annál (MCC: 56,1; a BERT eredménye a CoPÁ-n 70,6% abszolút pontosság). A GPT modellt nem sikerült a HuCoPA adatbázisokra finomhangolni, illetve a megfelelő promptot megtalálni a tanításához.

	HuCOLA (MCC)	HuCoPA (MCC)	HuRTE (MCC)	HuSST (Pontosság)	HuWNLI (Pontosság)
huBERT	70,9	**56,1**	48,7	79,4%	64,9%
XLM-R base[15]	55,9	3,2	33,3	66,1%	63,4%
PULI GPT-2[16]	49,9		41,5	72,8%	61,9%
PULI BERT-Large	**71,1**	41,4	**51,7**	**79,9%**	**65,7%**

1. táblázat. Néhány magyar nyelvmodell eredménye a HuLU egyes korpuszain. Az oszlopok az egyes korpuszokat és az azoknál alkalmazott kiértékelési metrikát, a sorok a modelleket jelölik.

[14] https://streamlit.io

Az 1. táblázatban látható számok a legelső eredmények a HuLU adatbázisain. Valójában új kutatási irányt jelent, hogy az egyes feladatokon hogyan tudunk az angol SOTA eredményekhez hasonlóakat elérni a magyar modellekkel, hiszen minden modell és feladat más-más hiperparaméter-beállítást igényel. Például egy nagyobb modell kisebb tanulási rátát igényel, de a *batch* méret is hatással van a teljesítményre. A HuSST és a HuCOLA megoldható egyszerű mondatszintű osztályozás módszerével. A HuCoPA egy többválasztásos (*multiple choice*) feladat. A HuRTE és a HuWNLI korpuszokat egy entitásorientált szentimentelemzéshez hasonló megoldással érdemes megközelíteni, ahol a két vizsgált mondat viszonyát szeretnénk megtudni.

A különböző feladatok esetében a hiperparaméterek mellett különböző bemeneti adatstruktúrákkal, úgynevezett promptokkal (Shin és mtsai, 2020) is érdemes kísérletezni, amelyek hatással lehetnek a teljesítményre.[17]

6. Összegzés

A nyelvtechnológia különböző feladatain a korábbi szabályalapú, vagy hagyományos statisztikai alapú modelleket rendre túlteljesítő neurális nyelvmodellek teljesítményének megfelelő mérésére és összehasonlítására szolgálnak az ún. *benchmark* adatbázisok és adatbázis-gyűjtemények. A HuLU-t az előtanítás során magyar szöveget is látó, és így magyarul (is) tudó nyelvmodellek kiértékelésére indítottuk útjára 2022-ben. Jelen cikkünkben bemutattunk két új adatbázist, amelyek kifejezetten nehéz, következtetési feladatot képviselő adatbázisként a HuLU részei lehetnek, tovább színesítve a HuLU-ban elérhető korpuszok széles palettáját. Bemutattuk a HuLU-hoz készített webszolgáltatást, amellyel a modellek eredményeinek kiértékelése könnyen elérhetővé válik akár csak egy-egy adatbázison is. Az eredmények összehasonlíthatóságát pedig egy dicsőségtábla segíti. A HuLU eddigi korpuszain végzett kiértékeléseink azt mutatják, hogy

[15] Conneau és mtsai (2019)

[16] Yang (2022)

[17] Azoknál a feladatoknál, ahol két mondat viszonyát szeretnénk megtanítani a modelleknek (például HuCoPA, HuRTE vagy HuWNLI), kísérleti irány lehet a vizsgálandó szövegek szeparátor címkével (`SEP`) való elválasztása, vagy egy egységes folyószöveggé való átalakítása. Egy-egy példa a két típusra a HuCoPA egy példájával:

- Eredeti:
  - *premise*: A bár bezárt.
  - *choice1*: Zsúfolt volt.
  - *choice2*: Hajnali három volt.
  - *question*: *cause*
- Szeparátor címkével: A bár bezárt. [SEP] Zsúfolt volt.
- Szöveggé alakítva: A bár bezárt. Mert zsúfolt volt.

A fenti példában a szeparátor címke lehet akár `[CLS]` vagy a GPT modelleknél `</s>` stb. A folyószöveggé való átalakítás során a fenti példában két mondatot látunk, de lehet egy mondatba is foglalni őket vesszővel elválasztva stb.

a magyar nyelvmodelleken még bőven van mit csiszolni, hogy az angolul tudó társaik angolnyelv-tudásához hasonló szintre érjenek a magyarnyelv-tudásukkal.

Hivatkozások

Baan, J., Aziz, W., Plank, B., Fernández, R.: Stop Measuring Calibration When Humans Disagree. CoRR abs/2210.16133 (2022)

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Venice, Italy. pp. 1–9 (2006)

Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the TAC Workshop (2009)

Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics (2015)

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish Pre-Trained BERT Model and Evaluation Data. In: PML4DC at ICLR 2020 (2020)

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale (2019), `https://arxiv.org/abs/1911.02116`

Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché Buc, F. (szerk.) Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment. pp. 177–190. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), `https://aclanthology.org/N19-1423`

Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.Gy., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. p. 1–9. RTE '07 (2007)

Hatvani, P.: A Corpus to Investigate Projection Methods: The Hungarian Commitment Bank. Szakdolgozat, Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar, Angol-Amerikai Intézet, Elméleti Nyelvészet Tanszék (2022)

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast Neural Machine Translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018)

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Waseem, Z., Stenetorp, P., Jia, R., Bansal, M., Potts, C., Williams, A.: Dynabench: Rethinking benchmarking in NLP. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4110–4124. Association for Computational Linguistics, Online (Jun 2021), https://aclanthology.org/2021.naacl-main.324

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., Schwab, D.: FlauBERT: Unsupervised language model pre-training for French. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 2479–2490. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.302

Levesque, H.J., Davis, E., Morgenstern, L.: The Winograd Schema Challenge. In: Proceedings of the Thitteenth International Conference on Principles of Knowledge Representation and Reasoning. p. 552–561. KR'12, AAAI Press (2012)

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, R., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J.H., Wu, W., Liu, S., Yang, F., Campos, D., Majumder, R., Zhou, M.: XGLUE: A New Benchmark Dataset for Cross-lingual Pretraining, Understanding and Generation. arXiv abs/2004.01401 (2020)

Ligeti-Nagy, N., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Laki, L.J., Vadász, N., Yang, Z.Gy., Váradi, T.: HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 431–446. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

de Marneffe, M.C., Simons, M., Tonhauser, J.: The CommitmentBank: Investigating projection in naturally occurring discourse. Proceedings of Sinn und Bedeutung 23(2), 107–124 (Jul 2019), https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601

Matthews, B.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure 405(2), 442–451 (1975), https://www.sciencedirect.com/science/article/pii/0005279575901099

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (2014)

Plank, B.: The „Problem" of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP (2022)

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016), `https://aclanthology.org/D16-1264`

Shavrina, T., Fenogenova, A., Emelyanov, A., Shevelev, D., Artemova, E., Malykh, V., Mikhailov, V., Tikhonova, M., Chertok, A., Evlampiev, A.: RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. arXiv preprint arXiv:2010.15925 (2020)

Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4222–4235. Association for Computational Linguistics, Online (Nov 2020), `https://aclanthology.org/2020.emnlp-main.346`

Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (2012)

Vadász, N., Ligeti-Nagy, N.: Winograd schemata and other datasets for anaphora resolution in hungarian. Acta Linguistica Academica 69(4) (2022), in press.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (2020)

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018), `https://aclanthology.org/W18-5446`

Williams, A., Nangia, N., Bowman, S.: A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).

pp. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), https://aclanthology.org/N18-1101

Wu, H., Leung, S.O.: Can Likert Scales be Treated as Interval Scales?—A Simulation Study. Journal of Social Service Research 43(4), 527–532 (2017), https://doi.org/10.1080/01488376.2017.1329775

Yang, Z.Gy.: "Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 463–476. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Yang, Z.Gy., Dodé, R., Ferenczi, G., Héja, E., Kőrös, Á., Laki, L.J., Ligeti-Nagy, N., Jelencsik-Mátyus, K., Vadász, N., Váradi, T.: Jönnek a nagyok! BERT-Large, GPT-2 és GPT-3 nyelvmodellek magyar nyelvre. In: XIX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged, Hungary (2023)

HunSum-1: an Abstractive Summarization Dataset for Hungarian

Botond Barta[1,2], Dorina Lakatos[1,2], Attila Nagy[1], Milán Konor Nyist[1], Judit Ács[2]
{botondbarta, dorinalakatos, acsjudit}@sztaki.hu,
attila.nagy234@gmail.com, nyist.milan78@gmail.com

[1]Department of Automation and Applied Informatics
Budapest University of Technology and Economics
[2]Institute for Computer Science and Control
Eötvös Loránd Research Network

Abstract. We introduce *HunSum-1*: a dataset for Hungarian abstractive summarization, consisting of 1.14M news articles. The dataset is built by collecting, cleaning and deduplicating data from 9 major Hungarian news sites through CommonCrawl. Using this dataset, we build abstractive summarizer models based on huBERT and mT5. We demonstrate the value of the created dataset by performing a quantitative and qualitative analysis on the models' results. The *HunSum-1* dataset, all models used in our experiments and our code[1] are available open source.

1 Introduction

Automatic summarization of documents (Nenkova et al., 2011) is a widely studied task in natural language processing. The goal of summarization is to reduce the original document to a short, concise text, such that it captures every key information from the input. Generally, there are two approaches to summarization: extractive and abstractive. Extractive methods directly use the input document to produce the summary by extracting relevant tokens or sentences explicitly. In contrast, abstractive summarization generates new text conditioned on the input. Extractive summarization methods are usually less complex from a computational perspective compared to their abstractive counterparts, however, they have limitations such as handling long sentences or ensuring coherence among the extracted sentences. Producing more sophisticated summaries requires abstractive summarization, where the model may generate words, which are not present in the original document.

In this paper, we introduce *HunSum-1*, an abstractive summarization dataset for Hungarian consisting of 1.14M news articles with leads. In Section 3 we discuss in detail how we extracted the data from CommonCrawl and performed a number of preprocessing steps to arrive at the current version of *HunSum-1*. We also train three baseline models on the dataset and evaluate the results both

[1] https://github.com/dorinapetra/summarization

quantitatively and qualitatively. To the best of our knowledge, this is the first publicly available Hungarian abstractive summarization dataset of this size. We hope this will serve as a good foundation for future abstractive summarization research in Hungarian.

2 Related work

Nallapati et al. (2017) introduce an RNN based extractive summarization model called *SummaRuNNer*, which outperformed several models, such as a feature-rich logistic classifier (Cheng and Lapata, 2016), an Integer Linear Programming based approach (Woodsend and Lapata, 2010) and various graph based models (Parveen et al., 2015; Wan, 2010). Traditionally there are two main steps in an extractive summarization model: sentence scoring and sentence selection. Zhou et al. (2018) integrate the two steps and directly predict the importance of the sentence based on the previous ones. Liu (2019) introduces BERTSUM, an end-to-end BERT-based (Devlin et al., 2019) model with inter-sentence Transformer layers. BERTSUM achieves better ROUGE-L scores than NEUSUM, the previous state-of-the-art model on the *CNN/Daily Mail* (Hermann et al., 2015) dataset.

For abstractive summarization, Paulus et al. (2018) introduce an encoder-decoder based neural intra-attention model and a new training procedure that combines supervised and reinforcement learning. Gehrmann et al. (2018) propose the *bottom-up* attention, a new technique for content selection: it uses a data-efficient content selector to over-determine the important phrases from the input. The BERT-based PRESUMM model (Liu and Lapata, 2019) uses a new fine-tuning schedule which adopts different optimizers for the encoder and the decoder. The current state-of-the-art model, PEGASUS was introduced by Zhang et al. (2020). During the training of PEGASUS, the important sentences are masked from the input document, and the models have to generate these sentences together as one output. This training objective eliminates the need for a supervised summarization dataset.

Raffel et al. (2020) introduce T5 (Text-To-Text Transfer Transformer) a Transformer based encoder-decoder which was pretrained on several downstream tasks including summarization, but only for English. mT5 (Xue et al., 2021) is the multilingual version of the T5 model. Hasan et al. (2021) fine-tune the mT5 model for multilingual summarization and achieve competitive results compared to monolingual models.

Hermann et al. (2015) create a dataset from pairing the summaries with CNN and Daily Mail articles. The *CNN/Dailymail* summarization corpus contains over 1 million samples. Hasan et al. (2021) introduce a similar dataset called *XL-Sum*, which has 1 million annotated data samples in 44 languages. They extracted the lead as the summary and the article as the input document from BBC news.

Yang et al. (2021) build a summarization corpus from Index and HVG articles and train several BERT models for abstractive summarization: multilingual cased BERT, huBERT (Nemeskey, 2020) and the HILBERT large mod-

els (Feldmann et al., 2021). The corpus is unfortunately not publicly available. Makrai et al. (2022) publish an encoder-decoder based model initialized with huBERT. They use the ELTE.DH corpus with 300k samples for training. Agócs and Yang (2022) used the PreSumm method for training multilingual and Hungarian BERT models. Yang (2022) finetunes BART models (Lewis et al., 2020) for abstractive summarization.

3 Summarization dataset

We built a dataset from Hungarian news articles using the Common Crawl corpus[2]. Our dataset can be used as training data for abstractive summarization and title generation tasks.

3.1 Common Crawl

The Common Crawl (CC) Foundation crawls web pages and provides periodic snapshots freely available to the public. The corpus contains petabytes of data collected since 2008. The data is available in WARC format on Amazon S3 and can be downloaded through HTTP or S3. We used the CommonCrawl Downloader package (Indig, 2018), which helped us download and deduplicate articles based on the given domains. We selected 9 large Hungarian news websites with plenty of content and downloaded pages from all Common Crawl segments till 2022-27. Our selection was limited to websites that use dedicated lead fields. The portals' topics cover politics, economics, sports and several others. We collected around 94 GB of raw HTML data.

3.2 Dataset

We extracted the title, the lead, the article, the date of its creation and the tags, if it had any from each article. We skipped articles if we did not find a lead during parsing. We removed every link, embedded social media, recommendations and image captions from each article that would not help the model in the summarization task. Articles that are part of a live blog are dropped, as they are too short and often cannot be considered as an independent article without the rest of the blog. We also dropped galleries because they usually do not have meaningful content without the images.

After scraping the pages we had about 1.88 million articles. However, we filtered the data to discard articles that would not be valid input for a summarization model (e.g. too short article). Table 1 shows the constraints that were applied to the length of the lead and the article. For word and sentence tokenization we used the `quntoken` package[3]. We also removed articles where the lead was longer than the article's text, this removed roughly 17k articles. The constraint about minimum and maximum article characters removed 24k articles.

[2] https://commoncrawl.org/
[3] https://github.com/nytud/quntoken

After filtering for maximum number of lead sentences and minimum number of lead tokens, 10k and 25k articles were removed respectively. The largest number of articles were dropped when we filtered for the minimum sentence number of the article text, this filtering step removed roughly 300k articles. We also removed 3k non-Hungarian articles, which we identified using a language detection model, FastText (Joulin et al., 2016b,a).

Constraint	**Value**
min article characters	200
max article characters	15000
min lead tokens	6
max lead sentences	5
min article sentences	6

Table 1: Length constraints of the article and lead.

After filtering, we deduplicated the articles using Locality Sensitive Hashing (LSH) to remove redundant articles. Two articles were considered equivalent if their article texts were 45% similar. When two articles were identified similar, the one without lead was dropped, if both of them had lead we kept the one with later crawl time. With this threshold about 363k articles were removed. The deduplicated dataset consists of 1.14M data samples. Table 2 shows the number of samples from each news site.

Site	**Count**
24.hu	359.4k
origo.hu	305.0k
hvg.hu	216.8k
index.hu	154.5k
nepszava.hu	56.7k
portfolio.hu	22.6k
m4sport.hu	17.9k
metropol.hu	11.1k
telex.hu	4.4k

Table 2: Number of data samples per site.

The dataset contains articles from every year between 1999 and 2022. Figure 1 shows the distribution of the articles among the years. The dataset has significantly fewer samples from 2022 despite using all Common Crawl segments up until July, 2022. This is probably caused by the fact that CC has some latency.

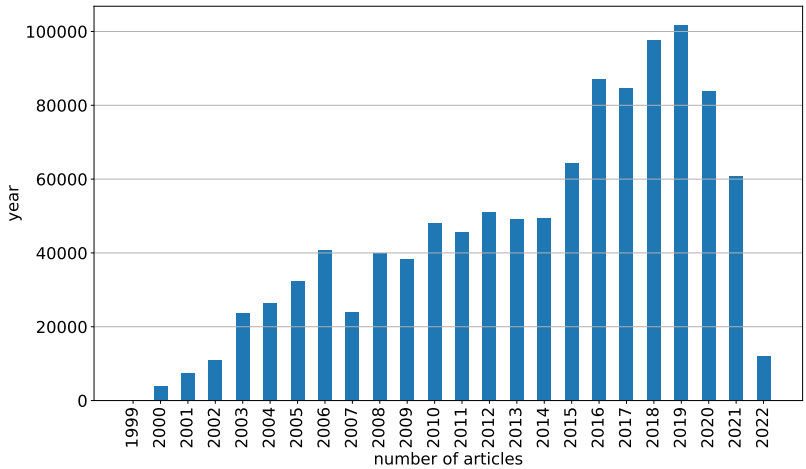


Fig. 1: Number of articles by year.

Table 3 describes the average number of tokens and sentences in the articles and leads. *Népszava* and *Telex* has significantly longer articles than the other sites.

	Article		Lead	
Site	**tokens**	**sents**	**tokens**	**sents**
24.hu	323	17	21	1.3
origo.hu	396	19	40	1.9
hvg.hu	378	17	30	1.5
index.hu	517	26	43	2.2
nepszava.hu	**869**	**39**	32	1.5
portfolio.hu	457	22	**52**	2.1
m4sport.hu	385	23	28	1.3
metropol.hu	305	17	25	1.4
telex.hu	839	38	49	**2.3**

Table 3: Average length of the articles and leads.

We compute a number of automatic quantitative metrics to estimate the quality of *HunSum-1*. Novel n-gram ratio (NNG-n) (Narayan et al., 2018) measures the percentage of n-grams in the summary that were not present in the original article. Bommasani and Cardie (2020) introduce a compression metric

(CMP), which is defined as:

$$\boldsymbol{CMP}(A, S) = 1 - \frac{|S|}{|A|} \tag{1}$$

where $|S|$ and $|A|$ are the length of the summary and the article. Finally, we compute a redundancy metric (RED-n) based on Hasan et al. (2021). It is defined as a ratio of redundant n-grams and the total number of n-grams in a summary S:

$$\begin{aligned} \boldsymbol{RED}(S) &= \frac{\sum_{i=1}^{m}(f_i - 1)}{\sum_{i=1}^{m} f_i} \\ &= 1 - \frac{m}{|S| - n + 1} \end{aligned} \tag{2}$$

where $\{f_1, f_2, ..., f_m\}$ denote the frequencies corresponding to unique n-grams $\{g_1, g_2, ..., g_m\}$. A summary is generally considered to be of high quality if it has a high novel n-gram ratio and compression with a low redundancy metric. The above metrics computed on the *HunSum-1* dataset can be seen in Table 4.

NNG-1	**NNG-2**	**NNG-3**	**CMP**	**RED-1**	**RED-2**
78.0	90.09	95.48	88.58	6.58	0.1

Table 4: Intrinsic evaluation of *HunSum-1*.

We created a train-dev-test split which will be available on the HuggingFace Dataset Hub under the name of `SZTAKI-HLT/HunSum-1`. The development and the test datasets both contain 2000 documents and are created using stratified sampling from every news page.

4 Models

We experimented with two language models: an encoder-decoder model, where both sides are composed of BERT models (Rothe et al., 2020). We initialized both sides with the weights of the Hungarian version of BERT (Devlin et al., 2019), huBERT (Nemeskey, 2020) (Bert2Bert) and the pretrained multilingual mT5 (Xue et al., 2021).

4.1 mT5

mT5 is the multilingual version of the Text-To-Text Transfer Transformer (T5) (Raffel et al., 2020) and it supports 101 languages including Hungarian. mT5 was trained on the mC4 dataset, totaling 6.3T tokens that was extracted from the Common Crawl dataset. Hungarian is the 22th most common language in this corpus and contains 37 million Hungarian pages. Unlike the T5 model, the mT5

was not trained on any downstream tasks, like abstractive summarization. mT5 has 5 variants: `mt5-small`, `mt5-base`, `mt5-large`, `mt5-xl` and `mt5-xxl`, with 300M, 580M, 1.2B, 3.7B and 13B parameters respectively. We decided to only experiment with the small and base variants due to computational constraints. Unlike the BERT family, T5 and mT5 are generative models.

5 Experiments

We trained 3 summarization models: Bert2Bert, mT5-small and mT5-base. We found that different generation parameters work best for the Bert2Bert and the mT5 models. The mT5 models also needed an additional parameter, because we observed that the mT5 models tended to copy the input sentences. To prevent this behaviour, we set the `encoder_no_repeat_ngram_size` argument to 4. Table 5 describes the training and the generation hyperparameters.

Parameter	**Bert2Bert**	**mT5-small**	**mT5-base**
batch size	13	16	12
learning rate	5e-5	5e-5	5e-5
weight decay	0.01	0.01	0.01
warmup steps	16000	3000	3000
epochs	15	10	10
`no_repeat_ngram_size`	3	3	3
`num_beams`	5	5	5
`early_stopping`	True	False	False
`encoder_no_repeat_ngram_size`	-	4	4

Table 5: Hyperparameters for training and generation.

The maximum length of the article is 512 subwords and for the generated lead it is 128. The models were trained on a single NVIDIA A100 GPU. The Bert2Bert was trained for 265 hours, the mT5-small was trained for 54 hours, while the mT5-base model took 119 hours. Our models are available on the Huggingface Model Hub under the names of `SZTAKI-HLT/Bert2Bert-HunSum-1`, `SZTAKI-HLT/mT5-small-HunSum-1` and `SZTAKI-HLT/mT5-base-HunSum-1`.

6 Results

We used the ROUGE scores (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) for automatic evaluation. ROUGE takes a reference summary and computes the word ngram overlaps with the models' output. The reference summary in our case is the real article lead. We calculate two versions, a normal and a stemmed one. As Hungarian is an agglutinative language, small changes

in inflection can decrease the ROUGE scores even if the summary is correct. We used the HuSpaCy (Orosz et al., 2022) model for stemming. Table 6 shows the ROUGE scores on the test set.

Model	Original			Stemmed		
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Bert2Bert	28.52	10.35	20.07	31.75	11.80	22.31
mT5-small	36.49	9.50	23.48	40.86	11.71	26.25
mT5-base	**37.70**	**11.22**	**24.37**	**41.97**	**13.65**	**27.23**
foszt2oszt[4]	23.62	5.05	16.57	26.22	5.95	18.37
hi-bart-base[5]	29.03	8.79	19.60	32.57	10.45	21.98
hi-bart[6]	30.03	8.96	20.50	33.23	10.45	22.71
nol-bart[7]	26.38	6.92	18.05	29.25	8.06	20.09

Table 6: ROUGE recall scores on the test set. ROUGE-1, ROUGE-2 and ROUGE-L scores are abbreviated as R-1, R-2 and R-L respectively.

Based on the ROUGE scores, the mT5-base model is clearly the best model and mT5-small is the second best, although not by every measure. Despite its name, mT5-small is not a very small model with its 300M trainable parameters. In contrast, Bert2Bert, which is the combination of two BERT models, only has 250M parameters. Aside from the higher parameter count, T5 models are generative, while BERT is not, which may explain why a multilingual model is better for this generative task than the monolingual huBERT.

We compared our models to the available Hungarian summarization models. The results can be seen in Table 6. Based on the scores on our test set, our mT5-base model outperformed the other Hungarian models.

6.1 Human evaluation

As the ROUGE scores do not always represent the quality of the generated summaries, we decided to evaluate the models manually. We sampled 80 articles from the test set with stratified sampling for human evaluation. We used the evaluation framework from Hasan et al. (2021) (**Property A** to **C**), where an evaluator has to answer the following questions with `Yes` or `No`. We also added a new question about grammatical correctness (**Property D**).

– **Property A:** Does the summary convey what the article is about?

[4] https://huggingface.co/BME-TMIT/foszt2oszt
[5] https://huggingface.co/NYTK/summarization-hi-bart-base-1024-hungarian
[6] https://huggingface.co/NYTK/summarization-hi-bart-hungarian
[7] https://huggingface.co/NYTK/summarization-nol-bart-hungarian

– **Property B:** If the answer to property A is `Yes`, does the summary only contain information that is consistent with the article?
– **Property C:** If the answer to property A is `Yes`, does the summary only contain information that can be inferred from the article?
– **Property D:** Is the summary grammatically correct?

The 3 generated leads were presented to the annotators in random order for each article and the model name was hidden. Every article was evaluated by three annotators, we computed the final answers by majority voting.

We took the average of the answers for every property-model pair, where 1 means `Yes` and 0 means `No` answer, Figure 2 shows the results of the human evaluation. According to the human annotators, the mT5-small model performs the worst among the three models. We also observe frequent inflection errors in this model. The mT5-base model gives the best summary based on the content selection, but gives inconsistent information with the article more often than the Bert2Bert model. The latter hallucinates less than the other two models. The average answer for the grammatical question is significantly higher than the others, which means our models rarely output grammatically incorrect summaries.

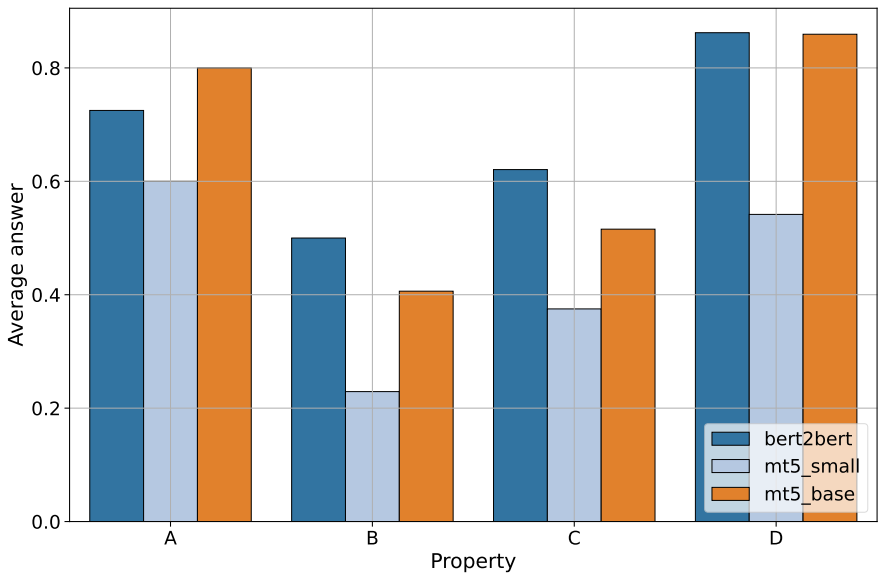


Fig. 2: The average answers for the properties by models.

We used the Cohen kappa (Cohen, 1960) to measure how much the annotators agreed on each question. We calculated this value for each annotator pair and took the average of it. The average pair-wised Cohen kappa is 56.77%. The

grammatical question has the lowest score (37.99%) because the guideline was not clear and some of the annotators considered unfinished sentences grammatically incorrect, while others did not.

7 Conclusion

We introduced *HunSum-1*, a Hungarian abstractive summarization corpus with over 1.1M data samples drawn from 9 major news outlets. We computed various intrinsic measures on the corpus quality. We then trained 3 summarization models: a Bert2Bert model and two mT5 models. We found that the larger mT5 model performed best both by automatic metrics and by human evaluation.

References

Agócs, Á., Yang, Z.Gy.: Absztraktív összefoglaló presumm módszerrel (2022)

Bommasani, R., Cardie, C.: Intrinsic evaluation of summarization datasets. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8075–8096. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.649

Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. CoRR abs/1603.07252 (2016), http://arxiv.org/abs/1603.07252

Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960), https://doi.org/10.1177/001316446002000104

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1423

Feldmann, Á., Váradi, T., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.Gy.: Hilbert, magyar nyelvű bert-large modell tanítása felhő környezetben. Berend G., Gosztolya G., Vincze V.(szerk.) XVII. Magyar Számítgépes Nyelvészeti Konferencia pp. 29–36 (2021)

Gehrmann, S., Deng, Y., Rush, A.: Bottom-up abstractive summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4098–4109. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1443

Hasan, T., Bhattacharjee, A., Islam, M.S., Mubasshir, K., Li, Y.F., Kang, Y.B., Rahman, M.S., Shahriyar, R.: XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4693–4703. Association for Computational Linguistics, Online (Aug 2021), https://aclanthology.org/2021.findings-acl.413

Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015)

Indig, B.: Közös crawlnak is egy korpusz a vége – korpuszépítés a commoncrawl .hu domainjából. In: Vincze, V. (ed.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018). p. 125–134. Szegedi Tudományegyetem Informatikai Intézet, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2018)

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016a)

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016b)

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020), https://aclanthology.org/2020.acl-main.703

Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013

Liu, Y.: Fine-tune BERT for extractive summarization. CoRR abs/1903.10318 (2019), http://arxiv.org/abs/1903.10318

Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1387

Makrai, M., Tündik, Á.M., Indig, B., Szaszák, G.: Towards abstractive summarization in hungarian (2022)

Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. p. 3075–3081. AAAI'17, AAAI Press (2017)

Narayan, S., Cohen, S.B., Lapata, M.: Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1797–1807. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1206

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)

Nenkova, A., Maskey, S., Liu, Y.: Automatic summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:

Tutorial Abstracts. p. 3. Association for Computational Linguistics, Portland, Oregon (Jun 2011), https://aclanthology.org/P11-5003

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit (2022)

Parveen, D., Ramsl, H.M., Strube, M.: Topical coherence for graph-based extractive summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1949–1954. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), https://aclanthology.org/D15-1226

Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=HkAClQgA-

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (2020)

Rothe, S., Narayan, S., Severyn, A.: Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics 8, 264–280 (2020), https://aclanthology.org/2020.tacl-1.18

Wan, X.: Towards a unified approach to simultaneous single-document and multi-document summarizations. In: Proceedings of the 23rd international conference on computational linguistics (Coling 2010). pp. 1137–1145 (2010)

Woodsend, K., Lapata, M.: Automatic generation of story highlights. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 565–574. Association for Computational Linguistics, Uppsala, Sweden (Jul 2010), https://aclanthology.org/P10-1058

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021), https://aclanthology.org/2021.naacl-main.41

Yang, Z.Gy.: Barterezzünk! messze, messze, messze a világtól, bart kísérleti modellek magyar nyelvre (2022)

Yang, Z.Gy., Agócs, , Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. Annales Mathematicae et Informaticae 53, 299–316 (01 2021)

Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 11328–11339. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/zhang20ae.html

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 654–663. Association for Computational

Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/P18-1061

Nyelvmodellek

Jönnek a nagyok! BERT-Large, GPT-2 és GPT-3 nyelvmodellek magyar nyelvre

Yang Zijian Győző, Dodé Réka, Ferenczi Gergő, Héja Enikő, Jelencsik-Mátyus Kinga, Kőrös Ádám, Laki László János, Ligeti-Nagy Noémi, Vadász Noémi, Váradi Tamás

Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
{vezetéknév.keresztnév}@nytud.hu

Kivonat Az utóbbi években rendkívüli mértékben felgyorsult a Transformer alapú nyelvmodellek méretének a növekedése. A globális technológiai cégek nagyobbnál nagyobb modelleket tanítanak, amelyek óriási erőforrást és tanítóanyagot igényelnek. Ezekkel a kísérletekkel azt próbálják bebizonyítani, hogy megfelelően nagy méretű modellek, megfelelően sok tanítóanyaggal képesek önmagukban akár finomhangolás nélkül bármilyen nyelvtechnológiai feladatot megoldani. Ebbe a versenybe nem igazán lehetséges beszállni, de arra van lehetőség, hogy az árnyékukban elkezdjünk kísérleteket végezni a nagyobb méretű modellek irányában. Kutatásunkban különböző nagy méretű nyelvmodelleket tanítottunk magyar nyelvre. Betanítottunk egy 6,7 milliárd paraméteres GPT-3, valamint egy GPT-2 és egy BERT-Large modellt magyar nyelvre. A modelleket különböző finomhangolással teszteltük. A BERT-Large modellünk több feladatban is felülmúlta a huBERT modellt, és elsőként hoztunk létre egynyelvű magyar GPT-3 modellt, amelyekkel tudomásunk szerint elsőnek végeztünk prompt kísérleteket *few-shot* tanulással magyar nyelvre.
Kulcsszavak: GPT-3, GPT-2, Megatron BERT, prompt programozás, *few-shot* tanulás

1. Bevezetés

Az utóbbi időkben minden évben jön ki egy újabb nyelvmodell, a nagy kutatóközpontok és cégek versenyt űznek abból, hogy nagyobbnál nagyobb méretű és paraméterszámú nyelvmodelleket tanítsanak. Amikor 2021-ben a Microsoft az NVIDIA-val karöltve létrehozta az 530 milliárd paraméteres Megatron-Turing NLG modellt (Smith és mtsai, 2022), megszületett a cikk[1], amelyik felteszi a kérdést, hogy vajon ez a verseny lenne az új Moore-törvény? Ezek a kutatások

[1] https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model

azt próbálják bebizonyítani, hogy ha elég nagy a modell és több adaton tanult, akkor képes ez az egy nagy modell minden nyelvtechnológiai feladatot megoldani finomhangolás nélkül, csupán prompt programozással. Ehhez a versenyhez óriási erőforrás szükséges, amit csak a legnagyobb cégek és kutatóközpontok engedhetnek meg maguknak.

Jelenleg magyar nyelvre a legjobban teljesítő nyelvmodell a huBERT (Nemeskey, 2021). A HILBERT (Feldmann és mtsai, 2021) teljesítménye, bár méretét tekintve a huBERT-nél nagyobb modell, az eddig rendelkezésre álló teszteken elmaradt a huBERT-étől; ennek oka az, hogy a huBERT-hez képest kevesebb adaton lett betanítva. 2022 júniusában mutatta be a HILANCO[2] konzorcium a HILANCO-GPTX 6,7 milliárd paraméteres angol-magyar kétnyelvű GPT-3 modellt.

Jelen cikkben a huBERT és HILBERT modelleknél nagyobb nyelvmodellek készítéséről számolunk be. A modellek tanításához nagy mennyiségű adatot is gyűjtöttünk. A jelen tanulmányban bemutatjuk az általunk összegyűjtött tanítóanyagot, majd bemutatjuk az ebből tanított modelljeinket. Nevezetesen betanítottunk egy HILANCO-GPTX modellhez hasonló 6,7 milliárd paraméteres GPT-3, egy GPT-2 és egy BERT-Large modellt magyar nyelvre, melyeket PULI névre kereszteltük: **PULI GPT-3SX**, **PULI GPT-2** és **PULI BERT-Large**. Mindhárom modellünk kutatás céljára szabadon elérhető a Hugging Face oldalunkonk[3]: NYTK/PULI-GPT-3SX, NYTK/PULI-GPT-2, NYTK/PULI-BERT-Large

2. Kapcsolódó irodalom

Jelenleg a világ egyik legnagyobb modellje a PaLM (Chowdhery és mtsai, 2022) (Pathways Language Model) a Google-tól, amely 540 millárd paraméteres. A méretek növelése mellett bevezették a Pathways architektúrát, amely azt a célt szolgálja, hogy egyszerre minél több feladatot tudjon tanulni a modell. A Pathway egy hagyományos, csak dekóderrel rendelkező transzformer (Vaswani és mtsai, 2017) architektúrát valósít meg néhány módosítással. Módosításaihoz az utóbbi évek fejlesztéseiből merít, úgy mint a SwiGLU activációs függvény (Shazeer, 2020), a párhuzamos rétegezés (Wang és Komatsuzaki, 2021) a transzformer blokkokban, a RoPE beágyazás (Su és mtsai, 2021) vagy a SentencePiece (Kudo és Richardson, 2018) használata. A PaLM modellhez képest méretben egy kicsivel lemaradva, komoly konkurens a már a bevezetésben is említett Megatron-Turing NLG modell (Smith és mtsai, 2022). Ebben a nagyságrendben elsikkad a figyelmünk a 10 milliárd paraméter különbség felett, de például magyar nyelvre még nem sikerült senkinek 10 milliárd paraméteres modellt tanítani. Paraméterszámát tekintve így is háromszor akkora, mint a mérföldkőnek számító GPT-3. A GPT-3 (Brown és mtsai, 2020) megjelenése óriási visszhangot váltott ki mind a sajtóban, mind a nyelvtechnológiai szakmai közösség körében. Újdonsága abban rejlett, hogy óriási mennyiségű adattal tanították, és az akkori viszonylat-

[2] https://hilanco.github.io
[3] https://huggingface.co/NYTK

ban óriási paraméterszámmal. A modellel olyan szöveget tudtak generálni, ami az emberi íráshoz volt hasonló. Emellett finomhangolás nélkül, az úgynevezett prompt programozással kevés (*few-shot*) bemeneti példával, vagy egyáltalán nem adva példát (*zero-shot*), meg tud oldani nyelvtechnológiai feladatokat. A GPT-3 több változata is elérhető kipróbálásra, mint a Davinci, Curie, Babbage vagy az Ada. Mindegyik változat más-más feladattípusban erős. Az eddig felsorolt modellek mellett érdemes megemlíteni a Wu Dao 2.0[4] modellt. A Wu Dao 2.0 modellt a Beijing Academy of Artificial Intelligence (BAAI) szervezet mutatta be 2021-ben. Ez jelenleg a legnagyobb neurális modell, amely 1750 milliárd paraméterrel rendelkezik. A modell összehasonlítása a többi nyelvmodellel annyiban nehéz, hogy nem csak szövegeken, hanem képeken is tanították. A modellt a Pile angol adathalmaz (Gao és mtsai, 2020) mellett további 1,2TB kínai szövegen és 2,5TB képen tanították. A modellt a FastMoE (He és mtsai, 2021) rendszerben tanították. Több feladatban is 'state-of-the-art' eredményt ért el. Az elmúlt években egymás után mutatták be a modelleket. De érdemes megemlíteni még a Megatron-Turing NLG elődjeit, a 17,2 milliárd paraméteres Turing-NLG-t[5] vagy a 8,3 milliárd paraméteres Megatron-LM (Shoeybi és mtsai, 2019a) modelleket. Ha milliárd paraméterszámról beszélünk, akkor a T5 (Raffel és mtsai, 2020) XXL modellje 13 milliárd, a GPT-2 (Radford és mtsai, 2019) pedig 1,5 milliárd paraméterével éppen belefértek még ebbe az összehasonlításba. Bár paraméterszámban már nagyságrenddel kisebb, de az utolsó említendő modell a BERT-Large (és általában a BERT család), amely a maga már szerénynek mondható 340 millió paraméterével a transzformer alapú nyelvmodellezés alapjait fektette le.

3. A tanítóanyag

Kutatásunk első része a tanítóanyag összeállítása volt. A korábbi kutatások alapján a nagy modellek tanításához nagy mennyiségű adatra is szükség van. Ehhez a feladathoz az alábbi forrásokból származó korpuszokat használtuk fel.

Az 1. táblázatban láthatóak az alkorpuszok főbb tulajdonságai. Nem tokenizáltuk a szöveget, a számokat a nyers szövegre mértük. A modellek tanításához később sem tokenizáltuk a szöveget. A korpuszban egy sor egy bekezdés, a dokumentumokat üres sorok választják el, nem bontottuk a szöveget mondatokra. A korpusz szövegei a következő forrásokból áll össze:

- **Webkorpusz 2.0**: A Webkorpusz 2.0 (Nemeskey, 2020b) korpuszt Nemeskey Dávid Márk gyűjtötte a Common Crawl[6] adattárból 2013 és 2019 április közötti időszakból. A korpusz több mint 9 milliárd tokenből áll. A kutatásunkhoz a nem tokenizált változatot használtuk fel.
- **Wikipédia**: A magyar Wikipédia, a Webkorpusz 2.0 része.

[4] https://towardsdatascience.com/gpt-3-scared-you-meet-wu-dao-2-0-a-monster-of-1-75-trillion-parameters-832cd83db484

[5] https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft

[6] https://commoncrawl.org

- **CC**: A szöveg legnagyobb részét a Common Crawl adattárból gyűjtöttük. Mivel a WebKorpusz 2.0 a 2019 áprilisig gyűjtött anyagot tartalmazza, a kutatásunkban az ezután létrejött adatokat gyűjtöttük össze. Az adatok letöltéséhez és *boilerplate*-tisztításához az Indig (2018) által módosított CC letöltő szkriptet[7] alkalmaztuk. A felhasznált szkript már tartalmaz különböző tisztítási és deduplikációs eljárásokat. Ennek ellenére, a különböző évek gyűjtései között lehetnek még duplikációk, ezeket a folyamat végén kezeltük. Fontos megjegyezni, hogy mind a Webkorpusz 2.0, mind a saját CC gyűjtés csak a '.hu' doménről származó szövegeket tartalmazza.
- **neticle**: A nyilvános közösségi média posztokból és kommentekből származó szöveganyag, melyet a Neticle Kft-től kaptunk meg korábban.
- **sajtó**: Online sajtóoldalakról letöltött tartalmak 2020 és 2021 közötti időszakból.
- **JSI**: A szlovén Jožef Stefan Institute az eventregistry.org címen futó webszolgáltatás céljaira 2013 óta számos nyelven gyűjti a híreket internetes forrásokból (RSS-ből). Ennek a magyar anyagát használtuk fel.
- **araneum**: Araneum Hungaricum Maium[8](Benko, 2014a,b; Rychlý, 2007) korpusz, amelyet Vladimir Benko állított össze.
- **hutenten**: A huTenTen korpusz a Lexical Computing Llc. által kifejlesztett TenTen korpuszcsalád (Jakubíček és mtsai, 2013) része és a SketchEngine platform magyar referenciakorpusza. A korpuszt a Lexical Computing LLC állította össze egy 2013-ban végzett gyűjtés (Suchomel és Pomikálek, 2012) alapján, magyar nyelvi elemzését az MNSZ1 (Oravecz és mtsai, 2014) kódjaival Oravecz Csaba, az emMorph (Novák és mtsai, 2016) kódjaival a Lexical Computing Llc. végezte.
- **hírportálok**: Korábbi kutatásainkhoz, főleg szöveg-összefoglaló generálás feladatához összegyűjtött adathalmaz. Különböző hírportálokról gyűjtött cikkek és azok leadjei. A források: index.hu; nol.hu; HVG. Természetesen lehetnek átfedések a CC-ben gyűjtött adatokkal, a duplikációkat a folyamat végén kezeltük.

A gyűjtés végén összekonkatenáltuk az összes forrásból összegyűjtött szövegeket, majd dokumentumszintű `jsonline` formátumra alakítottuk át, ahol egy sor egy json objektum, benne egy `text` mezővel, amiben egy dokumentum szövegei találhatóak, megőrízve a sortöréseket. Ezen a json fájlon végeztünk *dokumentumszintű* deduplikációt (`uniq`) és véletlenszerű keverést. Az 1. táblázatban az utolsó két sorban az összegzett értékeket látjuk, ahol az első sor az összeadott értéket, míg az utolsó sor a deduplikált végső korpusz értékeit mutatja. Ha összevetjük a csak simán összeadott számokat a deduplikált számokkal, akkor azt láthatjuk, hogy van körülbelül 4 milláard szó eltérés. Ez azt jelenti, hogy főleg a CC éves gyűjtései között nem kevés duplikáció szerepel.

Az adatok gyűjtése több lépésben történt. Először a Webkorpusz 2.0 korpuszból és a Common Crawl gyűjtésekből állítottuk össze az első nagy korpuszunkat. Ily módon első körben létrejött egy körülbelül 25 milliárd szavas korpusz. Ez

[7] https://github.com/DavidNemeskey/cc_corpus
[8] http://ucts.uniba.sk/aranea_about/_hungaricum.html

	Dokumentum	Bekezdés	Szó
Webkorpusz 2.0	9 240 709	171 239 297	8 051 677 190
Wikipédia	418 622	6 804 115	124 982 493
CC 2019	8 259 348	199 368 999	5 994 324 578
CC 2020	7 374 175	174 726 213	5 289 809 348
CC 2021	10 681 529	254 848 762	7 702 038 666
CC 2022	2 586 953	61 817 892	1 874 763 279
neticle	30 471 970	85 351 213	1 112 740 383
sajtó	1 682 151	4 103 234	625 098 614
jsi	4 023 083	32 363 186	1 077 066 597
araneum	3 727 984	31 721 824	1 329 200 470
hutenten	6 447 787	164 654 976	2 670 682 031
híportálok	1 326 922	8 503 669	433 558 050
Összesen	86 241 233	1 195 503 380	36 285 941 699
Összesen (uniq)	78 059 419	1 069 655 352	32 425 610 652

1. táblázat. A tanítókorpusz összetétele

látható az 1. táblázat első kettő blokkjában. Ezzel a korpusszal elkezdtünk különböző kísérletek végezni. Ebben a fázisban jöttek létre a későbbi modellek tanításához használt szótárak (*vocab*) is. Összesen kettő szótárat hoztunk létre:

- **BERT szótár**: Szóelem (WordPiece) szótár; méret: 32 203. Néhány specális karaktert adtunk hozzá manuálisan, valamint a szótár végére manuálisan hozzáadtunk 203 darab tokent, amelyet a huBERT szótárából (utolsó 203 elem) emeltünk át. Erre azért volt szükség, mert korábbi kutatásainkban észleltük azt a jelenséget, hogy ha a tokenizálás folyamatába a hagyományos tokenizáláshoz magyar tokenizáló eszközt használunk, mint például a quntoken (Mittelholcz, 2017) eszközt, akkor bizonyos esetekben az írásjel rajtamarad a szón (ami helyes), amit nem tud kezelni az eredeti szótár, mivel az eredeti BERT tokenizáló minden írásjelet leválaszt a szóról. Ennek az a következménye, hogy mivel nem szerepel az írásjel szóelem részeként (csak külön szóelemként), ismeretlen (`[UNK]`) szóelemként kezeli őket.
- **GPT szótár**: Byte-Pair-Encoding (BPE) szótár; méret: 50 000. Néhány specális karaktert adtunk hozzá manuálisan.

A 2. táblázatban láthatóak a tokenizálók összehasonlítása. Referenciaként a huBERT tokenizálóját adtuk meg. Az összehasonlíthatóság végett a Nemeskey (2021) tanulmányban megadott példákat használtuk. Látható, hogy a mi BERT szótárunk nagyon hasonlít a huBERT szótárához. Majdnem minden szó tokenizálása megegyezik, kivétel az "Andersen" szó, ahol a huBERT egybetartotta, míg a mi szótárunk szétszedte. A GPT szótárunk viszont már "töredezettebb" a BERT szótárakhoz képest.

A szótárak elkészítése után elkezdtük tesztelni a tanítószkripteket, kevés lépéssel sikerült betanítani modelleket. Miután sikerült a tanítás, következő lépésként további adatokat adtunk hozzá a korpuszhoz (lásd 1. táblázat harmadik blokk). Így jött létre a 32,4 milliárd szavas korpuszunk, amellyel végül beta-

	huBERT	BERT szótár	GPT szótár
Nemzeti	Nemzeti	Nemzeti	Nemzeti
Andersen	Andersen	And ers e n	A nder sen
labdarúgó	labdarúgó	labdarúgó	lab darúgó
zambiai	z amb iai	z amb iai	z amb iai
megmaradt	megmaradt	megmarad t	meg maradt
hétfő	hétfő	hétfő	hétfő
keddtől	kedd től	kedd től	ked d től
edényben	edény ben	edény ben	edény ben
Hétfőn	Hétfőn	Hétfőn	Hétfőn
tájékoztatták	tájékoztat ták	tájékoztat ták	tájékozt atták
leggazdagabb	leggazdagabb	leggazdagabb	leg gazdagabb
elpártolt	el párt olt	el párt olt	elp árt olt

2. táblázat. Tokenizálók összehasonlítása

nítottuk a modelljeinket. A szövegek egy részét normalizálni kellett. Bizonyos szövegek nem utf-8 kódolásúak voltak, ezeket kellett utf-8 formátumba konvertálni. Továbbá a *sajtó* szövegein metaadat szűrést kellett végrehajtani, mivel nyelvi annotációkat is tartalmazott a szöveg. A *neticle* szövegei esetén pedig bekezdésszintű deduplikációt is végre kellett hajtani.

4. A modellek bemutatása és előtanítása

Kutatásunkban három modellt tanítottunk: Megatron BERT, Megatron GPT-2 és a GPT-NeoX családhoz tartozó 6,7 milliárd paraméteres GPT-3 modellt.

A **PULI BERT-Large** a Megatron BERT (Shoeybi és mtsai, 2019b) magyar változata. A Megatron BERT a Megatron-DeepSpeed projekt része, ami az NVIDIA Megatron-LM modellek tanítását támogatja DeepSpeed technológiával. A projekt tartalmaz egy BERT-Large méretnek megfelelő (345 millió paraméter) BERT nyelvmodell előtanítási implementációt. Kutatásunkban azt a szkriptet használtuk, amelyik egyetlen GPU-n végzi a tanítást. Így a modelltanításhoz egyetlen NVIDIA A100 (80GB) GPU-t használtunk. A tanítás során a legtöbb hiperparaméteren nem változtattunk, kivéve:

– `split`: 994,5,1; kikapcsoltuk az fp16 kapcsolót; `micro-batch-size`: 40; `global-batch-size`: 320.

A vágáson (`split`) azért módosítottunk, hogy több anyagot használjunk fel a tanításhoz. A micro-batch-size méretet empirikusan választottuk ki, ennyi fért bele a 80 GB-os GPU memóriába. A `global-batch-size` méret választása esetében 8 darab GPU használatát szimuláltuk. A tanítást 750 000 lépésnél állítottuk le, ami körülbelül 2 és fél epoch környékén van. Az eredeti BERT modellek tanítása 256 batch méreten történt (Devlin és mtsai, 2019) körülbelül 40 epoch a 3,3 milliárd szavas korpuszon, ami átszámítva a mi méretünkre körülbelül 4 epochnak felel meg. A végső veszteségi értékek (*loss*):

– Nyelvmodell (*language model* - lm) validációs veszteség (`lm loss value`): 2,32; perplexitás (`lm loss PPL`): 10,19.
– Mondatsorrend predikció (*sentence order prediction* - sop) validációs veszteség (`sop loss value`): 0,14; perplexitás (`sop loss ppl`): 1,15.

A tanítási idő körülbelül két hónap volt. Végül az elkészült modellt a Hugging Face által közzétett szkripttel[9] konvertáltuk a Hugging Face által használt formátumra.

A **PULI GPT-2** a Megatron GPT-2 magyar változata. A Megatron GPT-2 szintén a Megatron-DeepSpeed projekt része, amely egy 345 millió paraméteres GPT-2 típusú modell. Hasonlóan a BERT modell tanításához, az egy GPU-s implementációt választottuk, vagyis egy NVIDIA A100 (80GB) GPU-t használtunk. A hiperparaméter beállításakor hasonlóan jártunk el, mint a BERT tanítása esetén:

– `split`: 994,5,1; kikapcsoltuk az fp16 kapcsolót; `micro-batch-size`: 13; `global-batch-size`: 52.

A tanítást 500 000 lépésnél állítottuk le, ami körülbelül 1 epoch. A végső veszteségi értékek (*loss*):

– Nyelvmodell validációs veszteség (`lm loss value`): 2,80; perplexitás (`lm loss ppl`): 16,46.

A tanítási idő 20 nap volt. Az elkészült modellt a Hugging Face által közzétett szkripttel[10] konvertáltuk Hugging Face által használt formátumra. A konvertáló szkriptben módosítani kellett a `vocab_size`, `bos_token_id` és az `eos_token_id` paramétereket, mivel saját szótárat használtunk.

A **PULI GPT-3SX** a GPT-NeoX (Andonian és mtsai, 2021; Black és mtsai, 2022) magyar változata. A GPT-NeoX az EleutherAI[11] projektje. A projekt célja, hogy nagy paraméterszámú nyelvmodelleket lehessen tanítani, mint a GPT-3. Egyik konkrét céljuk, hogy reprodukálják a GPT-3 modellt. Implementációjuk az NVIDIA Megatron-LM és a DeepSpeed technológián alapszik. Többféle GPT-3 jellegű konfigurációt állítottak össze, kicsi modellektől (pl.: 160 millió paraméter) a közepes méretűeken át (pl.: 20 milliárd paraméter) a nagy modellekig (175 milliárd paraméter). Kutatásunkban egy viszonylag kicsinek számító 6,7 milliárd paraméteres konfigurációt használtunk. A tanításhoz 8 darab NVIDIA A100 (80GB) GPU-t használtunk. A betanítást a hiperparaméterek nagy részének a módosítása nélkül végeztük. Egyedül a *batch* méretet módosítottuk, amely empirikus módon (ennyi fér bele a 80 GB-os GPU-ba) a 16-ra állítottunk. A tanítást 150 000 lépésnél állítottuk meg, ami számításunk alapán egy kicsivel több mint 1 epoch. A tanítási idő 21 nap volt. A végső veszteségi értékek (*loss*):

[9] https://github.com/huggingface/transformers/tree/main/src/transformers/models/megatron_bert
[10] https://github.com/huggingface/transformers/tree/main/src/transformers/models/megatron_gpt2
[11] https://www.eleuther.ai

– Nyelvmodell validációs veszteség (`lm loss value`): 2,17; perplexitás (`lm loss ppl`): 8,76.

A jövőben szeretnénk még további kísérleteket végezni a modellek tovább tanításával.

5. A modellek finomhangolása

A modellek kiértékeléséhez többféle módszert is választottunk. A Megatron BERT és a GPT-2 esetében a finomhangolás módszerével teszteltük. Ehhez először, az összehasonlíthatóság végett, a magyar szakirodalomban (Nemeskey, 2020a; Yang és Váradi, 2021; Yang Zijian Győző, 2022; Yang Zijian Győző, 2022) gyakran használt három tokenszintű osztályozó és kettő mondatszintű osztályozó korpuszt használtunk fel: NYTK-NerKor (NerKor) (Simon és Vadász, 2021), Szeged NER (SzNer) (Szarvas és mtsai, 2006), Szeged Treebank (NP) (Csendes és mtsai, 2005) és Magyar Twitter Szentiment korpuszt (Szabó és mtsai, 2016). A szentiment korpusz esetében a Laki és Yang (2021) által készített két alkorpuszt használtuk fel: 2 osztályos (2-o) és 5 osztályos (5-o). A kiértékelési metrikának az F1 és az abszolút pontosságot választottuk (accuracy - acc). A finomhangoláshoz a Hugging Face által közzétett függvényeket használtuk, ugyanazokat és ugyanolyan beállításokkal, mint Yang és Váradi (2021) a kutatásukban, annyi különbséggel, hogy a mi esetünkben 4 darab NVIDIA A100 GPU állt a rendelkezésünkre.

Következő lépésként a 2022-ben megjelent magyar benchmark korpuszon, a HuLU (Ligeti-Nagy és mtsai, 2022) korpuszon értékeltük ki. Kiértékelési metrikának az abszolút pontosságot választottuk (accuracy - acc), kivétel a HuRC, ahol a pontos egyezés és az F1 metrikákat. Időközben megjelent a HuLu hivatalos kiértékelő oldala[12], ahol némelyik korpusznál az MCC (Matthew's correlation coefficient) (Matthews, 1975) metrikát használják, így az itt elért MCC eredményeket is beillesztettük a tanulmányba. A HuCOLA, HuSST, HuWNLI és HuRTE feladatok finomhangolásához a Hugging Face szövegosztályozó szkriptjét használtuk (ugyanazt, mint amit a szentiment analízis modell finomhangolásához használtuk). A HuRC esetében a Hugging Face kérdés-válasz (question answering) szkriptjét[13], míg a HuCoPA feladathoz a többválasztós (multiple choice) szkriptet[14]. Minden finomhangoláshoz 4 darab NVIDIA A100 GPU állt a rendelkezésünkre.

A HuLU korpuszokon való finomhangolás részletei egy külön tanulmány témáját jelentik. Minden modell és feladat más-más hiperparaméter-beállításokat igényelt. Továbbá a HuWNLI, a HuCoPA, a HuRTE és a HuRC korpuszok esetében is különböző bemeneti adatstruktúrákkal, úgynevezett promptt kísérleteztünk, amelyek hatással voltak a teljesítményre. A két fő kísérleti irány, hogy

[12] https://hulu.nytud.hu

[13] https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering

[14] https://github.com/huggingface/transformers/tree/main/examples/pytorch/multiple-choice

a vizsgálandó szövegeket szeparátor címkével (SEP) válasszuk el, vagy egy egységes folyószöveggé (Szöveg) alakítsuk át őket. Egy-egy példa a két típusra:

- Eredeti:
  - *premise*: A sofőr felkapcsolta az autó fényszóróit.
  - *choice1*: Mennydörgést hallott.
  - *question*: cause
- SEP: A sofőr felkapcsolta az autó fényszóróit. [SEP] Mennydörgést hallott.
- Szöveg: A sofőr felkapcsolta az autó fényszóróit. Mert mennydörgést hallott.

A fenti példában a szeparátor címke lehet akár `[CLS]` vagy a GPT modelleknél `</s>` stb. A folyószöveggé alakítás során a fenti példában két mondatot látunk, de lehet egy mondatba is foglalni őket vesszővel elválasztva stb. Minden modellnél a különböző prompt különböző hatással van a teljesítményre. A GPT-3 modell esetében különösen nagy jelentősége van a megfelelő promptok kiválasztásában.

A legnehezebb feladat a GPT-3 modell kiértékelése volt. Mivel a GPT-3 finomhangolása nagy erőforrást igényelt volna, első körben prompt programozással (Reynolds és McDonell, 2021) végeztünk kísérleteket. A prompt programozás azt jelenti, hogy nincsen finomhangolás, csupán, a predikció során, a bemeneti adat manipulálásával érjük el, hogy a modell a kívánt kimenetet generálja. Az adott feladathoz a megfelelő prompt választása nem könnyű feladat, amelyre külön kutatások vannak (Shin és mtsai, 2020; Alivanistos és mtsai, 2022). A promptok előállítása történhet manuálisan és automatikus módszerekkel. A kutatások alapján az egyes nyelvmodellek eltérő módon reagálnak a különböző promptkészletekre, sőt a különböző feladatokra is (ahogy fent láthattuk). A promptok szerkezeti felépítése és a megfelelő promptok kiválasztása nagyban meghatározza a modell teljesítményét. A jövőben szeretnénk ennek egy külön kutatást szentelni. A mostani kutatásban manuálisan hoztuk létre a promptokat és empirikus módon állítottuk a paramétereket. Többféle prompttal és beállítással kísérleteztünk, végül a legjobbnak tűnő beállítást tartottuk meg. A generált válasz manipulálásában a következő paraméterekkel tudtunk kísérletezni: bemeneti prompt szöveg; hőmérséklet; top-k; top-p és kimeneti szöveg hossza. A kimeneti szöveg hossza minden esetben 3 volt megadva, hiszen mindig csak egy-egy osztálycímke, vagy egy-egy szót vártunk kimenetnek. Azért adtunk meg 1-nél nagyobb értéket, mivel a modell időnként sortöréssel kezdte a szöveget, illetve hogy tudjunk olyan kiemeneti szóval kísérletezni, amit esetleg több szóra darabolhat a tokenizáló. Ahhoz, hogy csökkentsük a modell 'kreativitását' a top-k, top-p és hőmérséklet értékét is alacsonyra kellett állítani. Első tapasztalataink alapján a top-k értékét 10 körüli értékre állítottuk, a top-p értékét 0,1-0,4 közé, a hőmérsékletet pedig 0,1-0,4 környékére. A prompt esetében vettük a tanítóanyag első X szegmensét, ahol minden szegmenshez egy szeparátor karakterrel elválasztva hozzákonkatenáltuk a kimeneti osztálycímkét. Az osztálycímkék esetében kipróbáltuk a szám (0;1) és különböző szöveges (igen; nem; pozitív; negatív; jó; rossz stb.) változatot. Az X kiválasztásánál az volt az elsődleges szempont, hogy lehetőleg lásson minden osztálycímkéből több példát is. Túl sok példát a bemeneti hossz korlátja miatt nem is tudtunk volna megadni. Tapasztalataink alapján 30-nál több példánál nem nagyon tudott jól osztályozni a modell. A modell az utolsó néhány

példára hajlamos jobban odafigyelni, ezért nem mindegy, hogy mi az utolsó példa amit lát. A 3. táblázatban látható egy példa egy lehetséges promptra. A dőlt betűvel szedett sor a tesztelni kívánt mondat. A modell a megadott szeparátor karakterre folytatja a szöveg generálását. Mivel 3 a kimeneti hossz értéke, ezért még folytatja a generálást a modell, így mi csak a generált szöveg első szavát vesszük figyelembe. Ahogy a példában látszik, egy sortörést és egy *A* betűt generált még a modell.

Hatásokkal teli, de túl langyos filmbiográfia. = negatív
Ha szeretsz időnként moziba menni, érdemes a Wasabi-val kezdeni. = pozitív
A szórakoztatás és az oktatás ritka kombinációját kínálja. = pozitív
Azon töpreng, hogy miért van szükségünk annyira a történetekre. = semleges
Szellemtelen és teljesen értelmetlen. =
Kimenet: negatív [sortörés] A

3. táblázat. Példa egy promptra

A következő beállításokat alkalmaztuk a különböző feladatokban:

- HuCOLA: prompt #: 25; hőmérséklet: 0,1; top-p: 0,12; top-k: 10;
- HuSST: prompt #: 29; hőmérséklet: 0,3; top-p: 0,1; top-k: 10;
- HuWNLI: prompt #: 15; hőmérséklet: 0,3; top-p: 0,3; top-k: 10;
- HuRTE: prompt #: 14; hőmérséklet: 0,3; top-p: 0.4; top-k: 10;

6. Eredmények

A 4. táblázatban láthatóak a NER, NP és szentimentanalízis eredmények, a 5. táblázatban pedig a HuLU eredmények. A mostani eredmények a jövőben változhatnak, ha tovább tanítjuk a modelljeinket.

A 4. táblázatban az látható, hogy a PULI BERT-Large modellünk az esetek nagy részében felülmúlta a huBERT modellt. Egyedül az NP feladatban marad le elhanyagolható mértékben. Ahogy az várható volt, az új PULI GPT-2 modellünk felülmúlta a kicsi méretű NYTK-GPT-2 modellt, azonban a BERT modelleket nem, ez sem meglepő, mivel a GPT típusú modelleknek nem az osztályozás az erősségük.

	NerKor (F1)	SzNER (F1)	NP (F1)	2-o (acc)	5-o (acc)
huBERT	90,18	97,51	**96,97**	85,92	68,50
NYTK-GPT-2 small	69,43	88,06	85,02	80,74	61,00
PULI BERT-Large	**91,06**	**97,55**	96,96	**86,29**	**68,99**
PULI GPT-2	71,45	88,25	87,73	81,48	63,75

4. táblázat. NER, NP és szentimentanalízis eredmények

A 5. táblázatban látható a HuLU korpuszain történő kiértékelés, a 6. táblázatban látható a HuCoPA, a HuCOLA és a HuRTE feladatokon mért MCC értékek. A GPT modellek esetében nem mindegyik feladatra tudtunk tesztelni. A HuCoPA egy úgynevezett többválasztásos feladat, amit nem tudtunk első körben finomhangolni a GPT-2 modellel. A GPT-3 esetében viszont még nem sikerült kitapasztalni a promptot erre a típusú feladatra. Hasonlóképpen a HuRC feladatára sem tudtuk most tesztelni a GPT modelleket. A számok alapján a legnehezebb feladat a HuWNLI volt, szinte mindegyik modell 66% alatt teljesített. Ha azt nézzük, hogy ez a feladat tulajdonképpen egy bináris osztályozó feladat, akkor a 66% rendkívül alacsonynak számít, ebből is az következik, hogy ez egy nehéz feladat. A huBERT és a Megatron BERT kimagaslóan teljesít a GPT modellekhez képest, ez várt eredmény, hiszen a GPT modelleknek nem erősségük az osztályozás. Természetesen a GPT-3 a leggyengébben teljesítő modell, mivel külön kutatás szükséges a megfelelő promptok használatára, a mostani kutatás nem erre fókuszált.

A HuLU már jobban árnyékolja a BERT modellek működését. Több feladaton is a huBERT vette át a vezetést. A HuCoPA és a HuRC esetében a Megatron BERT lemarad a huBERT modellhez képest. Ebből arra lehet következtetni, hogy az osztályozási feladatokra jobban be lehet tanítani a Megatron BERT modellt, de a komplexebb nyelvi feladatokban, mint a többválasztásos vagy a kérdés-válasz feladatokban van még hova fejlődnie. Úgy gondoljuk, hogy ez a modell továbbtanításával fejleszthető lehet.

	HuCoPA (acc)	HuCOLA (acc)	HuSST (acc)	HuRC (egyezés/F1)	HuWNLI (acc)	HuRTE (acc)
huBERT	**77,99**	91,43	79,40	**64,50/69,03**	64,93	74,05
PULI BERT-Large	76,59	**91,65**	**79,91**	60,38/65,57	**65,67**	**75,88**
PULI GPT-2	-	85,93	72,79	-	61,94	70,66
PULI GPT-3SX few-shot	-	74,18	68,84	-	63,43	56,33

5. táblázat. HuLU eredmények

	HuCoPA (MCC)	HuCOLA (MCC)	HuRTE (MCC)
huBERT	**0,561**	0,709	0,487
PULI BERT-Large	0,414	**0,711**	**0,517**
PULI GPT-2	-	0,499	0,415

6. táblázat. HuLU MCC eredmények

A GPT-3 teljesítménye rendkívül bizonytalan. Az adott paraméterek mellett minden predikciónál más-más eredményt ad. Valamennyire lehet ezt korlátozni és kontrollálni, de így is rendkívül érzékeny. A különböző promptokkal és

hiperparaméter-beállításokkal végzett többszöri futtatásból a legjobb eredményt írtuk be a 5. táblázatba. A jövőben külön kutatást szeretnénk szentelni a GPT-3 modell kiértékelésére.

Végül a 7. táblázatban látható néhány GPT-3 által generált szöveg különböző hőmérsékleten. A modellnek a következő kérdést tettük fel: 'Mit gondolsz a nyelvtechnológiáról?' A kiemeneti hossz 256 volt, azonban ahol úgy láttuk, hogy a modell témát vált, ott levágtuk a szöveg végét, így a válaszoknak az első néhány mondatai láthatóak csak. Láthatjuk, hogy mindegyik generált bekezdés jól-formált, jól-struktúrált és nyelvileg is helyes.

Mit gondolsz a nyelvtechnológiáról?
1. válasz (0,5 hőmérséklet):
A nyelvtechnológia egy olyan technológia, amely a nyelveket a mesterséges intelligencia segítségével közelíti meg.
2. válasz (0,6 hőmérséklet):
A nyelvtechnológia egy új technológia, ami lehetővé teszi, hogy a számítógépek közvetlenül az emberek agyát használják. Ha az agyat egy számítógéphez kapcsoljuk, akkor képes lesz megérteni a nyelvet. A nyelvtechnológia, a nyelvi feldolgozás és a nyelvtechnológia együtt képesek lesznek arra, hogy a nyelvet az emberek gondolkodásához igazítsák. Nem tudom, hogy hol tart a nyelvtechnológia, de azt tudom, hogy a nyelvtechnológia a következő lépés, és szerintem a nyelvtechnológia a következő nagy lépés.
3. válasz (0,7 hőmérséklet):
A nyelvtechnológia a mesterséges intelligencia egyik ága. A mesterséges intelligencia a gépi tanulással és a nyelvtechnológiával együtt fejlődve alakítja az emberi nyelvet és annak használatát.

7. táblázat. GPT-3 által generált szövegek

7. Összegzés

Kutatásunkban betanítottunk egy 6,7 milliárd paraméteres GPT-3 modellt, egy 345 millió paraméteres GPT-2 modellt és egy 345 millió paraméteres BERT-Large modellt. A GPT-3 modellünk az első egynyelvű GPT-3 modell magyar nyelvre. Több nyelvtechnológiai feladaton is teszteltük modelljeinket. A PULI BERT-Large modellünk a legtöbb feladatban felülmúlta a huBERT modellt. A PULI GPT-3SX modellünkkel magyar nyelvre elsőnek végeztünk prompt programozással, *few-shot* módszerrel kiértékelést. A PULI GPT-3SX számai még messze alulmaradnak a többi modellhez képest, azonban külön kutatást igényel a megfelelő promptok és hiperparaméterek beállítása az adott feladatokra. Azonban képes olyan szöveget generálni, ami hasonlít egy ember által írt szövegre.

Hivatkozások

Alivanistos, D., Santamaría, S.B., Cochez, M., Kalo, J.C., van Krieken, E., Thanapalasingam, T.: Prompting as Probing: Using Language Models for Knowledge Base Construction (2022)

Andonian, A., Anthony, Q., Biderman, S., Black, S., Gali, P., Gao, L., Hallahan, E., Levy-Kramer, J., Leahy, C., Nestler, L., Parker, K., Pieler, M., Purohit, S., Songz, T., Phil, W., Weinbach, S.: GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch (8 2021), https://www.github.com/eleutherai/gpt-neox

Benko, V.: Aranea: Yet Another Family of (Comparable) Web Corpora. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (szerk.) Text, Speech and Dialogue. pp. 247–256. Springer International Publishing, Cham (2014a)

Benko, V.: Compatible Sketch Grammars for Comparable Corpora. In: Abel, A., Vettori, C., Ralli, N. (szerk.) Proceedings of the 16th EURALEX International Congress. pp. 417–430. EURAC research, Bolzano, Italy (jul 2014b)

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U.S., Purohit, S., Reynolds, L., Tow, J., Wang, B., Weinbach, S.: GPT-NeoX-20B: An Open-Source Autoregressive Language Model. In: Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models (2022), https://arxiv.org/abs/2204.06745

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (szerk.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: PaLM: Scaling Language Modeling with Pathways (2022)

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (szerk.) Text, Speech and Dialogue. pp. 123–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.Gy., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., Leahy, C.: The Pile: An 800GB Dataset of Diverse Text for Language Modeling (2020)

He, J., Qiu, J., Zeng, A., Yang, Z., Zhai, J., Tang, J.: FastMoE: A Fast Mixture-of-Expert Training System (2021)

Indig, B.: Közös crawlnak is egy korpusz a vége – Korpuszépítés a CommonCrawl .hu domainjából. In: Vincze, V. (szerk.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018). p. 125–134. Szegedi Tudományegyetem Informatikai Intézet, Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2018)

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. In: 7th International Corpus Linguistics Conference CL 2013. pp. 125–127. Lancaster (2013)

Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018), https://aclanthology.org/D18-2012

Laki, L., Yang, Z.Gy.: Improving Performance of Sentence-level Sentiment Analysis with Data Augmentation Methods. In: Proceedings of 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021). pp. 417–422. IEEE, Online (2021)

Ligeti-Nagy, N., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Laki, L.J., Vadász, N., Yang, Z.Gy., Váradi, T.: HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. p. 431–446. JATEPress, Szeged (2022)

Matthews, B.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure 405(2), 442–451 (1975)

Mittelholcz, I.: emToken: Unicode-képes tokenizáló magyar nyelvre. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 61–69. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2017)

Nemeskey, D.M.: Egy emBERT próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged (2020a)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020b)

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Novák, A., Siklósi, B., Oravecz, Cs.: A New Integrated Open-source Morphological Analyzer for Hungarian. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)

Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners (2019)

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research 21(140), 1–67 (2020)

Reynolds, L., McDonell, K.: Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. CHI EA '21, Association for Computing Machinery, New York, NY, USA (2021)

Rychlý, P.: Manatee/Bonito - A Modular Corpus Manager. In: 1st Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 65–70. Masarykova univerzita, Brno (2007)

Shazeer, N.: GLU Variants Improve Transformer (2020)

Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4222–4235. Association for Computational Linguistics, Online (Nov 2020)

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B.: Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism (2019a)

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B.: Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism (2019b)

Simon, E., Vadász, N.: Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., Zhang, E., Child, R., Aminabadi, R.Y., Bernauer, J., Song, X., Shoeybi, M., He, Y., Houston, M., Tiwary, S., Catanzaro, B.: Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model (2022)

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: RoFormer: Enhanced Transformer with Rotary Position Embedding (2021)

Suchomel, V., Pomikálek, J.: Efficient Web Crawling for Large Text Corpora. In: Kilgarriff, A., Sharoff, S. (szerk.) Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 39–43. Lyon (2012)

Szabó, M.K., Vincze, V., Simkó, K.I., Varga, V., Hangya, V.: A Hungarian sentiment corpus manually annotated at aspect level. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 2873–2878. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016)

Szarvas, G., Farkas, R., Kocsor, A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: Todorovski, L., Lavrač, N., Jantke, K.P. (szerk.) Discovery Science. pp. 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is All you Need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)

Wang, B., Komatsuzaki, A.: GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax (May 2021)

Yang, Z.G., Váradi, T.: Training language models with low resources: RoBERTa, BART and ELECTRA experimental models for Hungarian. In: Proceedings of 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021). pp. 279–285. IEEE, Online (2021)

Yang Zijian Győző: BARTerezzünk! - Messze, messze, messze a világtól, - BART kísérleti modellek magyar nyelvre. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–29. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Yang Zijian Győző: "Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 463–476. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Látens szemantikus eloszlások használata a nyelvi modellek előtanítása során

Berend Gábor

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

Kivonat Cikkünk egy olyan variánsát mutatja be a nyelvi modellek előtanításának, amely során a maszkolás tárgyául nem a véletlenszerűen kiválasztott tokenek rekonstruálását, hanem azok szemantikus kategóriájának megállapítását tűzzük ki célul. A javasolt módon létrehozott modelljeink finomhangolását változatos benchmarkokon elvégezve azt találjuk, hogy azok szignifikánsan jobb eredmény elérésére képesek hagyományos társaikhoz képest.
Kulcsszavak: előtanítás; szemantikus klaszterek

1. Bevezetés

A komoly eredményeket felmutató modern természetesnyelv-feldolgozó modellek alapjául tipikusan a nyelvi modellezési feladat ellátására önfelügyelt módon létrehozott, transzformer architektúrát használó (Vaswani és mtsai, 2017) modellek szolgálnak. A modellek a nyelvi modellezés elsajátítása során először egy általános nyelvmodellező képességre tesznek szert, amit aztán egy-egy specializált feladat megoldására transzferálhatunk a célfeladat doménjéből jövő felügyelt tanítópéldák felhasználása segítségével. Az előbbi lépést szokás szerint előtanításnak, míg az utóbbit a célfeladatra történő finomhangolásnak hívjuk.

Az előtanítás során két fő megközelítést szokás alkalmazni: az autoregresszív, valamint a maszkolásos módszereket. Az autoregresszív modellek a megkezdett szövegek minél valósághűbb folytatását tűzik ki feladatukul, a céljuk tulajdonképpen a minél kisebb perplexitással rendelkező szövegek generálása.

A maszkolást alkalmazó módszerek ezzel szemben egy teljes mondatot (vagy szövegrészt) kapnak inputul a tanítás során, majd az ezekből a mondatokból véletlenszerűen kitakart szavak (illetve szótöredékek) rekonstruálását tűzik ki célukul az előtanítás során, ami gyakorlatilag az autoenkóderek működéséhez hasonló viselkedést eredményez – annyi különbséggel, hogy a maszkolásos nyelvi modellek nem folytonos értékek, hanem diszkrét szimbólumok rekonstruálására törekszenek.

Munkánkban a maszkolással tanított nyelvi modellek előtanításának egy alternatíváját kívánjuk bemutatni, amely során nem a konkrét kitakart szó(töredék)et, hanem annak (látens) szemantikai jellemzésének megállapítását adjuk a nyelvi modellezést elvégző neurális háló feladatául. Noha cikkünkben az autoregresszív modellekkel nem foglalkozunk, a javasolt eljárás azokra is kiterjeszthető lenne.

A javasolt módszert motiválandó, vegyük azt az egyszerű mondatot, hogy „`Mari süteményt eszik`”. Amennyiben az előtanítás során a kitakarásra véletlenszerűen kiválasztott szimbólum a „`süteményt`” szó, úgy a neurális modellünk veszteségfüggvénye abban az esetben lesz a lehető legkisebb *ezen példa mentén*, ha a háló a kitakart szó vonatkozásában kizárólag a „`süteményt`” szót tartaná behelyettesítésre alkalmasnak, míg más – hétköznapi tudásunkkal egyébként teljességgel összeegyeztethető – szavak (mint pl. „`sóletet`”, „`disznósajtot`”, *stb.*) vonatkozásában a kitakart szó helyén való előfordulás eshetőségét teljességgel kizárná.

A nyelvi modell végső célja szempontjából egy természetesebb, és vélhetőleg jobb mintahatékonysággal bíró – gyakorlati szempontból ugyanakkor jóval problematikusabb – célfüggvény a modell által egy-egy adott szó kitakarása mentén visszaadott, a szótár elemei fölött értelmezett eloszlást azzal a – jellemzően nem a teljes valószínűségi tömeget egy adott szó mentén tartalmazó – eloszlással vethetné össze, amely a kitakart szó adott kontextusa mentén az egyes szavak helyettesítőként való előfordulásának valószínűségét tartalmazná. Kellő megbízhatóságú tanítóadatot minderre a nyelvhasználatot jellemző adatritkaságból fakadóan szinte lehetetlen lenne találni.

Egy alternatív előtanítási folyamatot úgy is elképzelhetünk, hogy valamilyen szemantikai erőforrásra támaszkodva (pl. WordNet (Fellbaum, 1998; Miháltz és mtsai, 2008) vagy ConceptNet (Speer és Havasi, 2012)) a kitakart szavak ontologikus tulajdonságainak meghatározását várnánk el a nyelvi modellek előtanítása során. Ebben az esetben a korábban látott példamondat („`Mari süteményt eszik`”) esetén a `sütemény` szó kimaszkolása esetén nem a konkrét szó rekonstruálása lenne a feladatunk, hanem például annak meghatározása, hogy a kitakart szó helyén egy *ehető* fogalom állt. Egy ilyen módon működő előtanítás azonban feltételezi egy kellően expresszív tudásbázis meglétét, mi több, a nyelvi modell tanítására használt szövegek vonatkozásában az az információ is rendelkezésre kell álljon, hogy az egyes szóelőfordulásokra az éppen adott kontextusuk mentén a különböző lehetséges jelentéseik közül mely szemantikus tulajdonság(ok) teljesül(nek).

Ilyen szemantikus részletességgel annotált tanítóadatbázis meglétében az előtanításhoz szükséges mennyiségben aligha reménykedhetünk. Az általunk javasolt módszer éppen ennek a szemantikus annotációval ellátott előtanító anyag kiváltására tesz javaslatot oly módon, hogy a kontextuális szóreprezentációk alapján felügyelet nélküli módon látens tulajdonságok teljesülését rendeljük az egyes szóalakokhoz, megspórolva ezzel a módosított előtanításhoz szükséges szemantikus emberi annotáció elvégzésének szükségességét.

2. Kapcsolódó munkák

A különböző transzformer architektúrák (Vaswani és mtsai, 2017) mára meghatározó jelentőségűvé váltak a mesterséges intelligencia számos területén. A kontextusérzékeny jelentésreprezentációkat megalkotni képes BERT modell (Devlin és mtsai, 2019) mutatott rá először az angol nyelv vonatkozásában az – erede-

tileg gépi fordításra kifejlesztett – architektúra általános és változatos nyelfeldolgozási feladatokon való alkalmazhatóságára. A modell sikerein felbuzdulva, mára rengeteg nyelv támogatására készítettek BERT variánsokat (Vilares és mtsai, 2021; Martin és mtsai, 2020; Le és mtsai, 2020; Ulčar és Robnik-Šikonja, 2020). A nyelvspecifikus BERT modellek egy számunkra kifejezett fontossággal bíró képviselője a dedikáltan a magyar nyelv feldolgozására specializált HuBERT (Nemeskey, 2021).

A különböző forrásokból származó külső tudások előtanított nyelvi modellekbe történő integrálására számos kísérlet született már (Mihaylov és Frank, 2018; Bauer és mtsai, 2018; Peters és mtsai, 2019; Ye és mtsai, 2019; Yang és mtsai, 2019; Qiu és mtsai, 2019; Levine és mtsai, 2020; Liu és mtsai, 2020). A javasolt megoldások széles spektrumon mozognak aszerint, hogy a külső tudást milyen módon használják föl, az azonban közös bennük, hogy mind valamilyen explicit tudásra, pl. manuálisan létrehozott tudásbázisokra támaszkodnak.

A mi megközelítésünk alapvetően abban különbözik a korábbi megoldásoktól, hogy az nem igényel explicit külső szemantikus erőforrást, az előtanulás során használt szemantikus kategóriákat felügyelet nélkül hozzuk létre. Ezek az adatvezérelt szemantikus kategóriák bizonyos tekintetben hasonlítanak a Brown klaszterekre (Brown és mtsai, 1992), fontos különbség azonban, hogy míg a Brown klaszterek kialakítására a szóalakok szintjén kerül sor, addig a mi módszerünkkel a szó*előfordulások* szintjén kaphatunk egy szemantikus kategorizációt.

3. Módszer

Az egyes szóelőfordulásokhoz tartozó látens szemantikus kategóriákat kontextusérzékeny módon előállító eljárásunkat a Berend (2020) által javasol módon hajtottuk végre. Az eljárás első lépése, hogy a transzformer modell által a felügyelet nélküli modellalkotáshoz fölhasznált mondatok inputszimbólumaihoz rendelt, egységnyi hosszúvá normalizált h-dimenziós rejtett állapotaiból képzett $\boldsymbol{X} \in \mathbf{R}^{h \times n}$ mátrixra az alábbi optimalizálási feladatot oldjuk meg

$$\min_{\boldsymbol{D},\boldsymbol{\alpha} \in \mathbb{R}^{k \times n}_{\geq 0}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{\alpha}\|_F^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{1}$$

amely alapján egy későbbiekben látott $\boldsymbol{x}_i \in \mathbf{R}^h$ kontextuális reprezentációs vektorhoz már a

$$\min_{\boldsymbol{\alpha_i} \in \mathbb{R}^k_{\geq 0}} \frac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha_i}\|_2^2 + \lambda \|\boldsymbol{\alpha_i}\|_1 \tag{2}$$

módon határozható meg annak látens szemantikus jellemzését adó $\boldsymbol{\alpha_i} \in \mathbb{R}^k_{\geq 0}$ vektora.

Fontos különbség (1) és (2) között, hogy az utóbbi esetében $\boldsymbol{D}$-ben nem optimalizálunk, azaz a kontextuális modell használata során kapott rejtett vektorokat a korábbi lépésben meghatározott $\boldsymbol{D}$ mátrixra támaszkodva igyekszünk ℓ_1 regularizáció használata mellett minél pontosabban felírni a $\boldsymbol{D}$-beli vektorok lineáris kombinációjaként.

A regularizációs tag azt eredményezi, hogy a $\boldsymbol{\alpha_i}$-beli együtthatók között több is pontosan 0 értéket fog fölvenni, míg a nemnulla együtthatók elhelyezkedése olyan lesz, hogy abból szemantikus tulajdonságok olvashatók ki (Berend, 2020).

Mivel az $\boldsymbol{\alpha_i}$-beli együtthatókkal kapcsolatban egy nemnegativitási feltételt is kikötöttünk, $\boldsymbol{\alpha_i}$ vektorra (a benne található együtthatók összegével való normalizálást követően) viszonylag természetes módon egy olyan eloszlásként is tekinthetünk, ami azt az információt hordozza, hogy egy kérdéses szóelőfordulás mekkora valószínűséggel tekinthető a k különböző felügyelet nélküli módon az (1) segítségével meghatározott látens szemantikus kategóriába tartozónak.

A módosított előtanítási eljárás során a humán annotáció útján rendelkezésre álló szemantikus tudás hiányát ezzel a (normalizáláson átesett) $\boldsymbol{\alpha_i}$ vektorral helyettesíthetjük. Mindehhez a transzformer alapú nyelvi modellünk szótárába fölveszünk k új speciális szimbólumot, amelyek mindegyike egy-egy látens szemantikus kategóriának felel meg, és ahelyett, hogy a maszkolásos nyelvi modellezés során a kitakart szó(töredék) identitásának helyreállítására törekednénk, a k speciális szimbólumra támaszkodva azt vizsgáljuk meg a KL-divergencia használatával, hogy a modell által a k speciális szimbólumra visszaadott multinomiális eloszlása mennyire tér el az adott szóelőfordulás mentén a (2) alapján meghatározott $\boldsymbol{\alpha_i}$ szemantikus eloszlástól.

4. Kísérletek

Kísérleteinket a HuBERT modellen (Nemeskey, 2021) végeztük a `transformers` (Wolf és mtsai, 2020) könyvárra támaszkodva. Az (1) megoldásához a Hungarian Webcorpus 2.0 (Nemeskey, 2020) Wikipédiáról származó 100 000 véletlenszerűen kiválasztott mondatából azon szóelőfordulások kontextuális reprezentációiból alkottuk meg az $\boldsymbol{X}$ inputmátrixot, amelyek olyan szóalakokhoz tartoztak, amelyeknek az adott mintán belüli előfordulása nem haladta meg az 1 000 darabot.

A kiválasztott mondatokban szereplő mintegy $2,7$ millió szó(töredék) nagyjából kétharmadát, közel $1,8$ millió szó(töredéke)t használtunk föl a $\boldsymbol{D}$ mátrix meghatározásához. A λ, illetve k hiperparamétereket (Berend, 2020) alapján rendre 0,05-nek, illetve 3000-nek választottuk, a rejtett kontextuális reprezentációkat pedig a transzformer modell utolsó (tizenkettedik) rétegéből vettük.

Az előtanítást a Hungarian Webcorpus 2.0 20 millió véletlenszerűen kiválasztott mondatán hajtottuk végre gradiens akkumuláció használata mellett, 2048-as virtuális batchmérettel. Az előtanított modellt a HuBERT modell súlyaival inicializáltuk, és a javasolt előtanítási módszert használva 10 000 modellfrissítési lépést hajtottunk végre. Az előtanítás során a tanulási ráta, valamint az egyes tokenek kimaszkolási valószínűsége gyanánt az irodalomban gyakran alkalmazott 0,0001, illetve 0,15 értékeket használtuk.

Baselineként egy olyan, a klasszikus – szemantikus eloszlások helyett szótöredéken alapuló – maszkolt nyelvi modellezési feladattal továbbtanított modellt is létrehoztunk, ami a célfeladattól eltekintve az általunk javasolt modellel teljesen azonos módon került továbbtanításra.

A továbbiakban a szemantikus eloszlások mentén a KL-divergenciával továbbtanított modellünket HuBERTkl-ként, a klasszikus maszkolt nyelvi modellezéssel továbbtanított baseline modellünket pedig HuBERTmlm-ként fogjuk hivatkozni. A különböző célfüggvények használatával továbbtanított modellek összehasonlíthatóságát azzal teremtettük meg, hogy mindkét modellt ugyanazokkal a kezdeti súlyokkal inicializáltuk, a frissítésük során pedig megegyező tartalmú batcheket dolgoztunk föl.

A modellünk alapját képző szemantikus kategóriákat kialakító felügyelet nélküli módszert először kvalitatív szempontból vizsgáljuk, majd az erre építő előtanítással létrehozott modellünk kvantitatív kiértékelését mutatjuk be különböző feladatokon történő finomhangolás mentén.

4.1. Kvalitatív eredmények

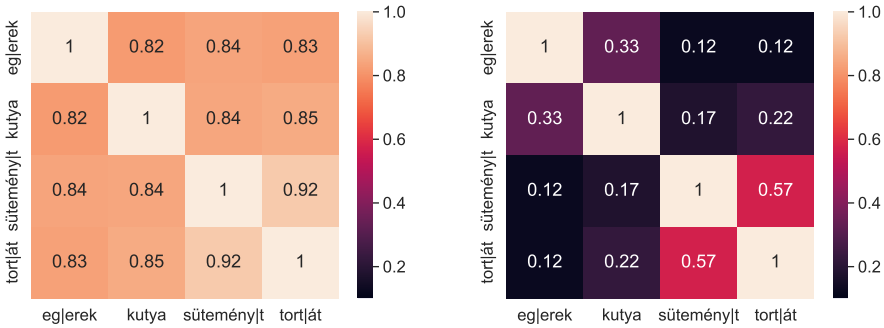
Ebben a fejezetben azt kívánjuk illusztrálni, hogy az (1) megoldásával előálló látens szemantikus kategóriák, illetve eloszlások alkalmasak a szóelőfordulások jelentéseik mentén történő csoportosítására, illetve a kontextuális hasonlóság mérésére. Kvalitatív vizsgálatainkat az 1. táblázatban található mondatokban kiemeléssel jelzett szóelőfordulások mentén közöljük. A mondatokban található | szimbólumok a tokenizálás során kialakuló szótöredékek közötti határokat jelölik. A több szótöredék alkotta szavak reprezentálására a szótöredékekhez tartozó kontextuális vektorok átlagát határoztuk meg.

Az **eg|erek** szeretik a sajtot|.
A **kutya** megker|get|te a macsk|át|.
Mari **sütemény|t** eszik|.
Klára nem kér több **tort|át|**.

1. táblázat. Példamondatok, amikben a kiemelt szavak viselkedését vizsgáljuk, a | pedig a szótöredékek határát jelzi.

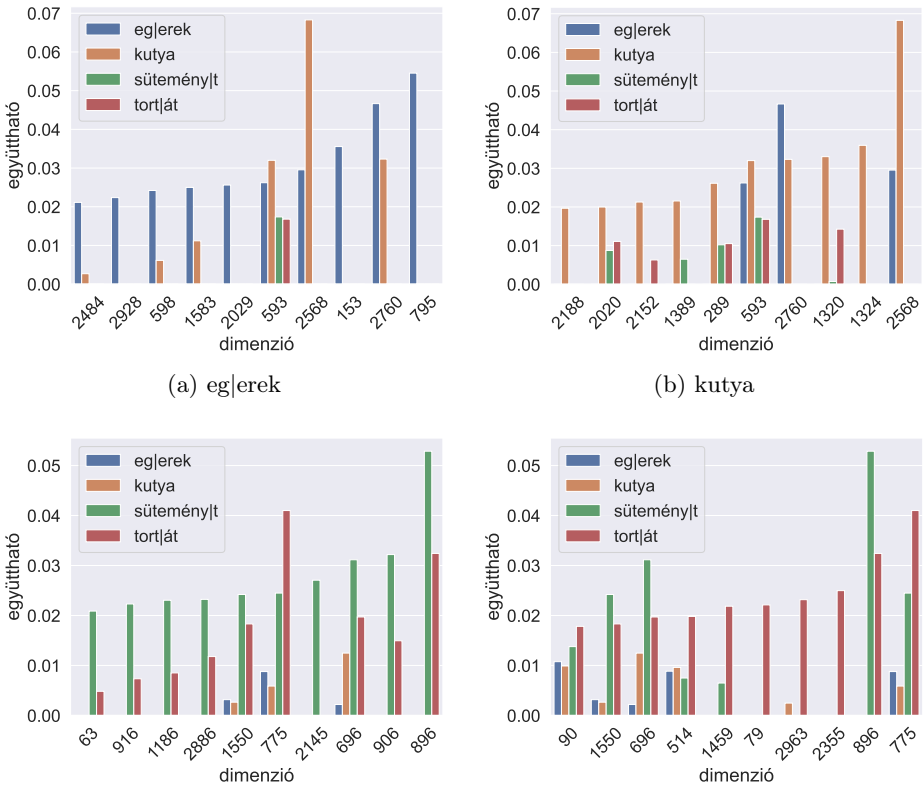
Megvizsgáltuk, hogy a példánkban vizsgált szópárok mentén miként alakulnak a koszinusz hasonlóságok, ha annak alapját az eredeti transzformer modell által létrehozott kontextuális reprezentációk, vagy a (2) alapján létrejövő $\boldsymbol{\alpha_i}$ ritka vektorok képzik. A páronkénti összehasonlítás eredményeit az 1. ábra foglalja össze. Jól látható, hogy míg a közvetlenül a transzformer modell rejtett állapotából jövő $\boldsymbol{x}_i$ vektorok mentén történő hasonlósági értékek homogén viselkedést mutatnak (1a. ábra), addig a ritka $\boldsymbol{\alpha_i}$ vektorok mentén történő hasonlóságszámítás esetén a szóelőfordulások szemantikus csoportjainak kialakulása figyelhető meg (1b. ábra).

A 2. ábrán az látható, hogy a vizsgált szavakhoz a (2) segítségével kapott szemantikus eloszlások mely látens dimenziók mentén veszik föl a legnagyobb értékeket, illetve, hogy ugyanezen dimenziók mentén a többi vizsgált szó milyen értékekkel rendelkezik. Jól kivehető, hogy a hasonló jelentéssel bíró szóelőfordulásokhoz hasonló szemantikus eloszlások kerültek meghatározásra.



(a) Az eredeti modell $\boldsymbol{x}_i$ vektorai alapján (b) A ritka $\boldsymbol{\alpha_i}$ vektorok alapján

1. ábra: A vizsgált szópárok közötti páronkénti koszinusz hasonlóságok.



(a) eg|erek (b) kutya

(c) süteményt (d) tort|át

2. ábra: A vizsgált szóelőfordulásokhoz rendelt szemantikus eloszlások domináns együtthatóinak vizsgálata. Minden ábrán az adott szóelőfordulás mentén kialakuló 10 legmagasabb együtthatóval bíró dimenziója látható, illetve a többi szó ugyanezen dimenziók mentén való viselkedése.

4.2. Kvantitatív eredmények

A létrehozott modelleket változatos célalkalmazási feladatokra való finomhangolás során is kiértékeltük. A kiértékelés során a szakirodalomból megszokott hiperparamétereket választottunk ki: az előtanított modellek finomhangolását 32-es batchmérettel, 0,00002 tanulási ráta mellett hajtottuk végre 3 epochon keresztül az összes vizsgált adatbázison. Mivel a neurális modellek finomhangolása során kapott eredmények nagy szórással rendelkeznek (Dodge és mtsai, 2020), célszerű a finomhangolás során kapott eredményeket több véletlenszerűen inicializált osztályozó fej tekintetében vizsgálni. Éppen ezért, minden vizsgált adathalmazon mindkét modellünket $50 - 50$ kiértékelésnek vetettük alá, eredményeinket több formában is közöljük.

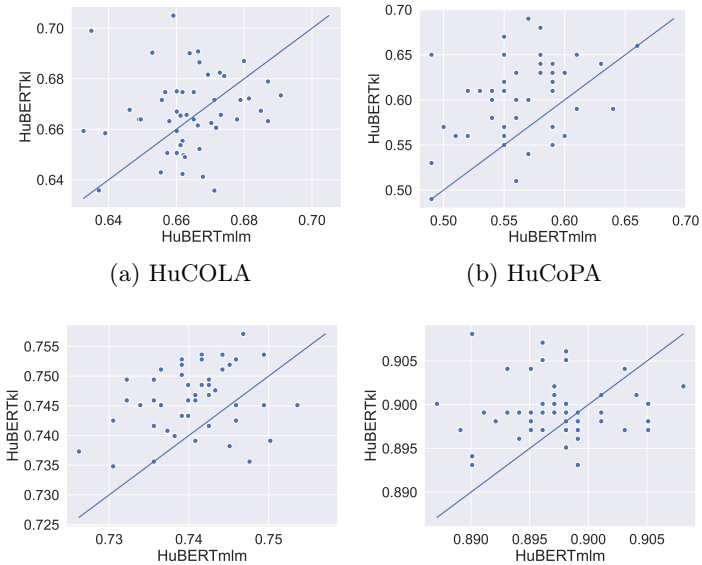
A HuLU (Ligeti-Nagy és mtsai, 2022) három alfeladatán (HuCOLA, HuCoPA, HuSST2), valamint az OpinHuBank (Miháltz, 2013) szentimentosztályozási feladatán az egyes modellek által elért eredmények összefoglalását a 2. táblázat tartalmazza. Láthatjuk, hogy az átlagos teljesítmény tekintetében a szemantikus eloszlások segítségével létrehozott HuBERTkl modell minden vizsgált feladaton jobban teljesít a baselineként használt HuBERTmlm modellnél, amely különbségek az input mondatok grammatikai helyességének eldöntésére irányuló HuCOLA esetét leszámítva szignifikánsnak tekinthetők.

Feladat	Metrika	HuBERTmlm	HuBERTkl	p-érték
HuCOLA	Matthew korreláció	0,664	0,667	0,282
HuCoPA	Pontosság	0,564	0,603	2,4*e-7
HuSST	Pontosság	0,741	0,746	3,1*e-7
OpinHuBank	Pontosság	0,897	0,900	0,005

2. táblázat. A modellek átlagos teljesítménye a vizsgált feladatonkénti lebontásban, és modellek teljesítménybeli különbségének szignifikanciájára vonatkozó p-értékek.

A 4. ábra az eltérő módon előtanított, de azonosan inicializált osztályozó fejekkel rendelkező modellpárosok finomhangolás során nyert eredményeinek feladatonkénti összehasonlítását teszi lehetővé. A HuCOLA, HuCoPA, HuSST és az OpinHuBank feladatokon elvégzett modelltípusonkénti 50 kísérletpáros közül rendre 25, 42, 41 és 34 alkalommal tapasztaltuk azt, hogy a HuBERTkl modellre építő finomhangolás megegyezően inicializált osztályozó fejekből kiindulva legalább olyan eredményt produkált, mint a HuBERTmlm modell.

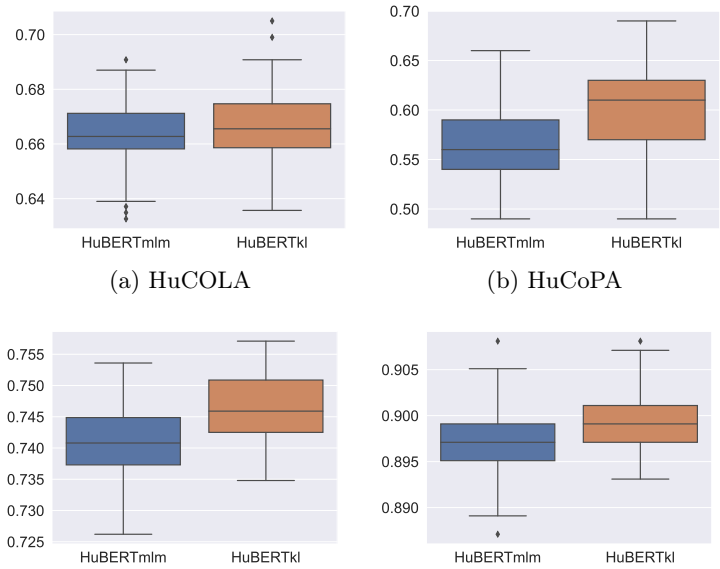
A különböző előtanítási megközelítések alkalmazása mellet létrehozott modellek egyes feladatokon a finomhangolási kísérletek során elért eredményességi mutatóinak eloszlását a 4. ábrán látható dobozdiagramok szemléltetik. Az ábra vizsgálatából kitűnik, hogy nem csak az átlagos teljesítmény tekintetében, hanem a medián teljesítmény vonatkozásában is eredményesebbek a HuBERTkl finomhangolásával nyert modellek.



(a) HuCOLA (b) HuCoPA

(c) HuSST (d) OpinHuBank

3. ábra: Az eltérő módon előtanított, azonosan inicializált osztályozó fejekkel rendelkező modellpárosok eredményei feladatonkénti lebontásban.



(a) HuCOLA (b) HuCoPA

(c) HuSST (d) OpinHuBank

4. ábra: A feladatonként és modellenkénti 50-50 kísérleti eredményeinek eloszlása.

5. Konklúzió

Cikkünkben egy olyan transzformer architektúrára tettünk javaslatot, ami az előtanítása során a konkrét szótöredékek rekonstruálása helyett (látens) szemantikus információk meghatározását tűzi ki célul. A javasolt módszert mind kvalitatív, mind pedig kvantitatív kiértékelésnek alávetettük, amelyek alapján úgy találtuk, hogy ígéretes alternatívát képes nyújtani a klasszikus szótöredék-alapú maszkolt nyelvi modellezéssel történő előtanításnak. Az elkészült modelljeinket[1,2], valamint a létrehozásukhoz használt forráskódot[3] egyaránt elérhetővé tettük.

Köszönetnyilvánítás

A kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében. A MILAB SZTE nyelvtechnológia projekt nevében köszönetet mondunk az ELKH Cloud (lásd: Héder és mtsai (2022); https://science-cloud.hu/) használatáért, ami hozzájárult a publikált eredmények eléréséhez.

Hivatkozások

Bauer, L., Wang, Y., Bansal, M.: Commonsense for generative multi-hop question answering tasks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4220–4230. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://aclanthology.org/D18-1454

Berend, G.: Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8498–8508. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.683

Brown, P.F., Della Pietra, V.J., deSouza, P.V., Lai, J.C., Mercer, R.L.: Class-based n-gram models of natural language. Computational Linguistics 18(4), 467–480 (1992), https://aclanthology.org/J92-4003

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://www.aclweb.org/anthology/N19-1423

[1] https://huggingface.co/SzegedAI/HuBERTmlm
[2] https://huggingface.co/SzegedAI/HuBERTkl
[3] https://github.com/szegedai/MLSM

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N.: Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping (2020), http://arxiv.org/abs/2002.06305, cite arxiv:2002.06305

Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)

Héder, M., Rigó, E., Medgyesi, D., Lovas, R., Tenczer, S., Török, F., Farkas, A., Emődi, M., Kadlecsik, J., Mező, Gy., Pintér, Á., Kacsuk, P.: The past, present and future of the ELKH cloud. Információs Társadalom 22(2), 128 (aug 2022), https://doi.org/10.22503/inftars.xxii.2022.2.8

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., Schwab, D.: Flaubert: Unsupervised language model pre-training for french. In: Proceedings of The 12th Language Resources and Evaluation Conference. pp. 2479–2490. European Language Resources Association, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.lrec-1.302

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., Shalev-Shwartz, S., Shashua, A., Shoham, Y.: SenseBERT: Driving some sense into BERT. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4656–4667. Association for Computational Linguistics, Online (Jul 2020), https://aclanthology.org/2020.acl-main.423

Ligeti-Nagy, N., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Laki, L.J., Vadász, N., Yang, Z.Gy., Váradi, T.: HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 431–446. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-BERT: Enabling language representation with knowledge graph. Proceedings of the AAAI Conference on Artificial Intelligence 34(03), 2901–2908 (Apr 2020), https://ojs.aaai.org/index.php/AAAI/article/view/5681

Martin, L., Muller, B., Ortiz Suárez, P.J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., Sagot, B.: CamemBERT: a tasty French language model. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7203–7219. Association for Computational Linguistics, Online (Jul 2020), https://aclanthology.org/2020.acl-main.645

Miháltz, M.: OpinHuBank: szabadon hozzáférhetö annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2013), http://acta.bibl.u-szeged.hu/58859/1/msznykonf_009_343-345.pdf

Miháltz, M., Hatvani, C., Kuti, J., Szarvas, Gy., Csirik, J., Prószéky, G., Váradi, T.: Methods and results of the Hungarian wordnet project. In: Proceedings of The Fourth Global WordNet Conference. pp. 311–321 (2008)

Mihaylov, T., Frank, A.: Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 821–832. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/P18-1076

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020), https://hlt.bme.hu/en/publ/nemeskey_2020

Nemeskey, D.M.: Introducing `huBERT`. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2021). pp. 3–14. Szeged (2021)

Peters, M.E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 43–54. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1005

Qiu, D., Zhang, Y., Feng, X., Liao, X., Jiang, W., Lyu, Y., Liu, K., Zhao, J.: Machine reading comprehension using structural knowledge graph-aware network. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5896–5901. Association for Computational Linguistics, Hong Kong, China (Nov 2019), https://aclanthology.org/D19-1602

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)

Ulčar, M., Robnik-Šikonja, M.: FinEst BERT and CroSloEngual BERT. In: Sojka, P., Kopeček, I., Pala, K., Horák, A. (szerk.) Text, Speech, and Dialogue. pp. 104–111. Springer International Publishing, Cham (2020)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)

Vilares, D., García, M., Gómez-Rodríguez, C.: Bertinho: Galician BERT representations. Proces. del Leng. Natural 66, 13–26 (2021), http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6319

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://aclanthology.org/2020.emnlp-demos.6

Yang, A., Wang, Q., Liu, J., Liu, K., Lyu, Y., Wu, H., She, Q., Li, S.: Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2346–2357. Association for Computational Linguistics, Florence, Italy (Jul 2019), https://aclanthology.org/P19-1226

Ye, Z., Chen, Q., Wang, W., Ling, Z.: Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. CoRR abs/1908.06725 (2019), http://arxiv.org/abs/1908.06725

Magyar nyelvű időjárásjelentések nyelvi modell alapú automatizált generálása

Knap Árpád[1], Dömsödi L Bíborka[2], Mogyorósi Pálma[2],
Szigeti Péter[3], Tóth Andor[3], Virág Attila[3] és Kmetty Zoltán[1]

[1] Társadalomtudományi Kutatóközpont
{knap.arpad, kmetty.zoltan}@tk.hu
[2] iStat Consulting Kft.
[3] Central Médiacsoport Zrt.

Kivonat: Tanulmányunkban bemutatjuk egy olyan eszköz első változatát, amely képes az Országos Meteorológiai Szolgálat adatai alapján magyar nyelvű időjárásjelentések automatizált megírására. Az eszköz első lépésben egy szabály-alapú rendszerben a nyers numerikus adatokból előállít egy listát a várható időjárási eseményekről, majd ezen listából az OpenAI GPT-3 nyelvi modelljére támaszkodva a megfelelő paraméterezés mellett kigenerál egy angol nyelvű időjárásjelentést, amit a DeepL Translate szolgáltatással magyar nyelvűre fordít. A kapott időjárásjelentések kifejezetten magas, 4 feletti értékelést kaptak a tesztelés során (1-5 skála) nyelvhelyesség, stílus és koherencia dimenziókban.

1 Bevezetés, célkitűzés

A természetes nyelv generálás (Natural Language Generation - NLG) nem számít új iránynak a számítógépes nyelvészeten belül, de az elmúlt években az egyre komplexebb tudású nyelvi modellek elterjedésével új lendületet kapott a terület. Az NLG-n belül két nagy kutatási és alkalmazási irány létezik: a szövegből szöveg generálás (text-to-text generation) és az adatból szöveg generálás (data-to-text generation) (Gatt és mtsai. 2018). Utóbbi irány elsősorban a robot újságírás kapcsán jelenik meg, különféle témaköröket érintve. Az egyik kiemelt terület a robot újságíráson belül az időjárásjelentések automatikus megírása (Goldberg és mtsai, 1994; Reiter és mtsai, 2005; Belz 2008; Turner és mtsai, 2007; Ramos-Soto és mtsai, 2014). Az időjárási hírek általában rövidek és jól felépíthetőek numerikus adatokból, ráadásul az adatok legtöbb esetben egy jól definiált és állandó struktúrában állnak rendelkezésre, ami jó alapot ad egy NLG projekthez.

A 24.hu, a TK[1] és az iStat Consulting egy közös projektet indított 2022 tavaszán, aminek az volt a célja, hogy magyar nyelvre előálljon az első olyan eszköz, amely automatikusan képes időjárásjelentéseket generálni OMSZ adatok alapján, nyelvi modell

[1] A publikációban szereplő kutatást, amelynek megvalósításában a Társadalomtudományi Kutatóközpont is részt vett, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

segítségével. Összességében tehát magyar nyelvű, szöveges időjárásjelentéseket kívántunk automatizáltan előállítani, amelyhez mintaként a 24.hu meglévő időjárásjelentéseit tekintettük, célunk ezek reprodukálása volt.

A tanulmány első felében bemutatjuk a projektben használt kiinduló adathalmazokat, illetve az ezeken végrehajtott transzformációs lépéseket és logikai műveleteket, amelyek segítségével az adatokból kiválasztjuk a jelentések megírásához szükséges adatpontokat. Ezt követően ismertetjük az általunk használt nyelvi modell sajátosságait, illetve kiértékeljük a rendszer működését emberek által írt időjárásjelentések szövegeinek és általunk generált jelentéseknek az összehasonlításával. Végezetül ismertetjük a projekttel kapcsolatban felmerült kihívásokat és limitációkat, illetve továbbfejlesztési irányokat is megfogalmazunk.

Munkánk jelenlegi, első fázisában az adott napra vonatkozó (reggeli) időjárásjelentések generálását tűztük ki célul. Ezek a jelentések jellemzően az adott nap reggelétől a következő nap reggeléig tartó időszakra tartalmaznak adatokat. A jelentések előállításához kizárólag a 24.hu számára rendelkezésére álló OMSZ adatszolgáltatás keretében érkező adathalmazokat használjuk fel, és ezek közül is kizárólag azokat, amelyek számszerű adatokat, és nem szöveges jelentéseket tartalmaznak. Célunk volt, hogy a létrejövő jelentésekben megkülönböztessük egymástól az eltérő napszakokat, tehát ne az egész napra adjunk egy általános képet, hanem eltérő időjárási események esetében az egyes napszakokra különböző megállapításokat tegyünk. Szintén fontosnak tartottuk, hogy ezt az elvet ne csak a napszakok esetében, hanem földrajzilag is kövessük, ezért a jelentésekben, ha szükséges, megkülönböztetjük egymástól az ország különböző régióit. A jelentések előállításához az OpenAI GPT-3 (Radford és mtsai, 2018; Floridi és Chiriatti, 2020) angol nyelvi modelljét használjuk, amely szabadszöveges instrukciókat fogad, így az adatok feldolgozása, és szöveges paranccsá alakítása angol nyelven történik, a létrejövő jelentéseket pedig automatizáltan, a DeepL Translate szolgáltatással fordítjuk magyar nyelvre.

2 Adatfeldolgozás

Ahogy említettük, az OpenAI GPT-3 modellje szabadszöveges instrukciókat fogad bemenetként. Az alábbiakban bemutatjuk azt a folyamatot, amely során az OMSZ által közölt XML formátumú állományokból számos lépésen keresztül angol nyelvű szöveges instrukciókat hozunk létre.

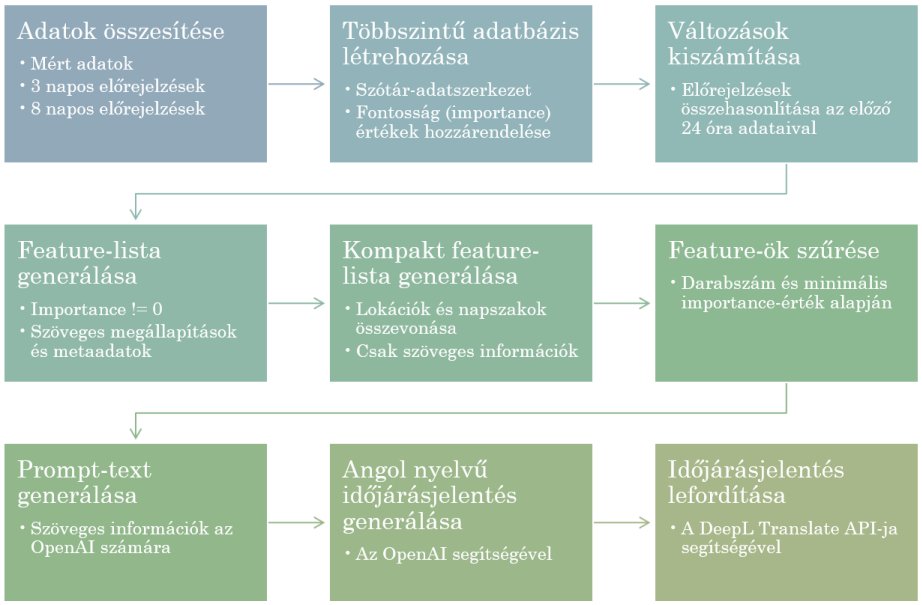


Fig. 1. az adatok transzformációjának és az időjárásjelentés megírásának folyamatábrája

2.1 Forrásadatok bemutatása

A 24.hu számára rendelkezésre álló OMSZ adatok közül három olyan állománytípust azonosítottunk, amely számszerű, tehát nem szöveges adatokat tartalmaz. Ezek az adatok (az adatformátum megváltozása miatt) 2022 január 10-e óta állnak rendelkezésünkre.

Kiinduló adataink első típusa a múltra vonatkozó tényadatokat tartalmaz. Ezekben az óránként kiadott XML állományokban egy-egy jelentés áll rendelkezésre 7 különböző településre (Budapest, Debrecen, Győr, Miskolc, Pogány, Sármellék, Szeged). Ami a fájlok adattartalmát illeti, időjárás kód (pl. derült, ködös, szitálás, eső, havazás), hőmérsékletérték, szélirány, szélerősség, légnyomás, továbbá relatív páratartalom érhető el az egyes településekre.

Kiinduló adataink második típusát a három napos előrejelzéseket tartalmazó fájlok jelentik. Ezeket minden nap két alkalommal bocsátja rendelkezésre az OMSZ: délben, valamint éjfélkor. Ezek az állományok óránkénti felbontású előrejelzéseket tartalmaznak, a következő három napra, 110 lokációra megadva. Az egyes adatpontok esetében hőmérsékletérték, illetve időjárás kód érhető el.

A harmadik kategóriába a nyolc napos előrejelzések tartoznak, amelyeket hasonlóan a három napos állományokhoz, szintén napi két alkalommal, délben, illetve éjfélkor adják ki, és ugyanarra a 110 helyszínre tartalmaznak adatokat. Ezekben a fájlokban a következő nyolc napra láthatunk előrejelzéseket, amelyek nem óránkénti, hanem napi felbontásúak, tehát a következő nyolc napra egy-egy adatpontot találunk. Ezeknek a

fájloknak az adattartalma is eltérő, itt 24 órás maximum és minimum hőmérséklettel, csapadékmennyiséggel, maximum szélerősséggel, továbbá csapadékvalószínűséggel dolgozhatunk.

2.2 Adatranszformációs lépések és logikai műveletek

Ahhoz, hogy az XML formátumú adathalmazokból eljussunk az OpenAI modellje számára inputként beadandó szabadszöveges instrukciók előállításáig, számos adattranszformációs lépést hajtunk végre. Elsőként az adatainkat táblázatos formátumúra hozzuk, majd ezt követően egy szótárszerkezetben tároljuk el őket. Ezután a szótárszerkezetből egy lapos listát hozunk létre, amelyben már megjelenik az adott adatpont szövegszerű reprezentációja. Ennek a listának a további transzformációjával állnak elő a szöveges adataink, amelynek szűrésével létrejön az az angol nyelvű prompt-szöveg, amelyet inputként használunk az OpenAI nyelvi modellben.

2.2.1 Az adatok összesítése

A folyamat a mért adatok összesítésével indul. Ennek során az a célunk, hogy az időjárásjelentés generálásához képest vett elmúlt 24 óra adatait összegezzük. Ehhez az adatokat az XML állományokból táblázatos formátumúra konvertáljuk, és napszakokra[2], valamint napszak/nagyrégió[3] bontásban aggregáljuk, kiszámítva a hőmérséklet és a szélerősség maximális, minimális, átlagos és medián értékeit, illetve az adott napszakra vonatkozó időjárás kódok, valamint az esetleges adathiányok darabszámát (pl. 6db borult, 1db szitálás, 21db eső, 2db zápor, 4db adathiány).

A következő lépésben a három napos előrejelzéseket összegezzük. Ennek során az adott napra, illetve a következő éjszakára vonatkozó adatokat dolgozzuk fel. Az adatokat napszakonként aggregáljuk négy különböző földrajzi felosztásban – nagyrégiókra, régiókra, megyékre, illetve a 110 lokációra, kiszámítva az adott napszakra vonatkozó maximális, minimális, átlagos, medián hőmérsékletet, valamint az időjárás kódok számát.

Ezt követően a nyolc napos előrejelzéseket dolgozzuk fel, hasonlóan a három napos állományokhoz. Mivel itt a kiinduló fájlok napi felbontásban tartalmaznak információkat, nem különböztetjük meg egymástól az egyes napszakokat, tehát az adott napra aggregáljuk adatainkat, nagyrégió, régió, megye, illetve lokáció bontásban. A folyamat során kiszámítjuk a hőmérséklet, csapadékmennyiség, szélerősség és csapadékvalószínűség adatok különböző szélső- illetve középértékmutatóit.

Az adatok összesítésével párhuzamosan elvégzünk néhány további műveletet is. Egyebeket mellett például egyszerűsítéseket végzünk az időjárás kódokon, pl. az ónos

[2] A projektben ötféle napszakot különböztetünk meg egymástól: 0:00-tól 6:00-ig éjszaka, 6:00-tól 9:00-ig reggel, 9:00-tól 12:00-ig délelőtt, 12:00-tól 18:00-ig délután, 18:00-tól 0:00-ig pedig este.

[3] Magyarországon a következő nagyrégiókat (NUTS-1 szint) különböztetjük meg egymástól: Dunántúl, Alföld és Észak, Közép-Magyarország.

szitálást és az ónos esőt egyaránt ónos esőre kódoljuk, az egyes mérési pontokhoz pedig hozzárendeljük a megyét, a régió, illetve nagyrégió elnevezéseket. Ezen kívül a szélerősségértékekhez hozzárendeljük a Beaufort-skála egy egyszerűsített verzióját[4].

2.2.2 Szótár létrehozása és fontosság értékek hozzárendelése

Ezzel előálltak a táblázatos adataink – a következő lépés annak eldöntése, hogy az egyes adatpontok közül melyek azok, amiket fel kívánunk használni az aktuális napi időjárásjelentés megírásához. Ennek eldöntéséhez egy többszintű szótárat (Python dictionary-t) használunk, amiben minden adatponthoz hozzárendelünk egy-egy fontosság (importance) értéket. A fontosság érték 0 és 1 között mozog, ahol az 1 nagyon fontos (kötelezően szerepelnie kell a jelentésben), a 0 pedig az egyáltalán nem fontos. A fontossági értékek bevezetésének az a célja, hogy az előálló időjárásjelentés ne legyen indokolatlanul hosszú, és csak olyan információkat tartalmazzon, amelyek érdemben elmondanak valamit az adott napról.

Az előálló szótár-adathalmaz kétféle típusú adatot tartalmaz: (1) egyrészt pusztán az előrejelzésekből származó adatokat, másrészt (2) a mért adatok és az előrejelzések összehasonlításán alapuló változásokat. Ami az első típust illeti, ide tartoznak a legalacsonyabb éjszakai és esti hőmérsékletre, valamint a legmagasabb nappali hőmérsékletre vonatkozó adatok, a különböző napszakokra vonatkozó időjáráskódok eltérő földrajzi bontásokban, továbbá a csapadékra és szélre vonatkozó adatok szintén különböző földrajzi bontásokban. Az előző napi mért adatok és az előrejelzések összehasonlításán alapuló adatok között jelenleg 12 féle megállapítást különböztetünk meg egymástól. Ilyen például az, ha derült, napos idő után felhős, borult időre számíthatunk, ha jelentős felmelegedés vagy lehűlés várható, vagy ha tartósan derült, esetleg borult idő valószínű.

2.2.3 Ismétlés helyett összevonás

A szótár adathalmaz létrehozásához és a fontosság értékek hozzárendeléséhez különböző logikai döntések meghozatalára van szükség. Az első lépés annak a megállapítása, hogy mik azok az adatok, amik „kötelezőek”, tehát mindenképp kellenek egy időjárásjelentésbe. Ezek közé tartozik például a legalacsonyabb éjszakai és a legmagasabb nappali hőmérséklet. Ezek 1-es fontosság értéket kapnak és a teljes országra vonatkoznak.

A következő lépés az egyes időjárási események fontossági kategóriákba való besorolása. Ezeket az időjárás kódok jelölik. Itt háromféle kategóriát különböztetünk meg egymástól. (1) A rendkívüli események 1-es fontosság értéket kapnak amennyiben megjelennek, függetlenül attól, hogy milyen arányban jelennek meg az adott bontásban (a következő időjárás kódok tartoznak ide: ködös, zápor, zivatar, ónos eső, hószállingózás, havazás, hózápor, havaseső, hózivatar, hófúvás). (2) A nem csapadékra, hanem derült vagy felhős időre vonatkozó kódok közül az adott bontásban a domináns kódot keressük, tehát azt, ami legalább egyharmados arányban jelenik meg (a következő kódok tartoznak ide: derült, kissé felhős, közepesen felhős, erősen felhős, borult). Ha több ilyen kód is van, akkor felsoroljuk ezeket a kódokat, és azt a megállapítást tesszük

[4] Ez a skála az eredeti 13 helyett 6 fokú.

például, hogy „kissé, illetve közepesen felhős idő várható”. Amennyiben nincsen domináns kód, de több is megjelenik a felsoroltak közül, akkor azt mondjuk, hogy változóan felhős idő várható[5]. (3) Ezeken kívül a szitálást és az esőt különböztetjük meg. Az erre vonatkozó kódoknak legalább 20 százalékos arányt kell elérniük az adott bontásban, hogy nullától eltérő fontossági értéket kapjanak.

A fontosság értékek meghatározásakor általánosságban azt az elvet követjük, hogy a legnagyobb fontosság értéket a rendkívüli időjárási eseményre vonatkozó kódok kapják, ezt követik a csapadékra vonatkozó egyéb kódok, utána pedig a nem csapadékra vonatkozó kódok következnek. Tehát abban az esetben, ha például a Dunántúlon kissé felhős idő lesz, akkor az erre vonatkozó megállapítás 0,4-es fontosság értéket kap, ha viszont eső várható, akkor az 0,7-es értéket. Ezen kívül a teljes országra vonatkozó megállapítások nagyobb fontosságot kapnak, mint a nagyrégiós és régiós, a nagyrégióra vonatkozó megállapítások pedig nagyobb fontosságot kapnak, mint a régiós bontású előrejelzések.

A logikai döntések következő kategóriája az országos, nagyrégiós és régiós felbontású adatok kezelésére vonatkozik. A cél az, hogy eltérő régiókra vonatkozó, de azonos tartalmú előrejelzések ne jelenjenek meg ismételten a jelentésekben, mivel ezek nagyon hosszúvá és repetitívvé tennék a keletkező szövegeinket. Itt azt az elvet követtük, hogy először az országos szintet vizsgáljuk az egyes napszakokra. Ha egy adott időjárási kód az előzőekben ismertetett logika szerint itt megjelenik, akkor 1-es fontosság értéket kap az adott napszakra. Ezután azt vizsgáljuk, hogy ez az esemény az adott napszakban hogyan jelenik meg a nagyrégiókban. (1) Ha mindhárom nagyrégióban megjelenik, akkor az egy országos esemény. Ilyenkor nincsen további teendőnk, mert országosnak jelöltük az eseményt. (2) Ha két nagyrégióban jelenik meg, akkor az nem egy országos esemény. Ilyenkor a releváns nagyrégiókhoz hozzárendeljük a nullától eltérő fontosság értékeket, és az országos szinten pedig nullát állítunk be. (3) Ha egy nagyrégióban jelenik meg, akkor az szintén nem egy országos esemény. Ilyenkor megvizsgáljuk azt, hogy az adott nagyrégió alá tartozó kisebb régiók közül hányban jelenik meg az adott esemény. Ha nem az összesben, akkor ezekhez a kisebb régiókhoz hozzárendeljük az fontosság értékeket, és az országos szinthez pedig a nulla értéket.

A csapadékkal kapcsolatos adatokat hasonlóan kezeljük, azzal a különbséggel, hogy mivel ezek a nyolc napos, napi felbontású előrejelzésekből származnak, nem bontjuk őket napszakokra. Mivel viszont az az információ rendelkezésünkre áll az időjárás kódokból, ha valahol csapadék várható, a csapadék adatokat arra használjuk fel, hogy a jelentős mennyiségű várható csapadékot előrejelezzük. A jelentős mennyiség alsó határát napi 10mm-ben definiáltuk.[6]

A következő lépésben a széllel kapcsolatos adatainkat összegezzük. A szélre vonatkozó előrejelzésekhez a fontosság értékeket az általunk létrehozott, hatfokozatú, egyszerűsített Beaufort-skála alapján rendeljük hozzá az adatpontokhoz. Ennek segítségével a széllökések maximális erősségét soroljuk be. Természetesen a magasabb

[5] Ezt csak abban az esetben tesszük, ha összességében ezek a kódok tényleg dominálnak, tehát legalább 50%-os arányt érnek el.

[6] Ahogyan az eddigiekből is látszik, a program működésének logikája tovább finomítható, bonyolítható. A cikkünkben tárgyalt első verzióban az említett határértékek rugalmasan változtathatók, emellett alkalmazásuk jelentősen egyszerűsítette például a jelentős mennyiségű csapadék előrejelzésének problémáját.

Beaufort-érték magasabb fontosság értéket eredményez. (1) Országos szintű adatként a (kis)régiókhoz hozzárendelt értékek móduszát, tehát a leggyakoribb értéket fogadjuk el. Ilyen módon elkerüljük az országos szélintenzitás felülbecslését. (2) Ha két nagyrégióban az országosnál magasabb Beaufort értéket látunk, akkor ezekhez a régiókhoz hozzárendeljük az adott fontosság értéket. (3) Amennyiben egy nagyrégióban, és azon belül több mint két régióban az országosnál magasabb Beaufort értéket látunk, akkor az adott nagyrégióhoz rendelünk hozzá nullától eltérő fontosság értéket. (4) Végezetül, ha egy nagyrégióban, és azon belül legfeljebb két régióban látunk az országosnál magasabb Beaufort értéket, akkor az adott régiókhoz rendeljük hozzá a megfelelő fontosság értékeket.

A logikai műveletek utolsó köre a napszakok kezelésére vonatkozik. Ennek fő célja – a régiókhoz hasonlóan – annak elkerülése, hogy ugyanazokat az eseményeket számos régió/napszak kombinációra újra és újra leírjuk a jelentésben, mivel az nagyon ismétlődővé és hosszúvá tenné a szöveget. Általánosságban azt az elvet követjük, hogy legfeljebb két napszak esetén megnevezzük a napszakokat. Ha az esemény ennél több napszakra igaz, akkor az egész napra teszünk megállapítást.

2.2.4 Úton a szövegszerű adatok felé

A következő lépésben a többlépcsős fa adatszerkezetből egy „lapos” attribútum-listát készítünk. A fenti logikai műveletek egy részét is ezen konverzió során hajtjuk végre. Ez a formátum egy köztes szintet képez a többszintű adathalmaz és a nyelvi modell számára előállítandó szöveges instrukciók között. Ezt azt jelenti, hogy itt még vannak számszerű adataink, de már megjelennek a szöveges adatok is. Az attribútum-lista már egy szűrt adatkészletet tartalmaz: ebbe csak a nullától eltérő fontosság értékkel rendelkező adatpontok kerülnek bele.

Ezt követően tovább egyszerűsítjük az adatszerkezetet, amelynek az eredményét kompakt attribútumoknak neveztünk el. Itt már kizárólag a szöveges információk és a hozzájuk tartozó fontosság értékek jelennek meg. Ebben a fázisban történik meg a napszakok és a régiók összevonása „egy sorra”. Ennek az a célja, hogy például a különböző napszakokra vonatkozó, de megegyező tartalmú megállapítások egy instrukcióként kerüljenek be a modellbe.

Ezt követően a keletkezett listát tovább szűrjük az alapján, hogy hány darab adatpontot szeretnénk felhasználni az adott időjárásjelentés megírásához. Ez a folyamat egy függvény segítségével rugalmasan változtatható kimenetellel hajtható végre aszerint, hogy milyen hosszú, mennyire részletes időjárásjelentést szeretnénk generálni. A szűrést az alábbi elvek alapján hajtjuk végre. (1) Az 1-es fontosság értékkel rendelkező adatok mindenképp megjelennek, akkor is, ha kevesebb attribútumot kértünk, mint ahány ilyen adatpont van. (2) A fennmaradó darabszámú helyre fontosság szerint csökkenő sorrendben kerülnek be a megállapítások. (3) A megegyező fontosságú megállapítások véletlen sorrendben kerülnek be. (4) A lista összeállításakor meghatározhatunk minimális fontosság értéket is.

Utolsó lépésként előállítjuk azt a szöveges instrukciót, amelyet az OpenAI GPT3 modellje számára beadunk, mint inputot – ezt a szöveget prompt-szövegnek neveztük el. Ez tartalmazza azokat a megállapításokat, amelyek a fenti lépések eredményeképpen előálltak. A prompt-szöveg minden esetben az alábbi mondattal kezdődik:

```
write a weather report for today using the following
information
```[7]

Ezt követően a leszűrt attribútumok felsorolása olvasható, vesszővel elválasztva. Fontos, hogy a prompt szövegben megadott adatok sorrendje visszaköszön a generált jelentésben is, emiatt érdemes ezeket olyan sorrendben megadni, ahogyan azok általában a valós jelentésekben is szerepelnek – mi is ezt a logikát követtük a projektben. Az 1. táblában egy példa olvasható egy prompt szöveggel és egy ebből előállt időjárásjelentéssel.

1. táblázat: prompt szöveg és az OpenAI GPT-3 modellje által generált időjárásjelentés magyarra fordított változata

Prompt	Generált időjárásjelentés	Magyarra fordított jelentés
lowest night temperature between -1.0 and 7.0 degrees, daytime temperature between 8.7 and 20.3 degrees, rain possible, very cloudy weather in the evening and night, thunderstorms possible in the Southern Great Plain and Northern Great Plain regions, sleet possible in the night in the Western Transdanubia and Southern Transdanubia regions, snow possible in the night in the Western Transdanubia region, large amounts of precipitation is possible in the Great Plain and North and Transdanubia regions, the weather warms up significantly, strong gale expected in the Great Plain and North and Transdanubia regions	The lowest night temperature will be between -1.0 and 7.0 degrees, and the daytime temperature will be between 8.7 and 20.3 degrees. It is possible that it will rain, and the weather will be very cloudy in the evening and night. There is a possibility of thunderstorms in the Southern Great Plain and Northern Great Plain regions, and sleet is possible in the night in the Western Transdanubia and Southern Transdanubia regions. Snow is possible in the night in the Western Transdanubia region. Large amounts of precipitation are possible in the Great Plain and North and Transdanubia regions, and the weather will warm up significantly. A strong gale is expected in the Great Plain and North and Transdanubia regions.	A legalacsonyabb éjszakai hőmérséklet -1,0 és 7,0 fok között, a nappali hőmérséklet pedig 8,7 és 20,3 fok között alakul. Elképzelhető, hogy esni fog az eső, este és éjszaka pedig erősen felhős lesz az idő. A Dél-Alföld és az Észak-Alföld térségében zivatarok kialakulására van kilátás, a Nyugat-Dunántúlon és a Dél-Dunántúlon pedig éjszaka havas eső, zápor is lehet. A Nyugat-dunántúli régióban éjszaka havazás lehetséges. Az Alföldön, valamint az Észak-Alföld és az Észak-Dunántúl térségében nagy mennyiségű csapadék hullhat, és jelentősen felmelegszik az idő. Az Alföldön és a Dunántúlon erős szélvihar várható.

[7] A kezdőprompt szövegének kiválasztásánál elsősorban az OpenAI dokumentációjában szereplő ajánlásokat tekintettük alapnak. A fentihez képest próbálkoztunk más, kissé eltérő megfogalmazású promptok alkalmazásával is, de a generált időjárásjelentések értékelése során ezt a megfogalmazást értékeltük a legjobbnak.

Ahogyan a fenti példában is látható, az elkészült magyar nyelvű jelentés jól olvasható, megfelelő magyarsággal íródott, tartalmi és szakmai szempontból elfogadható. Néhány apróbb hibára azonban felhívnánk a figyelmet. Ilyen például a régiók nevének kezelése, amely az Alföld és Észak (Great Plain and North) nagyrégió esetében nem megfelelően kerül lefordításra (a „Large amounts of precipitation are possible in the Great Plain and North and Transdanubia regions" mondatrészből "Az Alföldön, valamint az Észak-Alföld és az Észak-Dunántúl térségében nagy mennyiségű csapadék hullhat" lesz). Mivel az Alföld és Észak nagyrégió a köznyelvben kevésbé használatos, ezért érdemes lehet ehelyett a „keleti országrészben" szókapcsolat alkalmazása, amely egyébként is gyakrabban szerepel valós időjárásjelentésekben. Szintén megfontolandó a teljes országra vonatkozó adatok esetében ennek explicit módon való kiemelése. A kialakított rendszer kellően rugalmas ahhoz, hogy az ehhez hasonló finomításokat egyszerűen lehessen implemetálni a későbbiek során.

3 Időjárásjelentések generálása

Az időjárásjelentések generálása tehát az OpenAI API-ja segítségével történik. Jelenleg négy GPT-3-ra épített nyelvi modell érhető el az OpenAI felületén, az Ada, a Babbage, a Curie és a Davinci. Ezek a modellek árban, futási időben és teljesítményben is eltérnek egymástól. A modellek közül az Ada a leggyorsabb és legolcsóbb, de képességeit tekintve ez a legbehatároltabb, míg lista másik végén a Davinci van, ami lassabb és drágább, de képes bonyolult nyelvi feladatok megoldására is.

A jelentés megírásához jelenleg a text-davinci-002 modellt használjuk, mivel az alacsony token szám miatt ez a modell is gyorsan fut és alacsony költségen tartható (egy napi jelentés generálása kevesebb mint 10 centbe kerül). A modell széleskörűen paraméterezhető. Három olyan paramétert azonosítottunk, ami kifejezetten fontosnak bizonyult a szöveggenerálásnál. A „temperature" azt szabályozza, hogy mennyire legyen kreatív a nyelvi modell. Magasabb temperature értéknél olyan elemekkel gazdagítja a modell a szöveget, amik nincsenek benne a prompt szövegben, de kontextusában a szövegbe illenek. Például eső esetén belekerülhet az időjárásjelentésbe, hogy érdemes aznap esernyőt magunkkal vinni, ha kimegyünk az utcára. A „frequency penalty" paraméterrel büntethetjük a szóismétlést, míg a „presence penalty"-val azt szabályozhatjuk, hogy mekkora valószínűséggel jelenjenek meg új tokenek a szövegben.

Az időjárásjelentéseknél fontos az „adathűség", tehát nem szerencsés magasra állítani a temperature értéket, mert ezzel növeljük annak a kockázatát, hogy olyan részek is bekerülnek a szövegbe, amit az adatok nem támasztanak alá. Az általunk optimálisnak talált temperature paraméter esetében a tesztelés során azt tapasztaltuk, hogy a nyelvi modell nem ad hozzá olyan megállapításokat a szöveghez, amelyet a prompt szöveg nem tartalmaz. Ugyanakkor az is cél továbbá, egyben üzleti szempontból is indokolható, hogy ne legyen nagyon unalmas és repetitív az időjárásjelentés, tehát érdemes csökkenteni a szóismétléseket a szövegekben, hogy a látogatók szívesen olvassák azokat. A vonatkozó vizsgálataink alapján tehát úgy találtuk, hogy az alábbi paraméterek segítségével állíthatók elő a legjobb minőségű időjárásjelentések:

- temperature [0:1] = 0,45

- frequency_penalty [-2:2] = 0,5
- presence_penalty [-2:2] = 0

Az angol nyelvű jelentéseket a DeepL Translate API-ja segítségével fordítjuk végül magyarra. Alternatív megoldásként megvizsgáltuk a projekt keretében a Google Translate API-t használatát is. Utóbbi magyarról angolra jobban fordította az időjárás jelentéseket, de angolról magyarra a DeepL bizonyult megfelelőbbnek[8] a tesztjeink alapján.

3.1 A rendszer teljesítményének kiértékelése

A fenti folyamat eredményeképpen előálló időjárásjelentések minőségének kiértékelését 100 darab valós, a 24.hu által a rendelkezésünkre bocsátott, illetve 100 darab általunk generált időjárásjelentésen végeztük.[9] A jelentések generálásához a fent leírt OpenAI paramétereket használtuk, az attribútumok szűrése során pedig 10 darab adatpontot kértünk le, 0,5-ös minimális fontosság értékkel. A jelentéseket 100 darab véletlenszerűen kiválasztott napra generáltuk 2022 január 11. és 2022 július 20. között.

[8] A Google Translate tapasztalataink szerint nagyobb arányban fordította hibásan a különböző régióneveket tartalmazó mondatokat. Az 1. táblázatban közölt példa utolsó mondatát („A strong gale is expected in the Great Plain and North and Transdanubia regions”) hibásan a következőre fordította: „Erős szélvihar várható az Alföldön, valamint az Észak- és a Dunántúlon”.

[9] Ahogyan korábban is említettük, az általunk elérhető adatok köre és időintervalluma limitált volt, ezért nem állt módunkban ugyanazon napok esetében a valós időjárásjelentésekhez hasonlítani az általunk generált szövegeket. Az adatokon kívül további kihívást jelentett az, hogy nincsen információnk arról, hogy az újságíró vagy meteorológus pontosan melyik időpontban közzétett OMSZ modell, illetve az OMSZ által biztosított adathalmazok közül mely adattípusok alapján készítette el a közzétett jelentést, tehát ilyen összehasonlításoknál nem lehettünk volna biztosak abban, hogy ugyanazt a bemeneti adatot használjuk a jelentések generálásához, mint amelyikből a meglévő jelentést író személy dolgozott.

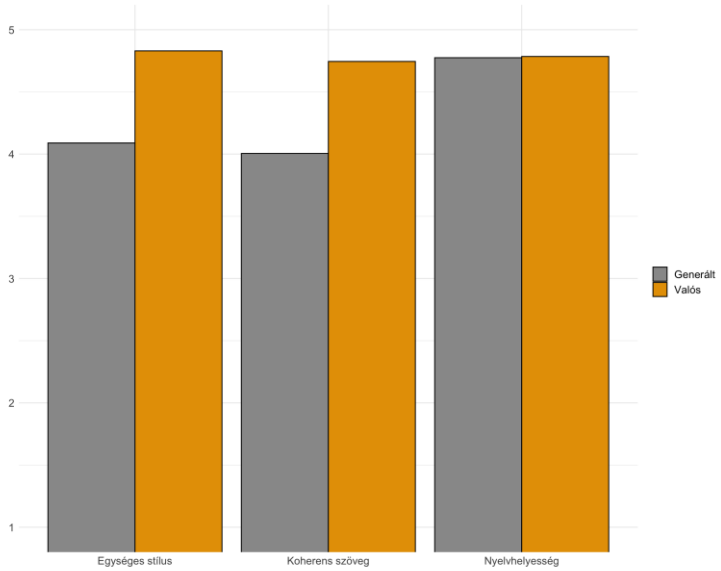


Fig. 2. a generált és a valós időjárásjelentésekhez tartozó minősítő szempontok átlagértékei

A jelentések értékelésére a projekten nem dolgozó személyeket kértünk fel, akik három szempont szerint minősítették az összesen 200 darab szöveget. A minősítés során az egységes stílust, a szöveg koherenciáját, illetve a szöveg nyelvhelyességét értékelték 1-től 5-ig terjedő skálán, ahol az 1-es érték a legrosszabb, az 5-ös pedig a legjobb minősítést jelentette. Minden időjárásjelentést egy annotátor értékelt.

Ahogyan a fenti ábrán látszik, az általunk generált jelentések – a nyelvhelyességet kivéve – nem érték el a valós időjárásjelentések minőségét, azonban nem is maradtak el azoktól jelentősen, és mindegyik értékelési szempont szerint meghaladták a 4,0-s átlagot.

4 Kihívások és limitációk, továbbfejlesztési lehetőségek

Az automatikus időjárásjelentés generálással kapcsolatos limitációk egy része az elérhető adatok körére vonatkozik. Jelentős limitációt jelent a mért adatokra vonatkozó fájlok adattartalma. Ahogy említettük, ezekben csupán hét település adatai szerepelnek, ami lehetetlenné teszi a régiós vagy megye szintű összehasonlításokat a mért értékek és az előrejelzések között. További problémát jelentenek az ezekben a fájlokban mutatkozó adathiányok az időjárás kódok esetében. Ezek Győr és Sármellék állomásoknál szinte teljesen hiányoznak a projektben felhasznált több mint fél éves adathalmaz alapján, Pogánynál pedig 21 százalékban mutatkozik adathiány. Mivel ezek a települések mindegyike a Dunántúlon található, a nagyrégiós összehasonlítások is rendkívül bizonytalanok lennének az adathiányok miatt. A fentiek miatt csak országos szintű összehasonlításokat tudtunk kalkulálni.

Ennél kisebb jelentőségű, de mindenképpen említésre méltó probléma, hogy a háromnapos előrejelzések óránkénti, a nyolcnapos előrejelzések pedig napi felbontásúak, azonban ahogy láttuk, ezek az állományok eltérő adatokat tartalmaznak. Ez azt jelenti, hogy a nyolcnapos előrejelzésekből származó, csapadékkal és széllel kapcsolatos kalkulációinkat csak napi szinte tudjuk kiszámítani, tehát itt nincsen lehetőségünk arra, hogy az egyes napszakokat megkülönböztessük. Így tehát az egyes napszakokra csak az időjárás kódok és a hőmérsékletértékek alapján tudunk előrejelzéseket adni. További korlátot jelent az elkészült időjárásjelentések információtartalmára vonatkozóan, hogy nem áll rendelkezésünkre olyan adat, amiből szélirányt vagy felhőzettípust tudunk előrejelezni, ezért ezek kimaradnak a program által generált jelentésekből. Ezen kívül szintén nincsenek olyan adataink, amelyek alapján például olyan, komplexebb kijelentéseket tehetünk, hogy „egy nyugatról érkező melegfront vonul át az ország felett, jelentős csapadékot hozva először a Dunántúlon, majd a középső országrészben".

Munkánk során arra törekedtünk, hogy az általunk megírt kódok könnyen bővíthetők legyenek további adattípusokkal (pl. további OMSZ adatokkal) és egyéb logikai műveletekkel. A kiterjesztés történhet például eltérő hosszúságú időszakokra (hétvégi vagy többnapos jelentések megírására), vagy különböző speciális tartalmú jelentésekre (pl. nyáron vizek hőmérséklete, télen hójelentés síterepekről, UV előrejelzés stb.)

A generált időjárásjelentések minőségének további javításához fontos irányt jelenthet a meteorológus szakértőkkel való közös munka, amelynek során az általunk meghatározott thresholdok és fontosság értékek tovább pontosítására nyílna lehetőség.

Szintén a további fejlesztések egy lehetséges irányát jelenti megfelelő méretű tanítóadatbázisok létrehozását követően az OpenAI-ban elérhető fine-tuning használata, amelynek során akár bizonyos újságokban szereplő időjárásjelentések speciális nyelvezetére is rátanítható lenne a nyelvi modell. Ez egyben az egyik lehetséges módja a generált szövegek további javításának. A másik utat a meglévő kódok magyar nyelvű nyelvi modellre való átültetése jelenti a jelenlegi angol helyett, amely az automatizált fordítás során megjelenő esetleges minőségromlást is kiküszöbölné. GPT-3 tudású magyar nyelvi modell (még) nem áll rendelkezésre, de GPT-2-re építve már voltak próbálkozások magyar nyelvű tartalmak előállítására (Yang Zijian, 2022).

Bibliográfia

Belz, A. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. Natural Language Engineering, 14(4). pp. 431–455. (2008)

Floridi, L., Chiriatti, M. GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30(4). pp. 681–694. (2020)

Gatt, A., Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research, 61. pp. 65–170. (2018)

Goldberg, E., Driedger, N., Kittredge, R. I. Using natural-language processing to produce weather forecasts. IEEE Expert, 9(2). pp. 45–53. (1994)

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. Improving language understanding by generative pre-training. (2018)

Ramos-Soto, A., Bugarin, A. J., Barro, S., Taboada, J. Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. IEEE Transactions on Fuzzy Systems, 23(1). pp. 44–57. (2014)

Reiter, E., Sripada, S., Hunter, J., Yu, J., Davy, I. Choosing words in computer-generated weather forecasts. Artificial Intelligence, 167(1-2). pp. 137–169. (2005)

Turner, R., Sripada, S., Reiter, E., Davy, I. P. Selecting the content of textual descriptions of geographically located events in spatio-temporal weather data. In: International conference on innovative techniques and applications of artificial intelligence. pp. 75–88. Springer, London (December 2007)

Yang, Z. Gy. „Az invazív medvék nem tolerálják az agressziót". Magyar GPT2 kísérleti modell. In*:* XVIII. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2022. január 27–28. pp. 464–476. Szegedi Tudományegyetem (2022)

Morfológia, előfeldolgozás

Korpusztisztítás és sorvégi kötőjelek kezelése karakteralapú neurális nyelvmodellel

Pethő Gergely, Sass Bálint, Simon László, Lipp Veronika

Nyelvtudományi Kutatóközpont
pagstudium@gmail.com
{sass.balint,simon.laszlo,lipp.veronika}@nytud.hu

Kivonat Cikkünk célkitűzése kettős: egyrészt bemutatunk egy olyan egyszerű és általános módszert, amellyel karakteralapú nyelvmodellek hasznosíthatóak egyebek mellett korpuszok tisztításában, másrészt ismertetünk egy olyan konkrét, tiszta magyar sajtónyelvi korpuszon tanított nyelvmodellt, amelyre építve jó eredményeket értünk el e módszer alkalmazásával. Továbbá nyilvánosan elérhetővé tesszük az akár karakter-, akár szószintű rekurrens neurális nyelvmodellek konfigurálását és (újra)tanítását szolgáló, Pythonban írt alkalmazást, amellyel a nyelvmodellünket tanítottuk, és amelynek segítségével akár ez a magyar sajtónyelvi modell hozzáigazítható más jellegű tanítókorpuszokhoz, akár új modell tanítható be. A bemutatott kétirányú LSTM-nyelvmodell erőforrásigénye aránylag szerény, és a javasolt módszert követve közvetlenül, vagyis az adott részfeladatra történő bármilyen további betanítás nélkül jól használható a korpusztisztítás során felmerülő feladatok széles körére, például idegen nyelvű, túl sok zajt tartalmazó szövegrészek azonosítására, szórványos OCR-hibák és hiányzó ékezetek javítására. A nyelvmodellt a sorvégi elválasztások egyértelműsítése feladatra értékeltük ki: a módszer teljesítménye ezen a feladaton meghaladta a nagyon magas baseline-t.
Kulcsszavak: karakteralapú nyelvmodell, n-gram-modell, LSTM, kétirányú nyelvmodell, autoregresszió, OCR, hibajavítás, korpuszok előfeldolgozása

1. Bevezetés

Az olyan kisebb méretű, kézzel összeállított korpuszok kivételével, mint amilyen például a *Magyar történeti szövegtár,* az *Ómagyar korpusz* vagy akár a *BNC* (Csengery és mtsai, 2016; Simon és Sass, 2012; Aston és Burnard, 1998), a „tisztítás", „zajszűrés" problematikája szinte minden esetben felmerül. Egy méretét tekintve szavak százmillióit tartalmazó szövegadatbázis rendszerint a webaratásnak nevezett technikával állítható elő, esetleg kisebb-nagyobb részkorpuszok kombinálásával vagy optikai karakterfelismertetéssel kapott szövegek felhasználásával. Ezekkel a módszerekkel gyorsan építhető nagyméretű korpusz, azonban annak anyagát – éppen a források sajátosságai miatt – szinte minden esetben több szempontból is javítani kell, illetve kiszűrni belőle a nem javítható részeket. Egy adott szövegadatbázis nyelvészeti, lexikográfiai célú felhasználása esetében az elvárások akár különösen szigorúak is lehetnek.

A korpusztisztításnak a tanulmányunkban részletesen ismertetett speciális részfeladatára a *visszaválasztás* (dehyphenation) terminust használjuk. Ezen a sorvégi elválasztójelek eltávolítását értjük oly módon, hogy helyreállítjuk az elválasztott szavak elválasztás előtti helyes formáját. A feladat nem triviális, négy lehetőség között kell dönteni: (1) Jelentős számban az egyszerű esettel találkozunk, mikor egyszerűen törölhetjük a kötőjel+sortörés kombinációt ('*kere-/tes*' → '*keretes*'). (2) Létezik ennek a digráfkezeléssel nehezített formája ('*hosz-/szú*' → '*hosszú*'). (3) További eset, mikor megmarad a kötőjel ('*egyszer-/kétszer*' → '*egyszer-kétszer*'). (4) Végül előfordul, hogy a sortörést szóközre kell cserélni: '*bal-/és jobboldali*' → '*bal- és jobboldali*'. Ez a probléma mindenhol felmerül, ahol a kiindulópont egy elválasztásokat tartalmazó nyomtatott mű.

A kötőjel (diviz, „-") írásjelnek csak az egyik funkciója a sorvégi elválasztás jelölése, a további lehetséges pozíciók feltérképezéséhez a magyar helyesírás részletszabályainak ismerete szükséges. A kötőjel a szövegekben előfordulhat a szóhatár jelöléseként toldalékok kapcsolásakor *(Tolouse-ban, 2022-ben)* vagy összetételekben *(balett-táncos, négy-öt)*, kettős családnevekben és földrajzi nevekben *(Batthyány-Strattmann, Decsi-Nagy-Holt-Duna)*, de szótagoláskor is *(á-bé-cé)*. Általában tapad az előtte álló szó utolsó és az utána következő szó első betűjéhez, de nem mindig, és ritkán megismétlődhet a sor elején (ha egy kötőjellel írt szó éppen a kötőjelnél kerül elválasztásra). A kötőjelnek a *„bükk-, tölgy- és kőrisfák"* vagy a *„gépgyártó, -szerelő és -javító műhely"* mellérendelő szerkezetekben való megjelenése tipográfiailag nagyon hasonló az elválasztójelként való használatához, ami szintén nehézséget okozhat az utóbbi beazonosításában.

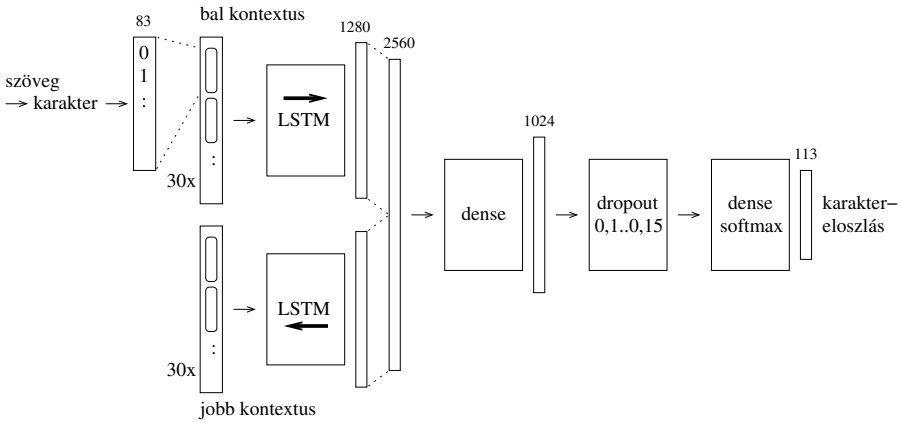
Annak részletezése, hogy a szöveges anyagokban található különféle hibákat és zajt mi okozza, és ezek hogyan javíthatók, szétfeszítené ennek a tanulmánynak a kereteit. Számos ilyen természetű probléma a szövegek OCR-ezéséből ered. Az OCR-hibák kézi javítása költséges és időigényes, ezért régóta kutatják az erre szolgáló automatikus módszereket. Ismert, hogy az írásjel képének önmagában való azonosítása esetében az output mindenképpen több hibát tartalmaz, mint akkor, ha az OCR-motor rendelkezik beépített nyelvmodellel (Mori és mtsai, 1999; Cheriet és mtsai, 2007). Laki és mtsai (2022) kutatásaik során OCR-hibákat tartalmazó szövegeket detektáló és javító modelleket implementáltak, valamint saját silver standard párhuzamos korpuszt építettek. Arra a megállapításra jutottak, hogy amikor kizárólag az OCR-hibák javítása volt a cél, akkor a modelljeiknek a Context-based Character Correction (CCC) modellel való kombinációja mutatkozott a leghatékonyabbnak.

2. A nyelvmodell kialakítása és jellemzői

Jelen tanulmányban a korpusztisztítási és -javítási feladatok megoldására nyelvmodellek által előállított valószínűségi predikciók közvetlen felhasználását javasoljuk a 3. részben leírt módon. Az alkalmazott nyelvmodellt Python nyelven, a TensorFlow gépi tanulási és a Keras mélytanulási keretrendszerben implementáltuk. A modell a jól ismert LSTM (Hochreiter és Schmidhuber, 1997) architektú-

rán alapul, hagyományos LSTM-rétegeket építettünk be egy céljainkra alkalmas saját fejlesztésű kétirányú autoregresszív neurális modellbe.

Amint ismert, az LSTM a rekurrens neurális hálók (RNN; Goodfellow és mtsai, 2016, 10. fejezet, Goldberg, 2017, 14. fejezet, Kamath és mtsai, 2019, 7. fejezet) egyik típusa. Az RNN-ek az előzményszekvenciának egy rögzített dimenzionalitású vektorreprezentációját állítják elő oly módon, hogy ezt a reprezentációt a k darab kontextuselem olvasása során lépésről lépésre, elemenként frissítik. Ez a reprezentáció az egyszerű Markov- vagy más néven n-gram-modellekkel (vö. Jurafsky és Martin, 2021, 3. fejezet) ellentétben nem diszkrét, hanem folytonos, és ennek köszönhetően a rekurrens neurális hálók sokkal sikeresebben képesek általánosítani a tanítóadatok alapján.



1. ábra: A modell architektúrája. A számok a dimenziót jelzik.

Az általunk betanított modell architektúrája az 1. ábrán látható. Bemenete nyers, UTF–8 kódolású szöveg, amelynek karaktereit első lépésben egy karakterkódoló egy-egy ritka numerikus vektorrá alakítja. A neurális háló bemeneti rétegét e vektorok egy $2k$ szélességű szekvenciája alkotja, ebből az első/második k vektor a végső kimenetként keresett középső karakter bal/jobb kontextusában szereplő karaktereket reprezentálja. Az első k vektort egy balról jobbra haladó, a második k vektort pedig egy jobbról balra haladó LSTM-réteg dolgozza fel. Mindkét LSTM-réteg kimenete egy-egy vektor, amely a bal, illetve jobb kontextus egészét reprezentálja. E két vektor konkatenációja a bemenete a következő perceptronrétegnek, ezt dropout és egy újabb perceptronréteg követi, amelynek softmax aktivációval kapott kimenete a lehetséges középső karakterek valószínűségi eloszlását közelíti. A nyelvmodell mérete mai mércével mérve aránylag kicsi, mintegy 11 millió paramétert tartalmaz. Ennek köszönhetően a modell gépigénye nem jelentős, memóriaigénye gyakorlatilag elhanyagolható, egy korszerű asztali gépen akár CPU-n is viszonylag jól futtatható, egy mondatnyi bemenetet érzékelhető késleltetés nélkül fel tud dolgozni. Geforce RTX 3060-as GPU-n 1 millió karaktert kb. 3,5 perc alatt prediktál, így hosszabb korpuszok feldolgozására akár egyetlen asztali gépen is alkalmas.

Tanítás során a szövegen egy $2k + 1$ karakter szélességű ablakot léptetünk egyesével úgy, hogy a középső karakternek megfelelő one-hot vektort kezeljük a neurális háló kimeneteként keresett célértékként, és a modell az általa adott előrejelzésnek e célértékhez viszonyított hibája alapján tanul kategorikus keresztentrópia veszteségfüggvénnyel. A rétegek jellemzői rendre a következők:

Karakterkódoló. A megszokottól eltérően nem véletlenszerűen inicializált és a modell betanítása során tanult karakterbeágyazást használtunk a bemenetben szereplő karakterek reprezentációjához, sem one-hot kódolást, hanem egy olyan ritka vektort, amely figyelembe veszi a magyar helyesírás és morfológia bizonyos relevánsnak feltételezhető sajátságait. A nyers karaktereket a következő bináris jegyek alapján jellemezzük: ±*betű*, ±*nagybetű*, ±*hosszú* (magánhangzó), ±*mellékjeles* (idegen betű), ±*szóköz*, ±*kötőjel*, ±*számjegy*. Ezt követően minden karaktert normalizálunk: kisbetűvé alakítjuk, a hosszú magyar magánhangzókat a rövid megfelelőjükké alakítjuk, az idegen mellékjeles betűkről eltávolítjuk a mellékjelet (pl. *ä*>*a*). Ezt a normalizálást persze a kimenetben már nem végezzük el, tehát a kimenetben az ékezetes és mellékjeles betűket helyesen prediktáljuk.

Ennek az inputreprezentációnak a motivációja, hogy a környező karakterek prediktálása szempontjából sokszor lényegtelen például, hogy egy magánhangzó rövid vagy hosszú változata jelenik meg éppen. Ennek oka lehet a tőtípusokban előforduló allofónia (*órakor* vs *órák*), az egyes rövid–hosszú párok (*i–í, u–ú, ü–ű*) gyakori tévesztése, vagy az idegen szavak helytelen írása (*Dalí* helyett *Dali*). Ugyanígy sokszor az sem lényeges, hogy egy betűnek a kis vagy a nagy változata (*Budapest* vs *budapesti*, mondatkezdő nagybetűk), vagy hogy pontosan melyik számjegy jelenik meg az adott ponton (*január 8.*). A betűk normalizálásának és jegyekre bontásának az a célja, hogy ezzel segítsük a hálózatot a releváns mintázatok megtanulásában, ami különösen a ritka és gyakran elrontott mellékjeles betűk szempontjából lesz várhatóan különösen hasznos. Amennyiben egy normalizált karakter gyakorisága a tanítókorpuszban nem ért el egy előre meghatározott küszöbértéket, a nyers karaktert a speciális OOV karakterként normalizáltuk. A 83 dimenziós karaktervektorok a 72-féle gyakori karakter, az OOV karakter, a három vezérlőkarakter (szöveg eleje, vége, maszkolás) one-hot kódolásából és a fenti bináris jegyeknek megfelelő plusz 7 dimenzióból épülnek fel.

Próbaképpen a kézi kódolás helyett szokványos, véletlenszerűen inicializált és a nyelvmodellel együtt tanult sűrű karaktervektorokkal is próbálkoztunk. Azt tapasztaltuk, hogy a megkülönböztetendő karakterek számánál (112) jelentősen alacsonyabb (16 vagy 32) dimenzionalitású ilyen beágyazásvektorokra építve markánsan kevésbé sikeresen prediktált a két egyébként azonos szerkezetű modell, és a karakterek számával közel azonos méretű, 64 dimenziós beágyazásvektorral sem teljesített a bináris vektorhoz képest jobban. Így az utóbbiak használata mellett döntöttünk, de az általunk elérhetővé tett alkalmazás igény szerint konfigurálható úgy is, hogy beágyazásvektorokkal tanuljon egy új modellt.

Bemeneti réteg. Két részből áll: a jobb és a bal kontextus karakterkódjaiból. A modellünkben bal és jobb kontextusként egyaránt 30–30 karaktert használtunk. Kísérleteinkben azt tapasztaltuk, hogy a 15 karakteres kontextus következetesen enyhén gyengébb predikciókat eredményezett, ugyanakkor a kontextus

50 karakterre növelésével a perplexitás egyáltalán nem javult tovább. Úgy tűnik tehát, hogy az ennyire tág kontextus vizsgálatával a modell nagyon ritkán jut a középső karakter predikcióját segítő információhoz, míg a tág kontextusnak a rekurrens rétegben kiszámolt reprezentációba beépülő zaja megnehezíti a hasznos mintázatok megtalálását a középső karakterhez közelebbi környezetben.

Kétirányú LSTM-réteg. Szintén két részből áll: az egyik LSTM-réteg a bal kontextuson balról jobbra halad végig, míg a másik a jobb kontextuson jobbról balra. Az input feldolgozása megáll a prediktálandó karakter előtti, illetve utáni karakternél, vagyis a prediktálandó középső karaktert értelemszerűen nem építi be a reprezentációvektorba. A két LSTM-réteg egymástól független, egymással nem kommunikálnak, és egymás inputját sem látják, kimenetük egy-egy 1280 dimenziós vektor. Kísérleteztünk kisebb LSTM-rétegekkel is, de Karpathy és mtsai (2015, 1. ábra) eredményeivel összhangban azt láttuk, hogy a prediktálóképesség nő a rekurrens réteg dimenzionalitásának növelésével.

Megjegyzendő, hogy az általunk választott kétirányú LSTM-architektúra eltér a szokásosan választott implementációktól. Kétirányú RNN-eket többnyire kétféle módon szoktak gépi tanulási modellekbe beépíteni. Szövegosztályozás vagy szekvenciagenerálás esetén szokásos alkalmazási mód, hogy a két rekurrens réteg ugyanazt a hosszabb szekvenciát dolgozza fel (balról jobbra, illetve jobbról balra), így a két RNN együttesen jól képes reprezentálni a szekvencia egészét. Elemenkénti címkézés esetén ellenben mind az előre, mind a hátra haladó RNN-réteg karakterenként generál egy-egy vektorreprezentációt oly módon, hogy az egyik az adott elem bal, míg a másik a jobb kontextusáról szóló információkat összegzi, beleértve az adott elemet is. Mivel esetünkben ezektől eltérő, autoregressziós feladatot kellett megoldanunk, ahol lényeges, hogy a prediktálandó karaktert magát nem olvashatjuk be, célszerűnek láttuk egy olyan hálózat felépítését, amely az említett módon karakterről karakterre halad, az adott karakternek csak egy szűkebb környezetét vizsgálja, és a két rekurrens réteg a középső karakter beolvasása nélkül középen „találkozva” prediktálja azt.

Sűrűn kapcsolt réteg. Bemenete a bal és a jobb LSTM-réteg végső állapotának konkatenációja, azaz egy $1280 + 1280 = 2560$ dimenziós vektor. Maga a sűrűn kapcsolt réteg 1024 neuront tartalmaz. Kísérleteztünk ettől eltérő mérettel is: ha növeltük a neuronok számát, nem tapasztaltuk a predikciós képesség további javulását, míg ha csökkentettük, akkor romlott.

Dropout. Ismert, hogy a dropout révén történő regularizáció növeli a mély neurális modellek predikciós képességét azáltal, hogy a modellt a hasznos általános mintázatok megtanulására kényszeríti a tanítóadatok „bemagolása” helyett. Az elsőként kipróbált 0,5-ös dropouttal a modell predikciós képessége a validációs adatokon mérve messze elmaradt a dropout nélküli azonos méretű modellétől. Ugyanakkor a dropout teljes elhagyása esetén a tanítás során túlillesztés lépett fel, az elszálló gradiensek hatására a részben betanult hálózat ugrásszerűen és helyrehozhatatlanul elromlott. Végül legsikeresebbnek a kezdetben 0,1-es (majd a második epochtól 0,15-re növelt) dropout bizonyult, ez ugyanis egyfelől elegendő volt a gradiensek elromlásának megelőzéséhez, másfelől pedig a perplexitás lehető legkisebb mértékű romlásával járt.

Kimeneti réteg. Softmax aktivációs sűrűn kapcsolt réteg. A 113 dimenzió mindegyike egy-egy one-hot kódolt prediktálandó középső karakternek felel meg: egy érték a ritka karaktereket magában foglaló OOV kategória, a többi egy-egy konkrét betűt vagy írásjelet kódol. A (természetesen normalizálatlan) outputkarakterek számát a bemeneti normalizált karakterek számához hasonlóan kaptuk: a 112-féle gyakori karakter az OOV karakterrel együtt adja ki a dimenziószámot.

A fentieket összegezve: miután a megszokott rácskereséses eljárással megvizsgáltuk a szóba jöhető modellkonfigurációk és hiperparaméterek széles körét, azon modell teljes betanítása és alkalmazása mellett döntöttünk a továbbiakban, amely egy kiválasztott, mintegy 1 millió szó terjedelmű, a tanítókorpusszal azonos műfajú és témájú (azaz magyar sajtónyelvi) validáló korpuszon a legjobb predikciós teljesítményt nyújtotta a legjobb feldolgozási idő mellett. Az így meghatározott modell az 1. ábrán látható. Az általunk kézzel meghatározott bináris jegyes karakterkódolást alkalmazza, a prediktálandó karakter 30–30 karakternyi kontextusát veszi tekintetbe balra és jobbra. Az LSTM rétegek 1280 cellát tartalmaznak, ezek konkatenációja szolgál bemenetként egy 1024 neuronból álló sűrűn kapcsolt rétegnek, amely eleinte 0,1, majd 0,15 arányú dropouttal tanul. Néhány azonos felépítésű LSTM-modellünk perplexitásértéke az 1. táblázatban látható.

1. táblázat. Néhány modell perplexitása különböző hiperparaméterek mellett.

LSTM	sűrű	dropout	epoch	perplexitás	pontosság
512	512	0,5	2	1,182	94,96
512	512	0,1	3	1,112	96,54
1280	1024	0,1→0,15	3	1,081	97,31

3. Módszer: alternatívák közötti döntés egy nyelvmodell által prediktált perplexitások alapján

Módszerünk lényege, hogy közvetlenül használja fel minden nyelvmodell tulajdonképpeni célját, raison d'être-ét: az arra való képességüket, hogy megadják egy szekvencia ismeretében a szekvencia lehetséges következő elemeinek a valószínűségi eloszlását. A nyelvmodellnek ezt a funkcióját a talán megszokottabb használati móddal ellentétben nem szekvenciák generálásában, hanem felismerésében hasznosítjuk. A nyelvmodell kimenetének „közvetlen" felhasználásán azt értjük, hogy nem építünk rá transzfertanulás jellegű eljárással összetettebb gépi tanulási modellt, valamint nem közlünk a nyelvmodellel semmilyen azzal kapcsolatos információt, hogy mire akarjuk használni a kimenetét, hanem úgymond tökéletesen „zero–shot" elven generáltatunk vele predikciókat.

Ez utóbbi az elképzelhető legegyszerűbb módon történik: a karakteralapú nyelvmodellünkkel karakterről karakterre haladva feldolgoztatunk egy karaktersorozatot úgy, hogy az lépésenként a bal és jobb 30–30 karakter alapján előrejelezze a pillanatnyi középső karakterpozíció valószínűségi eloszlását, majd ezt összevetjük a ténylegesen a karaktersorozatban az adott helyen szereplő karakterrel. E predikciókból kiszámoljuk az adott karaktersorozat egészének átlagos

karakterenkénti valószínűségét, illetve pontosabban (ami ezzel egyenértékű) a karaktersorozatnak a nyelvmodell által becsült perplexitását. A modell által adott válasz mindenekelőtt arra használható fel, hogy kiválasszuk a hibátlant vagy legalábbis a legkevésbé hibásat egyazon szövegrészlet több különböző változata közül. Ezeket az alternatívákat generálhatjuk mi magunk, amire a 5. részben mutatunk konkrét példát – hasonló jellegű alkalmazási módok az ékezetek pótlása, nem tapadó és rosszul formázott idézőjelek, gondolat- és kötőjelek visszatapasztása, valamint véletlenszerűen beszúródott szóközök által széttördelt szavak töredezettségmentesítése. Elképzelhető viszont az is, hogy a feldolgozandó szövegváltozatok eleve adottak, például ugyanazon nyomtatott dokumentum alapján több különböző OCR-motor által generált kimenetek. Lényeges szempont, hogy mivel a nyelvmodellünk karakterszinten értékeli ki a megadott változatokat, nem két vagy több teljes mondat vagy akár szöveg között kell választanunk, hanem ezek megfelelő illesztését feltételezve képesek vagyunk akár karakterenként válogatva a különböző inputokból összerakni a helyes vagy legalábbis a rendelkezésre álló leghelyesebb outputot. Mivel zajt jól tűrő neurális nyelvmodellt használunk, az többnyire olyan helyzetekben is képes használható választ adni, amikor nemcsak a javítandó elem hibás, hanem annak kontextusa is zajos.

Alább, az 5. fejezetben részletesen ismertetjük módszerünk alkalmazását és kiértékeljük predikcióinak minőségét egy esettanulmány kapcsán. Emellett itt röviden bemutatunk egy-két példa alapján több más olyan alkalmazási lehetőséget, amelyek tapasztalataink szerint könnyen implementálhatóak és jó eredményeket adnak. A modellek építéséhez és kiértékeléséhez szükséges kód, maguk a modellek és egyéb kapcsolódó információk elérhetők a https://github.com/ril-lexknowrep/character_language_models repozitóriumban.

3.1. Erősen hibás szövegrészek szűrése

Egy nyelvmodell értelemszerűen akkor fogad el egy neki átadott szekvenciát, azaz abban az esetben tulajdonít neki alacsony perplexitást, ha az az adott modell által reprezentált nyelven íródott. Konkrétabban a karakteralapú nyelvmodellünk lényegében bármely magyarul íródott helyes mondatnak 1 és 2 közötti, inkább 1-hez közeli perplexitást tulajdonít. Ettől drasztikusan eltér az erősen hibás vagy tisztán OCR-szemétből álló szekvenciák prediktált perplexitása, például az alábbié 15,3:

```
Nílusi hajóul közvetlen chalerjarultal.
5* m luxuslupóval: 129 900 Ftíőtől
```

Egy extrémebb, 2900 körüli perplexitású példa:

```
A a L ét • a  ffc Mkm bML A hWiWWHK1**(re a nMtjjww áHő
```

Könnyen kiszűrhetjük tehát az erőteljesen zajos mondatokat vagy szakaszokat azáltal, hogy meghatározunk empirikusan egy perplexitási küszöbértéket, amely nagy biztonsággal arra utal, hogy az adott szekvencia erősen hibás, és minden ilyen mondatot vagy szakaszt törlünk a korpuszból. Gondolhatnánk, hogy ez

a feladat a Pythonban rendelkezésre álló nyelvfelismerő könyvtárakkal könnyen megoldható, azonban ez nincs így. Kísérleteinkben például a `langid` sok esetben „magyarabbnak” értékelte a jelentősen zajos változatot, mint a hibátlant.

3.2. Idegen nyelvű szövegrészek szűrése

A karakteralapú nyelvmodellek egyik klasszikus alkalmazási területe a nyelvfelismerés (vö. 4. szakasz): ha egy mondatnak egy magyar nyelvmodell jelentősen alacsonyabb perplexitást tulajdonít, mint egy angol nyelvmodell, akkor biztosak lehetünk benne, hogy a mondat nem angolul van; és ha azt is biztosan tudjuk, hogy csak a két nyelv jöhet szóba, akkor biztosan magyarul van. Ugyan ez az előbbiekből nem következik, és nem is magától értetődő, de történetesen nem szükséges több nyelvmodell ahhoz, hogy idegen nyelvű részeket kiszűrjünk egy magyar nyelvű korpuszból, hanem elég az egyetlen magyar nyelvmodell is. Míg, mint említettük, a magyar mondatok perplexitása tipikusan az 1–2, nagyon szokatlan esetekben a 3–4 tartományban mozog, addig az idegen nyelvű szövegeké rendszerint 10 fölötti, így az utóbbiak ily módon egyértelműen elkülöníthetőek az előbbiektől. Az idegen nyelvű részek így gyakorlatilag „zajként” sorolhatók be és törölhetőek a rontott szövegekkel együtt, ugyanúgy egy perplexitási küszöbértéket mint szűrési feltételt használva.

3.3. OCR-javítás

A perplexitás alapján történő választás módszerének különösen ígéretes felhasználási területe az OCR-hibák javítása. Ha rendelkezésünkre áll egyazon digitalizált dokumentum több különböző OCR-motorral szöveggé alakított változata, ezek a nyelvmodellre támaszkodva összevethetőek, és a legalacsonyabb perplexitású változat tekinthető helyesnek. Egy tipikus példa:

```
V1: a község egyik hivatalnokát háza elõtt agyonszarták.
V2: a község egyik hivatalnokát háza elõtt agyonszurták.
V3: a község egyik hivatalnokát háza elõtt agyonszúrták.
```

A V1 változatot az ABBYY Finereader 12, a V2-t a FR 15 generálta (az utóbbi a digitalizált dokumentumban szereplő régies írásmódú alakot tartalmazza), míg a V3 általunk kézzel javított változat. Nyelvmodellünk az utolsó szavakra a következő összesített perplexitást prediktálja a fenti (egyébként azonos) kontextusokban: *agyonszarták* 10,16, *agyonszurták* 6,45, valamint *agyonszúrták* 1,72.

A nyelvmodell igen magabiztosan választja ki a helyes változatot a három lehetőség közül. Emellett figyelemre méltó, hogy a tartalmilag helyes, csupán az ékezetet nélkülöző *agyonszurták* sokkal jobb eredményt mutat, mint a helyesen írt, de tartalmilag értelmetlen *agyonszarták*. Vegyük észre, hogy a nyelvmodell a kontextusban jelen lévő zaj ellenére (a Finereader outputjában tildés *õ* szerepel *ő* helyett) képes meghozni a helyes döntést. További viszonylag kiterjedt kísérleteket is folytattunk e módszerrel végzett OCR-javítással, és azt tapasztaltuk, hogy az eljárás rendkívül sikeresen javít olyan típusú hibákat, amelyek egy szótáralapú helyesírás-ellenőrzőnek nehézséget okoznának, így különösen elrontott központozási jeleket (pl. pont helyett vessző).

4. A módszer szakirodalmi előzményei

A szakirodalmat áttekintve úgy látjuk, hogy az általunk követett egyszerű eljáráshoz hasonló megközelítést ritkán alkalmaznak; vagy ha mégis, akkor egyszerűsége miatt nem számolnak be róla. Az egyetlen általunk ismert terület, ahol jól dokumentálva van a karakteralapú nyelvmodellek (ezen belül mindenekelőtt az n-gram-modellek) által kiadott valószínűségi, illetve perplexitási predikciók közvetlen felhasználása, a nyelvfelismerés.

Amint korábban utaltunk rá, ennek az az alapötlete, hogy ugyanazon szöveg valószínűségét/perplexitását több különböző nyelv modelljével prediktáltatjuk, és amelyik modell a legvalószínűbbnek tartja az adott szöveget, azt azonosítjuk annak nyelveként. Ez a módszer az általunk itt javasolt módszernek pont a fordítottja: míg itt egyazon szöveg különböző változatainak a valószínűségére adunk becslést egyetlen modellt használva, és így választunk a szövegváltozatok közül, addig a nyelvfelismerés kapcsán ugyanazon szöveg valószínűségére több különböző nyelvmodell alapján adnak becslést, és így választanak a nyelvmodellek közül. Az n-gram-modellek ilyen használatáról szóló klasszikus szakirodalom Dunning (1994). A téma egy későbbi releváns feldolgozása Řehůřek és Kolkus (2009), amely jó áttekintést ad az egyéb kapcsolódó irodalomról is. Az itt javasolt megoldáshoz legközelebb Pethő és Mózes (2014) áll, amely viszont hagyományos Markov-modellt használ nyelvfelismerésre.

Karakteralapú megközelítésre az ékezetek és egyéb mellékjelek pótlása témakörében is találunk példát. Mihalcea és Nastase (2002) a prediktálandó karakter bal és jobb 5 karakter széles kontextusában elhelyezkedő karakterekre mint jegyekre épít döntési fát, illetve esetalapú tanulót. Ennek a megoldásnak nyilvánvaló hátulütője a nyelvmodell használatával szemben, hogy a betanított modell kizárólag azt a feladatot képes megoldani, amelyet megtanult.

5. Esettanulmány: visszaválasztás

5.1. Célkitűzés

A választott feladatunk a következő: sorvégi elválasztást tartalmazó anyagban előállítani az eredeti elválasztatlan szöveget. Amint a bevezetőben említettük, ez azért nem triviális, mert a kötőjel+sortörés kombinációt nem szabad egyszerűen törölni, hanem négy különböző esetet kell kezelni. Ezekre a következő kódokkal hivatkozunk: `{1}` egyszerű törlés; `{2}` törlés digráfkezeléssel; `{3}` a kötőjel megtartása; `{4}` a kötőjel megtartása és a sortörés szóközre cserélése. A 3. részben leírt módszert úgy alkalmazzuk, hogy legeneráljuk a fenti négy lehetséges alternatívát, és azt fogadjuk el jónak, amelyikre a legkisebb perplexitást kapjuk.

5.2. Anyagok

Mind az n-gram-, mind az LSTM-modellt, mind pedig a szabályalapú visszaválasztó eszközünk esetalapú komponensét azonos szövegkorpuszon tanítottuk

be, amely az MNSZ2 (Oravecz és mtsai, 2014) sajtónyelvi részén alapul. Ezt a részkorpuszt reguláris kifejezésekkel javítottuk és szűrtük, és ezáltal elhárítottunk benne különböző karakterkódolási hibákat és egyéb anomáliákat (pl. nem tapadó írásjelek). A tanítókorpusz kiválasztása során döntő szempont volt, hogy ne tartalmazzon optikai karakterfelismeréssel nyert, valamint eredetileg elválasztott formában tárolt szövegeket, tehát minőségi szempontból a lehető legtisztább legyen; sajtókorpusz legyen, miután magunk is sajtónyelvi korpusz tisztítására kívánjuk használni; valamint elég nagy méretű legyen. Tanítókorpuszunk terjedelme mintegy 250 millió szövegszó, illetve közel 2 milliárd karakter. E korpusz a 90-es évek végétől a 2010-es évek elejéig terjedő szövegeket tartalmaz, és 11 online és nyomtatott sajtótermék (egyebek mellett az Origo, a 168 óra, a HVG, a Népszabadság, a Népszava és a ComputerWorld) anyagaiból áll össze.

5.3. Kiértékelés

A neurális rekurrens modellel történő visszaválasztás kiértékelése céljából annak sikerességét egy nagyon egyszerű baseline mellett egy kombinált szabályalapú–esetalapú elven működő, szintén általunk írt visszaválasztó alkalmazással vetettük össze, valamint az általunk javasolt, perplexitások összehasonlításán alapuló módszert úgy is vizsgáltuk, hogy a kétirányú LSTM-modell helyett kétirányú n-gram-modellel számíttattuk a perplexitásokat.

A szabályalapú–esetalapú alkalmazásunk dióhéjban összefoglalva azt vizsgálva hoz döntést az inputszövegében szereplő kötőjelek feloldásának mikéntjéről, hogy a sor végén álló kötőjel két oldalán álló egy-egy szó a magyar `hunspell` szerint egybeírva egy szót alkot-e (ez az alkalmazás szabályalapú része), valamint hogy a feldolgozott tanítókorpuszban a két szó előfordult-e egybeírva, illetve kötőjellel írva, és amennyiben igen, e két írásmód egymáshoz képest mennyire volt gyakori (ez az esetalapú komponens). Ezeket és további szempontokat összekapcsolva az alkalmazás heurisztikusan egyértelműsíti a két szó közötti kötőjeleket.

2. táblázat. Az n-gram-modellek perplexitásértékei.

n	perplexitás	pontosság	memória
5	1,707	83,28	7 GB
6	1,495	87,52	25 GB
7	1,424	89,37	70 GB
8	1,404	90,17	180 GB

A neurális rekurrens modell alternatívájaként egy alapvetően hagyományos, bár kétirányú n-gram-modellt is kipróbáltunk. Különböző rendű, magunk által implementált n-gram-modellekkel próbálkoztunk, ezek rendje 5-től 8-ig terjedt. A kétirányú n-gram-modell úgy állt össze, hogy két egyszerű egyirányú, szükség esetén eggyel alacsonyabb rendű modellre visszalépő n-gram-modellel prediktáltuk a középső karakter valószínűségi eloszlását úgy, hogy az egyik modell balról jobbra, a másik jobbról haladt a középső karakter felé. A két n-gram-modell által prediktált valószínűségi eloszlásvektort elemenként összeszoroztuk, majd

a szorzatokat normalizáltuk 1-re, és ennek eredményét tekintettük a kétirányú n-gram-modell által előrejelzett valószínűségi eloszlásnak.

A különböző méretű n-gram-modelljeinkre a 2. táblázatban lévő perplexitás-értékeket mértük a validáló korpuszon. A kiértékeléshez a 7-gram-modellt használtuk a hasonló pontosságú, de 250 millió szós korpuszon tanítva kezelhetetlenül nagy memóriaigényű 8-gram-modell helyett. Összehasonlításképpen (ld. 1. táblázat) a kétirányú LSTM-modelleknek azonos korpuszon mért perplexitása ehhez képest jelentősen jobb volt, és memóriaigénye is elhanyagolható.

A kiértékelés a következőképpen történt. Tetszőleges sima szövegből kiindulva a szöveget szabályos automatikus elválasztással 40 karakter szélességű sorokra tördeltük (a `pyphen` csomag segítségével), szimulálva egy elválasztásokat tartalmazó nyomtatott mű felépítését. Minden sor végén a megfelelő kóddal megjelöltük, hogy az adott ponton az eredetiben melyik elválasztási eset szerepelt, azaz hogy mi lesz a helyes visszaválasztás. A kiértékelés során csak a sorok végére eső kötőjelekkel foglalkoztunk, a sorok belsejében lévőkkel nem. Az így annotált gold fájlokból az osztálykódokat leválasztva kaptuk a inputfájlokat, ezeken futtattuk a visszaválasztó algoritmusokat, melyeknek a feladata a kód prediktálása volt. A kiértékelési mérőszámok a két kódsor összevetésével egyszerűen adódtak.

Négy konfigurációt vetettünk össze.

1. **triviális** baseline: ez egyszerűen egyszerűen törli az az összes kötőjel+sortörés kombinációt, `{1}`-es kódot oszt ki minden sorvégi kötőjelnek, azaz minden esetben úgy tekinti, mintha egyszerű elválasztást kellene visszaalakítani
2. **szabály-** és esetalapú módszer (ld. fentebb)
3. perplexitásalapú döntés kétirányú **7-gram** nyelvmodellel (ld. fentebb)
4. perplexitásalapú döntés kétuirányú **LSTM** nyelvmodellel (ld. 2. és 3. rész)

A kiértékelést egy 100 000 sort tartalmazó korpuszon futtattuk le, mely 30 592 sorvégi elválasztást tartalmaz. Ennél a méretnél a legritkább kategóriára is van nagyjából 100 előfordulás. Ennek a korpusznak a felén kiértékelve gyakorlatilag ugyanazokat a számokat kapjuk, így nem érdemes tovább növelni a kiértékelő korpusz méretét.

5.4. Eredmények

A 3. táblázatban látható eredményeink azt mutatják, hogy a szabályalapú és a perplexitásalapú eljárások hasonlóan jó eredményt adnak: a probléma természetéből adódó nagyon magas triviális baseline fölött teljesítenek, a hibák számát 40-60%-kal csökkentik. A legnehezebb, `{4}`-es kategóriában a szabályalapú módszer kifejezetten rosszul teljesít, a két nehéz kategóriában az LSTM-modell adja a legjobb eredményt. Érdekes, hogy a szóban forgó alkalmazás tekintetében az n-gram modell kicsivel jobbnak tűnik az LSTM-nél. Látjuk azt is, hogy a kategóriák sorra egyre nehezebb feladatot jelentenek a módszereknek.

A validálókorpusz hibáit elemezve azt látjuk, hogy azon túl, hogy nagyon ritkán elrontja a digráf kezelését ('*hoszszú*'), csak a tényleg nehéz esetekben téved az LSTM-et használó eljárás. Helyesen kezeli például a '*lopottautó- vagy fegyverkereskedelemből*' és a '*harmincasok-negyvenesek*' kifejezést. Előfordul, hogy

lenyeli a kötőjelet ('*védelmipénzszedés*' a helyes '*védelmipénz-szedés*' helyett vagy '*diszkódrogfogyasztótáborral*' a helyes '*diszkódrogfogyasztó-táborral*' helyett), vagy meghagyja a szóközt ('*ex- Ybl-bankos*' az '*ex-Ybl-bankos*' helyett). A szintén helytelen eredményt jelentő '*Gelá-ban*' azért is érdekes, mert az effajta ritka és idegen földrajzi nevek toldalékolt alakjait az anyanyelvi nyelvhasználó is kötőjellel írja néha, hiába kell szabály szerint egybeírni.

3. táblázat. A négy konfiguráció kiértékelése.

		pontosság	fedés	F_1	accuracy
triviális	{1}	0,981	**1,000**	0,991	
	súlyozott átlag	0,963	0,981	0,972	
					0,981
szabály	{1}	**0,999**	0,994	0,997	
	{2}	1,000	**0,989**	**0,994**	
	{3}	0,718	0,654	0,685	
	{4}	0,125	0,163	0,142	
	súlyozott átlag	**0,994**	0,988	0,991	
					0,988
7-gram	{1}	0,997	0,998	**0,998**	
	{2}	1,000	**0,989**	**0,994**	
	{3}	**0,774**	0,692	0,731	
	{4}	0,316	**0,378**	0,344	
	súlyozott átlag	0,993	**0,993**	**0,993**	
					0,993
LSTM	{1}	0,996	0,997	0,997	
	{2}	1,000	0,749	0,856	
	{3}	0,697	**0,825**	**0,755**	
	{4}	**0,579**	0,337	**0,426**	
	súlyozott átlag	0,992	0,992	0,992	
					0,992

Köszönetnyilvánítás. A modellek tanítását az ELKH Cloud infrastruktúrán futtattuk. Köszönetet mondunk az ELKH Cloud (lásd: Héder és mtsai, 2022; https://science-cloud.hu/) használatáért, ami hozzájárult a publikált eredmények eléréséhez.

6. Konklúzió

Cikkünkben egy speciális karakteralapú kétirányú LSTM architektúrát mutatunk be, amely autoregresszió révén egy kérdéses karakter(szekvencia) perplexitását adja meg, annak bal és jobb oldali kontextusa alapján. Egy ilyen modell segítségével egy egyszerű módszert alkalmazva arról tudunk dönteni, hogy egy szöveg bizonyos változatai közül melyik a hibátlan (legkevésbé hibás) verzió. Ezt a módszert alkalmasnak tartjuk számos korpusztisztító lépés megvalósítására, ezt példákon mutattuk be. Részletesen a visszaválasztás problémájára értékeltük ki a módszert, azaz elválasztásokat tartalmazó szövegben állítottuk helyre az eredeti szöveget a kötőjelek, szóközök és szükség esetén a digráfok helyes kezelésével. Baseline-nal összevetve azt kaptuk, hogy ez a viszonylag kevés erőforrást igénylő neurális módszer jó teljesítményt nyújt a nehéz esetek kezelésében is.

Hivatkozások

Aston, G., Burnard, L.: The BNC Handbook: Exploring the British National Corpus with SARA. Edinburgh University Press, Edinburgh (1998)

Cheriet, M.E.A., Kharma, N.N., Liu, C.L., Suen, C.Y.: Character Recognition Systems: A Guide for Students and Practitioners. Wiley-Interscience, 605 Third Avenue New York, NY United States (2007)

Csengery, K., Lipp, V., Simon, L.: A magyar történeti szövegtár bővítése. korhatáremelés 2010-re. In: Ittzés, N.e. (szerk.) A magyar nyelv nagyszótára. VI. kötet. Di–ek., pp. 926–927. MTA Nyelvtudományi Intézet, Budapest (2016)

Dunning, T.: Statistical identification of language. Technical Report MCCS. New Mexico State University (1994)

Goldberg, Y.: Neural Network Methods for Natural Language Processing. No. 37 in Synthesis Lectures on Human Language Technologies, Morgan & Claypool (2017)

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org

Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)

Héder, M., Rigó, E., Medgyesi, D., Lovas, R., Tenczer, S., Török, F., Farkas, A., Emődi, M., Kadlecsik, J., Mező, G., Pintér, A., Kacsuk, P.: The past, present and future of the ELKH Cloud. Információs Társadalom 22(2), 128–137 (2022)

Jurafsky, D., Martin, J.H.: Speech and Language Processing, 3. kiadás. Kézirat. Stanford University, Stanford (2021), https://web.stanford.edu/~jurafsky/slp3/

Kamath, U., Liu, J., Whitaker, J.: Deep Learning for NLP and title = variable-length and variable-language texts variable-length and variable-language texts, Speech Recognition. Springer, Cham (2019)

Karpathy, A., Johnson, J., Li, F.: Visualizing and understanding recurrent networks. CoRR abs/1506.02078 (2015), http://arxiv.org/abs/1506.02078

Laki, L.J., Kőrös, Á., Ligeti-Nagy, N., Nyéki, B., Vadász, N., Yang, Z.G., Váradi, T.: OCR-hibák javítása neurális technológiák segítségével. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2022). pp. 417–430. Szegedi Tudományegyetem, Szeged (2022)

Mihalcea, R., Nastase, V.: Letter level learning for language independent diacritics restoration. In: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002) (2002), https://aclanthology.org/W02-2021

Mori, S., Nishida, H., Yamada, H.: Optical character recognition. J. Wiley, NY United States (1999)

Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014). pp. 1719–1723. European Language Resources Association (ELRA), Reykjavik, Iceland (2014)

Pethő, G., Mózes, E.: An n-gram-based language identification algorithm for variable-length and variable-language texts. Argumentum 10, 56–82 (2014)

Simon, E., Sass, B.: Nyelvtechnológia és kulturális örökség, avagy korpuszépítés ómagyar kódexekből. Általános Nyelvészeti Tanulmányok 24, 243–264 (2012)

Řehůřek, R., Kolkus, M.: Language identification on the web: Extending the dictionary method. In: Proceedings of the 10th International Conference on Intelligent Text Processing and Computational Linguistics. pp. 357–368 (2009)

Transformer-alapú HuSpaCy előelemző láncok

Szabó Gergő, Orosz György, Szántó Zsolt,
Berkecz Péter, Farkas Richárd

Szegedi Tudományegyetem, Informatikai Intézet
Szeged, Árpád tér 2.
{gszabo,szantozs,berkecz,rfarkas}@inf.u-szeged.hu
gyorgy@orosz.link

Kivonat Évről évre jelennek meg egyre pontosabb nyílt forráskódú természetes-nyelvfeldolgozó transformer-alapú nyelvi modellek. Viszont ezekre épülő, kifejezetten magyar nyelvre fejlesztett előelemző lánc, ami tartalmaz tokenizálást, mondatra bontást, lemmatizálást, szófaji egyértelműsítést, morfológiai címkézést, függőségi elemzést és névelem-felismerést, idáig nem létezett. Dolgozatunkban bemutatunk egy, a fent említett feladatokat megvalósító transformer-alapú előelemző láncot, amit több különböző magyarra elérhető nyelvi modellel is teszteltünk. Az elkészült rendszer a feladatok nagy részében state-of-the-art eredményeket ért el. Arra is kellő figyelmet fordítottunk, hogy megvizsgáljunk különböző erőforrás-igényű lehetőségeket és bemutassunk olyan módszereket, amelyekkel a leghatékonyabbnak bizonyult rendszerek memóriahasználata csökkenthető.
Kulcsszavak: transformer, huspacy, nyelvi előfeldolgozás

1. Bevezetés

Az elmúlt években nagyon elterjedtek a transformer-alapú modellek alkalmazása (például: `BERT` (Devlin és mtsai, 2018)). Ezek a mély neuronhálókat használó, nagy mennyiségű szövegen előtanított nyelvi modellek azonnal state-of-the-art eredményeket értek el a természetes-nyelvfeldolgozás különböző feladatain (Devlin és mtsai, 2018). Bár már magyar nyelvre is elérhetők ilyen modellek (Nemeskey, 2022) (Conneau és mtsai, 2019), olyan ezekre épülő, kifejezetten a magyar nyelv igényeit figyelembe vevő általános előelemző keretrendszer eddig nem készült, ami egy architektúraként foglalná össze a tokenizálást, a mondatra bontást, a szófaji egyértelműsítést, a morfológiai elemzést, a lemmatizálást, a függőségi elemzést és a névelem-felismerést. Célunk egy ilyen eszköz létrehozása volt.

Elemzőnk alapjául a spaCy (Honnibal, 2015) keretrendszer magyar nyelvű változatát, a HuSpaCy-t (Orosz és mtsai, 2022) választottuk. A HuSpaCy egy Python nyelvre építő, nyílt forráskódú, szabadon felhasználható nyelvfeldolgozási keretrendszer, ami a korábban felsorolt komponensek mindegyikét tartalmazza. Cikkünkben bemutatjuk, hogy a HuSpaCy-ben található szóbeágyazások lefedő konvolúciós rétegekre épülő neuronháló esetében, milyen javulás érhető el az

előtanított transformer-alapú modellek beépítésével. Kísérleteinkhez a kizárólag magyar adaton tanított huBERT-et (Nemeskey, 2022) és két többnyelvű modellt, a Multilingual BERT-et (Pires és mtsai, 2019) és az XLM-RoBERTa-t (Conneau és mtsai, 2019) használtuk.

A transformer modellek hátránya a nagy számításikapacitás- és a memóriaigényük. Utóbbi kifejezetten igaz a többnyelvű modellek esetén, hiszen ezen modellek szótára óriási mennyiségben tartalmaz olyan szótöredékeket, amik a magyar nyelvben nem fordulhatnak elő. Ennek a problémának a kezelésére megszűrtük az XLM-RoBERTa szótárát és megvizsgáltuk, hogy ez milyen hatással van a rendszer teljesítményére.

A további hatékonyság javítás érdekében a HuSpaCy címkézésért felelős neurális architektúrájában két ponton változtattunk. A korábban használt átmenet-alapú függőségi elemzőt kicseréltük egy gráf-alapú változatra, ezen felül a névelem-felismerőt is bővítettük a beam search (Medress és mtsai, 1977) beépítésével.

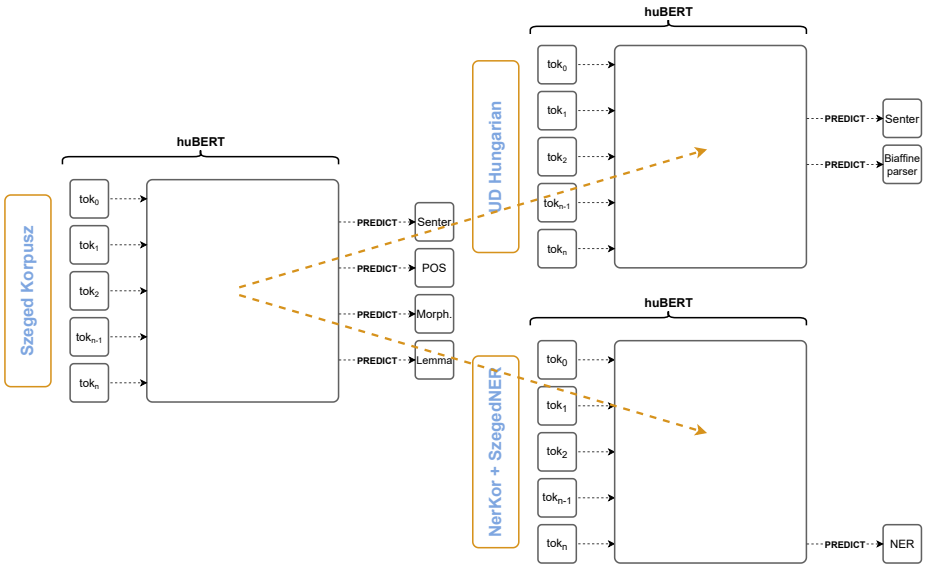
2. Kapcsolódó munkák

Kutatásunk fő célja, hogy megvizsgáljuk, hogy a legújabb transformer-alapú megoldások hogyan teljesítenek a korábbi megoldásokkal szemben. A magyar nyelvre már korábban is fejlesztettek teljes előelemző láncokat. Ilyen például a HuSpaCy-large (Orosz és mtsai, 2022) modellje, amely a WebCorpus 2.0-án (Nemeskey, 2020b) tanított statikus szóbeágyazásokat használ, illetve egy 4 rétegű CNN-t (*Convolutional Neural Networks*); a magyarlanc (Zsibrita és mtsai, 2013), amely a rejtett Markov modell-alapú PurePos-t (Orosz és Novák, 2013) használja és egy gráf-alapú függőségi címkézőt; a Stanza (Qi és mtsai, 2020), amely LSTM-alapú (*Long Short Term Memory*) és amely magas pontszámokat ért el az UD korpuszok terén; az UDPipe (Straka, 2018), amelyet alaprendszerként használnak a CoNLL versenyeken; a UDify (Kondratyuk és Straka, 2019), ami egy többnyelvű BERT-alapú modell és az emtsv (Simon és mtsai, 2020; Indig és mtsai, 2019a,b; Váradi és mtsai, 2018, 2017), amely kifejezetten a magyar nyelv igényeire lett fejlesztve, illetve amibe bele lett integrálva az emBERT (Nemeskey, 2020a) transformer-alapú NER rendszere. A feladatok tekintetében a HuSpaCy és a Stanza képes elemezni az összes eddig taglalt feladatot. Ezen túl az emtsv is szintén, viszont a függőségi elemzője nem Universal Dependencies formátumú. Az UDPipe-ból és a magyarlanc-ból a névelem-felismerő maradt ki.

3. Rendszer felépítése

3.1. Alapmodell felépítése

Rendszerünk alapja a HuSpaCy (Orosz és mtsai, 2022) cikkben bemutatott, többfeladatos tanulásra építő architektúra képezi, ami egy lépésben kezeli a különböző címkézési feladatokat. Az ebben szereplő 4 rétegű konvolúciós hálókra építő architektúrát lecseréltük egy előtanított transformer-alapú modellre.



1. ábra: Transfer learning ábrázolása.

Ahogy az 1. ábrán is látható, a rendszerünk 3 különálló modellből áll. Az első, Szeged Korpuszon (Csendes és mtsai, 2004) tanított modell végzi a szófaji egyértelműsítést, lemmatizálást és a morfológiai címkézést. Ezt a modellt lemásoltuk két példányban és továbbtanítottuk az egyiket függőségi elemzésre kiegészítve a mondatra bontás feladattal is, a másikat pedig névelem-felismerésre. Erre azért volt szükség, mert az egyes feladatok tanítására eltérő adatbázisok álltak rendelkezésünkre és így mégis képesek voltunk kihasználni az egyes részfeladatok közötti kapcsolatokat. Függőségi elemzésre a Szeged Korpusznál kisebb, viszont nemzetközileg elterjedt címkekészletet használó magyar Universal Dependencies (Nivre és mtsai, 2017) (UD) korpuszt használtuk. A névelem-felismerőt pedig a NerKor (Simon és Vadász, 2021) és a SzegedNER (Szarvas és mtsai, 2006a) összevont korpuszán tanítottuk.

Ez a módszer abban tér el a korábbi HuSpaCy architektúrától, hogy csak 2 modell készült. A Szeged Korpuszon tanult modell tovább volt tanítva az Universal Dependencies korpuszon, amely így már ki volt egészítve a függőségi elemző fejjel. Így nem volt egy külön modell a függőségi elemzésre. A változtatásra a transformer architektúra esetén azért volt szükség, mert a függőségi elemző tanítása közben azt tapasztaltuk, hogy a modell elfelejtette a korábban tanult feladatokat. A lemmatizálás, szófaji egyértelműsítés és morfológiai címkézés feladatok esetén a pontosság lezuhant az első néhány epochban, utána javult minimálisan, de például a morfológiai címkézés esetén 6, a lemmatizálás esetén pedig 27 százalékponttal romlott a pontosság a Szeged Korpuszon tanult modellhez képest.

3.2. Az alapmodell kiegészítései

A HuSpaCy rendszer által alapértelmezettként használt átmenet-alapú függőségi elemzőt lecseréltük a spaCy rendszer által biztosított gráf-alapú, biaffine figyelmi *(attention)* mechanizmussal rendelkező elemzőre (Dozat és Manning, 2016). Az eredeti cikkben alkalmazott Bi-LSTM rétegek helyett, ahogy a korábbi feladatokban, itt is transformer-alapú architektúrát használtunk. Az átmenet-alapú rendszerek a függőségi fát osztályozási problémák sorozataként építik fel, amikben lehetőség van két vizsgált szó között él behúzására vagy balról jobbra haladva a vizsgált szavak léptetésére. Ezek a megoldások a mohó, lokális döntéseik miatt rosszul tudnak teljesíteni hosszú mondatok esetén. Ezzel szemben a gráf-alapú elemzők a szavakból, mint csúcsokból felírt teljes gráfban keresik a maximális feszítőfát.

A névelem-felismerő a függőségi elemzőhöz hasonlóan egy átmenet-alapú megoldást alkalmaz, amit beam search használatával bővítettünk. Ennek lényege, hogy az elemző az egyes lépésekben több korábbi elemzést is figyelembe vesz. A lemmatizálás feladatban a Berkecz és mtsai (2023) cikkben használt megoldásokat alkalmaztuk.

A kifejezetten magyar adatokon tanított `huBERT` mellett kísérleteztünk más, többnyelvű modellekkel is, amelyek támogatják a magyar nyelvet is. Az egyik a `BERT-base-Multilingual` volt, illetve az `XLM-RoBERTa` két különböző méretű változata, az `XLM-RoBERTa-base` és az `XLM-RoBERTa-large`.

A `huBERT`, a `BERT-base-Multilingual` és az `XLM-RoBERTa-base` mindegyike 12 rétegből áll, amelyek egyenként 768 dimenziós vektorokat használnak. Az előbbi kettő összesen 110 millió paraméterrel rendelkezik, az `XLM-RoBERTa-base` pedig 125 millióval. Ezzel szemben az `XLM-RoBERTa-large` kétszer annyi réteget és 1024 dimenziós vektorokat használ és összesen 355 millió paraméterrel rendelkezik.

3.3. Többnyelvű modellek méretcsökkentése

A többnyelvű modellek egyik hátránya a nagyobb memóriaigényük, aminek az oka, hogy az eltérő nyelvek szavai miatt sokkal nagyobb szótárra van szükségük. Egynyelvű felhasználás esetén viszont a szótár egy jelentős részére szinte biztosan nincs szükségünk. Ilyenek például a kínai szimbólumok, török betűk, cirill betűk és az azokra épülő szavak, szótöredékek. Ennek kezelésére reguláris kifejezésekkel megszűrtük a két `XLM-RoBERTa` modell szótárát. A jelenlegi szűrés csak magyar nyelvben használt betűket és írásjeleket tartalmazó szótöredékeket hagyta meg.

4. Eredmények

Az összes kiértékelés (a névelem-felismerő kivételével) a Hungarian Universal Dependencies Corpus tesztkészletén (De Marneffe és mtsai, 2021) készült és a CoNLL 2018 Shared Taskon[1] alkalmazott kiértékelő szkript felhasználásával tör-

[1] `https://universaldependencies.org/conll18/conll18_ud_eval.py`

tént. A nem transformer-alapú modellek esetén összehasonlításképpen öt rendszert választottunk. Az első az `emtsv` (Simon és mtsai, 2020; Indig és mtsai, 2019a,b; Váradi és mtsai, 2018, 2017), második az `UDPipe` (Straka, 2018), a harmadik a `Stanza` (Qi és mtsai, 2020) és a negyedik a `UDify` (Kondratyuk és Straka, 2019). Ugyanakkor a 2021-ben publikált `HuSpaCy-large` (Orosz és mtsai, 2022) modelljével is összehasonlítottuk.

A `Stanza`[2] és `UDPipe`[3] szerzői már kimérték a rendszerüket az UD teszthalmazon, de ez nem mondható el az `emtsv`-ről. Itt a rendszert az alapértelmezett konfigurációs beállítások megtartásával lefuttattuk az általunk alkalmazott teszthalmazon. Ugyanakkor fontos kiemelni, hogy az `emtsv` eredményei olyan szempontból nem számítanak összevethetőnek, hogy nem lett újratanítva az általunk használt adatbázis vágásokkal, hanem csak egyszerű kiértékelés történt. Így a kiértékelés során előfordulhatott átfedés a tanító és a kiértékelő adathalmazok között. A `UDify` kiértékelése során gold adatot használt a tokenizálás és a mondatra bontás feladatokban.

A transformer-alapú modellek esetében felhasználtuk az eddigi tapasztalataink és további kísérleteket végeztünk néhány egyéb előtanított nyelvi modellel a HuSpaCy mostanra már alapértelmezett konfigurációjával. A következőkben részletesen bemutatjuk az eredményeket, valamint azt is, hogy mely nyelvi modellekkel sikerült elérni őket.

4.1. Tokenizálás és mondatra bontás

A mondatra bontás esetén látható, hogy minimálisan, de a transformer modellek meghaladják a nem transformer-alapú modelleket. Továbbá látható, hogy az `XLM-RoBERTa-large`-alapú modell szinte tökéletesen szegmentált, így megelőzve társait. A tokenizálás semmilyen technikában nem tért el a HuSpaCy cikkben ismertetett módszerektől.

	Token	Mondatra bontás
`Stanza`	**99,92%**	97,45%
`UDPipe`	99,80%	95,90%
`emtsv`	99,77%	98,67%
`UDify`	–	–
`HuSpaCy-large`	99,89%	98,55%
`huBERT`	99,89%	99,33%
`BERT-base-Multilingual`	99,89%	87,75%
`XLM-RoBERTa-base`	99,89%	99,33%
`XLM-RoBERTa-large`	99,89%	**99,67%**

1. táblázat. Mondatra bontás F1-score eredményei az UD teszthalmazán.

[2] `https://stanfordnlp.github.io/stanza/performance.html`
[3] `https://ufal.mff.cuni.cz/udpipe/1/models`

4.2. Morfo-szintaktikai elemzés

A 2. táblázat eredményei alapján elmondható, hogy a huBERT-alapú modell kiemelkedően teljesít minden területen. A morfológiai elemzésben sikerült több, mint fél százalékponttal javítani az eddigi legjobb eredményen. A szófaji címkézés esetében szintén sikerült javítani több, mint egy százalékponttal. Ezen felül a lemmatizálás feladatban is legalább 4 százalékponttal javított minden eddigi rendszerhez képest.

Ugyanakkor összesítve az XLM-RoBERTa-large-alapú modell ismét majdnem minden más modellt (lemmatizálásnál szinte fej fej mellett van a huBERT-alapú modellel) megelőz, morfológiai elemzés esetén is 3 százalékponttal felülmúlja a huBERT-alapú modellt.

	Lemma pontosság	PoS pontosság	Morf. pontosság
Stanza	94,19%	96,00%	93,62%
UDPipe	88,50%	90,60%	88,50%
emtsv	96,16%	89,19%	89,12%
UDify	90,19%	96,36%	86,16%
HuSpaCy-large	97,46%	96,89%	93,87%
huBERT	**98,68%**	97,27%	94,20%
BERT-base-Multilingual	98,48%	97,11%	91,76%
XLM-RoBERTa-base	98,55%	97,73%	96,61%
XLM-RoBERTa-large	98,67%	**98,01%**	**97,15%**

2. táblázat. Lemmatizálás, szófaji címkéző (PoS) és morfológiai elemző F1-score eredményei az UD teszthalmazán.

4.3. Gráf-alapú függőségi elemző

A 3. táblázatban látható az átmenet-alapú függőségi elemző, a gráf-alapú függőségi elemző, illetve a többi rendszer eredményei. Mivel az emtsv nem szolgáltat Universal Dependencies formátumú függőségi elemzést, így ez a modell nem mérhető ezzel a feladattal.

A meglévő rendszerek és a HuSpaCy-large modellje is alulmarad a huBERT átmenet-alapú modellel szemben, mind az UAS és a LAS tekintetében is. Az átmenet-alapú függőségi elemzővel rendelkező huBERT-alapú modell viszont alulmarad a már tárgyalt gráf-alapú megoldással szemben. Ezért ezt alkalmaztuk is a továbbiakban, illetve a többi transformer-alapú modellnél is ezt használtuk alapértelmezett elemzőként függőségi elemző esetén.

Viszont ebben a feladatban az XLM-RoBERTa-large pontatlanabb a huBERT gráf-alapú modellel szemben. Ez az egyetlen olyan komponens, ahol a huBERT-alapú modell sokkal jobban teljesít az XLM-RoBERTa-large-alapú modellnél.

	UAS	LAS
`Stanza`	84,19%	79,23%
`UDPipe`	72,80%	67,20%
`UDify`	89,69%	84,88%
`HuSpaCy-large`	82,53%	75,56%
`huBERT` - átmenet-alapú	89,95%	83,94%
`huBERT` - gráf-alapú	**91,57%**	**87,58%**
`BERT-base-Multilingual` - gráf-alapú	85,92%	81,93%
`XLM-RoBERTa-base` - gráf-alapú	89,47%	86,02%
`XLM-RoBERTa-large` - gráf-alapú	90,38%	86,88%

3. táblázat. Átmenet-alapú és gráf-alapú függőségi elemzés összehasonlítása a meglévő rendszerekkel, illetve a magyar nyelvre elérhető transformer-alapú nyelvi modellek kiértékelése.

4.4. Névelem-felismerés

Mivel az UD korpusz nem tartalmaz névelemeket, ezért erre a célra a NerKor (Simon és Vadász, 2021) és a SzegedNER (Szarvas és mtsai, 2006a) korpuszt használtuk. Mivel az `UDPipe`-ban nincs névelem-felismerő, így azt nem lehetett bevonni a vizsgálatba. Három korábbi névelem-felismerőt is bevettünk az összehasonlításba. Az egyik a Szarvas és mtsai (2006b), amely döntési fákat használ. A második a Simon (2013), amely egy lineáris modellt használ és rejtett Markov modelleket kombináló statisztikai címkézővel rendelkezik. A harmadik, újabb rendszer pedig az `emBERT`, amely egy transformer-alapú NER modell. A `Stanza` esetén viszont fontos kiemelni, hogy újra lett tanítva az általunk használt adatbázisokkal (Simon és mtsai, 2022).

	SzegedNER	NerKor	Kombinált
Simon (2013)	95,06%	–	–
Szarvas és mtsai (2006b)	94,77%	–	–
`emBERT`	97,40%	92,09%	**92,99%**
`Stanza`	91,78%	80,53%	83,75%
`HuSpaCy-large`	95,31%	80,75%	85,73%
`huBERT`	97,01%	88,27%	90,26%
`huBERT` + beam search	97,37%	89,13%	91,27%
`BERT-base-Multilingual` + beam search	–	–	89,07%
`XLM-RoBERTa-base` + beam search	–	–	90,94%
`XLM-RoBERTa-large` + beam search	–	–	91,86%

4. táblázat. F1-scoreon mért névelem-felismerés összehasonlítása a SzegedNER, a NerKor és a kombinált teszthalmazon is.

A NerKor adatbázis esetében a szerzők által meghatározott tanító-validáló-teszt halmazokra bontást alkalmaztuk. A SzegedNER esetében pedig, az összehasonlítás érdekében, a Szarvas és mtsai (2006b) cikk által meghatározott vágás lett felhasználva.

A meglévő rendszerek, az `emBERT` kivételével, alulmaradnak a `HuSpaCy-large` modellel és a `huBERT`-alapú modellel szemben. Viszont a `huBERT`-alapú + beam search modell jobb eredményt ért el, mint ahol nem volt alkalmazva a beam search, ahogy a 4. táblázatban is látható. A konklúzió tehát, hogy jobb a beam search, ezért a többi transformer-alapú modellnél is ezt használtuk alapértelmezett módszerként.

Így, mivel bevettük a vizsgálatba a régebbi rendszereket, ezért külön korpuszokra is le lettek mérve a modellek. Az `emBERT`-tel való összehasonlítás során a SzegedNER korpusz esetében szinte fej fej mellett van a két modell (`emBERT` és a beam search `huBERT`-alapú modell), de a NerKor esetében kicsit alulmarad a `huBERT`-alapú modell. A kombinált korpuszon is jobb az `emBERT`, mert a 12 réteg feletti Viterbi-algoritmus garantálja a jósolt címkeszekvencia konzisztenciáját. Ezen felül, ha megnézzük az összes transformer-alapú modellt és az `emBERT` eredményeit, kijelenthető, hogy az utóbbi továbbra is a leghatékonyabb.

4.5. `XLM-RoBERTa`-hu modell méretcsökkentése

Ahogyan a 3.3. fejezetben említettük, az `XLM-RoBERTa` modellek méretét, ha sikerülne csökkenteni megtartva a teljesítményét, akkor elérhető lenne egy rendkívül pontos modell jelentősen kevesebb memóriaigénnyel.

Az első fázisban reguláris kifejezések segítségével szűrtük ki azokat a tokeneket, amelyek nem fellelhetőek a magyar és az angol nyelvben. Így például a kínai szimbólumokat, török betűket, cirill betűket tartalmazó részszavakat és karaktereket kivettük. Ugyanakkor az Embedding layerből is eltávolítottuk a hozzátartozó vektorokat, így a paraméterszáma is csökkent a modellnek.

	Token	Mondatra bontás	PoS	Morf.	Lemma
`XLM-RoBERTa-base`	**99,89%**	99,33%	97,73%	96,61%	98,55%
`XLM-RoBERTa-base-hu`		98,56%	97,60%	96,27%	98,42%
`XLM-RoBERTa-large`	**99,89%**	**99,67%**	**98,01%**	**97,15%**	**98,67%**
`XLM-RoBERTa-large-hu`		97,24%	97,46%	95,84%	98,11%

5. táblázat. Módosított `XLM-RoBERTa` modellek F1-score eredményei az UD korpuszon. (Az aláhúzás az aktuális résztáblázatban a legjobb eredmény, a félkövérített pedig az adott egész táblázatban a legjobb eredmény.)

Ahogy a 5. és a 6. táblázatokban is észrevehető, a méretoptimalizált modellekkel hozzávetőlegesen sikerült tartani a teljesítményt. Jelenlegi kísérletek is hasznos eredményeknek számítanak, ugyanis az `XLM-RoBERTa-base-hu` esetén

	UAS	LAS	NER
XLM-RoBERTa-base	89,47%	86,02%	90,94%
XLM-RoBERTa-base-hu	89,21%	85,54%	89,89%
XLM-RoBERTa-large	**90,38%**	**86,88%**	**91,86%**
XLM-RoBERTa-large-hu	89,40%	85,46%	90,44%

6. táblázat. Módosított XLM-RoBERTa modellek UAS és LAS eredményei az UD korpuszon, a NER pedig a kombinált NER adatbázison lett mérve, amely F1-scoret mutatnak.

például sikerült a **felére csökkenteni** a méretet és az XLM-RoBERTa-large-hu modell pedig megközelítőleg **kétharmada az eredetinek**. Ezt a következő fejezetben jobban kifejtjük és összehasonlítjuk. Konklúzió, hogy a jelenlegi módszer segítségével képesek vagyunk csökkenteni a XLM-RoBERTa modell méretét minimális pontosságvesztéssel. Annak érdekében, hogy valóban hasznos legyen ez a modell és megtartsa a pontosságot, szükség lenne egy olyan heurisztikára, amely képes lenne ezt a feladatot ellátni. Jövőbeli tervünk, hogy az eldobandó szótöredékek halmazát statisztikai alapon válasszuk ki. Ehhez szeretnénk felhasználni a WebCorpus 2.0-át (Nemeskey, 2020b) és csak az abban előforduló leggyakoribb szótöredékeket megtartani, amivel a terveink szerint tovább csökkenthetnénk a szótár méretét és megtarthatnánk olyan speciális karaktereket is, amiket a kézi szabályokkal nem fedtünk le. A jelenlegi script elérhető publikus módon a GitHubon[4], ami lehetővé teszi egy új specifikus nyelvi modell készítését bármely nyelvre.

4.6. Memóriahasználat

A transformer architektúrák esetén fontosak az erőforrásigények, ezért ebben a fejezetben mélyebben megvizsgáljuk a futásidőket és a memóriaigényeket az összes eddigi modell szemszögéből. Az UDPipe, emtsv és az UDify rendszerek nem támogatják a névelem-felismerést, így ezeket a modelleket a névelem-felismerő nélkül mértük. A mérések az UD teszthalmazán készültek.

Először nézzünk meg a CPU-s oszlopot. A 7. táblázatban jól látható, hogy a transformer modellek a többi rendszerekhez képest több erőforrást igényelnek. Viszont kivételt képez például az emtsv gyorsasága a huBERT-alapú modellhez képest. A Stanza szintén lassabb, memóriaigények tekintetében pedig egy skálán mozognak. Az UDPipe ellenben mind a két esetben jól teljesített a futásidő és memóriaigények tekintetében, ugyanakkor ez a rendszer minden elemző esetén alulmarad hatékonyságban a többi rendszerhez képest. A többi transformer modell több paraméterrel rendelkezik, ezért és a többnyelvűség miatt is több erőforrást igényelnek.

Megfigyelhető az is, hogy a huBERT-alapú modell memóriahasználatban a HuSpaCy-large modelljéhez képest nem mutat jelentős eltérést. Ellenben az

[4] https://github.com/huspacy/huspacy-resources/blob/master/scripts/XLM-RoBERTa_size_reduction/XLM-RoBERTa_size_reduction.ipynb

	Gyorsaság (token/mp) CPU	Gyorsaság (token/mp) GPU	Memóriahasználat (GB)
Stanza	30	395	5,3
UDPipe	3175	–	1,3
emtsv	113	–	3,9
UDify	129	475	3,2
HuSpaCy-large	728	4685	3,2
huBERT	176	2605	4,8
BERT-base-Multilingual	138	2631	7,0
XLM-RoBERTa-base	151	2847	11,2
XLM-RoBERTa-base-hu	166	3265	6,2
XLM-RoBERTa-large	50	2353	18,9
XLM-RoBERTa-large-hu	59	2390	14,6

7. táblázat. A CPU-n (AMD EPYC 7F72) és GPU-n (NVIDIA A100 40 GB) összehasonlított rendszerek gyorsasága (token/másodpercben mérve) és a maximális memóriahasználatuk.

XLM-RoBERTa modellekre ez már nem teljesen igaz. Ahogyan szó volt a 3.1. fejezetben a modellek összetételéről, itt nyilvánul meg a három transformer modell mérete és az, hogy a többnyelvű modellek sokkal több memóriát igényelnek már önmagukban is.

A méretoptimalizált és az eredeti modellek között a futásidő azért nem változott drasztikusan, mert tulajdonképpen mind a két esetben ugyanazok a neuronok feleltek a futásidőért. Más szóval, a modell csak azon neuronokat használta fel a predikáláshoz, amelyekre a magyar nyelvhez szükség van. Az, hogy el lettek távolítva azok a részek, amelyeket a modell egyébként sem használt, nem befolyásolta a futásidőt.

5. Összegzés

Elkészítettünk a HuSpaCy-hez[5] egy huBERT-alapú modellt[6], ami minden hagyományos nyelvi elemzőt tartalmaz, vagyis tokenizálót, mondatra bontót, lemmatizálót, szófaji és morfológiai egyértelműsítőt, függőségi elemzőt, illetve ezeken felül névelem-felismerőt is és ezen eredmények teljes mértékben reprodukálhatóak.

Összességében a transformer-alapú modellek minden feladatban jobb eredményt érnek el a korábbi megoldásoknál. Ezek közül is a legtöbb feladatban a XLM-RoBERTa-large teljesített a legjobban, csupán a függőségi elemzés részfeladaton múlta felül a huBERT-alapú modell. Az alacsonyabb memóriaigényének és gyorsabb futásának köszönhetően a huBERT sok esetben jó választás lehet, de, ha a pontosságon van a hangsúly, a XLM-RoBERTa-large felhasználásával érhetjük el a legjobb eredményt. Viszont a state-of-the-art eredmények ellenére

[5] https://github.com/huspacy/huspacy/tree/develop
[6] https://huggingface.co/huspacy/hu_core_news_trf

is hátrányt jelenthet bizonyos végalkalmazásokban a nagyobb erőforrásigényük, például időérzékeny alkalmazásokban, illetve, ha korlátozott erőforrásokkal rendelkezünk. Ezekben a szituációkban megfelelő döntés lehet a `HuSpaCy-large` modell használata.

A függőségi elemzés és a névelem-felismerés problémákra több algoritmust is kipróbáltunk és a legmegfelelőbbet alkalmaztuk. Az eredmények alapján, a `huBERT`-alapú modell a függőségi elemző esetén hatalmas javulást mutat az elődeihez képest, így sikerült egy state-of-the-art eredményt elérni. Továbbá a morfológiai elemzés esetén is az `XLM-RoBERTa-large`-alapú modell képest volt 4 százalékponttal javulni az eddigi legjobb eredményhez képest.

Az elkészült modellek bárki számára szabadon igénybe vehetők a nyílt forráskódú HuSpaCy részeként.

Köszönetnyilvánítás

A kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében. Továbbá szeretnénk megköszönni Berend Gábor segítségét a nyelvi modellek méretcsökkentésével kapcsolatban.

Hivatkozások

Berkecz, P., Orosz, G., Szántó, Z., Szabó, G., Farkas, R.: Hybrid lemmatization in HuSpaCy. In: XIX. Magyar Számítógépes Nyelvészeti Konferencia (2023)

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)

Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: International Conference on Text, Speech and Dialogue. pp. 41–47. Springer (2004)

De Marneffe, M.C., Manning, C.D., Nivre, J., Zeman, D.: Universal dependencies. Computational linguistics 47(2), 255–308 (2021)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)

Honnibal, M.: Introducing spaCy (Feb 2015), `https://explosion.ai/blog/introducing-spacy`

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – The `emtsv` pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (aug 2019a), `https://www.aclweb.org/anthology/W19-4018`

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` — Egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019b)

Kondratyuk, D., Straka, M.: 75 languages, 1 model: Parsing universal dependencies universally. arXiv preprint arXiv:1904.02099 (2019)

Medress, M.F., Cooper, F.S., Forgie, J.W., Green, C., Klatt, D.H., O'Malley, M.H., Neuburg, E.P., Newell, A., Reddy, D., Ritea, B., és mtsai: Speech understanding systems: Report of a steering committee. Artificial Intelligence 9(3), 307–316 (1977)

Nemeskey, D.M.: Egy emBERT próbáló feladat (2020a)

Nemeskey, D.M.: Introducing huBERT. XVII. Magyar Számítógépes Nyelvészeti Konferencia pp. 3–14 (2022)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020b)

Nivre, J., Agić, Ž., Ahrenberg, L., Antonsen, L., Aranzabe, M.J., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., és mtsai: Universal Dependencies 2.1 (2017)

Orosz, G., 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. pp. 539–545 (2013)

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. arXiv preprint arXiv:2201.01956 (2022)

Pires, T., Schlinger, E., Garrette, D.: How multilingual is multilingual BERT? arXiv preprint arXiv:1906.01502 (2019)

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082 (2020)

Simon, E.: Approaches to Hungarian Named Entity Recognition. Ph.D.-értekezés, PhD School in Cognitive Sciences, Budapest University of Technology and Economics (2013)

Simon, E., Indig, B., Kalivoda, Á., Mittelholcz Iván, S.B., Vadász, N.: Újabb fejlemények az `e-magyar` háza táján. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–42. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2020)

Simon, E., Vadász, N.: Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Simon, E., Vadász, N., Lévai, D., Dávid, N., Orosz, G., Szántó, Z.: Az NYTK-NerKor több szempontú kiértékelése. XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)

Straka, M.: UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to

Universal Dependencies. pp. 197–207. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), https://www.aclweb.org/anthology/K18-2020

Szarvas, G., Farkas, R., Felfoldi, L., Kocsor, A., Csirik, J.: A highly accurate Named Entity corpus for Hungarian. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06) (2006a)

Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In: International Conference on Discovery Science. pp. 267–278. Springer (2006b)

Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholtz, I., Novák, A., Indig, B., Prószéky, G., Vincze, V.: Az e-magyar digitális nyelvfeldolgozó rendszer (2017)

Váradi, T., Simon, E., Sass, B., Mittelholcz, I., Novák, A., Indig, B.: E-magyar–A Digital Language Processing System (2018)

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of Recent Advances in Natural Language Provessing 2013. pp. 763–771. Association for Computational Linguistics, Hissar, Bulgaria (2013)

Hybrid lemmatization in HuSpaCy

Péter Berkecz, György Orosz, Zsolt Szántó,
Gergő Szabó, Richárd Farkas

Institute of Informatics, University of Szeged
2. Árpád tér, Szeged, Hungary
{berkecz,szantozs,gszabo,rfarkas}@inf.u-szeged.hu
gyorgy@orosz.link

Abstract. Lemmatization is still not a trivial task for morphologically rich languages. Previous studies showed that hybrid architectures usually work better for these languages and can yield great results. This paper presents a hybrid lemmatizer utilizing both a neural model, dictionaries and hand-crafted rules. We introduce a hybrid architecture along with empirical results on a widely used Hungarian dataset. The presented methods are published as three HuSpaCy models.

1 Introduction

Lemmatization is the process where the lemma - the dictionary form of a word - needs to be computed from an inflected word form. This can happen in various ways. One approach is that we are deriving the possible lemmata of a token using a morphological analyzer which relies on hand-crafted morphological rules then a statistical model is applied to disambiguate using the context. On the other hand, end-to-end statistical lemmatizers are using a training corpus exclusively to learn the transformation rules and the context disambiguation in a single model. Hence, they can track the language evolution and fit domain variations if a training dataset is available.

Several end-to-end statistical lemmatizers have been proposed (Müller et al., 2015; Straka, 2018) however, based on our experiments on the Hungarian subset of Universal Dependencies (Nivre et al., 2017), they were not accurate enough on their own. Studies (Orosz and Novák, 2013; Boudchiche and Mazroui, 2019) showed, that for morphologically rich languages hybrid architectures tend to perform better.

In this paper, we propose the following three-step hybrid architecture:

1. a dictionary-based first step for unambiguous cases, where we just replace the token with the lemma if the traits are the same,
2. end-to-end statistical lemmatizer for the highest accuracy in ambiguous cases
3. hand-made rules for special cases where the statistical approach did not see enough examples.

For end-to-end lemmatization, we investigated two algorithms that learn rules from training data, the Lemmy (Kristiansen, 2019) and the Edit-tree Lemmatizer (de Kok, 2021). Both of them use neural networks to find the best set of rules for a given word in a given context. We examined various word embedding and transformer-based networks to find the best-performing end-to-end lemmatizer for Hungarian.

Our tool is built on spaCy's foundation and is both versatile and easy to use in a multi-language environment. It is conveniently released under a Creative Commons BY-SA 4.0 license, making it an attractive option for organizations looking to incorporate natural language processing capabilities into their systems without any licensing issues.

Our hybrid lemmatizer is available in the open-source HuSpaCy toolkit, in the `medium`, `large`, and `transformer` models since version 3.4.

2 Related Work

2.1 Lemmatizing in spaCy

spaCy (Honnibal, 2015) is an industrial-strength natural language processing pipeline, which already supports lemmatization for nineteen languages. HuSpaCy (Orosz et al., 2022) is a Python package supporting Hungarian models for spaCy. Three different models are available for the tool, with two different embedding types. The `medium` and `large` model uses static word vectors, while the `transformer` model relies on `huBERT` (Nemeskey, 2020a), a fine-tuned BERT (Devlin et al., 2018) model for Hungarian. Up until now, HuSpaCy utilized a simple rule-learning algorithm it used a lemmatizer called Lemmy.

There are three different built-in lemmatizers available in spaCy. The first is the ruled-based one. It uses hand-crafted rules and part-of-speech tags to make a lemma from a token. Four official spaCy models use this, English, French, Spanish, and Macedonian.

The second type is a dictionary-based lemmatizer. It stores the tokens with their PoS tags, and it loads a lemma for a token during prediction. Only one official spaCy model uses this, the Catalan one.

The third and newest one is the Edit-tree Lemmatizer, which is an end-to-end neural lemmatizer. Fourteen official spaCy models are using this, for example, Greek, Polish, Danish, German, and Finnish.

2.2 Hungarian Lemmatization Approaches

In Hungarian, there are several approaches to lemmatizing texts, for example, `magyarlanc` (Zsibrita et al., 2013) and `emtsv` (Simon et al., 2020; Indig et al., 2019a,b). Both of them use symbolic methods to make lemma candidates and disambiguate between them based on the context of the word in another step.

The morphological parser in `magyarlanc` generates lemma candidates based on KR (Trón et al., 2006) grammatical rules and then uses the PurePOS part-of-speech tagger (Orosz and Novák, 2013) to select the correct one according to the sentence context.

`emtsv` is the result of a collective effort to integrate multiple existing NLP tasks into one application. For lemmatization, it uses the emMorph (Novák et al., 2016; Novák, 2014) morphological analyzer to create a lemma, part-of-speech tag, and morphological description candidates, and the emTag (PurePOS) tagger to disambiguate between them. During prediction, it uses Hungarian rules (Novák, 2015) for HFST (Lindén et al., 2009), then from the pool of lemma candidates, it selects the most probable one using the PurePOS tagger. We must note that emMorph has a very restrictive license, thus it cannot be used freely in a commercial application.

The lemmatizer of Stanza (Qi et al., 2020) combines a dictionary-based lemmatizer with a neural Seq2Seq lemmatizer. The dictionary is being used as a cache to predict lemmata quickly because the Seq2Seq lemmatizer is typically slower, this is also compiled from the training dataset.

UDPipe (Straka, 2018) handles lemmatization as a sub-task of tagging. The system generates a rule from tokens and their lemmata based on the longest common subsequence of two words and then selects them as a classification task.

3 Methods

Various studies (Orosz et al., 2022; Jongejan and Dalianis, 2009) have already shown how the different types of lemmatizers perform individually. In some cases of agglutinating languages, however, the best solution was usually achieved with a combination of these (Orosz, 2015). In the following section, we present the methods we have selected for our architecture.

3.1 Lemmy

Lemmy is an end-to-end statistical lemmatizer. It is an open-source Python implementation of the CST algorithm (Jongejan and Dalianis, 2009). This method automatically learns rules from the training data by detecting prefixes and suffixes to be transformed, so that the resulting token can be used to generate the lemma. First, it tries to lemmatize based on the existing rules, if this fails, it creates a new rule using their longest common subsequence and their differences. It creates a rule based on the token and its lemma's longest common subsequence, and stores this and the token's PoS-tag with the rule occurrence count. It had a few modifications over the original project to improve its accuracy. During prediction, it uses this frequency information and the predicted PoS-tag to select a rule, then applies it to the token to compute the lemma.

3.2 Edit-tree Lemmatizer

The Edit-tree Lemmatizer is another end-to-end statistical lemmatizer that exploits neural language models. It is a spaCy implementation of the Lemming lemmatizer (Müller et al., 2015), which learns rules (called edit-trees) from the training data to transform a word into its lemma.

For example, this system builds an edit-tree for the word *leghosszabb* (longest) this way:

1. Search for the longest common subsequence (LCS) between the token and its lemma: *leghosszabb* - *hosszú* (long)
2. Based on this, it splits into 3 parts, prefix - LCS - suffix
3. Finds the necessary transformations for the prefix and suffix to transform the inflected form into the lemma
   (a) In the prefix, we change *leg-* into an empty string (ϵ)
   (b) In the suffix, we change *-abb* into *-ú*

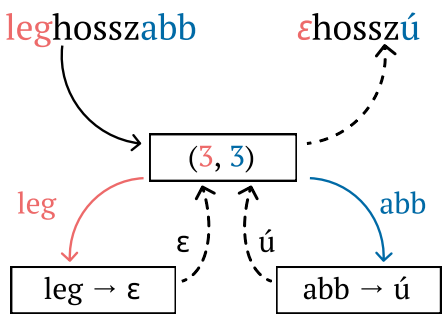


Fig. 1: Edit-tree for the word *leghosszabb*

When predicting, a neural network is used as a classifier to select the right edit-tree. In spaCy we can use multi-task learning and this can be inserted as a task, thus being able to access hidden representations in the underlying neural network. This way, for example, the tagger and the lemmatizer can work together in an even closer relationship, not just on the few tags that the tagger predicts.

3.3 Hungarian Language Models Used

The architecture of HuSpaCy (Orosz et al., 2022) consisted of a Word2Vec (Mikolov et al., 2013) static word embedding and CNN layers. The disadvantage of Word2Vec is that it only knows the words it saw during training and does not use the context information of the tokens.

To overcome these drawbacks, we experimented with word embeddings containing subword information and context-aware transformer networks.

Static word embeddings containing subword information (e.g. `fastText` (Bojanowski et al., 2017)) learn not just a single word during training, but several small parts of a word. This way, because a token is a sum of its character *n-gram* vectors, it performs better for words outside the vocabulary (OOV), be it a truly unknown word or a typo. It has the advantage of having a small storage footprint due to the smaller vocabulary size and because of that, it is quite fast when evaluated on the CPU. In our experiments, we have utilized `floret` (Boyd, 2021) which is the spaCy fork of the `fastText`. We trained two models on the Hungarian WebCorpus 2.0 (Nemeskey, 2020b), a 100-dimensional and a 300-dimensional one. The advantage of them is that they include character n-grams, and due to the hashing trick of `floret`, their storage size is much smaller than the original word embeddings.

On the other hand, transformer models (e.g. `BERT`) are able to take the full sentence context of the word into consideration when predicting, and thus can give a more accurate representation and achieve a high level of accuracy in HuSpaCy (Szabó et al., 2023). Their disadvantages are their large storage size and relatively slow CPU evaluation, but with GPU it can be fast.

3.4 Lookup Lemmatizer

Dictionary-based lemmatizers only perform a simple replacement operation. They "learn" to map tokens to lemmata from the training dataset by memorizing a word form with their morphosyntactic information associated with lemmata. If more than one lemma is associated with the same word with the same morphosyntactic labels, the most frequent one is learned. In prediction time, they simply return the matching lemmata if any. The disadvantage is that it can only lemmatize tokens that it has already seen.

3.5 Postprocessing Rules

There may be cases where the lemmatizer makes mistakes and these cases can be easily corrected by rules. These special hand-crafted rules are most often used when something has to be cut off the end of a token to get the correct lemma. You could actually call it the little brother of stemming. Beyond suffix cutting, these rules are also useful in certain cases for agglutinating languages.

We noticed that in a few rare cases where there are exclamation and question marks in a token they were left in a lemma, thus we've built a rule to remove them. Also, there were some numbers and dates that needed to be fixed. For example, for the token *4-6-os* the lemmatizers usually made *4-6-* as a lemma, but *4-6* would be the correct one. Because the predicted lemmata had unrecognizable patterns, this rule used the original token to remove any suffixes after a number or a date.

4 Results

4.1 Experimental Setup

To ensure comparability, we have used the most commonly available tools for training and testing. The evaluations were carried out on the test set of the Hungarian Universal Dependencies (UD) (Nivre et al., 2017) corpus using the CoNLL 2018 Shared Task (Zeman et al., 2018) evaluator script[1].

For training, we used parts of the Szeged Korpusz (Csendes et al., 2004) that are not included in any part of the UD corpus. As for the comparison, we chose the most popular open-source tools and used them as they were published for reproducibility.

4.2 Experiments

Lemmy was used in HuSpaCy, but it only relied on the part-of-speech information and did not have any other information about the token. The Edit-tree Lemmatizer was released in spaCy 3.3. Studies (de Kok, 2021) showed that it brings significant improvements for various languages, hence we investigated its usage for Hungarian (cf. Table 1).

Lemmy	Edit-tree Lemmatizer
95.53%	95.90%

Table 1. Results of the previous Lemmy lemmatizer and the Edit-tree Lemmatizer with the default configurations. We consider this as the baseline.

As we have shown in subsection 3.3, the underlying neural language model can be simply replaced by another spaCy-compatible module. The Word2Vec word embeddings previously shipped with HuSpacy models were replaced by a `floret` (Boyd, 2021) architecture. This not only improves our results but also reduces the storage requirements of the whole model.

We experimented with pretrained transformer language models as well. Based on our experiments (cf. Table 2), `huBERT` (Nemeskey, 2020a) gave a good accuracy-to-speed ratio, but with an `XLM-Roberta-Large` (Conneau et al., 2019) model we were able to achieve even higher accuracy at the expense of throughput.

`floret`(100d)	`floret`(300d)	`huBERT`	`XLM-Roberta`
96.56%	96.76%	98.53%	98.89%

Table 2. Results of the Edit-tree Lemmatizer with different language models.

[1] https://universaldependencies.org/conll18/conll18_ud_eval.py

The Edit-tree Lemmatizer has an important hyperparameter, which controls the number of edit-trees to try before giving up and choosing another one of the token's properties as a lemma. It defaults to 1, but increasing this value improved the accuracy. By adjusting this, the model can access more edit-tree candidates during prediction because the possible edit-tree is chosen from all of them. Increasing it to 3 improves the result (cf. Table 3) even though the pipeline is slower, but it's a worthy compromise. A higher `top_k` value makes the algorithm a bit slower (cf. Table 4) and has almost no effect on the results. Because of this, we've choosen 3 as our `top_k` value for the next experiments.

	`floret`(100d)	`floret`(300d)	`huBERT`	`XLM-Roberta`
`top_k` = 1	96.56%	96.76%	98.53%	98.89%
`top_k` = 3	96.87%	97.01%	98.63%	98.93%
`top_k` = 8	96.84%	97.09%	98.61%	98.83%

Table 3. Lemma accuracies of the Edit-tree Lemmatizer with increased `top_k` parameters.

	`floret`(100d)	`floret`(300d)	`huBERT`	`XLM-Roberta`
`top_k` = 1	7739	7255	3349	2429
`top_k` = 3	6740	6873	3166	2358
`top_k` = 8	6655	6571	3146	2347

Table 4. Throughput (token/s) of the Edit-tree Lemmatizer with increased `top_k` parameters.

In Lemmy, it was already noticeable that the casing of some words was wrong, this problem was also apparent here. So the same rule had to be built into the system (cf. Table 5). If the word is a proper noun or the first token of a sentence, it can start with a capital letter, in all other cases, the word is lowercase. We built this in so that the algorithm can be aware of these casings during training so that it can apply them later during prediction.

	`floret`(100d)	`floret`(300d)	`huBERT`	`XLM-Roberta`
+ Casing	96.29%	97.23%	98.63%	98.85%

Table 5. Results of the Edit-tree Lemmatizer taking casing into account.

Hybrid Lemmatizer Results Since the Edit-tree Lemmatizer doesn't give any prediction sometimes, we tested how a dictionary-based lemmatizer performs as a supplement. This became the second part of our architecture, but it became a pipeline step before the Edit-tree Lemmatizer in the whole spaCy pipeline.

We had two experiments with this setup, the first was about pairing the training data to the lemmata based on the token and PoS tag (cf. Table 6). However, this led to many incorrect predictions. Finally, to solve this, we paired the training data to the lemmata to be trained based on the token, the PoS-tag, and the morphological tags. During training and prediction, the number-like tokens are masked, resulting in fewer pairs, but this will match more tokens containing numbers. It can generalize better with this because for example the *1000-ben* will be *0000-ben* and it could be applied to *3000-ben* which may not have been included in the training set.

In a couple of cases, even combined – the dictionary-based lemmatizer with the Edit-tree Lemmatizer – they still made some errors, and we used post-processing rules to try to correct the lemma in the ways that we mention in subsection 3.5. Upon implementation of these rules, the accuracies demonstrated a slight decrease, however, the overall quality of the predicted lemmata showed noticeable improvement.

	`floret`(100d)	`floret`(300d)	`huBERT`	`XLM-Roberta`
+ Lookup	97.19%	97.48%	98.68%	98.94%
+ Rules	97.17%	97.46%	98.68%	98.94%

Table 6. Results of the hybrid architecture. The rows are an addition to the existing system. It can be observed that the transformer-based models were not really helped by the hybrid additions, but the floret-based models were. This is due to the fact that the larger models are more adept at selecting the edit-trees.

4.3 Final Results

In order to compare our architecture's results with already existing systems, we chose three popular systems that have built-in lemmatizers. `emtsv` is a Hungarian pipeline of state-of-the-art NLP components, UDPipe is commonly used as a baseline in CoNLL competitions, and Stanza has relatively high scores on UD test sets. It is important to point out that all of them were used off-the-shelf, i.e. not re-trained or fine-tuned.

The evaluation was performed on the Hungarian Universal Dependencies test set. The UDPipe and Stanza authors have made the results of their systems available, but the `emtsv` authors have not, so we used one of the default configurations to get UD-compatible output and then evaluate it. It is important to point out that comparing `emtsv`'s lemmatizer is probably not fair out-of-the-box, because the tool was trained on a different train-test split, which may conflict with ours. (There is a high probability that their training set contains sentences in our test set.)

In Table 7 one can see that all systems are quite accurate except UDPipe. Our hybrid lemmatizer consistently outperforms other systems, demonstrating a marked improvement in performance.

	Accuracy
Word2Vec *baseline*	95.90%
HuSpaCy + `floret` (100d)	97.17%
HuSpaCy + `floret` (300d)	97.46%
HuSpaCy + `huBERT`	98.68%
HuSpaCy + `XLM-Roberta`	98.94%
`emtsv`	96.16%
UDPipe	88.50%
Stanza	94.19%

Table 7. Lemmatization accuracies of the different NLP pipelines.

4.4 Resource Usage

It is also important to mention the resource usage of our models. These models contain a complete language pipeline, i.e. a single model of a system. In the case of HuSpaCy, these models include tagger, morphologizer, lemmatizer, parser, and named entity recognition.

	Storage Size	Throughput (tokens/sec) CPU	Throughput (tokens/sec) GPU	Memory Usage (GiB)
Word2Vec *baseline*	1.6 GB	854	7059	4.61
HuSpaCy + `floret` (100d)	125 MB	1862	5903	2.35
HuSpaCy + `floret` (300d)	455 MB	1186	6074	3.33
HuSpaCy + `huBERT`	1.8 GB	242	3055	4.84
HuSpaCy + `XLM-Roberta`	9.1 GB	78	2186	17.87
`emtsv`	3.6 GB	116	-	3.91
UDPipe	5 MB	3175	-	1.38
Stanza	246 MB	30	395	5.31

Table 8. Resource usages of the different models. The storage sizes contain a full natural language processing pipeline, not just the lemmatizer. During measurement, we've tried to run every benchmark three times, and take the best result with each tool. For each tool, we've tried to use only the parts that are needed to the lemmatizer, in the case of our models this means that we've run them without named entity recognition. We measured the throughput using an AMD EPYC 7F72 CPU and NVIDIA A100 (40GB) GPU.

Table 8 shows that the results do not really correlate with the storage requirements of a model. This is mostly due to the language model, as can be seen in the case of the Word2Vec model, which is 1.6GB but is outperformed by a 125MB model in accuracy. However, the two models use completely differ-

ent word embeddings, while Word2Vec uses a continuous bag-of-words model, `floret` uses CBOW with subword information but also hashes words so it is a bit slower.

For both `XLM-Roberta`-based and `huBERT`-based, we can see that the 5x storage requirement is almost unnecessary, with the result being only a few tenths of a percentage point better.

We must note that at `emtsv` it is the size of a Docker image which is a recommended way to use the tool, thus it may contain parts that are not part of the original `emtsv` pipeline.

5 Conclusions

We presented a hybrid architecture, which consists of three parts utilizing the advantages of neural language model-based statistical lemmatizers and dictionaries along with hand-crafted rules. We have shown the application of hand-crafted rules and morphological information-based dictionaries can improve end-to-end statistical lemmatizers. We examined different language models and we found the `XLM-Roberta-large` model achieves the best results, but the five times smaller `huBERT` reached comparable results too. We performed multiple experiments to extensively compare our tool with other systems. As the results show, our lemmatizer outperforms other models and the storage requirement is smaller with the smaller `floret` model than some of them.

Altogether, this paper has presented a new hybrid lemmatizer architecture that is freely available in three models of HuSpaCy. We made it open-source, under a permissive license.

Acknowledgements

The authors would like to thank Dávid Nemeskey and Dániel Lévai for their help in benchmarking `emBERT` and Stanza. HuSpaCy research and development is supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory.

References

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)

Boudchiche, M., Mazroui, A.: A hybrid approach for arabic lemmatization. International Journal of Speech Technology 22(3), 563–573 (2019)

Boyd, A.: floret (2021), https://github.com/explosion/floret

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale (2019), https://arxiv.org/abs/1911.02116

Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004. pp. 19–23 (2004)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018), https://arxiv.org/abs/1810.04805

Honnibal, M.: Introducing spaCy (Feb 2015), https://explosion.ai/blog/introducing-spacy

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – the emtsv pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (aug 2019a), https://www.aclweb.org/anthology/W19-4018

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: emtsv — egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019b)

Jongejan, B., Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre- , in- and suffixes alike. In: Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. pp. 145–153 (2009)

de Kok, D.: Neural edit-tree lemmatization for spaCy (Nov 2021), https://explosion.ai/blog/edit-tree-lemmatizer

Kristiansen, S.L.: lemmy: Lemmy a lemmatizer for Danish and Swedish (Apr 2019), https://github.com/sorenlind/lemmy

Lindén, K., Silfverberg, M., Pirinen, T.: Hfst tools for morphology – an efficient open-source package for construction of morphological analyzers. vol. 41, pp. 28–47 (08 2009)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)

Müller, T., Cotterell, R., Fraser, A., Schütze, H.: Joint lemmatization and morphological tagging with lemming. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2268–2274. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), https://aclanthology.org/D15-1272

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020a)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020b)

Nivre, J., Agić, Ž., Ahrenberg, L., Antonsen, L., Aranzabe, M.J., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., et al.: Universal dependencies 2.1 (2017)

Novák, A.: A new form of humor? mapping constraint-based computational morphologies to a finite-state representation (2014)

Novák, A.: A model of computational morphology and its application to Uralic languages. Ph.D. thesis, Pázmány Péter Katolikus Egyetem (2015)

Novák, A., Siklósi, B., Oravecz, C.: A new integrated open-source morphological analyzer for hungarian. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 1315–1322 (2016)

Orosz, G.: Hybrid algorithms for preprocessing agglutinative languages and less-resourced domains effectively. Ph.D. thesis, Pázmány Péter Katolikus Egyetem (2015)

Orosz, G., Novák, A.: PurePos 2.0: a hybrid tool for morphological disambiguation. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013). p. 539–545. INCOMA Ltd. Shoumen, Hissar, Bulgaria (2013)

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit (2022)

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020)

Simon, E., Indig, B., Kalivoda, Á., Mittelholcz Iván, S.B., Vadász, N.: Újabb fejlemények az `e-magyar` háza táján. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 29–42. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2020)

Straka, M.: UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 197–207. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), https://www.aclweb.org/anthology/K18-2020

Szabó, G., Orosz, G., Szántó, Z., Berkecz, P., Farkas, R.: Transformer-alapú HuSpaCy előelemző láncok. In: XIX. Magyar Számítógépes Nyelvészeti Konferencia (2023)

Trón, V., Halácsy, P., Rebrus, P., Rung, A., Simon, E., Vajda, P.: The annotation system of hunmorph. Tech. rep., Technical Report, The Media Research Center, Budapest University of ... (2006)

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., Petrov, S.: CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 1–21. Association for Computational Linguistics, Brussels, Belgium (Oct 2018), https://aclanthology.org/K18-2001

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of Recent Advances in Natural Language Processing 2013. pp. 763–771. Association for Computational Linguistics, Hissar, Bulgaria (2013)

Neural Morphological Generators for Hungarian

László János Laki, Noémi Ligeti-Nagy, Noémi Vadász, Zijian Győző Yang

Hungarian Research Centre for Linguistics
H-1068 Budapest, Benczúr u. 33.
{surname.forename}@nytud.hu

Abstract. Here we present a set of morphological generators for Hungarian that generate surface forms from emMorph and Universal Dependencies (UD) morphological tags with high accuracy. We experimented with two approaches: first, neural machine translation models were trained based on the morphological analysis as the source format and the corresponding surface form as the target format. Second, we tackled the problem as a text generation task, where the morphological analysis is followed by the correct word form. The corpus we used is a normalised version of Webcorpus 2.0 (Nemeskey, 2020). Marian MT proved to produce the best results, thus we evaluated its output manually on NerKor (Simon and Vadász, 2021). Our analysis shows that the generator achieves a high accuracy of 96.27% in the case of emMorph and 94.94% in the case of UD. After manual evaluation, we counted a more concise accuracy, which is 99.43% (emMorph) and 98.69% (UD). This model may be used for several NLP tasks, such as anonymisation and terminology translation.
Keywords: morphological generator, neural generator, emMorph, Universal Dependencies morphology

1 Introduction

A morphological generator is a program that performs the task of generating a word form based on the morphological analysis. Morphological generation may be considered an opposite task of morphological analysis. Here, given the description of a word in terms of number, category, stem, and so on, the original word is retrieved. For example, if root = *go*, part of speech = *verb*, tense= *present*, and if it occurs along with a third person and singular subject, then a morphological generator would generate its surface form, *goes*.

Despite being a core task of paramount importance for many NLP tasks, the number and the research of morphological generators is far below that of morphological analysers. Morphological generators, if built, are usually built together with morphological analysers. Morphological generation is a crucial task for languages with rich morphology in tasks like anonymisation and pseudonymisation, where the removed personal identifier must be replaced with another one, but in the correct surface form.

After a brief overview of the related works, we will present our experiments and then show the results of the evaluation and error analysis.

Our finetuned mT5 models are available on our Hugging Face page. The name of the emmorph tagset based model is *NYTK/morphological-generator-emmorph-mt5-hungarian*[1], while the name of the model for the UD tagset is *NYTK/morphological-generator-ud-mt5-hungarian*[2].

2 Related works

Minnen et al. (2000) presents a fast and robust morphological generator for English based on finite-state techniques. They illustrate the relevance of their generator on the automatic simplification of English newspaper texts.

As morphological generation is more crucial for languages with rich morphology, we found generators for agglutinative languages such as Tamil (Kengatharaiyer et al., 2021), Tatar (in Apertium, `https://github.com/apertium/apertium-tat`), and for many uralic languages (Prószéky and Novák, 2005; Novák, 2008a,b; Bakró-Nagy et al., 2010; Endrédy et al., 2010; Fejes and Novák, 2010)[3].

There are some morphological analyzers for Hungarian that can be used for generation as well. HuMor also has a Hungarian morphological generator mode (Novák, 2015) which was the basis for the MetaMorpho machine translation system (Novák et al., 2008) as well. HunSpell can be used for this task in two ways[4]: it generates word forms by typing the lemma and the features, or typing the lemma and an example word (e.g. *kutya* + *macskákkal* = *kutyákkal*). Hunmorph (Trón et al., 2005) and Morphdb.hu (Trón et al., 2006) are also suitable for morphological generation. Hunmorph-foma[5] uses the morphological tagset of HunMorph and it is based on the foma generator (Hulden, 2009). Our goal was to generate word forms using the tagsets of emMorph (Novák et al., 2016; Novák et al., 2017) and Universal Dependencies, therefore the generators referred here are not suitable for our needs, because they use other tagsets.

In contrast to the solutions described above, neural networks can also be used to solve the task of generation. E.g. Malouf (2016) uses a Long Short-Term Memory (LSTM) network to learn the paradigms of a morphologically complex language and the model generates the paradigms of Russian, Finnish, Irish, Maltese, and Khaling. Micher (2019) built a generator from the output of an existing analyzer for Inuktitut with a sequence-to-sequence neural network which transforms the underlying morphemes into the surface forms. Schwartz et al. (2019) created a generator for the case-inflected nouns in the polysynthetic language of Yupic. They utilized an existing FST morphological analyzer to

[1] https://huggingface.co/NYTK/morphological-generator-emmorph-mt5-hungarian

[2] https://huggingface.co/NYTK/morphological-generator-ud-mt5-hungarian

[3] The morphological analyser and word form generators for the languages mentioned in these papers can be found and used at `http://www.morphologic.hu/urali/index.php?lang=english`.

[4] See the methods of pyhunspell, the Python bindings for HunSpell here: `https://github.com/pyhunspell/pyhunspell/wiki/Documentation`

[5] `https://github.com/r0ller/hunmorph-foma`

create training data and treated morphological generation as a recurrent neural sequence-to-sequence task.

3 Corpora

For training neural models for the task of morphological generation, as a first step, a huge amount of morphologically analysed text was needed. Since Webcorpus 2.0 (Nemeskey, 2020) is a morphologically analysed corpus of 9 billion tokens, it seemed the right choice for this task. In the morphological generation process, using the lemma and its' morphological tags of a word, the model generates the correct surface form. Thus, from the analysed Webcorpus 2.0 the following columns are needed: FORM (original surface form), LEMMA (lemma), XPOS (emMorph tags). From these extracted columns, we created our emMorph corpus. In this corpus, one segment is one word and its forms: *munka* `[/N]` `[Acc]` *munkát*.

Our first experiment created two versions of this corpus: **unnormalised** and **normalised**. In the case of the unnormalised corpus, we simply applied the `uniq` function, so every $lemma + tag = form$ segment was represented only once. Using this method, 75,569,032 segments (and type at the same time) were created. But after the training and evaluation processes, many errors were recognized. For instance, there were many words without accents in this corpus. After `uniq` process, for example, the accusative form of *kutya* 'dog' can be *kutyat* and *kutyát* with the same probability. In this case, *kutyat* is the wrong form. Our models with the input `kutya [/N]` `[Acc]` generated the *kutyat* word form, which is incorrect. Thus, we needed to normalise our corpus. Our normalization steps are the following:

- The frequency of the $lemma + tag = form$ segments were calculated based on the Webcorpus 2.0
- A hard rule was used which filters out the less frequent segments with the same $lemma + tag$. This constrain had more than 95% precision to select wrong segments and there were only few cases where it removes correct word forms.
- The segments that has a final subtag `[Punct]` were filtered out. We decided to add this step as the generator shouldn't know which punctuation mark should be generated at the end of a word.
- The rare segments (less then 10 times presented in the corpora) were removed.
- The word frequency information would be integrated to the model so all segments were duplicated based on the logarithm of their word frequency.

After the normalization process, our corpus contained 12,371,157 segments and 6,830,804 types.

In our next step, we created our UD corpus. The Webcorpus 2.0 does not contain UD tags. Thus, we needed an emMorph→UD converter. For this task, we used emmorph2ud2 (Indig et al., 2019) converter[6].

[6] `https://github.com/vadno/emmorph2ud2`

4 Methods and Experiments

For the training of the morphology generator, we applied two approaches: machine translation and text generation.

Machine translation: Our first approach is solving the task as a translation task. Technically they are a sequence-to-sequence architectures, which contains not only a generator, but an encoder part as well. The source segment contains the lemma and the morphological tags, the target segment is the morphologically analysed surface word. An emMorph and a UD example are shown in Table 1.

Tagset	Source format	Target format
emMorph	munka [/N][Acc]	munkát
UD	munka NOUN Case=Acc|Number=Sing	munkát

Table 1. A sample of the format of the data

To train machine translation models, three different kinds of method were tried:

- **Marian NMT**: Marian NMT is a machine translation framework, which has written in C++ language. The biggest advantages of this seq2seq implementation are the optimized training and prediction time, as well as the low resource and hardware requirement. We trained models from scratch with the following hyperparameters: epoch: 10, dim-vocabs 800, learn-rate 0.00005, max-length 50, dim-emb 512. The system was trained on a single NVIDIA A100 (20GB) GPU. The training took 12 hours.
- **mT5**: We fine-tuned the pre-trained google/mt5-base model to the translation task. The hyper-parameters are the following: source prefix: 'morph:'; max source length: 64; max target length: 32; batch size: 128/GPU (8 GPU); source language: en; target language: hu; epoch: 10. The training took 14 hours.
- **M2M100**: Since this task was considered a translation task, we fine-tuned the facebook/m2m100_1.2B model that was pre-trained for multilingual machine translation tasks. The hyper-parameters are the following: max source length: 64; max target length: 32; batch size: 128/GPU (8 GPU); source language: en; target language: hu; epoch: 10. The training took 15 hours.

Text generation: Our second approach is solving the task as a text generation task. It means that one segment contains all the information: the lemma, the morphological tags and the morphological analysed surface word. To help the generative model to solve the task more precisely, separator and end-of-text tags were added to the segment. An emMorph and an UD example are shown in example 1.

(1) EM: munka [/N][Acc] </s> munkát <|endoftext|>
UD: munka NOUN Case=Acc|Number=Sing </s> munkát <|endoftext|>

To train the text generation model, a GPT-2 model was fine-tuned:

– **GPT-2**: We have fine-tuned the PULI GPT-2 model to this task. The hyperparameters are the following: block size: 64; batch size: 128/GPU (4 GPU); epoch: 10. In the fine-tuning script, we have modified the preprocess function. In the original function, the input texts were concatenated. We removed this concatenation method. As a consequence, one segment contains only one word and its forms. In our experiments we tried two kinds of separator tags: '</s>' and '='. During the measurements, using the '=' separator tag, we could gain higher results. Thus, in the results section, we presented the performance of models that used the '=' separator. The training took 8 hours.

The first experiment was to train the model on the 'raw' version of our corpora. It means the models were trained on the unnormalised corpus. The Marian on this corpus achieved 94.5% word-based accuracy, but on the NerKor corpus the Marian model could gain only 76.3% accuracy. This experiment showed that the Webcorpus contains many erroneous word forms. Thus, our next step was normalizing the corpus, then the models were retrained on the normalised corpus.

All of our models were trained on NVIDIA A100 (80GB) GPUs.

5 Results and Evaluation

To evaluate our models, we used the word-base accuracy metrics. Table 2 shows the results of our models on the morphologically analysed subpart of NerKor. In general, all models could achieve more than 93% accuracy, which means all of our models could learn this task. Among these models, Marian could achieve the highest performance. This indicates that in this task the pre-training process could not add any extra knowledge. The surprising result is that the GPT-2 gained the lowest values. In our hypothesis, this task fits the text generation task.

	emMorph (%)	UD (%)
Marian real accuracy	**99.43**	**98.69**
Marian	**96.27**	**94.94**
mT5	95.53	94.66
M2M100	95.04	93.83
GPT2	93.78	93.43

Table 2. Results

Since Marian achieves the best results, our evaluation processes were applied to the output of the Marian models.

5.1 The evaluation of the emMorph-based generation

For the evaluation, the morphologically analyzed subpart of NerKor was used. The test cases are the tokens generated based on the lemmas and the disambiguated emMorph tags from the corpus. These generated tokens were compared to their counterparts found in the corpus. The generated token was identical to the token in the corpus in 53,942 cases. The remaining 2,006 tokens were checked manually to explore the reasons for the difference between the generated tokens and the ones in the corpus.

- *erroneous reference*: It turned out that in some cases (actually, in 1,317 cases) the generator was not at fault. On the one hand, despite NerKor being a gold standard corpus in which the morphological tags were checked and corrected by two annotators and were curated by a third, some annotation errors may naturally occur in the corpus.[7] On the other hand, the annotation scheme of NerKor caused some different tokens compared to the generated ones, especially in the cases of named entities. Due to the annotation scheme of NerKor the morphological tags of the named entities do not reflect the internal structure of the tokens of named entities. Only the last token of multiword named entities got the full morphological tag, all other tokens got the noun tag without case suffix (`[/N]`).
- *allomorphy*: In 382 cases the token in the corpus and the generated one were also correct (e.g. *estig, estéig*, *panelban, panelben*, *tietek, tiétek*).
- *actual errors*: only 306 tokens turned out to be a real mistake made by the generator.

5.2 The evaluation of the UD-based generation

Of the 55,282 tokens in the test set of NerKor, 52,484 tokens were correctly generated by the model. The 2,798 instances, where there was a difference between the original word form and the model's output, were manually checked. Three cases are differentiated:

- *erroneous reference*: 1,728 instances are extracted from our evaluation as the reference morphological features (*használjuk*, lemma: `használ`, features: `Definite=Ind|Mood=Imp|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act`)[8] or the lemma (*írt* 'wrote', lemma: `írt`, features: `Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act`) are erroneous, or the reference word form contains punctuation marks as the result of false tokenisation (*használatát.*), or the output of the generator is correct, but the reference word form is not normalised (*károsító*).

[7] A useful by-product of the error analysis is the list, based on which the incorrect tags of NerKor can be corrected.

[8] The false tag in the feature set is highlighted with red.

- *allomorphy*: In 349 cases both the reference word form and the generated word form are correct. This is mainly caused by the allomorphs in Hungarian, such as *azzal* = *avval*. As the two essive suffixes, *essivus-modalis* and *essivus-formalis* are uniformly tagged as `CASE=ESS`, the word forms ending in *-ként* and *-ul/-ül* are not differentiated. Parallelly, the two types of plural suffices (plural *kutyá**k*** 'dogs' and familiar plural *Nagy**ék*** 'the Nagys') are not differentiated either. These are the other main cause of this type of error in the test set.
- *actual errors*: In 720 cases the reference was correct, and the model generated a false word form. This category includes tokens like emojis, foreign segments and various punctuation marks.

The biggest challenge for the generator seems to be the case of numerals. In the test set, many different surface forms bear the exact same lemma and morphological features, see for example Table 3. Since Universal Morphology (McCarthy et al., 2020) allows a more subtle distinction between different types of numerals,[9] it would be useful, not only for the morphological generator but also for other models fine-tuned on this corpus, to refine these.

reference	**lemma**	**POS**	**features**	**generated**
sok	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
legöbb	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
legtöbb	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
legtöbben	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
sokan	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
sokszor	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
tobb	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
több	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
többen	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
többször	sok	NUM	Case=Nom|Number=Sing|NumType=Card	legtöbbször
többre	sok	NUM	Case=Sub|Number=Sing|NumType=Card	legtöbbre
legtöbbön	sok	NUM	Case=Sup|Number=Sing|NumType=Card	többön

Table 3. A sample of numerals in the test set of NerKor. *Reference* column shows the word forms in the corpus; *lemma*, *POS* and *features* are the analysis of the word form in the corpus; *generated* column shows the word forms generated by the model based on the *lemma*, *POS* and *features*.

In Table 4, you can see the error types of the Marian models. In both cases more than 60% of errors are reference errors and more than 10% of errors are allomorphy, which are actually correct. Thus, according to the error ratios, our Marian emMorph model could achieve **99,43%** and Marian UD model could gain **98,69%** real accuracy (see Table 2).

[9] `https://universaldependencies.org/u/pos/NUM.html`

Error Type	**Ratio (%)**	**Sample** lemma (`tag`) – reference – generated
		emMorph
Reference	65.70	tápoldat (`[/N][Acc]`) – tépoldatot – tápoldatot
Allomorphy	19.04	panel (`[/N][Ine]`) – panelben – panelban
Model error	**15.28**	te (`[/N|Pro][All][2Sg]`) – hozzád – tédhez
		UD
Reference	61.78	vég (`Case=Nom`)[10] – véget – vége
Allomorphy	12.47	agresszív (`Case=Ess`) – agresszívan – agresszíven
Model error	**25.73**	végtelen (`Pos=NUM`) – végtelen – végtelen-

Table 4. Error types of the Marian models

6 Summary

Here we presented a morphological generator for Hungarian that generates surface forms from emMorph and UD morphological tags with high accuracy. We tackled the problem with two approaches: first, neural machine translation models were trained based on the morphological analysis as the source format and the corresponding surface form as the target format. Second, we experienced with text generation, where the morphological analysis is followed by the correct word form. The corpus we used is a normalised version of Webcorpus 2.0. From the above-mentioned methods, Marian MT proved to produce the best results: 96.27% in the case of emMorph and 94.94% in the case of UD. We evaluated its output manually on NerKor. Our detailed analysis shows that the generator achieves a high accuracy of 99.43% (emMorph) and 98.69% (UD). This model may be used for several NLP tasks, such as anonymisation and terminology translation. Our model called *HuMorGen* is freely available on GitHub.[11]

Bibliography

Bakró-Nagy, M., Endrédy, I., Fejes, L., Novák, A., Oszkó, B., Prószéky, G., Szeverényi, S., Várnai, Z., Wagner-Nagy, B.: Online morfológiai elemzők és szóalak-generátorok kisebb uráli nyelvekhez. In: Attila, T., Veronika, V. (eds.) VII. Magyar Számítógépes Nyelvészeti Konferencia: MSZNY 2010. pp. 345–349 (2010)

Endrédy, I., Fejes, L., Novák, A., Oszkó, B., Prószéky, G., Szeverényi, S., Várnai, Zs., Wágner-Nagy, B.: Nganasan - Computational Resources of a Language on the Verge of Extinction. In: Sarasola, K., Tyers, F.M., Forcada, M.L. (eds.) Creation and Use of Basic Lexical Resources for Less-Resourced Languages: 7th SaLTMiL Workshop (LREC-2010). pp. 41–44 (2010)

[11] Link in the final version of the paper.

Fejes, L., Novák, A.: Obi-ugor morfológiai elemzők és korpuszok. In: Attila, T., Veronika, V. (eds.) VII. Magyar Számítógépes Nyelvészeti Konferencia: MSZNY 2010. pp. 284–291 (2010)

Hulden, M.: Foma: a Finite-State Compiler and Library. In: Proceedings of the Demonstrations Session at EACL 2009. pp. 29–32. Association for Computational Linguistics, Athens, Greece (Apr 2009), `https://aclanthology.org/E09-2008`

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., Makrai, M.: One format to rule them all – The `emtsv` pipeline for Hungarian. In: Proceedings of the 13th Linguistic Annotation Workshop. pp. 155–165. Association for Computational Linguistics, Florence, Italy (2019)

Kengatharaiyer, S., Dias, G., Butt, M.: ThamizhiMorph: A morphological parser for the Tamil language. Machine Translation 35, 1–34 (04 2021)

Malouf, R.: Generating morphological paradigms with a recurrent neural network. San Diego Linguistics Papers 6, 122–129 (09 2016)

McCarthy, A.D., Kirov, C., Grella, M., Nidhi, A., Xia, P., Gorman, K., Vylomova, E., Mielke, S.J., Nicolai, G., Silfverberg, M., Arkhangelskiy, T., Krizhanovsky, N., Krizhanovsky, A., Klyachko, E., Sorokin, A., Mansfield, J., Ernštreits, V., Pinter, Y., Jacobs, C.L., Cotterell, R., Hulden, M., Yarowsky, D.: UniMorph 3.0: Universal Morphology. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 3922–3931. European Language Resources Association, Marseille, France (May 2020), `https://aclanthology.org/2020.lrec-1.483`

Micher, J.: Bootstrapping a Neural Morphological Generator from Morphological Analyzer Output for Inuktitut (2019)

Minnen, G., Carroll, J., Pearce, D.: Robust, applied morphological generation. In: Proceedings of the First International Conference on Natural Language Generation - Volume 14. p. 201–208. INLG '00, Association for Computational Linguistics, USA (2000), `https://doi.org/10.3115/1118253.1118281`

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)

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'16). pp. 1315–1322. European Language Resources Association (ELRA), Portorož, Slovenia (may 2016), `https://aclanthology.org/L16-1209`

Novák, A.: Creating a Morphological Analyzer and Generator for the Komi language. In: Carson-Berndsen, J. (ed.) Proceedings of the SALTMIL Workshop at LREC 2004. pp. 64–67 (2008a)

Novák, A.: Language resources for Uralic minority languages. In: Williams, B., Forcada, M.L., Sarasola, K. (eds.) Proceedings of the SALTMIL Workshop at LREC 2008: Collaboration: interoperability between people in the creation of language resources for less-resourced languages. pp. 27–32 (2008b)

Novák, A.: A Model of Computational Morphology and its Application to Uralic Languages. Ph.D. thesis (2015)

Novák, A., Rebrus, P., Ludányi, Zs.: Az emMorph morfológiai elemző annotációs formalizmusa [Annotation format of the emMorph morphological analyzer]. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 70–78. Szegedi Tudományegyetem, Szeged (2017)

Novák, A., Tihanyi, L., Prószéky, G.: The MetaMorpho Translation System. In: Proceedings of the Third Workshop on Statistical Machine Translation. pp. 111–114. Association for Computational Linguistics, Columbus, Ohio (2008)

Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Antti, A., Carlson, L., Linden, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., Yli-Jyrä, A. (eds.) Inquiries into Word, Constraints and Contexts. (Festschrift in the Honour of Kimmo Koskenniemi on his 60th Birthday), pp. 150–157 (2005)

Schwartz, L., Chen, E., Hunt, B., Schreiner, S.L.: Bootstrapping a neural morphological analyzer for St. Lawrence Island Yupik from a finite-state transducer. In: Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers). pp. 87–96. Association for Computational Linguistics, Honolulu (Feb 2019), `https://aclanthology.org/W19-6012`

Simon, E., Vadász, N.: Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: Ekstein, K., Pártl, F., Konopík, M. (eds.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Trón, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: open source word analysis. In: Jansche, M. (ed.) Proceedings of the ACL 2005 Software Workshop, pp. 77–85. ACL, Ann Arbor (2005)

Trón, V., Halácsy, P., Rebrus, P., Rung, A., Vajda, P., Simon, E.: Morphdb.hu: Hungarian lexical database and morphological grammar. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06). European Language Resources Association (ELRA), Genoa, Italy (May 2006), `http://www.lrec-conf.org/proceedings/lrec2006/pdf/683_pdf.pdf`

Gondolatok a gondola-tokról

Morfológiai annotációt javító módszerek tesztelése gold standard korpuszon

K. Molnár Emese[1,2], Dömötör Andrea[1,3]

[1]Digitális Örökség Nemzeti Laboratórium
1088 Budapest Múzeum krt. 6-8.
[2]ELTE BTK Nyelvtudományi Doktori Iskola
1088 Budapest Múzeum krt. 6-8.
[3]PPKE BTK Nyelvtudományi Doktori Iskola
1088 Budapest, Mikszáth Kálmán tér 1.
mesii@student.elte.hu, domotor.andrea@btk.elte.hu

Kivonat Kutatásunk célja az volt, hogy csökkentsük az emberi annotációs munka mennyiségét a gold standard korpusz projektünkben. A cikkben (egy baseline mellett) három egyszerűen kivitelezhető módszert vizsgáltunk meg, amely alkalmas lehet az annotálási hibák megtalálására. A cél elsősorban a magas fedés: olyan módszert keresünk, amely úgy szűkíti le a második annotátor által áttekintendő tokenek körét, hogy a lehető legtöbb hibát lefedje. A három módszer eredményeit összegezve azt az eredményt kaptuk, hogy a tokenek 31,65%-át újraannotálva a szövegben előforduló annotálási hibák arányát 2%-ra csökkenthetjük, melynek közel fele csak a részletes (HuMor) elemzést érinti. Ez a munkaterhelésben lényeges csökkentést jelent, miközben magas minőséget is biztosít.
Kulcsszavak: annotáció, morfológia, korpusz, gold standard

1. Bevezetés

Jelenleg magyar nyelvre kevés megbízható minőségű elemzett korpusz áll rendelkezésre, ezért egy új, függőségi elemzést is tartalmazó korpusz (treebank) hiánypótló munka lenne az egész magyar NLP-közösség és a korpuszfelhasználó nyelvészek számára is. A kézzel készült, ellenőrzött annotáció előállítása azonban rendkívül idő- és erőforrásigényes. Kutatásunk ezért azt vizsgálja, hogyan lehetséges minőségi annotációt előállítani kevesebb humán erőforrás felhasználásával.

A klasszikus gold standard módszertan szerint minden szöveget 2 annotátornak kell annotálni, majd egy harmadiknak felülvizsgálni az eltéréseket. Tekintve azonban, hogy a mai gépi elemző eszközök eleve elég magas pontossággal működnek, úgy gondoljuk, hogy nincs szükség ilyen mennyiségű emberi annotációra.

A cikkben (egy baseline mellett) három egyszerűen kivitelezhető módszert vizsgáltunk meg, amely alkalmas lehet az annotálási hibák megtalálására. A cél elsősorban a magas fedés: olyan módszert keresünk, amely úgy szűkíti le a

második annotátor által áttekintendő tokenek körét, hogy a lehető legtöbb hibát lefedje.

Mivel gold standardről beszélünk, az első körös emberi annotációt nem akarjuk megspórolni. Abból indulunk ki tehát, hogy egy annotátor végigannotálja a szöveget, és az ő munkájában keresünk hibajelölteket.

2. Motiváció

Kutatásunk motivációját egy gold standard korpuszépítési projekt adta. Ennek célja egy olyan többszintű nyelvi annotációt tartalmazó általános referenciakorpusz létrehozása, amely adatvezérelt kutatások adatbázisaként, valamint természetes nyelvet feldolgozó gépi tanulásos algoritmusok tanítóanyagaként egyaránt felhasználható. Terveink szerint a korpusz morfológiai, szintaktikai (függőségi), tagmondat- és NP-annotációt is tartalmazni fog. A munka első (jelenlegi) fázisában a morfológiai annotálást végezzük.

A folyamatosan bővülő korpusz jelenleg 350 ezer tokennyi szöveget tartalmaz egy annotátor által (morfológiailag) annotálva. Ennek nagy része blogokból, valamint tudományos-ismeretterjesztő és kulturális weboldalakról származik, illetve megtalálható még a korpuszban néhány 20. századi irodalmi szöveg is. A későbbiekben további műfajok, például sajtó-, hivatalos és szakszövegek felvételét is tervezzük.

Az annotáció munkafolyamata Vadász (2020) és Simon és Vadász (2021) módszertanát követi. Az annotáláshoz az emMorph (Váradi és mtsai, 2018) kódkészletét használtuk. A szövegeket emtsv-vel (Indig és mtsai, 2019) előelemeztük, az annotátorok az emMorph modul kimenetét javították. Korpuszunk morfológiai annotációja tartalmazza a szótőt, a szófajcímkét és a részletes morfológiai elemzést (Novák, 2014) is, az annotátoroknak ezek mindegyikét javítaniuk kellett. A kézi annotáláshoz egy saját fejlesztésű Java alapú felületet használtunk. Ezen az annotátornak lehetősége van a tokenizálás és a morfológiai elemzés javítására is. A tokenizálás javítása során az annotátor egy tokent összevonhat az előtte és az utána lévő tokennel, szétszedhet egy tokent tetszőleges helyen, illetve ha szükséges, át is írhatja a tokent.

A program felületén előhívható egy szó összes, az e-magyar által megadott lehetséges elemzése, amelyek egymás alatt, listaszerűen sorolódnak fel ("Elemzés választása" menüpont). A lehetséges elemzések megjelenítései tartalmazzák a lemmát, valamint a szónak az e-magyar által megadott részletes és egyszerű morfológiai elemzését is. Az e-magyar egyértelműsítő modulja által helyesnek tartott elemzés zöld pipával jelenik meg. Amennyiben nem értünk egyet az e-magyar által helyesként megadott elemzéssel, akkor a listában egy másik lehetséges elemzésre kattintva megváltoztathatjuk azt. Ha az e-magyar által megadott lehetséges elemzések közül egyiket sem tartjuk helyesnek, akkor megadható egy teljesen új elemzés is egy erre szolgáló üres mező kitöltésével. Az annotáló felület bármikor képes az e-magyartól új elemzést lekérni egy-egy tokenre, ha a tokenizálás javítása során a token vagy a mondat megváltozik. A programnak ez a

funkciója jelentősen leegyszerűsíti a munkafolyamatot, hiszen így egy lépésben lehet elvégezni a tokenizálást és a morfológiai elemzést.

Az annotálás gyorsításának érdekében a Szeged Treebank alapján létrehoztunk egy olyan listát, amely az egyértelmű szavakat tartalmazza, tehát azokat, amelyekhez minden esetben csak egyféle elemzés tartozhat. A listát nyelvészek ellenőrizték. A listán szereplő szavakat az előfeldolgozás után az annotáló felület már ellenőrzött tokenként jeleníti meg, így az annotátornak nincs vele munkája[1]. A lista jelenleg 31881 szót tartalmaz, ami általában lefedi az annotálandó szövegek egyharmadát.

A korpusz eredetileg XML formátumú, de a hibakereső kísérletekhez készítettünk egy egyszerűsített TSV formátumot is. Terveink szerint a későbbiekben (a függőségi elemzés bevezetése után) a korpuszt CoNLL-U+ formátumban is közzétesszük.

3. Vizsgált módszerek

A vizsgálandó módszerek kiválasztásakor Ide–Pustejovsky nyelvi annotációs kézikönyvének vonatkozó fejezete (Dickinson és Tufis, 2017) volt a kiindulópontunk. Ez a fejezet számos, az annotáció minőségének javítását célzó módszert foglal össze. Külön alfejezetekben tárgyalja azokat a módszereket, amelyek kész korpuszokra alkalmazhatók jól, és azokat, amelyek folyamatban lévő annotálás esetén is használhatók az inkonzisztens annotációk (és esetleg a séma hibáinak) szűrésére.

Kutatásunkban egy intuitív baseline mellett három, a kézikönyv által ihletett módszer hatékonyságát teszteltük. Mindegyik módszer lényege az, hogy hibajelöltek keresésével csökkentsük a második annotátor által átnézendő tokenek számát úgy, hogy a hibák minél nagyobb részét lefedjük.

3.1. Baseline: többértelmű és egyedi szavak mint hibajelöltek

Baseline módszerünk abból a gondolatból indul ki, hogy azok a szavak, amelyek többször, következetesen ugyanazzal az annotációval szerepelnek a korpuszban, nagy valószínűséggel nem hibásak. Ennek megfelelően hibajelöltnek tekintünk minden olyan tokent, amely a már elkészült korpuszban többféle elemzéssel is előfordul. Ezen kívül hozzá kell adni a hibajelöltekhez minden hapaxot is, hiszen ezeknél nem tudjuk mérni az annotáció következetességét. Így várhatóan sok hibajelöltünk lesz ugyan, de bízhatunk a magas fedésben. A módszer előnye továbbá, hogy mindhárom elemzési szintre (egyszerű elemzés, részletes elemzés, szótövesítés) használható.

3.2. Gépi és emberi annotáció összevetése

A kézikönyv több olyan kísérletet is említ, ahol a szerzők egy automatikus elemző eszköz kimenetével vetették össze az emberi annotációt arra alapozva, hogy a

[1] Kivéve ha a szó egy cím vagy tulajdonnév része, l. 4.2.

gépi eszköz alapvetően konzisztens viselkedésre van betanítva, így az általa adott elemzésektől való eltérés alkalmas lehet az inkonzisztenciák detektálására.

Az egyik ehhez kapcsolódó kísérletünkben az annotálási munkafolyamatban kiindulásnak használt emtsv eredeti elemzéseit vetettük össze az annotátorok által javított változattal abban a reményben, hogy a hibák nagy része azokból az esetekből fog származni, ahol az annotátor módosított a gépi előelemzésen.

Mivel az emtsv nem képes a részletes elemzések egyértelműsítésére (erről bővebben l. 4.3. pont), így ez a módszer csak az egyszerű elemzésekre és a szótövekre alkalmazható.

3.3. Annotáció összevetése másik elemző kimenetével

Az előző módszer továbbfejlesztéseként a kézi annotációkat összehasonlítottuk egy másik, az annotációs munkafolyamattól független elemző kimenetével is. Ehhez az emtsv-be integrált UDPipe és Stanza modulok, illetve a HuSpaCy (Orosz és mtsai, 2022) jött szóba. Ezek mindegyike az UDv2 kódkészletét használja, ezért az összehasonlíthatóság kedvéért a kézi annotációkat az emtsv emmorph2ud2 moduljával konvertáltuk az UDv2 címkekészletre.

Az összehasonlítás során azok a tokenek lettek hibajelöltek, ahol a kézi elemzés és az elemző kimenete eltérő volt. Külön gyűjtöttük a szótő, és külön a morfológiai annotációk eltéréseit. Utóbbi esetben mind a szófajcímke, mind a morfológiai jegyek eltéréseit figyelembe vettük, hiszen ezek együttesen feleltethetők meg az emMorph egyszerű elemzéseinek.

A UDPipe és a Stanza esetén váratlanul nagyszámú eltérést tapasztaltunk. Ennek az volt az oka, hogy az elemzők kimenetében használt kódkészlet nem volt teljesen kompatibilis az emmorph2ud2 által konvertált címkékkel. (Az elemzők jellemzően többféle morfológiai jegyet jelenítettek meg.) Emiatt az eredményeknél csak a HuSpaCy-vel kapott eredményeket ismertetjük.

3.4. Érvénytelen bigram módszer

Květoň és Oliva (2002) módszerének lényege az, hogy érvénytelen szófajcímke-bigramokat keres a javítandó korpuszban. Ehhez egy kisebb, validált korpuszra van szükség, amelyből az érvényes bigramok kinyerhetők. Esetünkben a 4.1. pontban leírt tesztkorpusz felelt meg erre a célra.

A tesztkorpusz eredeti annotált és javított (gold standard) változatát is 10 részre osztottuk, és minden iterációban a gold standard 9 részéből készítettük el az érvényes bigramok listáját (csak az egyszerű elemzést figyelembe véve), és a fennmaradó egy rész eredeti annotált változatából kigyűjtöttük a hibajelölteket. A hibajelöltek ebben az esetben azok a tokenek voltak, amelyek egy érvénytelen (azaz az érvényes bigramok listáján nem szereplő) bigram részei voltak.

4. Eredmények

4.1. Tesztkorpusz

Tesztkorpusznak 6 szöveget választottunk ki a korpuszunk eddig elkészült részéből. Ezek összesen 14147 tokent tartalmaznak. A válogatásnál szempont volt, hogy minél több annotátortól kerüljön be szöveg a tesztkorpuszba, illetve a kiválaszott szövegek között legyenek régebben és újabban annotáltak is. Ez utóbbi szempontra azért volt szükség, mert a több mint egy éve tartó annotálási munkafolyamat során előfordultak változtatások az annotálási útmutatóban (ezeket a következő pontban fejtjük ki részletesebben).

A kiválasztott szövegeket a klasszikus gold standard módszer szerint még egy annotátor annotálta az első annotátortól függetlenül. A két annotáció eltéréseit részletesen megvizsgáltuk, és közös megegyezéssel döntöttünk a végleges annotációkról. Az így létrejött gold standardet tekintjük referenciának a hibakereső módszerek kiértékelésénél. A megtalálandó hibák listáját az első annotátorok annotációi alapján állítottuk össze.

4.2. Hibatípusok a tesztkorpuszban

A hibákat egyaránt vizsgáltuk a lemmák, az összetett és az egyszerűsített elemzések esetében is. A különböző elemzési lehetőségek hibái részben átfedik ugyan egymást, azonban a részletes kiértékelés és az egyes hibatípusok pontos azonosítása indokolttá tette, hogy minden csoportot külön is vizsgáljunk. Összesen 1261 hibát azonosítottunk az első annotátorok és a gold standard változatok összevetésével. Ebből 196 esetben a lemma volt hibás, 762 esetben az összetett elemzésben, míg 303 esetben az egyszerűsített elemzésben volt hiba.

Az egyes elemzések hibáit kategorizáltuk és altípusokba osztottuk. Az elemzési szintek hibatípusainak összefoglalását példákkal és gyakoriságértékekkel az 1–3. táblázatok tartalmazzák.

A lemmáknál a hibás szótövesítést (`lem`), a kis- illetve nagybetűk tévesztését (`lett`), az elütéseket (`typ`), a tokenizálási hibákat (`tok`), a szövegben előforduló karakterhibákat (`cerr`), valamint az annotálás közben történt sémaváltozások nyomán kialakuló hibákat (`schem`) különböztettük meg. A hibás szótövesítés esetén az annotáror hibásan határozta meg a szó szerkezetét, ezért nem megfelelő szótő került az elemzésbe. A kis- és nagybetűk tévesztése főként a tulajdonnevek esetén fordult elő, ugyanis az emMorph bizonyos esetekben következetesen kisbetűvel jelölt egyes tulajdonneveket, amelyeket az annotátorok nem minden esetben javították. Elütésként határoztuk meg, ha egy-egy betű volt téves a szótövön belül. A tokenizálási hiba esetén a szövegben előforduló gépelési vagy helyesírási hibákat az annotátor nem javította, így ez hibát eredményezett a szótőben. A szövegben megjelentek olyan karakterhibák is, amelyek az írásjeleket és a számjegyeket érintették, ezeket az annotátorok nem minden esetben javították. A sémaváltozás pedig azokat az esetek jelöli, ahol az annotációs séma (útmutató) időközben megváltozott, ezért az első annotátor változata eltér

a gold standardtől. Ez főként a tulajdonnevek és az elváló igekötős igék annotálásában jelent változást. A tulajdonnevek esetében a nyelvi egység státusz (Tolcsvai Nagy, 2008) jelölése érdekében az annotálás során úgy határoztunk, hogy csak az utolsó tagon jelöljük az esetragot, a megelőző tagokat eset nélküli főnévnek vagy melléknévnek címkézzük, tehát például `Pázmány[/N] Péter[/N] Katolikus[/Adj] Egyetemen[/N][Supe]` formában. Ezen belül a címeket speciális főnévi csoportoknak tekintjük, ezért a címek utolsó szavát minden esetben főnévi (`[/N]`) címkével jelöljük a megfelelő esetraggal, a cím többi szava `[None]` (korábban `[/X]`), azaz nem elemzett címkét kap, így például `Jojo[None] + nyuszi[/N] + ban[Ine]` formában elemezzük. Ha az igekötő elválik az igéjétől, az igét, Pethő és mtsai (2022) javaslata alapján igekötős igeként elemezzük. Az igekötő megjelenik az ige lemmájában és zéró morféma formájában a részletes elemzésben is: `kinéz | ki[/Prev]=[] + néz[/V] + [Prs.NDef.3Sg]`.

LEMMA			
Hibakód	**Helytelen**	**Helyes**	**Gyakoriság**
lem	*szó*	*szóval*	70
lett	*nap*	*Nap*	15
typ	*Millenniumi*	*Milleniumi*	5
tok	*nem-megújuló*	*nem*	3
cerr	*15_000*	*15000*	7
schem	*Csillagok*	*[None]*	96
		Összesen:	**196**

1. táblázat. A szótövek hibatípusainak összefoglalása

Az összetett elemzés esetében a helytelen címkekiosztásból származó hibás elemzéseket (`err`), az egyes kifejezések túlelemzéséből (`oana`) vagy alulelemzésből (`uana`) származó hibákat, a szövegben hibásan szereplő tokenek nyomán létrejövő szóhibákat (`word`), hiányzó elemzéseket (`miss`), az annotáció szintaxisában megjelenő hibákat (`syn`), egyéb egyedi hibákat (`x`), az emMorph sémától eltérő esetekben ejtett hibákat (`emM`) és a sémaváltozás folytán kialakuló hibákat (`schem`) azonosítottunk. A hibás elemzésnél az annotátor rosszul határozta meg a szófajt vagy szószerkezet valamelyik elemét, ezért helytelen az összetett elemzés. Azokban az esetekben, amelyekben a szófaj alapvetően megfelelően lett meghatározva, azonban például az összetételek vagy a képzett szavak esetén tovább vagy nem elég szóelemre bontott az annotáror, túlelemzésről vagy alulelemzésről beszélhetünk. Bizonyos szavaknál az is előfordult, hogy az elemző nem kínált fel összetett elemzést, az annotátorok pedig nem pótolták ezeket. Továbbá a megfelelően meghatározott morfológiai elemzések ellenére hibát okozott az is, ha az annotátor az annotáció szintaxisában rontott. Főként az írásjeleket – kötőjeleket és gondolatjeleket – érintő hibákat egyéb egyedi hibáknak tekintettük. Az annotáció során a megváltozott séma következtében felmerülő eltéréseket sémaváltozásként szintén külön kategóriaként kezeltük. Az annotálás kezdetétől két esetben egyértelműsítettük az emMorph sémáját: egyrészt a létigék töveinek meghatározásában, másrészt a számok annotálásában. A létigék esetén minden

esetben szigorúan morfológiai alapon határozzuk meg a szótövet, tehát a *le-* kezdetű létigéket mindig a *lesz* szótőre vezetjük vissza, sohasem a *van*-ra. A többjegyű számok esetén az emMorph bizonyos esetekben egy számjegyként kezeli a számokat, más esetekben viszont külön karakterekre bontja őket. Az egységes és következetes sémakialakítás miatt mi az előbbi megoldást választottuk, tehát a többjegyű számjegyeket egyben elemeztük, nem bontottuk karakterekre és egy címkét kaptak.

RÉSZLETES ELEMZÉS			
Hibakód	**Helytelen**	**Helyes**	**Gyakoriság**
err	mód[/N] + on[Supe]	módon[/Adv]	217
oana	hossz[/N] + ú[_Adjz:Ú/Adj] + [Nom]	hosszú[/Adj] + [Nom]	167
uana	ökoszisztéma[/N] + [Nom]	öko[/CmpdPfx] + szisztéma[/N] + [Nom]	117
word	Malayam[/Adj|nat] + ból[Ela]	Malayalam[/Adj|nat] + ból[Ela]	13
miss		android[/N] + [Nom]	33
syn	Tejút=Tejut + at[Acc]	Tejút[/N]=Tejut + at[Acc]	14
x	IndiaPass[/N] + -[Hyph:Hyph] + t[Acc]	IndiaPass[/N]=Indiapass- + t[Acc]	12
emM	van[/V]=le + gyen[Sbjv.NDef.3Sg]	lesz[/V]=le + gyen[Sbjv.NDef.3Sg]	28
schem	ír[/V] + ni[Inf]	le[/Prev]=[] + ír[/V] + ni[Inf]	161
		Összesen:	**762**

2. táblázat. A részletes elemzések hibatípusainak összefoglalása

Az egyszerűsített elemzésnél különbséget tettünk a szófajtévesztésből származó (`lerr`), a csak a morfológiai jegyekben jelentkező hibák (`morph`), a szövegben hibásan szereplő tokenek nyomán létrejövő szóhibák (`word`), az annotáció szintaxisában mutatkozó hibák (`syn`) és sémaváltozásból adódó hibák (`schem`) között. A szófajtévesztésnél az annotátor alapvetően rossz szófajba sorolta be az adott szót. Hibás morfológiai jegyek esetén azonban a szófaj megfelelő volt, viszont egyéb morfológiai jegyek, mint például a főneveknél az eset vagy az igéknél a szám és személy meghatározása volt hibás. Emellett az egyszerűsített elemzésnél szintén előfordultak olyan hibák, amelyek abból adódtak, hogy az annotátorok nem javították a szövegben szereplő helyesírási hibákat. Továbbá itt is megjelentek a címkék szintaxisában történő tévesztések. Az annotáció közben változó séma az egyszerűsített elemzésben is okozott eltéréseket az első annotátor és a gold standard változat között.

EGYSZERŰ ELEMZÉS			
Hibakód	**Helytelen**	**Helyes**	**Gyakoriság**
lerr	*elhunyt [/V][Pst.NDef.3Sg]*	*elhunyt [/N][Nom]*	150
morph	*sejlik [/V][Prs.NDef.3Sg]*	*sejlik [/V][Prs.Def.3Pl]*	40
word	*rossztanulókat [/N][Pl][Acc]*	*rossz [/Adj][Acc]*	14
syn	*Keralából /N][Ela]*	*Keralából [/N][Ela]*	2
schem	*Fanny [/X]*	*Fanny [None]*	97
		Összesen:	**303**

3. táblázat. Az egyszerű elemzések hibatípusainak összefoglalása

4.3. Az emtsv és az annotátorok teljesítményének kiértékelése

Az emtsv által adott elemzéseket összevetettük a tesztkorpuszunkkal. Az egyszerű elemzésekben 989 (6,99%), a szótövekben pedig 711 (5,03%) hibát találtunk. A részletes elemzések helyességét nem értékeltük ki, hiszen ezekhez nem tartozik egyértelműsítő modul, és mivel sok szóalaknál azonos tő és azonos egyszerű elemzés mellett is lehetséges többféle részletes elemzés (1. ábra), így az emtsv eszközeivel lehetetlen ezek közül választani.

1. ábra: Példa konkurrens részletes elemzésekre

Az annotátorok teljesítményét aszerint értékeltük ki, hogy mennyit találtak meg és javítottak ki az emtsv hibáiból (azaz az eltéréseket vetettük össze az emtsv hibalistájával). Az eredményeket a 4. táblázat tartalmazza. A pontosságértékek szerint a módosítások nagyjából 80-90%-a sikeres javítást jelentett. (Egyébként minden módosított token hibás volt az emtsv elemzésében, csak az annotátorok által megadott javított verziók sem voltak feltétlenül helyesek.) A fedéseredmények pedig azt mutatják, hogy az emtsv hibáinak bő egyharmadával egyetértettek az annotátorok, vagy ha meg is találták őket, nem jól javították ki. A kézi annotáció így összességében 2-3%-ot javított az emtsv eredeti elemzésein.

	Pontosság	**Fedés**
Lemma	81,50%	59,48%
Egyszerű elemzés	91,23%	62,95%

4. táblázat. Az annotátorok eredményei az emtsv hibáinak megtalálásában

Mindez megerősíti azt a kiinduló kérdésfelvetésünket, miszerint a dupla annotáció és az eltérések áttekintése nem túlságosan pazarló módszer-e egy olyan annotálási munkafolyamatban, ahol már egy eleve jól teljesítő gépi elemző eszköz kimenetéből indulunk ki.

4.4. Az annotációt javító módszerek eredményei

Bár a vizsgált módszereknek nem mindegyike alkalmas minden annotációs szintünk ellenőrzésére, a különböző szintű hibák átfedéseinek nagy száma miatt

mindegyik módszert kiértékeltük mindegyik szinten. Az eredményeket elemzési szintenként elkülönítve az 5–7. táblázatok tartalmazzák. Összességében elmondható, hogy a legjobb fedéseredményeket minden esetben a baseline módszerrel kaptuk, de a többi módszerhez képest jóval nagyobb számú átnézendő hibajelölt árán.

	Hibajelöltek száma	**Megtalált hibák**	**Pontosság**	**Fedés**
Baseline	8796	247	2,81%	83,17%
e-magyar	535	99	18,50%	33,33%
HuSpaCy	2714	147	5,42%	49,49%
Érvénytelen bigram	967	130	13,44%	43,77%

5. táblázat. A hibakereső módszerek eredményei az egyszerű elemzésekre

Az egyszerű elemzéseknél a legjobb eredményt a HuSpaCy érte el: az emberi annotációk és a HuSpaCy kimenetének összevetésével a korpusz 20%-ának áttekintése elegendő a hibák közel felének megtalálásához.

A részletes elemzésekre értelemszerűen rosszabb eredményeket kaptunk, hiszen a baseline-on kívül egyik módszer sem alkalmas ennek az elemzési szintnek az ellenőrzésére. Az itt kapott találatok így annak köszönhetőek, hogy ahol az egyszerű elemzés hibás, ott általában a részletes is.

	Hibajelöltek száma	**Megtalált hibák**	**Pontosság**	**Fedés**
Baseline	8284	678	8,18%	88,98%
e-magyar	1476	295	19,99%	38,71%
HuSpaCy	2714	258	9,51%	33,86%
Érvénytelen bigram	967	159	16,44%	20,81%

6. táblázat. A hibakereső módszerek eredményei a részletes elemzésekre

A szótövek gyenge eredményei azonban meglepőek, úgy tűnik a szótőhibák detektálása (önmagában) különösen nehéz feladat. Ebben is a HuSpaCy-vel értük el a legjobb fedést.

	Hibajelöltek száma	**Megtalált hibák**	**Pontosság**	**Fedés**
Baseline	5099	160	3,14%	81,63%
e-magyar	365	32	8,77%	16,33%
HuSpaCy	801	65	8,11%	33,16%
Érvénytelen bigram	967	42	4,34%	21,43%

7. táblázat. A hibakereső módszerek eredményei a szótövekre

Megvizsgáltuk azt is, hogy az egyes módszerek milyen típusú hibákat találnak meg a legsikeresebben. Ezek elemzési szintek szerinti összefoglalását a 8–10. táblázatok tartalmazzák.

	e-magyar	**HuSpaCy**	**Bigram**
Szófajtévesztés	38,00%	48,00%	48,00%
Morfológiai jegyek	20,00%	55,0%	70,00%
Szintaxis	100%	50,0%	100%
Sémaváltozás	29,90%	50,52%	25,77%
Szóhiba	21,43%	21,43%	21,43%

8. táblázat. Az egyszerű elemzések megtalált hibáinak aránya hibatípusok szerint

Az egyszerű elemzéseknél a szintaktikai hibák megtalálása mind az e-magyar, mind az érvénytelen bigram módszerekkel könnyűnek bizonyult, ami nem is meglepő, hiszen szintaktikai hiba csak akkor fordulhat elő az annotációban, ha azt az annotátor kézzel írta be (tehát ezek értelemszerűen el fognak térni az eredeti e-magyar elemzéstől). Az érvénytelen bigram módszer emellett kiemelkedően teljesít a morfológiai jegyek hibáinak detektálásában.

	e-magyar	**HuSpaCy**	**Bigram**
Szintaxis	100%	71,43%	71,43%
Túlelemzés	11,38%	15,57%	7,19%
Alulelemzés	82,05%	23,93%	14,53%
Hibás elemzés	34,10%	41,47%	34,10%
Sémaváltozás	45,96%	35,40%	15,53%
Eltérés az emMorphtól	3,57%	71,43%	42,86%
Szóhiba	38,46%	23,08%	23,08%
Hiányzó elemzés	6,06%	48,48%	39,39%
Egyéb hiba	83,33%	66,67%	16,67%

9. táblázat. A részletes elemzések megtalált hibáinak aránya hibatípusok szerint

A részletes elemzéseknél szintén a szintaktikai hibák megtalálása volt a legkönnyebb. Ezen kívül az e-magyartól való eltérések nagy arányban jeleztek alulelemzést és egyéb hibát, a HuSpaCy segítségével pedig viszonylag jól megtalálhatók az emMorph sémától való eltérések (ennek okát nem vizsgáltuk meg részletesen).

A szótöveknél a kis- és nagybetűk hibái nagy arányban megtalálhatók a HuSpaCy segítégével. Ezen kívül az e-magyar eltérésekkel értünk el viszonylag jó eredményt az elütések detektálásában.

Látható tehát, hogy az egyes hibakereső módszerek különböző hibatípusok megtalálásában sikeresek, ezért érdemes őket kombinálni. Elvégeztük a három módszer (a baseline-t a nagyszámú hibajelölt miatt kihagytuk) összes hibajelölt-

	e-magyar	HuSpaCy	Bigram
Elütés	60,0%	40,0%	20,0%
Karakterhiba	14,29%	42,86%	0%
Hibás szótő	31,43%	55,07%	38,57%
Kis-/nagybetű	0%	86,67%	13,33%
Sémaváltozás	6,25%	8,33%	12,50%
Tokenizálás	0%	0%	0%

10. táblázat. A szótövek megtalált hibáinak aránya hibatípusok szerint

jének[2] összevetését az összes hibával, ezúttal már nem annotációs szintenként, hanem tokenenként. Ez utóbbi összevonást azért is érdemes megtenni, mert láthattuk, hogy a szótőhibák specifikus keresése nem volt sikeres, ám mivel a szótőhibák gyakran járnak együtt a morfológiai elemzés(ek) hibájával, így az elemzések hibajelöltjei egyúttal kiadhatják a szótőhibákat is. Másrészről, az annotálási munka során az egyes annotációs szinteket együtt ellenőrizzük, így a hibajelöltek kiválasztásánál az az elsődleges feladat, hogy határozzuk meg azokat a tokeneket, amelyek potenciálisan hibás annotáció(ka)t tartalmaznak.

Az eredmények összesítését a 11. táblázat tartalmazza. Az annotált szövegekben összesen 777 token volt hibás (a korpusz 5,49%-a). A hibakereső módszerek együttesen 4478 hibajelöltet adtak ki, és a hibák 63,71%-át találták meg. Ez összességében azt jelenti, hogy az egy annotátor által annotált korpusz 31,65%-ának átnézésével a hibás tokenek arányát 2%-ra tudjuk csökkenteni.

Hibajelöltek		**Összes hiba**		**Megtalált hibák**		**Megmaradt hibák**	
Száma	**Aránya**	**Száma**	**Aránya**	**Száma**	**Fedés**	**Száma**	**Aránya**
4478	31,65%	777	5,49%	495	63,71%	282	2,0%

11. táblázat. A vizsgált módszerek eredményeinek összesítése

Az előbbi eredményeket tovább árnyalja, ha megnézzük, hogy miyen típusú hibák maradtak a korpuszban a hibakeresések összesítése[3] után (12. táblázat). Ebből látható, hogy a megmaradt hibák közel fele túl- vagy alulelemzés, ami sok esetben meglehetősen szubjektív megítélésű, és a többi elemzési szintet (szótő, egyszerű elemzés) nem érinti. Az annotálási útmutató változásaiból fakadó hibák (schem) szintén nem jelentenek nagy problémát, hiszen ezek csak a szöve-

[2] A hibajelöltek között megtartottuk azokat a szavakat is, amelyek szerepelnek a 2. pontban említett egyértelmű szavak listáján, sok esetben ugyanis éppen a címekben és egyéb tulajdonnevekben hibáztak az annotátorok, mely esetekre a lista kivételesen nem érvényes.

[3] Több hiba esetén elsősorban az egyszerű elemzés, másodsorban a szótő, és harmadsorban a részletes elemzés hibáit vettük figyelembe. Így például az `oana` típusba sorolt tokenek annotációja biztosan nem tartalmaz más hibát, míg a `morph` típusba soroltaknál több hiba is lehetséges.

gek egy részét érintik, és mivel a sémaváltozások ismertek, ezekre az eltérésekre valószínűleg érdemes lehet specifikusan keresni.

Hibakód	**Megmaradt hibák száma**
oana	118
schem	69
err	37
lerr	30
morph	6
lem	6
uana	4
miss	4
word	3
emM	3
typ	1
cerr	1

12. táblázat. A megmaradt hibák típus szerinti eloszlása

5. Összegzés

Kutatásunk célja az volt, hogy csökkentsük az emberi annotációs munka mennyiségét a gold standard korpusz projektünkben. Három hibakereső módszert teszteltünk részletesen. Ebből kettő az emberi és a gépi annotáció összevetésén alapul. Az annotált tesztkorpuszunkat összehasonlítottuk az előfeldolgozásra is használt e-magyar, és az annotálási munkafolyamattól teljesen független HuSpacy kimenetével is. A harmadik vizsgált módszer, az érvénytelen bigram módszer más megközelítést alkalmaz. Ennek lényege, hogy egy kisebb validált korpuszból kinyerjük a lehetséges szófajcímke-bigramokat, és az annotációk ezzel nem kompatibilis bigramjait alkotó tokeneket tekintjük hibajelöltnek.

Bár önmagában egyik módszer sem adott kiemelkedő eredményt, a megtalált hibák részletesebb vizsgálatából kiderült, hogy a különböző módszerek jellemzően különböző típusú hibákat képesek megtalálni. A három módszer eredményeit összegezve azt az eredményt kaptuk, hogy a tokenek 31,65%-át újraannotálva a szövegben előforduló annotálási hibák arányát 2%-ra csökkenthetjük. Ez a munkaterhelésben lényeges csökkentést jelent, miközben magas minőséget is biztosít.

Korpuszunk és a kutatáshoz használt annotált szövegek megtalálhatók a githubon[4]. További terveink között szerepel specializált módszerek kidolgozása a gyakori hibatípusok megtalálására és javítására.

[4] Gold standard korpusz projekt: https://github.com/ELTE-DH/gold-standard
A cikkben ismertetett kutatás: https://github.com/ELTE-DH/gold-standard-eval

Hivatkozások

Dickinson, M., Tufis, D.: Iterative enhancement. In: Ide, N., Pustejovsky, J. (szerk.) Handbook of Linguistic Annotation, pp. 257–276. Springer (2017)

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: emtsv – Egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019)

Květoň, P., Oliva, K.: Achieving an almost correct pos-tagged corpus. In: Sojka, P., Kopeček, I., Pala, K. (szerk.) Text, Speech and Dialogue. pp. 19–26. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)

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. (szerk.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 1068–1073. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014), aCL Anthology Identifier: L14-1207

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit (2022)

Pethő, G., Sass, B., Kalivoda, A., Simon, L., Lipp, V.: Igekötő-kapcsolás. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) MSZNY 2022, XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 77–91. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Tolcsvai Nagy, G.: A tulajdonnév jelentése. In: Bölcskei, A., N. Császi, I. (szerk.) Név és valóság. A VI. Magyar Névtudományi Konferencia előadásai. pp. 30–41. Károli Gáspár Református Egyetem BTK Magyar Nyelvtudományi Tanszéke, Budapest (2008)

Vadász, N.: KorKorpusz: kézzel annotált, többrétegű pilotkorpusz építése. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020). pp. 141–154. Szegedi Tudományegyetem, TTIK, Informatikai Intézet, Szeged (2020)

Váradi, T., Simon, E., Sass, B., Mittelholcz, I., Novák, A., Indig, B., Farkas, R., Vincze, V.: E-magyar – A Digital Language Processing System. In: chair), N.C.C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., Tokunaga, T. (szerk.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 7-12, 2018 2018)

Poszter, laptopos bemutató

A beszéd artikulációs mozgásának predikciója agyi jel alapján – kezdeti eredmények

Csapó Tamás Gábor[1], Arthur Frigyes Viktor[1], Nagy Péter[2,3], Boncz Ádám[3]

[1]Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
[2]Budapesti Műszaki és Gazdaságtudományi Egyetem,
Méréstechnika és Információs Rendszerek Tanszék
[3]Természettudományi Kutatóközpont, Kognitív Idegtudományi és
Pszichológiai Intézet, Hang- és Beszédészlelési Kutatócsoport, Budapest
{csapot,arthur}@tmit.bme.hu,
nagy.peter.ssprg@ttk.hu, adam.boncz@gmail.com

Kivonat Az augmentatív és alternatív kommunikációs technológiák (pl. agy-gép interfész, BCI) közvetlenül olvashatják az agyi jeleket, hogy pótolják az elvesztett beszédképességet. Nemzetközi szinten végeztek már kezdeti kutatásokat agyi jel (pl. EEG, sEEG, ECoG) és beszéd alapú BCI kidolgozására, azonban hiányoznak azok a kombinált módszerek, amelyek a nem invazív EEG-t, az artikulációt és a beszédjeleket összevontan vizsgálnák, és elemeznék az agyban zajló tervezési folyamat, az artikulációs mozgás, és a keletkezett beszédjel kölcsönhatását. A jelen kutatásban ismertetett multimodális (EEG, nyelvultrahang és beszéd) analízis és szintézis révén túlmutatunk a legkorszerűbb nemzetközi trendeken. A beszéd közbeni agyi jelek elemzését nyelvultrahang-alapú artikulációs adatokkal bővítjük, hogy több összehasonlítható bioszignál álljon rendelkezésre. Az EEG-vel mért agyi jelből mély neuronhálóval az artikulációs mozgásra vonatkozó információt (nyelvultrahang képek) prediktálunk. Az eredmények szerint az EEG és nyelvultrahang közötti kapcsolat kimutatható. A jelen kutatás hosszútávú célja, hogy hozzájáruljunk a beszéd alapú agyszámítógép interfészekhez: az eredmények potenciálisan alkalmazhatók lehetnek pl. beszédsérülteknek szánt kommunikációs segédeszközként.
Kulcsszavak: beszédtechnológia, ultrahang, EEG, deep learning

1. Bevezetés

Az agy-számítógép interfészek (Brain-Computer Interface, BCI) lehetővé tehetik a számítógépek közvetlen, fizikai aktivitás nélküli vezérlését. Az augmentatív és alternatív kommunikációs (Augmentative and Alternative Communication, AAC) technológiák (pl. BCI) közvetlenül olvashatják az agyi jeleket, hogy pótolják az elvesztett beszédképességet (Chang és Anumanchipalli, 2020). A jövőben a beszédneuroprotézisek alkalmazása segíthet a neurológiai vagy beszédhibás betegeken.

Az agyi jel rögzítésére többféle technológia is rendelkezésre áll: például elektroenkefalográfia (EEG, McFarland és Wolpaw, 2017), sztereotaktikus mély elektródák (sEEG, Verwoert és mtsai, 2022), intrakraniális elektrokortikográfia (ECoG, Buzsáki és mtsai, 2012), magnetoenkefalográfia (MEG, Dash és mtsai, 2021), lokális mezőpotenciál (LFP, Buzsáki és mtsai, 2012). Az idegrendszeri jel rögzítési módok közül a BCI számára az EEG lehet a legmegfelelőbb, mivel elérhető árú, lényegesen kisebb kockázattal jár, mint az invazív módszerek, és hordozható is lehet (Casson, 2019). Nemzetközi szinten végeztek már kezdeti kutatásokat EEG és beszéd alapú BCI kidolgozására (Krishna és mtsai, 2020; Verwoert és mtsai, 2022; Arthur és Csapó, 2022a; Luo és mtsai, 2022), azonban ez még nem eredményezett jól érthető beszédet. Mivel az EEG csak a skalpon méri a jelet, ezért kevésbé pontosan lehet következteni egyes agyi régiók aktivitására/tevékenységére, mintha közvetlenül az agyban mérnénk. Invazív módszerekkel már sikerült beszéd-szerű szintetizált beszédet létrehozni agyi jelek alapján, pl. ECoG (Herff és mtsai, 2015; Anumanchipalli és mtsai, 2019; Le Godais, 2022) és sEEG (Angrick és mtsai, 2021; Verwoert és mtsai, 2022; Arthur és Csapó, 2022b), de a fenti hátrány (invazív jelleg) miatt utóbbiak széles körű elterjedése nem várható.

1.1. Az agyi jelek és artikulációs mozgás kapcsolata

Az artikulációs mozgást még nem vizsgálták az agyi jelekkel párhuzamosan. A legtöbb kapcsolódó kutatás csak származtatott adatokat használ, azaz a beszédjelből visszakövetkeztetett artikulációs információt veszik figyelembe (Carey és mtsai, 2017; Anumanchipalli és mtsai, 2019; Favero és mtsai, 2022; Le Godais, 2022). Carey és mtsai (2017) kutatásában például ugyanazon beszélőkkel vettek fel MRI-t az artikulációs csatornáról, és funkcionális MRI-t az agyról. Mivel a két jel egyszerre nem rögzíthető, ezért a beszélők ugyanazt a stimulust többször megismételték a két modalitáshoz, így a beszédjelen „keresztül” vizsgálható az agyi jel és artikuláció kapcsolata. Anumanchipalli és mtsai (2019) az artikulációs csatornára vonatkozó kinematikus információt (pl. ajakmozgás, nyelvmozgás, és állkapocs állása), valamint az egyéb fiziológiai jellemzőket (pl. artikuláció mechanizmusa) a beszédjelből becsüli, akusztikum-artikuláció inverziós (Acoustic-to-Articulatory Inversion, AAI) módszerekkel. Az AAI modell betanítására a MOCHA-TIMIT adatbázist használták (az agyi jeles felvételektől független beszélőkkel), melyben elektromágneses artikulográffal rögzítették az artikulációs mozgást (Wrench, 2000). Favero és mtsai (2022) szintén AAI jellegű módszereket alkalmaztak, de itt az artikulációs információ nem valós mérésen alapul, hanem a beszédjelből számított ún. TADA jellemzőkön (Nam és mtsai, 2004). Le Godais (2022) kortikális aktivitásból beszéd dekódolása során lineáris módszerekkel próbálkozott az artikulációs információ hozzáadásával. Az artikulációt beszédből származtatta, dinamikus idővetemítés alapján, független beszélők artikulációs mozgását felhasználva.

A fenti tanulmányok konklúziója és távlati célja, hogy olyan betegek számára, akiknél az artikuláció agykérgi feldolgozása még érintetlen, a beszédalapú BCI dekóder intuitívabb lehet és könnyebben megtanulható a használata.

1.2. A jelen kutatás célja

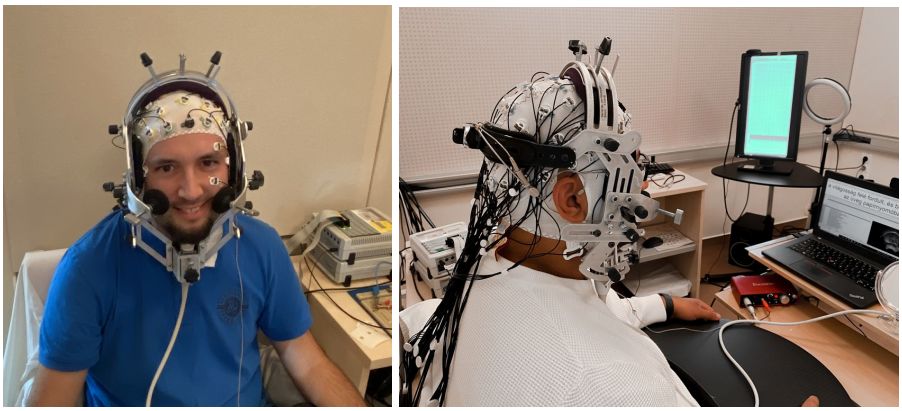
A jelen fejezet áttekintése szerint egyelőre hiányoznak azok a kombinált módszerek, amelyek a nem invazív EEG-t, az artikulációt és a beszédjeleket összevontan vizsgálnák, és elemeznék az agyban zajló tervezési folyamat, az artikulációs mozgás, és a keletkezett beszédjel kölcsönhatását. A jelen kutatásban a beszéd közbeni agyi jelek elemzését nyelvultrahang-alapú artikulációs adatokkal bővítjük, hogy több összehasonlítható bioszignál álljon rendelkezésre. Az EEG-vel mért agyi jelből mély neuronhálóval az artikulációs mozgásra vonatkozó információt prediktálunk, nyelvultrahang képek formájában.

2. Módszerek

2.1. Felvételek

A felvételek az ELKH TTK egyik csendes szobájában készültek. Az EEG jelet 64 csatornás Brain Products actiCHamp típusú erősítővel rögzítettük, actiCAP aktív elektródák felhasználásával. Négy csatorna a horizontális és vertikális szemmozgás követését szolgálta. Az elektródákat a nemzetközi 10-20-as elrendezés szerint helyeztük el (Klem és mtsai, 1999). Az elektródák impedanciáját 15 kOhm alatt tartottuk. A felvétel során az FCz elektróda töltötte be a referencia elektróda szerepét. A jelet 1000 Hz frekvencián mintavételeztük.

A nyelv középvonalának (szagittális) mozgását a „Micro” rendszerrel rögzítettük (AAA v220.02 szoftver, Articulate Instruments Ltd.) egy 2–4 MHz frekvenciájú, 64 elemű, 20 mm sugarú konvex ultrahang-vizsgálófejjel, 81,67 fps sebességgel, és rögzítő sisakot is alkalmaztunk (Csapó és mtsai, 2017a). A fém sisakot az EEG érzékelők fölé helyeztük el úgy, hogy az eszközök lehetőleg ne zavarják egymást. A felvételi elrendezésre az 1. ábra mutat példákat.

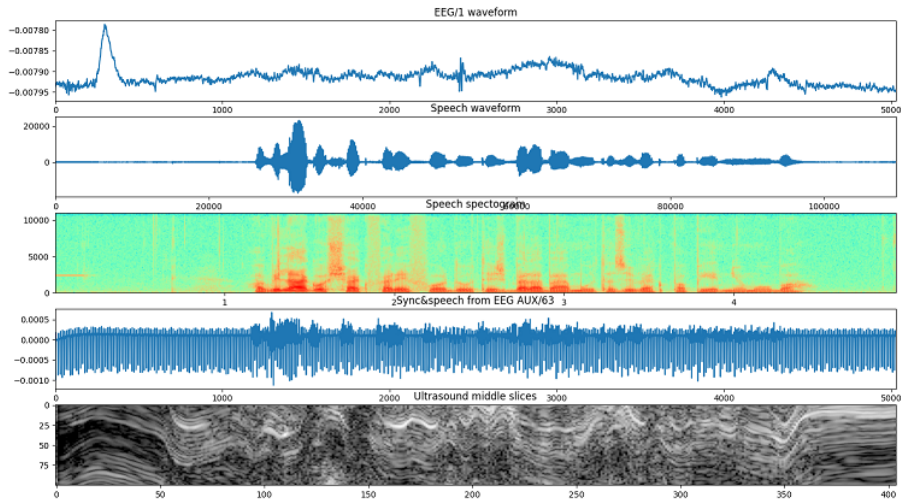


1. ábra: Felvételi elrendezés: EEG, nyelvultrahang, mikrofon és webkamera.

A beszédet egy Beyerdynamic TG H56c tan omnidirekcionális kondenzátor mikrofonnal vettük fel, és M-Audio M-Track 2x2 / FocusRite Scarlett 2i2 USB-s külső hangkártyával digitalizáltuk, 44 100 Hz-en. A beszélő arc- és szájmozgását egy Logitech C925e webkamerával, 1920x1080 pixel felbontásban rögzítettük.

A hangkártya kimenetét (amely a „Micro” ultrahang „frame sync” szinkronizáló jelét és a mikrofonból származó beszédjelet együttesen tartalmazza) rákötöttük az EEG AUX csatornájára – így az agyi és artikulációs jeleket ugyan külön számítógépeken rögzítettük, de mégis tudjuk utólag szinkronizálni az adatokat (ld. 2. ábra). Az EEG jelet folyamatosan vettük fel, míg a nyelvultrahangot és beszédet mondatonként. Mivel a beszédjel és a nyelvultrahang szinkronizációs jel (így az adott mondat felvételének kezdete és vége) megjelenik az EEG egyik csatornáján is, ezért a jeleket automatikusan szinkronizálni tudjuk utólag.

Az cikk beküldésének idejéig három magyar anyanyelvű férfi beszélővel rögzítettünk beszélőnként kb. 10–10 percnyi felvételt (a PPBA adatbázisból származó mondatokat (Olaszy, 2013)), melyet a későbbiekben további beszélőkkel bővítünk.

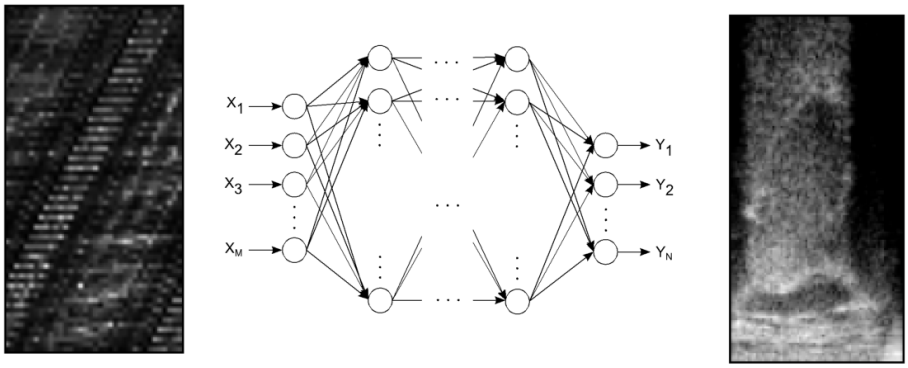


2. ábra: Példa a szinkronizált EEG, beszéd, és nyelvultrahang felvételre. a) EEG / 1. csatorna, b) beszédjel, c) beszéd spektrogram, d) ultrahang szinkornizálójel és beszédjel (EEG AUX-on), e) nyelvultrahang képek középső vonalának időbeli változása.

2.2. Az adatok előfeldolgozása

Az EEG jel előfeldolgozását Verwoert és mtsai (2022) alapján végeztük, a rendelkezésre álló szkriptekből kiindulva (`https://github.com/neuralinterfacinglab/SingleWordProductionDutch/blob/main/extract_features.py`). Az EEG jel minden csatornájára (az EEG AUX kivételével) kiszámítjuk a Hilbert burkolót négy frekvenciasávban: 1–50 Hz, 51–100 Hz, 101–150 Hz, és 151–200Hz. A burkolót 50 ms-onként átlagoltuk, és 12 ms-os eltolással számoltuk, hogy a nyelvultrahanggal összhangban legyen (melynek sebessége 81,67 fps volt). Ahhoz, hogy az időbeli információt is figyelembe vegyük, 4 megelőző és 4 követő blokkot is felhasználtunk a Hilbert-transzformált EEG jelekből. Az így előállított bemeneti jelre a 3. ábra bal oldala mutat példát.

A nyelvultrahang képeket 8 bites szürkeárnyalatos pixelekként használtuk fel, a „Micro” rendszer nyers ultrahang formájában. Az eredetileg 64x842 pixeles képeket átméreteztünk 64x128 pixelre (3. ábra, jobb oldal), mivel ez nem okoz jelentős információ veszteséget (Csapó és mtsai, 2022), de így kevesebb a feldolgozandó adat mennyisége.



3. ábra: A neuronháló bemenete (bal oldalon) és kimenete (jobb oldalon).

2.3. Artikuláció predikciója EEG bemenetből

A kutatás kezdeti fázisában egy egyszerű kísérletet végeztünk: teljesen kapcsolt (fully connected, FC-DNN) mély „egyenirányított” (rectifier) neurális hálózatot (Glorot és mtsai, 2011) tanítottunk, melynek során a nyelvultrahang képeket predikáltuk, a Hilbert-transzformált EEG bemenetből (3. ábra). A tanítás során az átlagos négyzetes hibafüggvényt (MSE) alkalmaztuk. Kísérleteink során egy 5 rejtett réteges, rétegenként 1000 neuront tartalmazó neuronháló struktúrát használtunk, ReLU aktivációval és lineáris kimeneti réteggel (hasonlóan az első nyelvultrahang-beszéd szintézis tanulmányunkhoz, (Csapó és mtsai, 2017b)). A

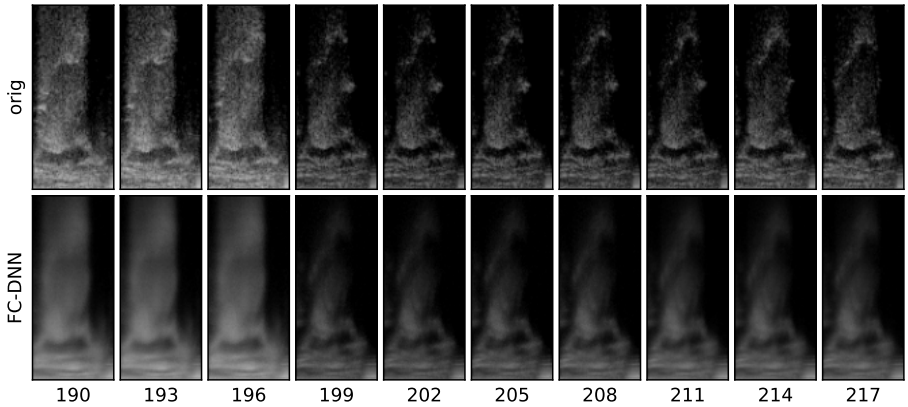
bemeneti EEG értékeket és a kimeneti ultrahang pixeleket a tanítás előtt 0–1 közé normalizáltuk. 100 epochig tanítottunk, de korai leállást alkalmaztunk, azaz ha a validációs hiba nem csökkent 3 epoch-on keresztül, a tanítást leállítottuk.

3. Kísérletek és eredmények

A rendelkezésre álló felvételekből beszélőfüggő tanításokat végeztünk, az adatok 80%-át használtunk a neuronháló tanítására, 10%-ot validációra, a maradék 10%-ot pedig tesztelésre (FF1 beszélő esetén: 25 600 / 3200 / 3200 mintapont).

3.1. Demonstrációs minták

A DNN tanítás után a teszthalmazon EEG-ből nyelvultrahang predikciót végeztünk. A 4. ábra az FF1 beszélőtől mutat néhány eredeti és EEG alapján becsült nyelvultrahang képet, az ultrahang gép „nyers” reprezentációjában. A nyelvultrahang kontúrja az eredeti képeken sem minden esetben látszik jól – ennek oka a nyelvultrahang beszélőfüggősége (Csapó, 2022). Az EEG alapján becsült képeken a nyelv kontúrja elkent, és a nyelv helyzetének képkockát közötti változása is nehezen kivehető - azaz a DNN az általános nyelvalakot (az átlagos képet) meg tudta tanulni, de a nyelvmozgásra vonatkozó finom részletek nem kivehetőek.



4. ábra: Demonstrációs minta: eredeti (felül) és EEG alapján becsült (alul) nyelvultrahang képek az FF1 beszélőtől, „nyers” ultrahang reprezentációban.

Ugyanezt a képsorozatot mutatja az 5. ábra, „szétterített” reprezentációban (a transzformációt az `https://github.com/UltraSuite/ultrasuite-tools` eszközzel végeztük). A szürkeárnyalatos képeken hasonló tendencia vehető észre, mint a 4. ábrán: az eredeti nyelvultrahang képeken még sejthető a nyelv felső

határvonala, de az EEG alapján becsült képeken már elmosódnak az ultrahang pixelek, és a nyelv kontúrja nem látható. Viszont a 196 és 199 időpillanat között a fényerősség váltása a DNN-predikált esetben is észrevehető.

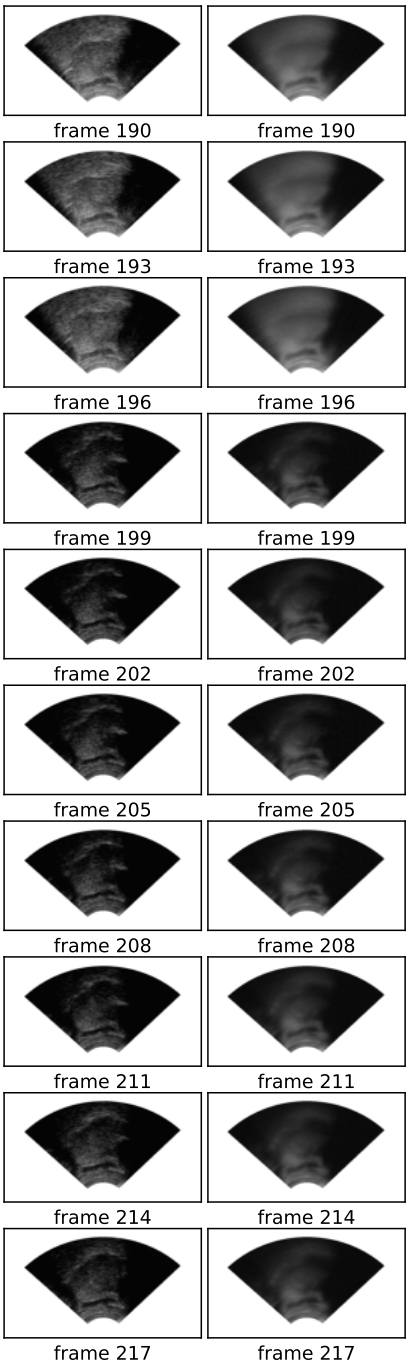
Ha képenként nézzük az eredményeket (mint a 4. és 5. ábrákon), akkor kevésbé látszik a hosszabb időtávlatú tendencia. Emiatt egy másik elrendezésben is ábrázoljuk az eredményeket: minden nyelvultrahang képből kivágtuk a középső függőleges vonalat (kb. ez megfelel a nyelv közepének), és ezen vonal időbeli változását ábrázoltuk, egy spektrogramhoz hasonlóan. A 6. ábra ennek eredményét mutatja: felül a beszédhez tartozó spektogram, középen az ugyanehhez bemondáshoz tartozó nyelvultrahang középvonal időbeli változása, alul pedig a DNN által prediktált nyelvultrahang középvonal látható. Az a) mel-spektrogram és b) artikulációs mozgás közötti hasonlóság egyértelműen észrevehető: a beszédben lévő formánsmozgások, és a nyelv függőleges mozgása nagyjából kivehető az ábrákon. A c) DNN-predikált nyelvultrahang középvonalon viszont a nyelv mozgása nem látszik, azaz a DNN nem tudta megtanulni az EEG és a nyelvultrahang közötti összefüggést. Ugyanakkor valamilyen információ mégis látszik a DNN-predikált képeken: a 195. időpillanat végén az egyik mondatnak vége van, és a következő elkezdődik, ami a b) eredeti nyelvultrahangon jól kivehető, és a c) becsült nyelvultrahangon is látszik.

3.2. Objektív mérések

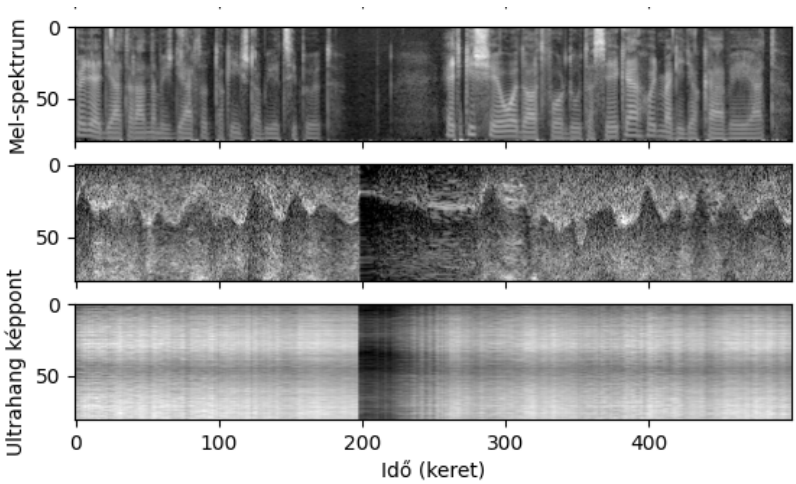
A fenti FC-DNN hálózattal elért átlagos négyzetes hiba (MSE) értékek az FF1 beszélő esetén: validációs hiba: 0,0053, teszthiba: 0,0055. Az érték önmagában nehezen értelmezhető, és az sem egyértelmű, milyen jóságot ír le, de pl. korábbi akusztikum-artikuláció inverziós kísérleteinkben (melynek során a beszédjelből becsültünk nyelvultrahang képeket (Porras és mtsai, 2019; Csapó és Sepúlveda, 2021)) a kapott NMSE validációs hiba értékek 0,0053–0,0088 nagyságrendben voltak; és ez esetben a beszédből generált nyelvulrahang kép közelítette az eredeti artikulációt. Ebből is látható, hogy az MSE érték a jelen kutatásban nem elegendő az eredmények jóságának megítélésére, mindenképp szükség van a vizuális vizsgálatra. A korábbi ultrahangos kutatásaink esetén kísérleteztünk más hibamértékek vizsgálatával is, úgymint Structural Similarity Index (SSIM) (Wang és mtsai, 2004), és Complex Wavelet Structural Similarity (CW-SSIM) (Sampat és mtsai, 2009), ultrahangon (Xu és mtsai, 2016; Csapó és mtsai, 2020; Csapó és Sepúlveda, 2021). Azonban a fenti vizuálisan gyenge eredmények miatt az SSIM-et és CW-SSIM-et itt az EEG-ből kiinduló nyelvultrahang becslés esetén egyelőre nem vizsgáltuk.

4. Diszkusszió és következtetések

A kutatás jelen kezdeti fázisában ismertetett multimodális (agy, beszéd és artikuláció) analízis és szintézis révén túlmutatunk a legkorszerűbb nemzetközi trendeken.



5. ábra: Demonstrációs minta: eredeti (balra) és EEG alapján becsült (jobbra) nyelvultrahang képek az FF1 beszélőtől, „szétterített” ultrahang reprezentációban.



6. ábra: Demonstrációs minta: a) eredeti beszédminta 80-dimenziós mel-spektrogramja, b) eredeti nyelvultrahang felvétel középvonalának időbeli változása, c) a becsült nyelvultrahang középvonalának időbeli változása.

A szakirodalomban számos korábbi kísérletet láttunk EEG (vagy más eszközzel mért agyi jel) alapján készült kezdeti beszéd BCI kutatásokra (Herff és mtsai, 2015; Anumanchipalli és mtsai, 2019; Krishna és mtsai, 2020; Arthur és Csapó, 2022a,b), azonban egyelőre az agyi jel alapú beszédszintézis esetén nem sikerült jól érthető beszédet létrehozni. Kézenfekvő megoldásnak tűnik az artikuláció, mint az agyi jel és végső beszéd közti köztes reprezentáció vizsgálata, mellyel a jelen cikkben is foglalkoztunk. Korábbi kutatásokban az artikulációs adatokat csak származtatott módon, nem közvetlenül mérve tudták felhasználni az agyi jel és beszéd vizsgálata során (Carey és mtsai, 2017; Anumanchipalli és mtsai, 2019; Favero és mtsai, 2022; Le Godais, 2022). Bár ez a közvetett artikulációs információs is segített az eredmények pontosításában, az artikuláció valós eszközökkel történő mérése további javulást eredményezhet.

A jelen kutatásban az EEG-vel mért agyi jel és mikrofonnal rögzített beszéd vizsgálatát egészítettük ki nyelvultrahanggal mért artikulációs felvételekkel. Mély neuronháót (FC-DNN) taníottunk EEG bemenet alapján nyelvultrahang becslésére. Az eredmények szerint a generált nyelvultrahang képsorozat még távol van az eredeti nyelvultrahangtól, de az EEG és a nyelvultrahang közötti kapcsolat egyértelműen kimutatható.

A jelen kutatás hosszútávú célja, hogy hozzájáruljunk a beszéd alapú agyszámítógép interfészekhez. Az eredmények potenciálisan alkalmazhatók lehetnek rehabilitáció során, pl. kommunikációs segédeszközként.

A fent bemutatott DNN-es kísérletek keras implementációja a következő címen érhető el: `https://github.com/BME-SmartLab/EEG-to-UTI`

Köszönetnyilvánítás

A kutatást a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal OTKA programja támogatta (FK 142163 projekt). Csapó Tamás Gábor kutatásait az MTA Bolyai János kutatói ösztöndíja, valamint az Új Nemzeti Kiválóság Program Bolyai+ (ÚNKP-22-5-BME-316) pályázata támogatta.

Szeretnénk köszönetet mondani Béres Lucának és Várkonyi Emesének az EEG felvételekben nyújtott segítségért, az ELKH TTK-nak az EEG eszközök biztosításáért, valamint az MTA-ELTE „Lendület" Lingvális Artikuláció Kutatócsoportnak a nyelvultrahang eszközök rendelkezésre bocsátásáért.

Hivatkozások

Angrick, M., Ottenhoff, M.C., Diener, L., Ivucic, D., Ivucic, G., Goulis, S., Saal, J., Colon, A.J., Wagner, L., Krusienski, D.J., Kubben, P.L., Schultz, T., Herff, C.: Real-time synthesis of imagined speech processes from minimally invasive recordings of neural activity. Communications Biology 4(1), 1055 (2021), https://doi.org/10.1038/s42003-021-02578-0

Anumanchipalli, G.K., Chartier, J., Chang, E.F.: Speech synthesis from neural decoding of spoken sentences. Nature 568(7753), 493–498 (apr 2019), https://www.nature.com/articles/s41586-019-1119-1

Arthur, F.V., Csapó, T.G.: Deep learning alapú agyi jel feldolgozás és beszédszintézis előkészítő munkálatai. In: MSZNY 2022. pp. 185–198. online (2022a)

Arthur, F.V., Csapó, T.G.: Speech synthesis from intracranial stereotactic Electroencephalography using a neural vocoder. In: submitted (2022b)

Buzsáki, G., Anastassiou, C.A., Koch, C.: The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes. Nature Reviews Neuroscience 13(6), 407–420 (may 2012), https://www.nature.com/articles/nrn3241

Carey, D., Miquel, M.E., Evans, B.G., Adank, P., Mcgettigan, C.: Vocal Tract Images Reveal Neural Representations of Sensorimotor Transformation During Speech Imitation. Cerebral Cortex 27(5), 3064–3079 (2017)

Casson, A.J.: Wearable EEG and beyond. Biomedical engineering letters 9(1), 53–71 (feb 2019), https://pubmed.ncbi.nlm.nih.gov/30956880/

Chang, E.F., Anumanchipalli, G.K.: Toward a Speech Neuroprosthesis. JAMA 323(5), 413–414 (feb 2020), https://jamanetwork.com/journals/jama/fullarticle/2758116

Csapó, T.G.: A nyelvmozgás ultrahangos vizsgálata és az automatikus elemzés alkalmazási lehetőségei a beszédtechnológiában, pp. 197–219 (2022), https://m2.mtmt.hu/api/publication/33232043

Csapó, T.G., Deme, A., Gráczi, T.E., Markó, A., Varjasi, G.: Szinkronizált beszéd- és nyelvultrahang-felvételek a SonoSpeech rendszerrel. In: MSZNY 2017. pp. 339–346. Szeged (2017a)

Csapó, T.G., Gosztolya, G., Tóth, L., Shandiz, A.H., Markó, A.: Optimizing the Ultrasound Tongue Image Representation for Residual Network-Based Articulatory-to-Acoustic Mapping. Sensors 22 (2022)

Csapó, T.G., Grósz, T., Tóth, L., Markó, A.: Beszédszintézis ultrahangos artikulációs felvételekből mély neuronhálók segítségével. In: MSZNY 2017. pp. 181–192 (2017b)

Csapó, T.G., Sepúlveda, A.: Ultrasound Tongue Image Generation for Acoustic-to-Articulatory Inversion using Convolutional and Recurrent Deep Neural Networks. submitted to Multimedia Tools and Applications (2021)

Csapó, T.G., Xu, K., Deme, A., Gráczi, T.E., Markó, A.: Transducer Misalignment in Ultrasound Tongue Imaging. In: 12th International Seminar on Speech Production (2020)

Dash, D., Ferrari, P., Babajani-Feremi, A., Borna, A., Schwindt, P.D., Wang, J.: Magnetometers vs Gradiometers for Neural Speech Decoding. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference 2021, 6543–6546 (nov 2021), https://pubmed.ncbi.nlm.nih.gov/34892608/

Favero, P., Berezutskaya, J., Ramsey, N.F., Nazarov, A., Freudenburg, Z.V.: Mapping Acoustics to Articulatory Gestures in Dutch: Relating Speech Gestures, Acoustics and Neural Data. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 802–806 (2022)

Glorot, X., Bordes, A., Bengio, Y.: Deep Sparse Rectifier Neural Networks. In: Gordon, G.J., Dunson, D.B. (szerk.) Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS). vol. 15, pp. 315–323. Journal of Machine Learning Research - Workshop and Conference Proceedings, Ft. Lauderdale, FL, USA (2011), http://www.jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf

Herff, C., Heger, D., de Pesters, A., Telaar, D., Brunner, P., Schalk, G., Schultz, T.: Brain-to-text: decoding spoken phrases from phone representations in the brain. Frontiers in Neuroscience 9, 217 (2015), https://www.frontiersin.org/article/10.3389/fnins.2015.00217

Klem, G.H., Lüders, H.O., Jasper, H.H., Elger, C.: The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology. Electroencephalography and clinical neurophysiology. Supplement (jan 1999), https://www.scienceopen.com/document?vid=5960cfa8-7fde-441c-8592-35fdb9841499

Krishna, G., Tran, C., Han, Y., Carnahan, M., Tewfik, A.H.: Speech Synthesis Using EEG. In: Proc. ICASSP. pp. 1235–1238. online (2020)

Le Godais, G.: Decoding speech from brain activity using linear methods. Ph.D.-értekezés, Université Grenoble Alpes (2022), https://tel.archives-ouvertes.fr/tel-03852448

Luo, S., Rabbani, Q., Nathan, ·., Crone, E.: Brain-Computer Interface: Applications to Speech Decoding and Synthesis to Augment Communication. Neurotherapeutics 2022 1, 1–11 (jan 2022), https://link.springer.com/article/10.1007/s13311-022-01190-2

McFarland, D.J., Wolpaw, J.R.: EEG-based brain–computer interfaces. Current Opinion in Biomedical Engineering 4, 194–200 (dec 2017)

Nam, H., Goldstein, L., Saltzman, E., Byrd, D.: TADA: An enhanced, portable Task Dynamics model in MATLAB. The Journal of the Acoustical Society of America 115(5), 2430 (apr 2004), https://asa.scitation.org/doi/abs/10.1121/1.4781490

Olaszy, G.: Precíziós, párhuzamos magyar beszédadatbázis fejlesztése és szolgáltatásai [Development and services of a Hungarian precisely labeled and segmented, parallel speech database] (in Hungarian). Beszédkutatás 2013 [Speech Research 2013] pp. 261–270 (2013)

Porras, D., Sepúlveda-Sepúlveda, A., Csapó, T.G.: DNN-based Acoustic-to-Articulatory Inversion using Ultrasound Tongue Imaging. In: International Joint Conference on Neural Networks. pp. N–19221. Budapest, Hungary (2019), http://arxiv.org/abs/1904.06083

Sampat, M.P., Wang, Z., Gupta, S., Bovik, A.C., Markey, M.K.: Complex Wavelet Structural Similarity: A New Image Similarity Index. IEEE Transactions on Image Processing 18(11), 2385–2401 (nov 2009), http://ieeexplore.ieee.org/document/5109651/

Verwoert, M., Ottenhoff, M.C., Goulis, S., Colon, A.J., Wagner, L., Tousseyn, S., van Dijk, J.P., Kubben, P.L., Herff, C.: Dataset of Speech Production in intracranial Electroencephalography. Scientific Data 2022 9:1 9(1), 1–9 (jul 2022), https://www.nature.com/articles/s41597-022-01542-9

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (apr 2004), http://ieeexplore.ieee.org/document/1284395/

Wrench, A.A.: A Multichannel Articulatory Database and its Application for Automatic Speech Recognition. In: Proc. 5th Seminar on Speech Production: Models and Data. pp. 305–308. Kloster Seeon, Bavaria, Germany (2000), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.195.1846

Xu, K., Csapó, T.G., Roussel, P., Denby, B.: A comparative study on the contour tracking algorithms in ultrasound tongue images with automatic re-initialization. The Journal of the Acoustical Society of America 139(5), EL154–EL160 (may 2016), http://scitation.aip.org/content/asa/journal/jasa/139/5/10.1121/1.4951024

Magyarcentrikus többnyelvű gépifordító rendszerek létrehozása

Laki László János, Yang Zijian Győző

Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
{laki.laszlo, yang.zijian.gyozo}@nytud.hu

Kivonat Kutatásunkban magyarcentrikus többnyelvű gépifordító rendszereket hoztunk létre. Egy konkrét projekt folytatásaként terjesztettük ki a kétnyelvű kísérleteinket többnyelvű modellekre. A projekt célja az volt, hogy 12 különböző nyelvről lehessen magyarra fordítani lehetőleg egy neurális modell alkalmazásával. Kutatásunk során betanítottunk egy előtudás nélküli Marian NMT gépifordító rendszert, valamint sikerült finomhangolnunk az M2M100 és az NLLB-200 többnyelvű előtanított modelleket. Az általunk létrehozott modelleket végül egymással, valamint különböző ipari rendszerekkel hasonlítottuk össze. A saját teszt korpuszunkon az M2M100 finomhangolt modell érte el a legjobb eredményt. A előtanított modelljeinket nyelvenként csupán 3 millió szegmenspárral és 1 epoch értékkel hangoltuk, további tanítóadat vagy tanítási lépés hozzáadásával tovább növelhető a minőségük. Kutatásunkban bemutattuk, hogy az angolcentrikus többnyelvű modellek áthangolhatóak magyarcentrikusra.
Kulcsszavak: neurális gépi fordítás, Marian-NMT, M2M100, NLLB-200

1. Bevezetés

A gépi fordítás a nyelvtechnológia történetének kezdetétől az egyik legkutatottabb tudományterület. A mai napig nem megoldott feladat az emberi fordítóval összemérhető minőségű fordítórendszer létrehozása. A mesterséges neurális hálózat, azon belül is a transzformer (Vaswani és mtsai, 2017) architektúra megjelenésével egy újabb határozott lépéssel lettünk közelebb ehhez a célhoz. A számítási kapacitások növekedésével lehetőség nyílik a neurális modellek méretének drasztikus növelésére, ami a modell által megtanult tudás méretére és komplexitására van hatással. Ez a folyamat indikálta a többnyelvű modellek létrejöttét is. A megfelelő nagyságú paraméterszámú modellek képesek akár több, mint 200 különböző nyelv közötti fordítást elsajátítani. Így született meg az NLLB-200 (NLLB Team és mtsai, 2022) modell is. Azonban ezek a modellek erősen angolcentrikusak, vagyis első sorban a nagyobb nyelvekre, azon belül is az angol nyelvre teljesítenek rendkívül magas minőségben. Továbbá a közös fordítási modell alkalmazásának előnye, hogy transzfer tanulással a nyelvek képesek a többi nyelvből profitálni, ami a kevés erőforrású nyelvek esetén hasznos igazán.

Mivel ezek az előtanított modellek általában angolról, vagy angolra fordítanak ezért sokkal kisebb hangsúlyt fektetnek a többi nyelvek közötti fordításra. Másik probléma, hogy amíg az angol-magyar nyelvpárok esetén viszonylag sok tanítóanyag érhető el, addig ez a mennyiség jelentősen kevesebb a többi nyelv esetén. Kutatásunk elsődleges célja az volt, hogy megvizsgáljuk képesek vagyunk-e ezeket a igen nagy tudással rendelkező többnyelvű modelleket – mint M2M100 vagy NLLB-200 – adaptálni magyarcentrikusra. Továbbá megvizsgáltuk hogyan teljesít az a fordítórendszer, amit saját magunk nulláról építünk.

Kutatásunkban három magyarcentrikus többnyelvű modellt tanítottunk, amelyek 12 nyelvről képesek magyar nyelvre fordítani. Modelljeink megtalálhatóak a Hugging Face[1,2] oldalunkon.

2. Neurális gépifordító rendszerek

A cikkünkben összehasonlítottunk kutatásra szánt és ipari gépifordító rendszereket egyaránt. Az ipari termékek közül nem mindegyik rendszernél sikerült felderíteni a modell architektúráját és paramétereit. Ebben a fejezetben részletesen bemutatjuk a vizsgált rendszereket.

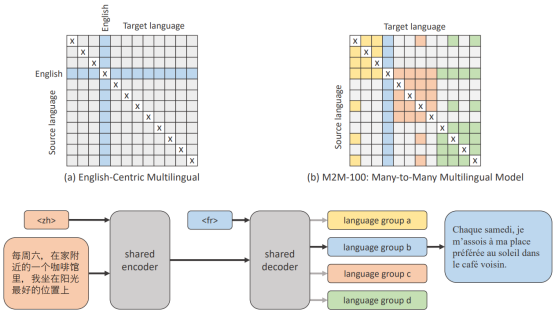
2.1. Marian NMT

A Marian NMT (Junczys-Dowmunt és mtsai, 2018) az egyik legnépszerűbb gépifordító keretrendszer, ami C++ nyelven íródott és szabadon hozzáférhető. Legnagyobb előnye, hogy memória- és erőforrás-optimális implementációjának köszönhetően rendkívül gyors. A Marian NMT egy transzformer (Vaswani és mtsai, 2017) modellel támogatott enkóder-dekóder architektúrájú neurális gépifordító modell. A Marian keretrendszer előtanított nyelvmodellek használata nélkül is gyors futási idejű tanítást eredményez. Kutatásunkban egy nagyméretű modellt tanítottunk: 6 réteg enkóder és 6 réteg dekóder; 16 figyelmi fej; 1024 szóbeágyazás dimenzió; 1024 bemeneti hossz; előre csatolt háló méret: 4096.

2.2. M2M100

Az M2M100 (Aharoni és mtsai, 2019) a Meta AI többnyelvű gépi fordítás projektje. A többnyelvű gépi fordítás (Multilingual Machine Translation) célja egy olyan modell létrehozása, amellyel sok nyelvről sok nyelvre képes fordítani. Az M2M100 projekt keretében elkészült egy 100 nyelvű tanítóanyag gépifordító modellek tanítására. A tanítóanyag erősen angolcentrikus. A kutatás egyik célja, hogy egyetlen modell tanításával a kevés erőforrással rendelkező nyelvpárok tanulni tudnak transzfer módon a többi nyelvpárok tudásaiból. Az M2M100 tartalmazza a magyar nyelvi tudást. Több méretben tanítottak M2M100 modellt, mi egy nagyobb modellel az M2M100 1.2B (1,2 milliárd paraméter) modellel kísérleteztünk.

[1] https://huggingface.co/NYTK/translation-m2m100-1.2B-multi12-hungarian
[2] https://huggingface.co/NYTK/translation-nllb-200-3.3B-multi12-hungarian



1. ábra: Az M2M100 architektúrája (Fan és mtsai, 2021)

2.3. NLLB-200

Az NLLB-200 (NLLB Team és mtsai, 2022) az M2M100 folytatása. Kutatásukban tovább növelték a nyelvek számát több mint 200 nyelvre. Létrehozták a Flores-200 gépi fordításra szánt adathalmazt, amin tanították a modelljeiket. A modell méretének és adatbázis méretének növelése mellett az architektúra módosításával is tettek kísérletet. Betanítottak egy 54,5 milliárd paraméteres modellt, ami egy MoE (Sparsely Gated Mixture of Experts) réteggel növeli a modell hatékonyságát. A transzformer architektúrában az előrecsatolt hálózatot cserélik le MoE rétegre. Céljuk, hogy a nagy mennyiségű nyelvpárok közötti interferenciát csökkentsék, amellyel a különböző erőforrást igénylő nyelvpárok számára egyformán tegyék előnyössé a tanítás. Az NLLB modell egyik legnagyobb előnye, hogy a kis erőforrású nyelvek számára is megfelelő minőségű fordítást eredményez. A modell természetesen tartalmazza a magyar nyelvi tudást, kísérletünkben az egyik legnagyobb modellel, a NLLB-200 3.3B (3,3 milliárd paraméter) modellel kísérleteztünk.

2.4. Google Fordító

A Google Fordító (Wu és mtsai, 2016) (Google Translate) napjaink egyik legnépszerűbb szabadon elérhető többnyelvű gépifordító rendszere. A Google Fordító a kezdetben (2003-ban) még a statisztikai gépi fordítás módszert alkalmazta, azonban 2016-ban felváltotta a neurális hálózaton alapuló megoldás. A neurális hálón alapuló módszer lényegesen javított a fordítás minőségén. A Google Fordító az idők során sokat fejlődött. Jelenleg több lefordított kimenetet is ad, így például az angolban nem specifikált, de más nyelvben (például franciában vagy spanyolban) jelölt nemek különböző verzióiból (Rescigno és mtsai, 2020) tud a felhasználó választani. A Google Fordító jelenleg 132 különböző nyelvet képes kezelni.

2.5. Microsoft fordító

A Microsoft fordító (Microsoft Translator) a Bing fordító utódja, amely a Micorsoft Cognitive Services termékcsalád része. Jelenleg 112 különböző nyelven

tud fordítani. 2018 előtt hasonlóan a Google Fordítóhoz, statisztikai gépifordító rendszer volt. Azóta átálltak transzformer alapú neurális gépi fordításra, számtalan kutatást végeznek, hogy növeljék a fordítás minőségét. Többek között tudás desztillációval (knowledge distillation) (Bucila és mtsai, 2006) és többnyelvű modellekkel (Tan és mtsai, 2019) kísérleteznek. Az eTranslation rendszernek több modellje is van, általános és specifikusabb (például jogi, pályázati vagy orvosi) egyaránt. Több mint 1 milliárd szegmensen tanították a modelleket, összesen 24 nyelven tud fordítani. Tudomásunk szerint a modellek kétnyelvű modellek.

3. Mérések

3.1. Felhasznált korpusz

A magyarcentrikus többnyelvű modellek tanításához párhuzamos korpuszokat kellett létrehozni magyar célnyelvvel. A kutatásunkban 12 különböző forrásnyelvvel dolgoztunk, így a párhuzamos korpuszok forrásnyelvei a következőek voltak: bolgár (bg), cseh (cs), német (de), angol (en), horvát (hr), lengyel (pl), román (ro), orosz (ru), szlovák (sk), szlovén (sl), szerb (sr), ukrán (uk).

A nulláról való gépifordító modell tanításához nagy mennyiségű párhuzamos korpusz szükséges. Ehhez a feladathoz olyan párhuzamos korpuszt építettünk, amely 8 millió (8M) szegmenst tartalmaz nyelvenként (kivétel az ukrán, ahol adat hiányában összesen csak 5.805.144 szegmenst tartalmaz a korpuszunk). A finomhangoláshoz kevesebb adat is elégséges, ezért létrehoztunk a 8M korpuszból egy részkorpuszt, amely 3 millió szegmenst (3M) tartalmaz nyelvenként. A párhuzamos mondatokat az OPUS Tiedemann (2012) adatbázisból gyűjtöttük, amely az alábbi részkorpuszokat tartalmazza: Bible, Bible-uedin, Books, CCAligned, CCMatrix, DGT, ECB, ELITR, ELITR-ECA, ELRC_2922, ELRC_2923, ELRC_3382, EMEA, EUbookshop, EUconst, Europarl, GNOME, GlobalVoices, JRC, JRC-Acquis, KDE4, KDEdoc, MultiCCAligned, MultiParaCrawl, OpenSubtitles, PHP, ParaCrawl, QED, TED2020, Tatoeba, TildeMODEL, Ubuntu, WMT-News, WikiMatrix, Wikimedia, Wikipedia, XLEnt.

A különböző nyelvpárok különböző összeállítást tartalmaznak a részkorpuszokból.

Az 1. táblázatban láthatóak a korpuszok (8M és 3M) fontosabb tulajdonságai nyelvekre bontva.

3.2. Felhasznált modellek és gépifordító rendszerek

Kutatásunk során betanítottunk egy saját **Marian NMT** big modellt a 8M korpuszon. A tanításhoz a keretrendszer által alapértelmezetten biztosított paraméter beállításokat alkalmaztuk[3]. „Subword” tokenizáláshoz a beépített Sentence Piece (Kudo és Richardson, 2018) tokenizálót használtuk. A szótér mérete: 32.000; 66 epoch értékig tanítottuk. A nyelvek megkülönböztetése végett a forrásnyelvi szövegek elé beillesztettük az ISO szabványú nyelvkódot: __lang__.

[3] https://github.com/marian-nmt/marian-dev/blob/master/src/common/aliases.cpp

	Token		Type		token/sent	
	8M	3M	8M	3M	8M	3M
bg	101 701 016	38 149 260	998 060	586 926	12,71	12,72
hu	93 370 875	35 023 413	1 843 452	1 057 434	11,67	11,68
cs	96 854 637	36 345 169	1 369 081	797 557	12,11	12,12
hu	96 313 811	36 125 748	2 008 769	1 141 009	12,04	12,04
de	123 826 131	46 407 141	1 708 615	957 634	15,48	15,47
hu	113 026 306	42 365 265	2 215 093	1 267 205	14,13	14,12
en	118 593 896	44 440 629	1 112 914	593 035	14,82	14,81
hu	104 287 145	39 072 921	2 375 910	1 331 924	13,04	13,02
hr	78 932 860	29 601 947	1 075 070	631 246	9,87	9,87
hu	78 540 254	29 445 821	1 685 025	961 367	9,82	9,82
pl	97 533 671	36 584 480	1 350 775	793 299	12,19	12,20
hu	98 984 434	37 126 013	2 062 157	1 166 764	12,37	12,38
ro	110 276 300	41 357 056	952 906	555 642	13,79	13,79
hu	93 431 714	35 058 265	1 906 878	1 091 748	11,68	11,69
ru	88 227 629	33 085 548	1 376 699	807 518	11,03	11,03
hu	85 205 960	31 956 481	1 838 741	1 049 578	10,65	10,65
sk	122 935 150	46 085 577	1 567 148	920 586	15,37	15,36
hu	123 016 834	46 105 105	2 225 916	1 278 686	15,38	15,37
sl	106 838 393	40 042 349	1 195 476	703 052	13,36	13,35
hu	106 714 770	40 013 573	1 973 244	1 138 862	13,34	13,34
sr	72 647 210	27 237 077	1 185 523	710 495	9,08	9,08
hu	71 058 803	26 642 218	1 446 568	832 887	8,88	8,88
uk	70 816 656	36 581 363	1 306 774	927 544	12,20	12,19
hu	69 564 268	35 933 267	1 556 554	1 088 340	11,98	11,98

1. táblázat. A korpusz tulajdonságai

Továbbá a kutatásunkhoz kipróbáltunk különböző kutatásra és ipari alkalmazásra szánt gépifordító rendszereket és modelleket, amelyek képesek angolról magyarra fordítani. Kísérleteinkben az alábbi rendszereket, modelleket próbáltuk ki:

- **M2M100**: Két modelljük is elérhető, egy kisebb (418M) és egy nagyobb (1.2B) modell. A fordításhoz a Hugging Face Transformers könyvtár M2M100-Tokenizer és M2M100ForConditionalGeneration függvényeit használtuk. Kutatásunkban a nagyobb 1.2B modellt használtuk, amely 1,2 milliárd paraméteres.
- **NLLB-200** (NLLB): Több modelljük is elérhető, egy kisebb (1.3B - 1,3 milliárd) és egy nagyobb (3.3B - 3,3 milliárd) modell, valamint ennek a két modellnek a desztillált változatai. A fordításhoz a Hugging Face Transformers könyvtár M2M100Tokenizer és M2M100ForConditionalGeneration függvényeit használtuk. Kutatásunkban a nagyobb 3.3B modellt használtuk, amely 1,2 milliárd paraméteres.

- **eTranslation**: Akadémia számára ingyenesen elérhető szolgáltatás. Regisztráció után beküldtük a teszt fájlunkat, amelynek a fordítását e-mailben kaptuk meg.
- **Google**: Az online webhelyfordító funkcióját használtuk, a mondatokat betettük egy weboldalra sortörésekkel, majd az oldal címét megadva a Google szolgáltatás lefordította.
- **Microsoft**: Az Azure Translator szolgáltatást használtuk, amely elérhető a Microsoft Word szoftverből.

Végül finomhangoltunk kettő kutatásra szánt többnyelvű gépifordító modellt magyarcentrikusra. Az alábbi kettő modellt finomhangoltuk a Hugging Face által implementált gépifordító szkriptjével[4]:

- **M2MF**: M2M100 1.2B modelljét finomhangoltuk a 3M korpuszon a következő hyper-paraméterekkel: 5e-5; batch méret: 16; forrás- és célnyelvi szöveg hossza: 256; epoch: 1.
- **NLLBF**: NLLB-200 3.3B modelljét finomhangoltuk a 3M korpuszon a következő hyper-paraméterekkel: 5e-5; batch méret: 3; forrás- és célnyelvi szöveg hossza: 256; epoch: 1; gradient accumulation steps: 4.

A modelleket 8 darab NVIDIA A100 (80GB) videokártyán tanítottuk. A tanító szkriptet kellett egy kicsit módosítani, mivel az szkript eredetileg egy nyelvpárral dolgozik, azonban a mi korpuszunk 12 nyelvpár szegmenseit tartalmazza véletlen sorrendben. Az előfeldolgozó függvényt kellett átírni úgy, hogy az aktuális szegmensből olvassa ki a nyelvkódot, majd annak megfelelően inicializálni a tokenizálót, mivel a tokenizáló az adott nyelv nyelvkódját elhelyezi a megfelelő helyre a szövegbe.

4. Eredmények

A különböző modellek, rendszerek kiértékelésére a SacreBLEU (Papineni és mtsai, 2002; Post, 2018) és a chrF (Popović, 2015) metrikákat használtuk. A BLEU metrika mellett azért választottuk a chrF metrikát, mert az karakteralapú, ami a ragozó nyelvek esetében, mint magyar nyelv, pontosabb kiértékelést eredményez. A chrF kiértékelésnél a 3-gram értékeket mértük a (Laki és Yang, 2022) kutatása alapján.

A 2. táblázatbal láthatóak a modellek eredményei a saját teszt korpuszunkon mérve. Az utolsó sor az átlagot (átl.) mutatja. Az eredményeinkben látható, hogy az M2MF modell érte el a legjobb eredményt, felülmúlva az összes többi modellt, minden nyelven. Ezzel bebizonyítottuk, hogy a finomhangolással kevesebb erőforrásból (csupán 1 epochal és 3 millió per nyelvvel) jobb eredményt tudtunk elérni, mint a Marian NMT, ami 66 epochig tanult. Az NLLB bár nagyobb modell, de a kétszer annyi nyelvi tudása miatt kevésbé koncentráltabb a korpusz,

[4] https://github.com/huggingface/transformers/tree/main/examples/pytorch/translation

így a magyar nyelvű fordítás minősége is gyengébb. Ennek következménye, hogy az 1 epochig tanított NLLBF minősége gyengébb, mint az M2M100F.

Várt eredmény, hogy az általunk tanított modellek a saját korpuszunkon jobban teljesítettek, mint az ipari alkalmazások, mint a Google Fordító vagy a Microsoft fordító.

	Marian	**M2MF**	NLLBF	M2M100	NLLB	Google	Microsoft	eTrans
bg	21,3/49,4	**26,6/52,9**	25,4/52,1	20,7/47,8	17,1/45,1	20,0/48,6	20,8/49,3	22,3/50,5
cs	22,5/51,3	**28,9/55,1**	27,4/53,8	23,5/50,9	19,0/47,5	22,6/50,3	23,1/50,9	24,7/52,3
de	21,9/51,5	**28,3/56,1**	27,0/54,9	24,4/53,5	19,6/49,3	22,7/52,9	22,8/52,8	24,0/53,6
en	27,7/54,7	**34,4/59,2**	33,2/58,6	21,6/50,9	23,4/52,7	25,3/54,1	26,3/50,3	28,3/56,0
hr	19,2/48,2	**26,2/52,2**	25,2/51,3	20,0/47,7	15,5/42,8	19,6/47,8	20,1/48,5	20,9/48,9
pl	21,2/50,8	**28,3/55,0**	27,2/53,6	22,7/50,4	18,6/47,1	21,4/50,5	22,2/50,8	23,9/52,2
ro	19,5/49,6	**26,4/53,8**	25,6/53,2	21,8/50,4	18,3/47,1	21,0/50,2	21,8/51,0	23,6/51,9
ru	19,7/49,7	**25,1/53,0**	23,2/51,7	20,2/49,8	17,4/47,1	19,8/49,8	21,0/51,0	20,3/50,2
sk	23,1/54,2	**30,9/58,5**	28,0/57,1	25,6/55,0	18,6/49,1	22,6/52,6	23,1/53,5	26,4/55,5
sl	22,7/51,1	**27,7/55,1**	26,3/53,8	24,5/51,7	19,8/47,1	14,4/40,3	21,5/50,0	26,0/53,1
sr	18,0/46,3	**23,4/50,0**	21,7/49,1	17,6/44,7	16,2/44,2	18,0/46,3	19,2/47,1	-/-
uk	24,2/55,4	**32,6/59,9**	30,3/58,6	25,3/54,4	18,7/48,7	21,8/52,3	23,3/53,6	22,9/53,6
átl.	21,8/51,0	**28,2/55,1**	26,7/54,0	22,3/50,6	18,5/47,3	20,8/49,6	22,1/50,7	23,9/52,5

2. táblázat. gépifordító rendszerek eredményei a saját korpuszunkon (BLEU/chrF++)

Az eredményünk validálása érdekében kimértük modelljeink teljesítményét egy külső korpuszon, a Flores 101 (Goyal és mtsai, 2022) korpuszon. A kiértékeléshez a hivatalos Flores 101 devtest[5] adathalmazt használtuk. Várakozásunknak megfelelően idegen korpuszon már gyengébben teljesítettek a modelljeink. Ez a mérés reálisabban árnyékolja a modelljeink teljesítményét. Modelljeink közül továbbra is az M2MF teljesített a legjobban és az eTranslation modelljeinél továbbra is magasabb eredményt ért el. Ez azért fontos eredmény, mert az eTranslate kétnyelvű modelleket alkalmaz, míg az M2MF egyetlen egy többnyelvű modellel ért el jobb eredményt. Összességében a Google és Microsoft modelljeiket nem sikerült felülmúlni, azonban lengyel (pl), szlovén (sl) és szerb (sr) nyelveken sikerült a Microsoft gépifordító rendszerét felülmúlni. Az eredmény azért sem meglepő, hiszen kevés tudásunk van arról, hogy a Google vagy a Microsoft pontosan milyen adatokat használt fel a tanítás során, de az biztos, hogy óriási mennyiségű adatot.

A 4. táblázatban mutatunk egy angol-magyar példát. Az első blokkban áll a forrásnyelvi (angol) mondat és a hozzá tartozó referencia (magyar). A második blokkban található az ipari alkalmazások fordításai. A harmadik blokkban szereplnek az M2M100 és az NLLB-200 modellek fordításai. Végül az utolsó blokkban láthatóak a kutatásunkban tanított és finomhangolt modellek kimenetei.

[5] https://www.statmt.org/wmt21/large-scale-multilingual-translation-task.html

	Marian	M2MF	NLLBF	M2M100	NLLB	Google	Microsoft	eTrans
bg	12,1/50,5	21,6/56,1	20,8/55,1	20,5/55,0	17,5/52,0	**24,1/57,9**	21,9/56,5	21,2/55,8
cs	9,9/45,8	22,8/56,8	21,3/55,3	21,1/55,0	17,3/51,1	**23,7/57,5**	23.4/57,6	21,4/55,5
de	7,9/42,8	23,0/57,0	21,8/55,9	21,8/55,8	18,9/53,1	**23,7/57,6**	23,9/58,4	22,3/56,3
en	11,9/46,9	27,0/60,3	25,5/59,0	25,9/59,0	24,2/57,7	29,5/62,4	**30,2/63,2**	27,7/61,0
hr	11,2/49,0	21,7/55,8	20,7/54,6	20,4/54,7	17,4/51,2	**23,0/56,7**	21,2/55,7	20,2/54,7
pl	8,8/46,2	18,3/52,8	17,5/52,0	16,6/51,0	14,7/48,7	**18,4/53,1**	17,9/53,4	16,6/51,7
ro	7,7/43,9	22,7/57,0	21,4/55,7	21,4/55,8	18,6/52,6	**24,8/58,3**	23,3/58,1	22,2/56,4
ru	12,2/50,1	20,0/54,6	19,1/53,2	18,6/53,0	16,5/50,8	**21,5/55,5**	20,4/55,3	19,1/53,6
sk	11,8/50,3	22,0/56,5	21,2/55,5	20,4/54,8	17,5/51,4	**23,4/57,1**	22.2/57,1	20,4/54,9
sl	8,9/46,4	20,6/55,4	19,6/54,0	19,6/54,0	16,2/50,9	**21,8/56,1**	20,1/54,7	19,3/54,2
sr	12,4/50,1	22,4/57,3	21,2/56,0	21,4/56,1	18,4/52,3	**24,7/58,5**	22,4/57,2	-/-
uk	12,0/50,0	21,6/55,6	20,6/54,7	19,4/53,7	17,3/51,5	**23,1/56,8**	22.2/56,4	20,1/54,7
átl.	10,6/47,7	22,0/56,3	20,9/55,1	20,6/54,8	17,9/51,9	**23,5/57,3**	22,4/57,0	21,0/55,4

3. táblázat. gépifordító rendszerek eredményei a Flores 101 korpuszon (BLEU/chrF++)

forrás	This may not make much sense to you, sir, but I'd like to ask your permission to date your daughter.
referencia	Szeretném megragadni az alkalmat uram, hogy az engedélyét kérjem, hogy találkozhassak a lányával.
Google	Lehet, hogy ennek nincs sok értelme, uram, de szeretném az engedélyét kérni, hogy randevúzhassak a lányával.
Microsoft	Lehet, hogy ennek nincs sok értelme számodra, uram, de szeretném engedélyt kérni a lányoddal való randevúzáshoz.
eTranslate	Ennek talán nincs sok értelme, uram, de szeretném kérni az engedélyét, hogy randizzon a lányával.
M2M100	Lehet, hogy ez nem sok értelme neked, uram, de szeretném kérni az engedélyét, hogy dátumot a lányod.
NLLB	Lehet, hogy önnek nincs sok értelme, uram, de szeretném kérni az engedélyét, hogy randizzak a lányával.
M2MF	Talán nem sok értelme van, uram, de szeretném megkérni az engedélyét, hogy randizhassak a lányával.
NLLBF	Lehet, hogy nem sok értelme van, uram, de szeretném megkérni az engedélyét, hogy randizzak a lányával.
Marian	Lehet, hogy nem sok értelme van ennek, uram, de szeretném megkérni az engedélyét, hogy adja meg a lányának.

4. táblázat. Egy angol-magyar példa a fordító rendszerek összehasonlítására

Az első észrevétel, hogy már a referencia sem pontos. Továbbá a Google és a Microsoft is jó fordításokat eredményeztek, bár a megfogalmazás lehet egy kicsit szokatlan. Az eTranslate szemantikai hibát vétett. Az eredeti M2M100 a *date* szót rosszul fordította, így a mondat vége is értelmetlenné vált, de a mondat eleje sem helyes. Az NLLB által fordított mondat eleje hiányos és kétértelmű lehet. Ezekhez képest a finomhangolt modellek kimenetei helyesek egyedül a *megkérni* szó lehet szokatlan. A Marian által fordított mondat vége nem helyes. A példa jól tükrözi a számokkal kimért minőségi különbségeket.

hu	A kutatók azt vetették fel, hogy bár ez egy fiatal dinoszaurusz farka, a minta felnőtt tollazatot, nem pedig fióka tollazatot sejtet.
bg	Изследователите предполагат, че макар това да е опашката на млад динозавър, пробата показва оперение на възрастен, а не пух на млад индивид. A kutatók azt sugallják, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt tollát mutatja, nem pedig egy fiatal egyén tollát.
cs	Vědci předpokládají, že i když se jedná o ocas mladého dinosaura, představuje vzorek opeření dospělého zvířete a nikoli prachové peří mláděte. A tudósok feltételezik, hogy bár egy fiatal dinoszaurusz farkáról van szó, a minta egy felnőtt állat, nem pedig egy kölyök poros tollát reprezentálja.
de	Die Forscher behaupteten, dass, obwohl dies der Schwanz eines jungen Dinosauriers ist, die Probe das Gefieder eines Erwachsenen und keine Kükendaunen zeigt. 'A kutatók azt állították, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt tollát mutatja, és nem csirke sebeket.
en	The researchers suggested that, even though this is the tail of a young dinosaur, the sample shows adult plumage and not a chick's down. 'A kutatók azt javasolták, hogy bár ez egy fiatal dinoszaurusz farka, a minta felnőtt szilárdságot mutat, és nem egy csaj le.
hr	Istraživanje je pokazalo da unatoč tome što se radi o repu mladog dinosaura, uzorak je dio perja boje odrasle ptice, a ne paperja pilića. A tanulmány kimutatta, hogy annak ellenére, hogy ez egy fiatal dinoszaurusz farka, a minta egy felnőtt madár színű toll része, nem pedig egy csirke paprika.
pl	Badacze zasugerowali, że mimo iż jest to ogon młodego dinozaura, upierzenie próbki wskazuje na dorosłego osobnika, a nie na pisklę. A kutatók azt javasolták, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt példányra utal, nem pedig egy csajra.
ro	Cercetătorii au sugerat că, deși aceasta este coada unui dinozaur tânăr, eșantionul arată un penaj adult și nu puful unui pui. A kutatók azt javasolták, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt kopoltyút mutat, nem pedig egy csirke bolháját.
ru	Исследователи предположили, что, несмотря на то, что это хвост молодого динозавра, данный образец имеет оперение взрослой особи, а не пух цыплёнка. A kutatók azt feltételezték, hogy annak ellenére, hogy ez egy fiatal dinoszaurusz farka, ez a példány egy felnőtt példány tollával rendelkezik, nem pedig egy csirke tollával.
sk	Aj keď ide o chvost mladého dinosaura, na vzorke je podľa vedcov perie dospelého jedinca a nie páperie mláďaťa. Bár egy fiatal dinoszaurusz farkáról van szó, a tudósok szerint a mintán egy felnőtt egyedek tollai vannak, nem pedig egy kölyök tollai.
sl	Čeprav gre za rep mladega dinozavra, pa po mnenju raziskovalcev vzorec kaže, da gre za perje odraslega primerka in ne puh mladiča. Bár egy fiatal dinoszaurusz farkáról van szó, a kutatók szerint a minta azt mutatja, hogy egy felnőtt példány tolláról van szó, nem pedig egy bolyhos kölyökről.
sr	Истраживачи су навели да, иако је ово реп младог диносауруса, узорак показује перје одрасле јединке а не паперје птића. A kutatók megjegyezték, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt egyedek tollát mutatja, nem pedig egy madár papírjait.
uk	Дослідники припустили, що, хоча це хвіст молодого динозавра, зразок демонструє доросле оперення, а не пух дитинчати. A kutatók azt feltételezték, hogy bár ez egy fiatal dinoszaurusz farka, a minta egy felnőtt tollat mutat, nem pedig egy kisgyermek szőrét.

5. táblázat. Egy többnyelvű példa az M2MF fordítására

A 5. táblázatban látható egy-egy példa az M2MF különböző nyelvekre való fordításairól. Az első blokkban található a referencia fordítás. Utána minden blokkban először a forrásnyelvi szöveg követi, utána az M2MF által fordított szöveg. Látható, hogy a modell bizonyos nyelvekre (pl.: bg, cs, sk, sl) egész jól fordít és bizonyos nyelvekre pedig kevésbé (pl.: en, de, ro, sr). Ez összefüggésben lehet a tanítóanyag minőségével. Továbbá a példa alapján a 'fióka' szó nagyban nehezítette a fordítást. A különböző nyelveken máshogy próbálták kifejezni ezt a szót, ami össze is zavarta a fordító modellünket.

A példákból egyértelműen az látszik, hogy nem kiegyensúlyozott a minőség a különböző nyelvekre. Következő feladatként azokra a nyelvekre kell további minőségjavítási módszereket alkalmazni, amelyekre gyengébben teljesít a rendszer. A módszerek lehetnek adattisztítási, és/vagy adatbővítési eljárások egyaránt.

Végül, de nem utolsó sorban a finomhangolt modelleinket egy olyan korpuszon tanítottuk, ami mindössze 3 millió szegmenst tartalmaz nyelvpáronként. Ezen a területen bőven van még terünk a fejlődésre.

5. Összegzés

Kutatásunk során 12 nyelvű magyarcentrikus gépifordító modelleket tanítottunk. Kísérleteinkben betanítottunk egy Marian NMT modellt a nulláról, valamint az M2M100 és az NLLB-200 modelleket finomhangoltuk. Méréseinkben kiértékeltük az általunk felhasznált modelleket eredetileg publikált változatukban, valamint összehasonlítottuk ipari alkalmazásokkal, mint a Google vagy a Microsoft fordító modelljével. Kutatásunkkal megmutattuk, hogy finomhangolással magyarcentrikussá tudjuk adaptálni a többnyelvű modelleket. Sőt ezzel a módszerrel, sokkal kevesebb erőforrással tudtunk jobb eredményt elérni, mint a nulláról tanított Marian gépifordító modell. Továbblépésként mind az M2M100, mind az NLLB-200 modellt szeretnénk még több forrásnyelvvel megtámogatni.

Hivatkozások

Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3874–3884. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Bucila, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 535–541. KDD '06, Association for Computing Machinery, New York, NY, USA (2006)

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Auli, M., Joulin, A.: Beyond english-centric multilingual machine translation. Journal of Machine Learning Research 22(107), 1–48 (2021), http://jmlr.org/papers/v22/20-1307.html

Goyal, N., Gao, C., Chaudhary, V., Chen, P.J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., Fan, A.: The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. Transactions of the Association for Computational Linguistics 10 (2022)

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018)

Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018)

Laki, L., Yang, Z.G.: Jobban fordítunk magyarra, mint a Google! In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 357–372. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

NLLB Team, Costa-jussà, M.R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G.M., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K.R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N.F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H., Wang, J.: No language left behind: Scaling human-centered machine translation (2022)

Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002), https://aclanthology.org/P02-1040

Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation. pp. 392–395. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015), https://aclanthology.org/W15-3049

Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers. pp. 186–191. Association for Computational Linguistics, Brussels, Belgium (Oct 2018)

Rescigno, A.A., Monti, J., Way, A., Vanmassenhove, E.: A case study of natural gender phenomena in translation: A comparison of Google Translate, bing Microsoft translator and DeepL for English to Italian, French and Spanish. In: Workshop on the Impact of Machine Translation (iMpacT 2020). pp. 62–90. Association for Machine Translation in the Americas, Virtual (Oct 2020)

Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., Liu, T.Y.: Multilingual neural machine translation with knowledge distillation. In: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019). New Orleans, LA, USA (2019)

Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR abs/1609.08144 (2016)

Többnyelvű modellek és PEGASUS finomhangolása magyar nyelvű absztraktív összefoglalás feladatára

Yang Zijian Győző

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

Kivonat Napjaink egyik legfontosabb és legkutatottabb nyelvtechnológiai területe az absztraktív szövegösszefoglaló készítése. Mind a kutatásban, mind az iparban egyre nagyobb igény keletkezik a feladat megoldására. Az elmúlt években magyar nyelven is elindultak a kutatások ezen a területen, voltak különböző kísérletek magyar és többnyelvű előtanított neurális nyelvmodellek finomhangolásával. Jelen kutatásomban elsősorban a többnyelvű modellek finomhangolására tettem a hangsúlyt. Arra kerestem a választ, hogy a más nyelvekre, akár feladatokra előtanított modellek hogyan teljesítenek magyar nyelvre, illetve azok a többnyelvű modellek, amelyek angol vagy más nyelven a legjobb eredményt érték el absztraktív összefoglalás területén, adaptálhatóak-e magyar nyelvre. A kísérletem kiterjedt a manapság rendkívül népszerű mT5-re, a magyar nyelvi előtudással nem rendelkező mBART modellre és az M2M100 gépi fordítás feladatára előtanított 100 nyelvű neurális modellre. Az utóbbi két modell esetén a kérdés, hogy egy modell, amely nem rendelkezik magyar tudással a finomhangolás során meg tud-e tanulni magyarul megoldani egy feladatot, illetve, bár rendelkezik magyar tudással, de gépi fordításra tanított modell módosítható-e absztraktív összefoglaló generálás feladatára. Végül, de nem utolsó sorban, az angol nyelvre egyik legjobban teljesítő PEGASUS modellt finomhangoltam magyar absztraktív összefoglaló feladatra. Ezzel a kutatással kísérletet tettem egy angol nyelvű modellt magyar nyelvre adaptálni és arra kerestem a választ, hogy vajon ez lehetséges-e és van-e értelme. Eredményeim azt mutatják, hogy mindegyik modell finomhangolható és adaptálható magyar nyelvre, sőt az mT5 és az mBART esetében sikerült felülmúlni az eddigi legjobban teljesítő magyar BART modellt.
Kulcsszavak: absztraktív összefoglalás, mT5, mBART, M2M100, PEGASUS

1. Bevezetés

Napjaink egyik legnépszerűbb nyelvtechnológiai feladata a hosszú szövegek összefoglalása. Mind a kutatásban, mint az iparban kiemelt terület. Külön cégek[1]

[1] https://www.getabstract.com

alakulnak a feladat megoldására. Kétféle szövegösszefoglaló módszer létezik: extraktív és absztraktív. Az extraktív összefoglalás esetében a feladat megjelölni, kiemelni a szövegben lévő fontosnak ítélt szövegrészeket. Az absztraktív összefoglalás esetében viszont a bemeneti szöveg alapján egy új szöveg megfogalmazása a cél, amely tartalmazza a bemeneti szöveg tartalmi lényegét. Mindkettő feladat nehéz, mivel emberek között sincsen minden esetben teljes egyetértés abban, hogy mi a fontos egy szövegben, vagy mi a legfontosabb mondanivalója egy szövegnek. A feladatot tovább nehezíti, hogy az is számít milyen célra készül az összefoglaló. Egy marketing célra használt, figyelemfelkeltő összefoglaló mást tart fontosnak, mint egy szakmai összefoglaló. Mindezen szempontok mellett az absztraktív összefoglaló feladata annyiban nehezedik, hogy a meglévő szöveg alapján egy összefüggő, logikusan felépített, nyelvtanilag helyes szöveget kell generálnia. Azonban egy jól működő automatikus absztraktív összefoglaló szoftver rengeteg időt és energiát megspórolhat egy cég számára.

Kutatásomban első sorban az absztraktív összefoglalás módszerére tettem a hangsúlyt. Kísérleteimben a témában népszerű mT5 és mBART többnyelvű modelleket finomhangoltam magyar nyelvre. A feladat érdekessége, hogy az mBART modell nem rendelkezik magyar tudással, azonban a szóelem tokenizálásnak köszönhetően a modell könnyen be tudja olvasni a magyar szöveget is. Emellett kísérletet tettem egy nem összefoglalásra szánt modell finomhangolására. Kutatásomban egy gépi fordító modellt, az M2M100 modellt próbáltam ki. Az M2M100 rendelkezik magyar tudással, azonban gépi fordítás feladatára tanították. Kutatásomban arra kerestem a választ, hogy kell-e magyarul tudnia egy modellnek, hogy magyar feladatot oldjon meg, illetve számít-e a feladat, vagy inkább a modell architektúrája a fontosabb. Végül, de nem utolsó sorban kísérletemben azt kutattam, hogy egy angol nyelvre előtanított modell, a PEGASUS, tovább finomhangolható-e magyar nyelvű feladatra, illetve milyen módszereket kell alkalmazni, hogy egyáltalán képes legyen valamennyire magyarul megtanulni.

Modelljeim elérhetőek a Hugging Face oldalunkon[2]: NYTK/summarization-hi-mbart-large-50-hungarian, NYTK/summarization-hi-mt5-base-hungarian és NYTK/summarization-hi-pegasus-hungarian.

2. Kapcsolódó irodalom

Az utóbbi években mind a nyelvtechnológiában, mind az absztraktív összefoglalás területén, a transzformer (Vaswani és mtsai, 2017) architektúra hozta a nagy áttörést. A transzformer architektúra megjelenésével egy időben a nyelvtechnológiai feladatok folyamata is megváltozott. Az új irány a kétlépcsős tanítás lett. Első lépésként egy általános nyelvmodellt tanítanak, amellyel a modellt felruházzák általános nyelvi tudással, ez az előtanítás folyamata, majd második lépésként az előtanított nyelvmodellt tovább tanítják, vagy másnéven finomhangolják egy adott konkrét feladatra. Bizonyos esetekben, hogy jobb eredményt érjenek el,

[2] https://huggingface.co/NYTK

az előtanítás folyamata során olyan feladatokkal tanítják a modellt, hogy azt könnyebb legyen egy adott feladatra, például összefoglaló generálásra, tanítani.

A szövegösszefoglaló generálás egy szöveggenerálási feladat, ami megkövetel a modelltől egy generálásra alkalmas architektúrát, amit dekódernek hívnak. A csak dekóderrel rendelkező modellek, mint például a GPT (Radford és Narasimhan, 2018; Radford és mtsai, 2019; Yang, 2022a), bár képesek szövegösszefoglalásra, nem teljesítenek olyan jól, mint az enkóder-dekóder architektúrájú modellek. Az első sikeres próbálkozások is enkóder-dekóder architektúrával történtek. 2019-ben a PreSumm (Liu és Lapata, 2019) módszerrel érték el a legjobb eredményt. A PreSumm módszer lényege, hogy egy előtanított BERT modellhez (Devlin és mtsai, 2019), csatlakoztatnak egy üres dekódert, amelyet finomhangolnak. Magyar nyelvre is ezzel a módszerrel érték el az első sikereket (Yang és mtsai, 2021), ahol enkódernek a huBERT (Nemeskey, 2021) modellel sikerült a legjobb eredményeket elérni.

Későbbiekben olyan autoregresszív modellekkel érték el a legjobb eredményeket, ahol már az előtanítás során is egy enkóder-dekóder architektúrát alkalmaztak. Az egyik legjobb eredményt elért modell 2020-ban a BART (Lewis és mtsai, 2020) modell volt, ahol az előtanítást úgy végezték el, hogy az az összefoglaló generálás finomhangolásának kedvezzen. Ehhez hasonló a PEGASUS is, ami csak az összefoglaló generálás feladatára alkalmas és több versenyben is a legjobb eredményt produkálta (Zhang és mtsai, 2020).

A BART angol nyelvű modell, ezért, hogy kiterjesszék a módszer előnyeit más nyelvekre is, elkészítették először az mBART (Liu és mtsai, 2020), majd az mBART-50 (Tang és mtsai, 2020) modelleket. A sima mBART 25, míg az mBART 50 nyelv tudását tartalmazza, azonban nincsen közöttük a magyar nyelv.

2021-ben Hasan és mtsai (2021) létrehozták az XL-Sum összefoglaló generálásra szánt korpuszt, ami 44 nyelvet tartalmaz. A korpusz közel egymillió BBC cikket és azok leadjeit tartalmazza. A korpusz segítségével a kevés forrással rendelkező arab nyelvekre sikerült jó minőségű szövegösszefoglaló modelleket tanítaniuk. A magyar nyelv azonban nincsen közöttük.

Pfeiffer és mtsai (2020) többnyelvű transzfer (cross-lingual transfer) tanuláson alapuló módszerekkel kísérleteztek többnyelvű modellek segítségével (Pfeiffer és mtsai, 2021, 2022).

Magyar nyelvre transzformer alapú neurális összefoglaló modelleket először Yang és mtsai (2021) készítettek a PreSumm módszerrel. Később Yang (2022b) végzett kísérleteket magyar nyelvű BART és GPT-2 (Yang, 2022a) modellek előállítására, amelyeket absztraktív összefoglalás feladatára is finomhangolta. Emellett Makrai és mtsai (2022) végzett sikeres kísérleteket huBERT alapú összefoglaló modellek tanítására.

Kutatásomban többnyelvű modellekkel és az angol nyelvű PEGASUS modellel kísérleteztem. Az mT5, az mBART, az M2M100 és a PEGASUS modellt finomhangoltam magyar nyelvű absztraktív szövegösszefoglaló generálás feladatára.

3. Korpusz és modellek bemutatása

A modellek tanításához a HI (Yang és mtsai, 2021; Yang, 2022b) korpuszt használtam fel. Az összehasonlíthatóság végett a korpuszom megegyezik a Yang (2022b) által használt HI korpusszal és annak tanító és teszt halmazával:

– HI (HVG + index.hu): 559 162 szegmens; cikk: 147 099 485 token; lead: 16 699 600 token; átlagos tokenszám a cikkekben: 263,07; átlagos tokenszám a leadekben: 29,87; tesztanyag: 3000 szegmens.

A kutatásomban négy modellel végeztem kísérletet: mT5, mBART, M2M100, PEGASUS.

Az **mT5** (Xue és mtsai, 2021) a T5 (Text-To-Text Transfer Transformer) (Raffel és mtsai, 2020) többnyelvű változata. A T5 a Google modellje, amellyel a transzfer tanulás tulajdonságát kutatták. A kutatásban több különböző nyelvtechnológiai feladatot tanítottak be egyszerre egy modellnek. Minden feladatot szövegből szöveg (sequence-to-sequence) feladatként kezeltek. Mivel egy modellnek tanították be őket, a feladatok segíthetik egymást. A T5 modell többnyelvű változata az mT5, azonban az mT5 esetében az előtanítás során nem tanították be a modellt különböző nyelvtechnológiai feladatokra, hanem BERT modellhez hasonlóan tanították elő. A modellt nagy mennyiségű adaton, a 101 nyelvből álló, általuk erre a célra létrehozott mC4 korpuszon tanították elő. A modell rendelkezik magyar tudással. A modellnek több különböző mérete van, a kutatásomban az mT5-base (580 millió paraméter) modellt használtam.

Az **mBART** (Liu és mtsai, 2020) a BART modell (Lewis és mtsai, 2020) többnyelvű változata. A BART modell előtanításához olyan feladatokat határoztak meg, amellyel később könnyebben tudják összefoglaló generálás feladatára finomhangolni a modellt. A feladatok a következőek: token maszkolás, token törlés, szövegrész maszkolás, mondatok összekeverése, dokumentum rotáció (egy véletlenszerűen kiválasztott tokennél elválasztják a szöveget, majd úgy forgatják a szöveget, hogy a kiválasztott token lesz az első tokenje a dokumentumnak). Az mBART esetében többnyelvű szövegen végezték el a leírt előtanítási módszert. Az mBART első változatához 25 nyelvet alkalmaztak a CC25 (Wenzek és mtsai, 2020; Conneau és mtsai, 2020) korpusz segítségével. Később ezt a modellt kiegészítették további 25 nyelvvel, így jött létre az mBART-50 (Tang és mtsai, 2020), amellyel egy 50 nyelvből 50 nyelvre fordító gépi fordító modellt tanítottak. A magyar nyelv nem része az 50 nyelvnek. Azonban a SentencePiece tokenizálásnak (Kudo és Richardson, 2018) köszönhetően fel lehet vele dolgozni a magyar nyelvű szöveget is. A kutatásomban az mBART-large-50 modellt (610 millió paraméter) alkalmaztam.

Az **M2M100** (Aharoni és mtsai, 2019) a Fairseq többnyelvű gépi fordító modellje. A modell tanításához 100 nyelvet használtak fel. Létrehoztak a feladatra egy angol centrikus 100 nyelvű párhuzamos korpuszt. Architektúrában enkóder-dekóder transzformert használtak, a többnyelvűség eléréséhez nyelvi kóddal látták el a szöveg elejét, mind az enkóder, mind a dekóder részben. A modell rendelkezik magyar tudással. Kutatásomban az M2M100_1.2B modellt (1,2 milliárd paraméter) alkalmaztam.

A **PEGASUS** (Zhang és mtsai, 2020) modell tanításával azt célozták meg, hogy már az előtanítás során olyan feladatokat adtak a modellnek (hasonlóan az BART modellhez), amelyekkel az összefoglaló készítésének képességét erősítik. Az előtanítás folyamatában egyszerre kettő feladatot adtak a modellnek: szómaszkolás és célmondat maszkolás. A szómaszkolás hasonló a BERT modellhez, véletlenszerűen lemaszkoltak kettő szót. Az új megközelítés a célmondat maszkolás volt. Extraktív összegzés módszerével kiválasztották azt a mondatot, amely leginkább jellemez egy bekezdést, majd ezt a mondatot lemaszkolták. Ezzel a módszerrel a modell megtanulta azt, hogy melyek a fontos részek egy szövegben. A PEGASUS finomhangolással 12 különböző összefoglaló feladaton ért el piacvezető teljesítményt. Kutatásomban a PEGASUS-large modellt (568 millió paraméter) alkalmaztam.

4. Kísérletek

Első kísérletként megvizsgáltam, hogy a különböző modellek tokenizálói hogyan tudták feldolgozni a magyar szövegeket. Az összehasonlíthatóság végett azokra a szavakra néztem meg, amelyeket Nemeskey (2020) vizsgált a cikkében. Az átláthatóság kedvéért kivettem a speciális karaktereket, mint a nyelvi kódokat az M2M100 (__hu__) és az mBART (en_XX) esetében, valamint a szegmens végét jelölő címkét (</s>), amelyek mind a négy esetben voltak. A 1. táblázatban láthatóak a többnyelvű modellek által tokenizált szavak. Ami kiemelendő, az az mBART modell tokenizálása, ami nagyban hasonlít a huBERT tokenizálójához. Ez azért meglepő, mert direkt módon nem tettek bele magyar tanítóadatot az előtanítás során, azonban a tokenizálásból arra következtettem, hogy mégis sok magyar szöveg kerülhetett bele. A három modellből az M2M100 a legtöredezettebb, ami várható volt, hiszen a szótárban 100 nyelv szavaival kell osztozkodniuk a magyar szavaknak. Akkor mutatkozik meg a többnyelvű modellek hátránya, amikor több magyar specifikus betű, mint 'é', 'ú', 'ó' vagy 'ő', szerepel a szóban, ilyenkor a tokenizáló jobban széttördeli a szavakat.

A 2. táblázat mutatja a PEGASUS tokenizálójának kimenetét. A 'PEGASUS eredeti' oszlopban található az eredeti modell tokenizálása. Látható, hogy a modell nem ismeri a magyar ékezetes magánhangzókat. Ez nem meglepő, hiszen a modellt egy válogatott korpuszon tanították elő, amiben csak angol nyelvű szöveg volt. Ezért ahhoz, hogy kezelni tudja a magyar szavakat, hozzá kellett adni a hiányzó magyar ékezetes magánhangzókat. Az eredeti szótárban egyedül a kis 'é' betű szerepelt, így a szótárhoz első körben a következő magánhangzókat adtam:

– á, í, ó, ö, ő, ú, ü, ű, Á, É, Í, Ó, Ö, Ő, Ú, Ü, Ű

Azonban a tokenizáló tesztelése során vettem észre, hogy a SentencePiece esetében (mivel ez nem szó alapú, hanem szövegalapú) ha egy token önmagában szerepelt a szótárban, az azt jelentette, hogy ez a token egy szó része volt és nem az adott szó kezdőtokenje. Vagyis, ha egy szó ékezetes magánhangzóval kezdődött, akkor hozzá kellett tenni egy speciális karatert az ékezetes magánhangzó

	huBERT	mT5	mBART	M2M100
Nemzeti	Nemzeti	Nem zeti	Nemzeti	Nemzeti
Andersen	Andersen	Andersen	Andersen	Andersen
labdarúgó	labdarúgó	labda rúg ó	labdarúgó	lab dar ú gó
zambiai	z amb iai	z ambia i	z ambia i	z amb iai
megmaradt	megmaradt	meg marad t	megmarad t	meg mar adt
hétfő	hétfő	hét fő	hétfő	hét fő
keddtől	kedd től	ked d től	kedd től	k edd től
edényben	edény ben	ed ény ben	e dé ny ben	ed ény ben
Hétfőn	Hétfőn	H ét fő n	H ét fő n	H ét fő n
tájékoztatták	tájékoztat ták	tájékoztat ták	tájékoztat ták	tájé kozt att ák
leggazdagabb	leggazdagabb	leg gazdag abb	leg gazda ga bb	leg gaz dag abb
elpártolt	el párt olt	el párt olt	el pár to lt	el p árt olt

1. táblázat. Többnyelvű modellek tokenizálása

elé, ami azt jelölte, hogy előtte egy szóköz szerepelt. Így a következő tokeneket is hozzá kellett adni a szótárhoz:

– _á,_í,_ó,_ö,_ő,_ú,_ü,_ű,_Á,_É,_Í,_Ó,_Ö,_Ő,_Ú,_Ü,_Ű

A magyar ékezetes magánhangzók hozzáadásával egy újabb problémát generáltam. Mivel az eredeti szövegben nem voltak ékezetes magánhangzók, ezért olyan szóösszetétel sem, amiben ékezetes betűk lettek volna. Így a tokenizáló leválasztotta az ékezetes magánhangzókat a körülötte lévő betűkről. Ennek az lett a következménye, hogy ahol két ékezetes magánhangzó között volt egy betű, az magára maradt, mint önálló token. Vagyis, egy szóközt illesztett be a magára maradt betű elé. A szótárban nem szerepelt egy-egy szóköz nélküli betű, ezért hozzá kellett adni őket, hogy helyesen tudja tokenizálni a magyar ékezetes betűvel rendelkező szavakat. Így a teljes magyar ábécé betűinek kicsi és nagy változatait, valamint a speciális karakterrel ellátott (space jelölő) ékezetes magánhangzókat hozzáadtam a szótárhoz.

A magyar ábécé betűinek hozzáadására több módszer is adott volt, azonban a meglévő függvényekkel való hozzáadása túlságosan széttöredezte a szöveget, gyakorlatilag karakteralapúvá alakította a tokenizálót. Ez azért történt, mert mind a token hozzáadása (add_token), mind a speciális token hozzáadása (add_special_token) függvény valószínűség nélkül a szótár elejére teszi be a hozzáadott tokeneket. Így a tokenizálás ezekkel az újonnan hozzáadott betűkkel kezdi a tokenizálást, vagyis karakter alapon kezdi a tördelést. Ezért bele kellett nyúlnom a szótárba. A fent említett tokeneket valószínűségekkel a szótár végére adtam hozzá. Az eredeti szótár utolsó token valószínűségnél egyenletesen kisebb valószínűségeket rendeltem az új tokenekehez. A transzformer finomhangoló implementációja[3] alapján a modell az új tokenekhez automatikusan véletlenszerűen rendel súlyokat, majd a modellt átméretezi az új szótárnak megfelelően. Ilyen

[3] https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization

módon az eredetileg 96.103 tokenből álló szótárból készítettem egy 96.200 méretűt. A módosítást a 'tokenizer.json' nevű fájlban tudtam elvégezni, amivel csak a PegasusTokenizerFast függyvény tud dolgozni, ezért a finomhangoló szkriptet át kellett írni, hogy a PegasusTokenizerFast függvényt használja.

A 2. táblázat 'PEGASUS kiegészített' oszlopában látható a magyar ékezetes magánhangzók hozzáadása utáni tokenizálás eredménye. Bár nagyon töredezett, de tudja kezelni a magyar szöveget. Az 'edényben' szóban látható, hogy az 'y' betűt leválasztotta a 'ny' betűről. Mint fent említettem, az 'é' betű szerepelt az eredeti szótárban is és megvizsgálva az 'én' szóelem benne van a szótárban. Hiába tettem bele a 'ny' betűt, az 'én' szóelemnek nagyobb a valószínűsége, ezért egybetartotta a tokenizáló, és leválasztotta az 'y'-t.

	PEGASUS eredeti	PEGASUS kiegészített
Nemzeti	Nem ze ti	Nem ze ti
Andersen	Andersen	Andersen
labdarúgó	lab dar <unk> g <unk>	lab dar ú g ó
zambiai	zambia i	zambia i
megmaradt	me g mara d t	me g mara d t
hétfő	hé tf <unk>	hé tf ő
keddtől	ked d t <unk> l	ke dd t ő l
edényben	ed én y ben	ed én y ben
Hétfőn	Hé tf <unk> n	Hé tf ő n
tájékoztatták	t <unk> j é ko z t att <unk> k	t á j é ko z t att á k
leggazdagabb	leg gaz dag abb	leg gaz dag abb
elpártolt	el p <unk> r tol t	e lp á r to lt

2. táblázat. PEGASUS modell tokenizálása

Kísérleteim során a kiválasztott modelleket absztraktív összefoglaló feladatra finomhangoltam a HI korpuszon. A finomhangoláshoz a Hugging Face szkriptjét[4] alkalmaztam. A mérésekhez 4 darab NVIDIA A100 (80GB) GPU-t használtam az alábbi paraméterekkel:

- mT5: batch méret: 4; bemeneti szöveghossz: 1024; kimeneti szöveghossz: 256; epoch: 40; tanulási ráta: 0,0005; prefix: 'summarize: ';
- mBART: batch méret: 4; bemeneti szöveghossz: 1024; kimeneti szöveghossz: 256; epoch: 40; tanulási ráta: 0,00002;
- M2M100: batch méret: 4; bemeneti szöveghossz: 1024; kimeneti szöveghossz: 256; epoch: 40; tanulási ráta: 0,00002; src_lang: hu; tgt_lang: hu; warmup: 15 000;
- PEGASUS: batch méret: 4; bemeneti szöveghossz: 1024; kimeneti szöveghossz: 256; epoch: 40; tanulási ráta: 0,00002; warmup: 15 000;

[4] https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization

Az összehasonlíthatóság végett, az epoch számot meghagytam ugyanannyinak (40) mint Yang (2022b) kutatásában. A modellek méretei miatt 4-es batch mérettel tudtam dolgozni. Az előnye ezeknek a nagyobb modelleknek, hogy képesek nagyobb 1024-es bemeneti szöveghosszal dolgozni. A HI cikkek átlagos tokenszáma 263, így nem szükséges longformer modellek alkalmazása.

Az mT5 finomhangolása során a modell tulajdonsága miatt, hozzá kellett adni egy prefixet. Ezt leszámítva gyakorlatilag az alapbeállításokkal lehetett finomhangolni.

Az mBART modell esetében már módosítani kellett a paramétereken. A tanulási rátát kisebbre kellett venni, hogy jobban konvergáljon a modell, azonban mást nem kellett módosítani rajta.

Az M2M100 modellel még több módosítást kellett végezni. Először a modell tulajdonságának megfelelően be kellett állítani, hogy mind a forrásnyelv, mind a célnyelv magyar legyen. Az első kísérlet után mindössze 20,13 ROUGE1 értéket sikerült elérni a HI korpuszon, ezért 0,0005 tanulási rátát lejjebb vettem 0,00002-re (több tanulási rátát is kipróbáltam, ez bizonyult a legjobbnak). Ezzel sikerült 28,91 ROUGE1 értéket elérni. Ezután a warmup funkciót is bekapcsoltam, amivel végül a legjobb eredményt sikerült elérnem.

Végül a PEGASUS modellel kellett a legtöbb módosítást végezni. Először is a szótárat kellett kiegészíteni, hogy fel tudja dolgozni a magyar nyelvű szöveget. Ezen kívül az M2M100 tapasztalataiból kiindulva kisebbre állítottam a tanulási rátát és hozzáadtam a warmup lépést is.

5. Eredmények

A modellek kiértékeléséhez a ROUGE (Lin, 2004) metrikákat használtam. A gép által generált összefoglalókat összehasonlítottam az eredeti leadekkel. A 3. táblázat mutatja a modellek eredményeit. Az eredmények a következő formátumban láthatóak: ROUGE-1/ROUGE-2/ROUGE-L. Az összehasonlíthatóság végett betettem Yang (2022b) legjobban teljesítő modelljét a BART-base-1024 modellt. Olyan szempontból is releváns ez a modell, hogy ez is 1024 bemeneti hosszal dolgozott.

Az eredményekben megfigyelhető, hogy az mT5 és az mBART magasan felülmúlta a magyar modellt. Viszont az M2M100 és a PEGASUS modelleknek nem sikerült. Ez igazából várt eredmény, hiszen az M2M100 gépi fordításra előtanított modell, míg a PEGASUS finomhangolás nélkül nem tud egyáltalán magyarul. A meglepő eredmény mégis az, hogy versenyképes teljesítményt nyújtottak. A PEGASUS esetében a legváratlanabb az eredmény, hiszen egy modell ami nem tudott egyáltalán magyarul a finomhangolás során megtanult magyarul is és közben nem felejtette el a szövegösszefoglaló generálás tudását sem. Valamint az mBART modell eredménye is meglepő, hiszen közvetlenül nem tettek bele magyar tudást, azonban a tokenizálásból azt lehetett leszűrni, hogy látott már magyar szöveget. Ez segíthette a modellt a magyar nyelvre való finomhangolásra.

A 4. táblázatban mutatok egy-egy példát a modellek összefoglalóiról. Egy olyan példát választottam, ahol a legtöbb modell hallucinált vagy rosszul kom-

	HI
BART-base-1024	31,86/14,59/23,79
mT5	33,30/15,97/24,65
mBART	**35,17/16,46/25,61**
M2M100	30,84/13,21/22,54
PEGASUS	30,36/13,11/21,57

3. táblázat. Absztraktív összefoglaló modellek eredményei

binált információkat, hogy lássuk a modellek gyengeségeit. A BART-base-1024 modell: 'Magyar Fotográfiai Központ', '160 fotóművész'; mT5: 'Magyar Nemzeti Galéria', 'Krisztina körúti épület'; mBART: 'Capa Központban nyílik a kiállítás', M2M100: 'Városi Múzeum'. A hallucinációk mellett a BART-base-1024 modell esetében a fogalmazás területén is vannak hibák. A példákból az látszik, hogy valóban a BART-base-1024 összefoglalója a legrosszabb. Az mT5 összefoglalója a legrészletesebb és leghosszabb, azonban a hallucinációi miatt fenntartásokkal kell kezelni, de a hosszúsága miatt nagyobb fedést tud produkálni, ez magyarázhatja a magas F1 mértékét is. Az mBART és az M2M100 a legszűkszavúbbak. A PEGASUS eredménye elég jó, de ha közelebbről megnézzük, akkor inkább extraktív módon készítette az összefoglalót. Az első mondat a cikk egyik hosszabb mondatának részmondata, amit levágott és egy ponttal lezárta, talán ez a lezárás az egyetlen absztraktív művelet benne. A második mondat egy az egyben szerepel az eredeti cikkben. Az eredmény nem meglepő, hiszen az előtanított modell részben extraktív feladatra volt tanítva. A beam érték feljebb állításával ez módosulna, de akkor az összehasonlíthatóság elveszne. Viszont az extraktív tulajdonsága miatt nincsen hallucináció a PEGASUS kimenetében. Ilyen szempontból ha a számok területén a leggyengébb is, mégis a legjobbnak mondható, mivel nincsen benne félrevezető információ. A jövőben szeretném továbbfejleszteni a kutatást abba az irányba, hogy egy angol nyelvű és egy magyar nyelvű szótár egyesítésével létrehozzak egy új, kétnyelvű szótárat.

6. Összegzés

A kutatásom során különböző többnyelvű és egy angol nyelvű neurális nyelvmodellt finomhangoltam magyar nyelvű absztraktív összefoglaló generálás feladatára. A kutatás érdekessége, hogy olyan modelleket is kipróbáltam, mint az mBART, ami ugyan összefoglaló generálásra való, azonban nincsen benne közvetlenül magyar nyelvi tudás. Illetve az M2M100 modellt, amelyben ugyan van magyar tudás, de gépi fordításra tanították elő. A kísérleteimben megmutattam, hogy mindegyik modellt lehet finomhangolni magyar összefoglaló generálás feladatára. Az eredményeimben nem várt módon, az mBART érte el a legjobb eredményt a HI korpuszon. Ez azt a kérdést veti fel, hogy mennyi magyar tudás szükséges az előtanításban, hogy utána magyar nyelvű feladatokra tudjuk adaptálni. A tokenizálás kísérletben láthattuk, hogy bár közvetlen módon nem tettek bele az mBART modellbe magyar anyagot, azonban mégis feltételezhetjük, hogy

	Példa
cikk	A több mint 160 fotót felvonultató tárlat azt szeretné megmutatni , ki volt Robert Capa - mondta el a hétfői sajtóbemutatón Fisli Éva , a tárlat kurátora . A látogatók így általa és róla készített felvételek , valamint korhű installációk segítségével a magyar származású Capát egyaránt megismerhetik mint emigránst , fotóriportert és haditudósítót , mint játékost és hőst . A 2014. január 12-ig látogatható tárlathoz Doku-tér is tartozik , ahol a látogatók elmélyülhetnek a Capa-irodalomban is . A fotográfus születésének 100. évfordulójához kapcsolódóan a 3-as metró kocsijainak hirdetőfelületein is látható jelenleg egy válogatás Capa-fotográfiákból , október 21-én pedig megnyílik az Ernst Múzeum épületében a Capa Központ .
lead	A világhírű magyar származású fotográfus " szerepeit " kutatja a Magyar Nemzeti Múzeum (MNM) Robert Capa - A játékos című , kedden nyíló kiállítása . Capa képei láthatók a 3-as metrón is , októberben pedig nyílik az Ernst Múzeum épületében a Capa Központ .
BART-base-1024	A Magyar Fotográfiai Központban több mint 160 fotóművész várja újra Robert Capa fotóit , köztük a korhű magyar származású fotográfiát .
mT5	A világhírű magyar származású fotográfus , Robert Capa születésének 100. évfordulója alkalmából nyílt kiállítás a Magyar Nemzeti Galéria budapesti , Krisztina körúti épületében .
mBART	Robert Capa születésének 100. évfordulója alkalmából fotókiállítás nyílik hétfőn Budapesten , a Capa Központban .
M2M100	Robert Capa világhírű fotográfusról nyílik kiállítás kedden Budapesten , a Városi Múzeumban .
PEGASUS	Több mint 160 fotót felvonultató tárlat azt szeretné megmutatni, ki volt Robert Capa. A látogatók így általa és róla készített felvételek, valamint korhű installációk segítségével a magyar származású Capát egyaránt megismerhetik mint emigránst, fotóriportert és haditudósítót, mint játékost és hőst.

4. táblázat. Különböző modellek összefoglalói

került bele magyar szöveg. Végül, de nem utolsó sorban, a PEGASUS angol nyelvű modellt adaptáltam magyar nyelvre. Megmutattam a kísérletemben, hogy a szótár bővítésével a modell megtanítható, finomhangolható magyar nyelvre.

Az mT5 és az mBART modellekkel felülmúltam és 'state-of-the-art' eredményeket tudtam elérni a magyar nyelvű absztraktív összefoglaló területén.

Hivatkozások

Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

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

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Hasan, T., Bhattacharjee, A., Islam, M.S., Mubasshir, K., Li, Y.F., Kang, Y.B., Rahman, M.S., Shahriyar, R.: XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4693–4703. Association for Computational Linguistics, Online (Aug 2021)

Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018)

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)

Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)

Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (2019)

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., Zettlemoyer, L.: Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics 8, 726–742 (11 2020)

Makrai, M., Tündik, Á.M., Indig, B., Szaszák, G.: Towards abstractive summarization in hungarian. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 505–219. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Nemeskey, D.M.: Egy emBERT próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged (2020)

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Pfeiffer, J., Goyal, N., Lin, X., Li, X., Cross, J., Riedel, S., Artetxe, M.: Lifting the curse of multilinguality by pre-training modular transformers. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3479–3495. Association for Computational Linguistics, Seattle, United States (Jul 2022)

Pfeiffer, J., Vulić, I., Gurevych, I., Ruder, S.: MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7654–7673. Association for Computational Linguistics, Online (Nov 2020)

Pfeiffer, J., Vulić, I., Gurevych, I., Ruder, S.: UNKs everywhere: Adapting multilingual language models to new scripts. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 10186–10203. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021)

Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research 21(140), 1–67 (2020)

Tang, Y., Tran, C., Li, X., Chen, P.J., Goyal, N., Chaudhary, V., Gu, J., Fan, A.: Multilingual translation with extensible multilingual pretraining and finetuning (2020)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)

Wenzek, G., Lachaux, M.A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., Grave, E.: CCNet: Extracting high quality monolingual datasets from web crawl data. In: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France (May 2020)

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021)

Yang, Z.G., Agócs, Á., Kusper, G., Váradi, T.: Abstractive text summarization for hungarian. Annales Mathematicae et Informaticae 53, 299–316 (2021)

Yang, Z.G.: "Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 463–476. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022a)

Yang, Z.G.: BARTerezzünk! Messze, messze, messze a világtól, BART kísérleti modellek magyar nyelvre. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–28. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022b)

Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences forabstractive summarization. In: Thirty-seventh International Conference on Machine Learning (2020)

Inzulinrezisztencia betegség jelenségének felismerése és osztályozása orvosi dokumentumokban

Yang Zijian Győző

Sightspot Network Kft.
4034 Debrecen, Vágóhíd u. 2.
Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
yang.zijian.gyozo@nytud.hu

Kivonat A jelen cikkben egy kutatás-fejlesztés projekt első fázisának részleteit mutatjuk be, amelynek keretében az inzulinrezisztencia betegség kialakulásának veszélyét szeretnénk előre jelezni a nyelvtechnológia eszközeivel. A kutatásunk kétmillió magyar nyelvű kórházi kórlap feldolgozásával történt a modern neurális nyelvtechnológia segítségével. A feladatot osztályozási feladatként értelmeztük, amelyben három különböző esetet különböztettünk meg: inzulinrezisztenciás betegek, nem inzulinrezisztenciás páciensek és gyanús esetek. A gyanús esetek közé azokat a pácienseket soroltuk, akik a kórlapjuk alapján nem inzulinrezisztenciások, de közben tudjuk, hogy azok. A feladat nehézsége, hogy a programunknak fel kell ismernie a gyanús eseteket úgy, hogy a kórlapon nem szerepel az inzulinrezisztencia betegség. A probléma ily módon egy háromosztályos klasszifikáció feladatként oldható meg. A kórlapok zajossága és félig strukturáltsága miatt, rendkívül nehéz belőle egységes releváns tulajdonság jegyeket kinyerni, ezért a probléma megoldására egyedül a modern nyelvi modellek jöhettek csak számításba, amelyek automatikusan nyerik ki a számukra relevánsnak számító nyelvi jegyeket. A kutatásunkban felhasználtunk egy statikus és egy környezetfüggő neurális nyelvi modellt. Az eredményeink alapján, a modelljeink közel 80%-os pontossággal tudta megbecsülni, hogy az adott kórlap a fent említett három kategóriából melyikbe tartozott. Az általunk létrehozott osztályozási modellekkel orvosi támogatást tudunk nyújtani, amelynek során a gép jelezni tudja azon eseteket, ahol, bár a beteg másféle kivizsgáláson vesz részt, a kórlap alapján az adott páciensnél felmerülhet az inzulinrezisztencia betegségének veszélye.
Kulcsszavak: inzulinrezisztencia, neurális szövegosztályozás, fastText, huBERT

1. Bevezetés

Az inzulinrezisztencia korunk egyik jelentős, sok embert érintő, de annál nehezebben előre jelezhető betegsége. A kórházba látogató páciensek kórlapjai az évek során temérdek mennyiségben halmozódtak fel. Ezek feldolgozása nem kis

feladat. A digitalizálás fejlődésével az egészségügyi intézetek elkezdték adatbázisokba menteni a dokumentumaikat. Ez a folyamat lehetőséget ad arra, hogy automatikus módszerekkel elemezzük őket. Azonban jelentős probléma, hogy ezek a dokumentumok kevésbé strukturáltak. Ez nagyban megnehezíti a szövegfeldolgozást. Továbbá az a tény is nehezíti a feladatot, hogy a különböző páciensek különböző betegségei is szerepelnek a kórlapokon. Ez megnehezíti az egységes feldolgozást. A nyelvtechnológia fejlődésével, különösen a neurális nyelvmodellek megjelenésével áthidalhatóak ezek a problémák. Az új generációs transzformer alapú nyelvmodellek képesek egy adott szövegben lévő összefüggéseket felismerni, független azok strukturális felépítésétől. Ezeknek a modelleknek a finomhangolásával képesek vagyunk a nyers strukturálatlan szöveget elemezni és számunkra hasznos információkat kinyerni.

Kutatásunkban kétféle neurális modellel végeztünk kísérletet. Egy statikus szóbeágyazással tanított fastText modellel, illetve egy kontextuális finomhangolt magyar nyelvű BERT modellel.

2. Kapcsolódó irodalom

Az utóbbi években a nyelvtechnológia fejlődésével számos kutatás célozta meg az orvosok támogatását a nyelvtechnológia segítségével. Chen és mtsai (2017) kutatásukban konvolúciós hálózattal prediktáltak betegségeket. Nyelvi osztályozó modellük tanításához 2013-2015 között gyűjtött kórházi elektronikus egészségügyi dokumentációkat használtak. Nori és mtsai (2015) a betegségek közötti hasonlóságot vetették össze az elektronikus egészségügyi dokumentumok közötti hasonlósággal, ezzel betegség specifikus összefüggéseket tudtak beépíteni a halálozási modellezésbe, amellyel pontosabb prediktív modelleket tudtak létrehozni. Szintén konvolúciós hálózatot használt Yao és mtsai (2018), akik klinikai szövegeket osztályoztak szabályalapú jegyek hozzáadásával. Geraci és mtsai (2017) rekurrens hálózat segítségével kerestek alkalmas jelölteket a fiatalkori depresszió kutatásához, ehhez strukturálatlan szövegekből azonosították a fiatalkori depresszió tüneteit.

Magyar nyelvre Siklósi és Novák (2014); Orosz és Prószéky (2014) végeztek kutatásokat klinikai szövegek normalizálására. Papp és mtsai (2014) az Alzheimer-kórban szenvedő páciensek beszédeit elemezték, hogy a korai Alzheimer-kórra jellemző nyelvi tüneteket detektálják. Bagi és mtsai (2019) kísérleteiben a szkizofréniát azonosították spontán beszéd temporális paraméterei alapján. Kicsi és mtsai (2020) kutatásukban radiológiai leletek szövegében azonosítottak testrészeket, elváltozásokat és azok kapcsolatait. Jenei és Kiss (2020) a depressziós állapot automatikus detektálását tűzték ki célul konvolúciós neurális hálózatok segítségével. Vetráb és mtsai (2022) szekvenciális autoenkódert használtak enyhe kognitív zavar automatikus felismerésére.

3. Osztályozó modelleink

A kutatásunkban felhasználtunk egy statikus és egy környezetfüggő, vagy másnéven kontextuális neurális nyelvi modellt.

A **fastText** (Joulin és mtsai, 2017, 2016) a Meta Research (korábban Facebook Research) csapatának fejlesztése, melynek célja a szóreprezentációs és a szövegosztályozó modellek effektív tanítása. A módszerrel szóalapú és n-gramm karaktereken alapúló 'skip-gram' és 'cbow 'modellt lehet tanítani. A rendszer legnagyobb előnye, hogy C++ nyelven implementált, ezért gyors és hatékony megoldást kínál anélkül, hogy előfeldolgozásra vagy felügyeletre lenne szükség (Bojanowski és mtsai, 2017). Nincsen szüksége videókártyára sem a tanításhoz. Szövegosztályozás szempontjából más mély tanulás alapú megoldásokkal összevethető a teljesítménye, és egy lényegesen gyorsabb megoldás tanítás és kiértékelés szempontjából (Joulin és mtsai, 2017). A platformon előre tanított szóvektorok érhetőek el 158 különböző nyelvre, ezáltal egy nagyon kézenfekvő és lehetőségekkel teli eszköznek számít a többnyelvű nyelvfeldolgozás terén is.

A **BERT** (Devlin és mtsai, 2019) (Bidirectional Encoder Representations from Transformer) egy kétirányú transzformer enkóder (Vaswani és mtsai, 2017). A BERT modellt két nyelvmodellezési feladaton tanították elő: szómaszkolás és következő mondat perdikciója. A szómaszkolás során a tanításhoz használt korpuszban a szavak 15%-a véletlenszerűen maszkolásra kerül, a rendszernek pedig ki kell találnia a kimaszkolt szavakat. A következő mondat predikciója során pedig a feladat annak kitalálása, hogy két kiválasztott mondat a szövegben egymást követő mondatok-e vagy csak két véletlenül kiválasztott mondat. A szótár méretének csökkentése érdekében a BERT modell szóelem (word pieces) tokenizáló algoritmust (Schuster és Nakajima, 2012) használ. A BERT előtanítása során általános nyelvi tudásra tesz szert, ezért is hívjuk nyelvmodellnek, majd ezt követően, a modellt finomhangolással további specifikus feladatokra tanítható. A kutatásunkhoz a magyar nyelvű **huBERT** (Nemeskey, 2021) modellt használtuk, amely a 9 milliárd szavas Webkorpusz 2.0 korpuszon (Nemeskey, 2020) lett előtanítva. A huBERT jelenleg a legjobban teljesítő magyar nyelvű BERT modell.

4. A Korpusz

Kutatásunkban összesen 2 000 000 orvosi dokumentummal dolgoztunk, amelyekből 1 000 000 dokumentum az inzulinrezisztencia kategóriába tartozott és 1 000 000 a nem inzulinrezisztencia kategóriába. Az orvosi dokumentumok a következő csoportba tartoznak: kórlap, zárójelentés, xamba, ambuláns lap, egyéb rövid jelentések. A 1. ábrán mutatunk egy példát egy nyers ambuláns lapról, ami az adatbázisban található. Látható, hogy a dokumentum félig strukturált, amelyben a strukturált rész inkább személyes adatokból áll. Továbbá az is látható, hogy az ambuláns lap szabad szöveges részében találhatóak a fontosabb információk a páciensről. Végül az is megfigyelhető, hogy a szöveg anonimizált.

```
VESZ Egészségügyi Szolgáltató Kht.
Gyermek Rehabilitációs Központ
#CIM                    .
Telefon:#T
0921P7101

                                AMBULÁNS LAP

Név:    #AEXTRA
Leánykori név:
TAJ:    #AEXTRA
Szül.dátum: #AEXTRA
Születési hely:     #AEXTRA
Anyja neve: #AEXTRA
Lakcím: #AEXTRA
A megjelenés ideje: 2007#      09:32
Naplószám:      02000449

PSZICHOLÓGIAI VÉLEMÉNY

Előzmények

    Az elmúlt tanévben havi 1 alkalommal vett részt nálam a kisfiú játékterápiás foglalkozáson.
    2007. júliusban kizárták izombetegség meglétét a #NEV, mely miatt születése óta (az anya
    betegsége miatt) rendszeres orvosi kontrollokra járt ill. gyógytornán is részt vett.

Vizsgálatok

    Viselkedésmegfigyelés és kontroll tesztvizsgálat történt (B. Binet, figyelemvizsgálat).

    Eredmény: #NEV viselkedése különösen a 2006. októberi státuszhoz képest sokat változott.
    Csapongása sokat mérséklődött, figyelme, kitartása életkorának megfelelő szintű. B. Binet
    tesztben teljesítménye életkorát 3 hónappal meghaladja, IQ=105. Általános ismeretei
    korának megfelelőek, az analógiás gondolkodás alakulóban, rövid távú verbális memóriája 3
    elemű, konstrukciós készségei, téri-vizuális észlelése jó. Enyhe grafomotoros ügyetlenség
    jellemzi, hangzóhibái még vannak.
    Szomatizációs problémái mérséklődtek; viselkedése érettebb.

Vélemény, javaslat

    A grafomotoros készségek javítását próbálják otthon formamásolással, közös rajzolással
    fejleszteni; a hangzóhibákat óvodai logopédia keretében lehetne korrigálni. Javaslom a
    Színezd ki és rajzolj te is...; és Színezd ki és számolj te is .. feladatlapok otthoni
    használatát.
    Kontroll: 1 év múlva, friss háziorvosi beutalóval.

Dátum:  Debrecen,  2007#      09:32

#NEV            gy.klin. mentalhig.szakpszich.

#NEV          !

Szíveskedjen minden további jelentkezéskor korábbi orvosi dokumentációját magával hozni!
Köszönjük, hogy megtisztelt bizalmával!
```

1. ábra: Példa egy nyers ambuláns lapról

A 2. ábra egy rövid dokumentumot mutat be. Rövid felsorolás jellegű, rövidítésekkel és kulcsszavakkal. A két példából az figyelhető meg, hogy nagy a különbség a dokumentumok között, ezért nehéz egységesen feldolgozni őket.

```
Víztiszta cervixnyák. Váladékvizsgálat: I. tiszt.fok.
Mikrobiológiai mintavétel. Gombavizsgálat: negatív.
TVS: Uterus afv-ben. Kismedencében kóros nem ábr.
```

2. ábra: Példa egy rövid orvosi dokumentumról

Az adatbázisban az orvosi dokumentumok egy része további meta adatokat tartalmaztak, amelyekből az egyik információ a betegségek nemzetközi osztályozására használt azonosító, az úgynevezett BNO kód. Az inzulinrezisztencia betegség kódjai 'E10' karaktersorozattal kezdődnek és 0-9 számokkal folytatódnak: E100, E101, E102, E103, E104, E105, E106, E107, E108, E109. Maga a kórházi adatbázishoz sajnos nem volt hozzáférésünk, mi már szűrt adatokat kaptunk kézhez. Tudomásunk szerint a szűrés a fent említett BNO kódok alapján történt. Az első egymillió kórlap tartalmazza valamelyik inzulinrezisztenciával kapcsolatos BNO kódot. A második egymillió kórlap a fent említett inzulinrezisztenciával kapcsolatos BNO kódokat nem tartalmazza. Miután a szűrés megtörtént, adatbiztonsági okokból a személyes adatokat törölték a kórlapokból. A törlés szabály alapon történt, a személyes adatok megjelenése a dokumentumokban valamennyire szabályosságot mutatott, például a nevek előtt megjelent a 'név:' vagy a születési dátumok előtt a 'szül. dátum' kifejezések. A személyes adatok törlésében szintén nem vettünk részt és nem is kaptunk részletes információt arról, hogy pontosan milyen adatokat töröltek ezért csak a kapott szövegek alapján tudtunk következtetéseket levonni. A kapott szövegek elemzése alapján a következő adatokat törölték:

- Beteg személyes adatai: név, születési hely és dátum, születési név, anyja neve, lakcím, TAJ szám, munkahely, halál dátuma
- Kezelő orvos személyes adatai: kezelőorvos neve, vezető főorvos neve
- Általános információk (beteg, orvos vagy intézmény): telefonszám, fax

A szabály nem terjed ki minden esetre, ha egy adat nem illeszkedett pontosan a megadott szabályra, akkor az a személyes adat benne maradt. Ilyen például, amikor a dátumba '#' jel került: 2018#, vagy amikor szabad megfogalmazásban jelent meg a személyes adat.

Az így kapott adatokat további szűrések alá vetettük, végeztünk rajta tokenizálást a huSpacy (Orosz és mtsai, 2022) eszközzel és a szövegből szabály alapján kitöröltünk minden inzulinnal kapcsolatos BNO kódot: E10, E100, E101, E102, E103, E104, E105, E106, E107, E108, E109, E11, E110, E111, E112, E113, E114, E115, E116, E117, E118, E119.

Így első feladatként a kétmillió kórlapot konkatenáltuk és létrehoztunk egy bináris osztályozó modellhez egy korpuszt. A korpuszt véletlenszerűen megkevertük és kivettünk belőle 10%-ot tesztelő anyagnak. Így létrejött egy 90%-10% vágás a tanító és a teszt halmaz létrehozásához. A korpusz részletesebb tulajdonságai az 1. táblázatban láthatóak.

Következő feladatként a kapott adathalmazból kiválogattuk azokat a kórlapokat, amelyek bár nem lettek ellátva inzulinrezisztencia BNO kóddal, mégis tudjuk a paciensről, hogy inzulin rezisztenciával rendelkezik. Ehhez a feladathoz kigyűjtöttük az összes kórlaphoz tartozó páciens azonosítót, majd megkerestük a metszetet. A metszet alkotja azokat a betegek azonosítóját, akiknek a kórlapjai szerepelnek mind az inzulinrezisztenciával rendelkező, mind a nem inzulinrezisztenciával rendelkező adathalmazban. Ezt a halmazt elneveztük 'GYANÚS' halmaznak. Az adathalmazok további kialakításában a metszet/GYANÚS halmaz (szűkössége miatt) mennyisége lett a mérvadó. Így összesen igazodva a GYANÚS

halmazhoz, kerekítve mind a három osztályhoz 400 000 kórlapot választottunk ki véletlenszerűen a tanítóanyaghoz és 15 000 kórlapot a tesztanyaghoz. Az 1 táblázatban találhatóak a korpusz részletes kvantitatív tulajdonságai.

	Szegmens	Token	Type	Átlagos tokenszám / kórlap	Átlagos mondatszám / kórlap
2-osztályos tanító	1 800 000	290 254 939	730 914	161,25	20,69
2-osztályos teszt	200 000	32 423 375	270 045	162,12	20,79
3-osztályos tanító	1 200 000	181 306 262	631 334	151,09	19,33
3-osztályos teszt	45 000	6 853 141	132 040	152,29	19,45

1. táblázat. Korpuszok tulajdonságai

Végül utolsó feladatként, mielőtt a modell tanítása történt, minimális tisztítást végeztünk az adatokon. A következő lépéseket végeztük el:

- Ismétlődő írásjelek normalizálása: A szövegben számtalan helyen díszítősor jellegű szövegrészek találhatóak, ilyen a sok pont, gondolatjel és kérdőjel egymás után. Ezekből egyet-egyet hagytunk meg.
- Néhány szó és rövidítés normalizálása: Az Ambuláns lap cím számtalan eseten a következőképpen szerepelt: A M B U L Á N S L A P. Ezt konvertáltuk "AMBULÁNS LAP" formába, illetve: 'k . m . n .', 'K f t .'.
- Személyes adatok: Mivel a személyes adatokat törölték, ezért töröltük a hozzátartozó helyőrzőket és megnevezéseket is: #NEV, #EMAIL, szül.ideje:, TAJ száma, #CIM, #AEXTRA, #DOC, #T.
- Whitespacek normalizálása: Ahol több whitespace állt egymás után, azokat lecseréltük egy szóközre.

5. Mérések és eredmények

A kutatásunk során felhasználtuk a fastText statikus és a BERT környezetfüggő neurális nyelvi modellt az osztályozó modellek tanítására. Mind a fastText keretrendszerrel, mind a BERT modellel két predikciós modellt tanítottunk: egy 2-osztályos és egy 3-osztályos modellt. A modellek tanításához a következő hiperparamétereket használtuk:

- fastText: wordNgrams: 3; tanulási ráta: 0,8; bucket: 200000; dim: 200; epoch: 10.
- huBERT: tanulási ráta: 5e-5; batch méret: 64; epoch: 10.

A BERT modell tanításához a Hugging Face által közzétett szövegosztályozó szkriptet[1] alkalmaztuk.

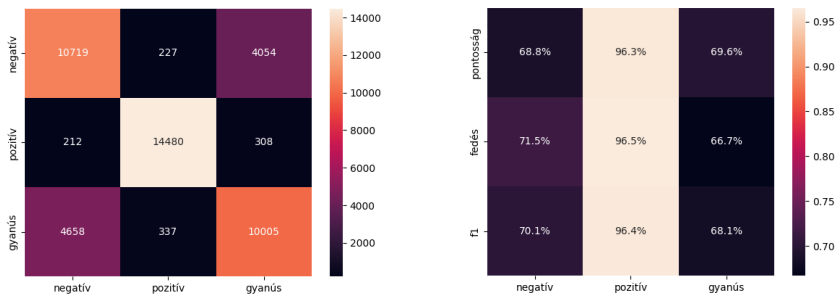
A 2. táblázatban láthatóak a modellek eredményei. A kiértékeléshez a pontosság (accuracy) metrikát használtuk.

[1] https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification

	2-osztályos	3-osztályos
fastText	97,58%	78,23%
huBERT	**98,19%**	**81,34%**

2. táblázat. Az osztályozó modellek eredményei

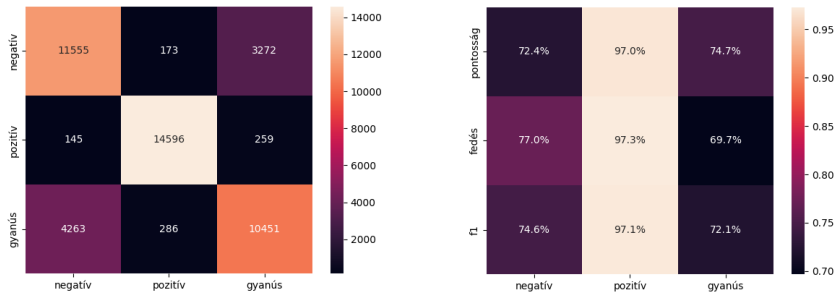
A 3. és a 4. képen láthatóak a 3-osztályos fastText és huBERT modellek teljesítményei címkékre bontva. A képek bal oldalán van a tévesztési mátrix, míg a jobb oldali ábrákon a pontosság, fedés és F1 mértékek találhatóak. A modellek a pozitív eseteket tudták könnyen megállapítani. Ez nem meglepő, hiszen a pozitív eseteknél csak a BNO kódokat töröltük a szövegből, de egy ilyen esetben többféle információ is utalhat magára a betegségre. Leggyengébben a gyanús esetek osztályozásánál teljesítettek a modellek, ez is várt eredmény, hiszen ezekben az esetekben kevés vagy semmilyen jel nem utal az inzulinrezisztenciára, azonban mégis azok lehetnek. A hőtérképek alapján a két modell rendkívül hasonlóan teljesített, ami a számokban is látszik. A huBERT fedése magasabb a negatív eseteknél, vagyis több tényleges negatív esetet tudott detektálni, ami a gyanús esetek predikciójának bizonytalanságát csökkenti. Az ábrán is látszik, hogy a gyanús eseteknél a pontosság magasabb, mint a fedés, ami a mi esetünkben azt jelenti, hogy amikor gyanús esetet mond a modell, azt magabiztosabban teszi.



3. ábra: A 3-osztályos fastText modell teljesítményének hőtérképei

6. Összegzés

Kutatásunkban az inzulinrezisztencia betegség kialakulásának kockázatát szeretnénk előre jelezni orvosi dokumentumok alapján. A feladat megoldásához az új neurális nyelvtechnológia eszközeit használtuk fel. Kísérleteinkben a szövegek osztályozásához a statikus fastText és a kontextuális huBERT modelleket tanítottuk be. A modellek tanításához kétmillió orvosi dokumentumot használtunk



4. ábra: A 3-osztályos huBERT modell teljesítményének hőtérképei

fel. Három esetre bontottuk a dokumentumokat: negatív, pozitív és gyanús. A gyanús esetek azok, akikről tudjuk, hogy kialakult nála az inzulinrezisztencia betegség, de az adott kórlapon ez nem volt feltüntetve, mivel lehet, hogy másféle vizsgálaton volt az illető. Eredményeink azt mutatják, hogy a huBERT modellel 81,34% pontossággal tudta megoldani ezt a 3-osztályos klasszifikációs feladatot. Rendelkezésünkre áll további hatmillió orvosi dokumentum, amelyeket következő lépésként szeretnénk feldolgozni.

Hivatkozások

Bagi, A., Gosztolya, G., Szalóki, S., Szendi, I., Ildikó, H.: Szkizofrénia azonosítása spontán beszéd temporális paraméterei alapján – egy pilot kutatás eredménye. In: XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 189–202. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2019)

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)

Chen, M., Hao, Y., Hwang, K., Wang, L., Wang, L.: Disease prediction by machine learning over big data from healthcare communities. IEEE Access 5, 8869–8879 (2017)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Geraci, J., Wilansky, P., de Luca, V., Roy, A., Kennedy, J.L., Strauss, J.: Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression. Evidence-Based Mental Health 20(3), 83–87 (2017)

Jenei, A.Z., Kiss, G.: Depresszió detektálása korrelációs struktúrán alkalmazott konvolúciós hálók segítségével. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 59–71. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2020)

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics, Valencia, Spain (Apr 2017)

Kicsi, A., Szabó, L.K., Pusztai, P., Németh, P., László, V.: Entitások azonosítása és szemantikai kapcsolatok feltárása radiológiai leletekben. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–27. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2020)

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2021)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)

Nori, N., Kashima, H., Yamashita, K., Ikai, H., Imanaka, Y.: Simultaneous modeling of multiple diseases for mortality prediction in acute hospital care. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 855–864. KDD '15, Association for Computing Machinery, New York, NY, USA (2015)

Orosz, G., Prószéky, G.: Hol a határ? Mondatok, szavak, klinikák. In: X. Magyar Számítógépes Nyelvészeti Konferencia. pp. 177–187. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2014)

Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., Farkas, R.: Huspacy: an industrial-strength hungarian natural language processing toolkit. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 59–73. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Papp, P.A., Rácz, A., Vincze, V.: Automatikus morfológiai elemzés a korai Alzheimer-kór felismerésében. In: X. Magyar Számítógépes Nyelvészeti Konferencia. pp. 199–207. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2014)

Schuster, M., Nakajima, K.: Japanese and korean voice search. In: ICASSP. pp. 5149–5152. IEEE (2012)

Siklósi, B., Novák, A.: Rec. et exp. aut. Abbr. mnyelv. KLIN. szöv-ben – rövidítések automatikus felismerése és feloldása magyar nyelvű klinikai szövegekben. In: X. Magyar Számítógépes Nyelvészeti Konferencia. pp. 167–176. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2014)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett,

R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)

Vetráb, M., José, V.E.L., Balogh, R., Imre, N., Hoffmann, I., Tóth, L., Pákáski, M., Kálmán, J., Gosztolya, G.: Enyhe kognitív zavar automatikus felismerése szekvenciális autoenkóder használatával. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 175–184. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022)

Yao, L., Mao, C., Luo, Y.: Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. In: 2018 IEEE International Conference on Healthcare Informatics Workshop (ICHI-W). pp. 70–71 (2018)

huBERT alapú sziámi neurális háló architektúrák elemzése ügyfélszolgálati emailek klasszifikációjára

Vándor Péter[1] és Csáki Csaba[1]

[1] Budapesti Corvinus Egyetem, Fővám tér 8,
H-1093 Budapest, Magyarország
peter.vandor@stud.uni-curvinus.hu
csaki.csaba@uni-curvinus.hu

Kivonat: A digitális gazdaságban megnőtt az ügyfélszolgálatok szerepe és az ügyfelek nem-közvetlen kommunikációs csatornák (pl. email) esetén is gyors választ és hatékony megoldást várnak. Az ügyfélszolgálatoknak ezt úgy kell elérniük, hogy közben a szöveges (email, közösségi média stb.) üzenetek száma rohamosan növekszik. A szöveges adathalmazok természetes nyelvi feldolgozásának egyik új technikája a BERT, melynek huBERT néven magyar változata is elérhető. Az itt bemutatott kutatás célja kettős: egyrészt egy valós ügyfélszolgálati probléma (nagy mennyiségű emailek automatikus kategorizálásának) mesterséges intelligencia alapú informatikai támogatása a hatékonyabb feldolgozás érdekében; másrészt a felhasznált BERT-alapú MI architektúrák szisztematikus vizsgálata tapasztalat szerzés céljából. A Python nyelven megírt és huBERT modulokat is használó sziámi architektúrára épülő 3 illetve 10 kategóriát felismerő, ügyfél emaileket feldolgozó megoldás paramétereinek változtatásával növelhető volt a pontosság és az alkalmazott legösszetettebb struktúrával a kezdeti alig 70%-os helyett közel 95%-os teljesítmény volt elérhető.

1 Ügyfélszolgálati kihívások és kapcsolódó NLP technikák

A hálózatos digitális gazdaság egyik hozadéka, hogy megnőtt az ügyfélszolgálatok szerepe mind az ügyfélelégedettség mind a színvonalasabb szolgáltatás nyújtás vonatkozásában (Cassandra és mtsai., 2019). A változó feladatokkal és újabb technológiákkal és kommunikációs csatornákkal (email, web űrlapok, közösségi média, videó hívások stb.) bővülő felelősségi kör komoly terhelést helyezett erre a vállalati funkcióra (Zendesk, 2020). Az új csatornák, és különösen az email általános elérhetősége miatt megnövekedett a beérkező üzenetek, kérések, panaszok, és kérdések száma. Ráadásul egyes 'kampány' időszakokban különösen kiugró lehet a kezelendő megkeresések száma. Ezeket a beérkező üzeneteket sokszor rövid határidővel kellene rendezni, mivel az ügyfelek nem közvetlen kommunikációs technikák (pl. email) esetén is gyors választ és hatékony megoldást várnak. A megnövekedett ügyfélszolgálati terhelés és elvárások együttes hatásának kivédésére egyre nagyobb hangsúlyt kapnak a mesterséges intelligencia (MI) alapú megoldások, azon belül is a természetes nyelv feldolgozás (angol rövidítésével NLP) valamilyen formája vagy technikája (Borg és mtsai., 2021) – így az ügyfélszolgálatok digitalizációjának része lett az NLP (Indurkhya és Damerau, 2010).

A mesterséges intelligencia technikák egyik ígérete, hogy komplex, nagy adathalmazokban is képesek az embernél hatékonyabban mintákat felismerni (Ferrario és Loi, 2022). A legelterjedtebb (legkézenfekvőbb) lehetőség az ún. chatbotok alkalmazása (Mnasri, 2019; Nguyen, 2019), mely lehet szabály alapú vagy nyílt szöveges megoldás. Ezen túlmenően felmerülhet telefonos beszélgetések értelmezése, közösségi média bejegyzések elemzése (pl. sentiment vagy topic analysis, vagy legalább a bejövő üzenetek tartalom alapján történő kategorizálása, előkészítése, vagy akár megválaszolása (Hardalov és mtsai., 2018; Hardalov és mtsai., 2019; Borg és mtsai., 2021). Ezek azonban sokszor nem elegendők önmagukban (Hardalov és mtsai., 2018; Saberi és mtsai, 2017). Ezért e technikák mellett egyre nagyobb figyelmet kapnak az összetetteb szöveg elemző technológiák, mint pl. a GPT-3 vagy újabban a BERT modellek (Devlin és mtsai.; 2018; Finardi és mtsai., 2021). Ezek lényege, hogy mélytanulásos megoldást alkalmaznak ún. 'transformer' hálózati felépítéssel és a 'figyelem'-re (attention) épülő modell alapelemekkel. A betanítás során igen nagyszámú (akár tízmilliárdos nagyságrendű) bemeneti paramétert alkalmazva, autoregresszív módon építik fel egy adott nyelv modelljét úgy, hogy a nyelvet reprezentáló adatok (a corpus) több tíz gigabájtnyi méretű is lehet (pl. rengeteg online weboldal, vagy hatalmas mennyiségű irodalmi mű), a céltól függően. Ebből adódik, hogy a modell kialakítása gépi erőforrás és idő igényes. Az elkészült modell azonban több célra használható (pl. szövegből hiányzó szavak beillesztésére), jellemzően úgy, hogy a modell köré összetettebb architektúrát építenek további, egyszerűbb modellekből. Előnyük, hogy egyrészt hosszabb (bár tokenizált) szövegeket is kezelni tudnak, másrészt ezt úgy teszik, hogy információt képesek kinyerni az adott szöveg vagy szöveg részek kontextusára vonatkozóan is, és ennek megfelelően javasolnak megoldást (a feladattól vagy kérdéstől függően). Egy ilyen modellt adott nyelvre elő kell készíteni, ami hatalmas erőfeszítés – viszont huBERT néven már magyar változat is elérhető (Nemeskey, 2021). Ugyanakkor a BERT modelleket más neurális háló megoldásokkal is lehet kombinálni a feladattól függően (Lu és mtsai.; 2020; Yang és Cui, 2021).

Jelen tanulmányban bemutatott kutatás a SZTAKI által kidolgozott huBERT alapmodellre (Nemeskey, 2021) építve vizsgált különböző architektúrákat és méreteket egy konkrét cég ügyfélszolgálatára beérkező (különböző hosszúságú) email üzenetek klaszszifikációs előkészítése céljából. A munka számos lépésben igyekezett egyre jobb teljesítményű architektúrát kialakítani.

2 Az eset háttere és adatok

A kutatás alapja egy email kategorizáló megoldás kidolgozása egy óvodai, iskolai étkeztetés adminisztrációját automatizáló cég (E-Menza, e-menza.hu) számára (Netkir, 2021). Regisztrált szülők a cég jelszóval védett webes felületén meghatározott beosztás szerint adhatják le a menzai étkezések megrendelését. A rendeléssel kapcsolatos kommunikáció (allergia, lemondás, stb.) illetve technikai jellegű kérések (pl. elfelejtett jelszó vagy egyéb bejelentkezési problémák) intézése alapvetően emailen keresztül történik. A rendelés időszaki jellege miatt az érkező e-mailek számában napi szinten igen kiugró értékek is lehetnek. Az is előfordulhat, hogy egy emailben több kérés vagy bejelentés is szerepel. Az e-maileket azok tartalmától függően kell elosztani az

ügyfélszolgálati referensek között, akik aztán döntenek más dolgozók (pl. pénzügyesek vagy IT technikusok bevonásáról).

A központi postafiókba összesen több mint 600 000 e-mail érkezett, azonban ezek többsége reguláris kifejezésekkel kezelhető volt (például 365 084 db rendelés visszaigazolás, 146 604 kézbesíthetetlen). A 10 000 körüli maradék halmazból csak az ügyfél indító levelét vettük figyelembe, a kommunikáció többi részét kiszűrtük. Így összesen 657 e-mail maradt. A postafiókba beérkező leveleket kézzel osztályozták a cég ügyfélszolgálati munkatársai és 10 kategóriát alakítottak ki a leggyakrabban előforduló problémakörökből: allergia, bankkártya, cafeteria, iskolaváltás, jelszó, kedvezmény, már beküldött kedvezmény, lemondás, regisztrációs kód és utalás. A betanításhoz mindegyik osztályba nagyjából 50 db e-mail került, legfeljebb 3-4 db email eltéréssel. Így egy kiegyensúlyozott bemeneti adathalmaz keletkezett, ami összesen 508 db e-mailt tartalmazott. A teljes maradék halmazból kiszűrtük az egyedi, sokféle kategóriába besorolható elemeket és így a 10 kategória eloszlása ott is hasonlóan egyenletes képet mutatott. Egyik kategóriában sem volt szignifikánsan több elem mint a többiben.

A feladat tehát elsőre egyszerű besorolásnak tűnhet (pl. a tárgy alapján). Ugyanakkor hamar kiderült, hogy az e-mailek címe nem mindig egyértelmű, az egyes ügyfelek szóhasználata széles skálán mozog. Sőt, a felhasználók több esetben a rendszertől kapott figyelmeztető levelekre válaszolva írták meg problémáikat egy teljesen más tárgykörben. Az e-mailek szövege igen eltérő hosszúságú is lehet. A fentiekből adódott, amit néhány egyszerű, hagyományos klasszifikációs gépi tanuló technika alkalmazásával végrehajtott teszt igazolt is (1. táblázat), hogy összetettebb NLP megoldás alkalmazása lehet célravezető. Az előzetes teszt eredményeként az e-mail tárgyát végül ki is kellett hagyni az elemzésből. A rendelkezésre álló kevés adat miatt nem tudtuk finomhangolni a BERT modellt, a teszteléshez pedig csak két halmazra bontottuk az adatokat és ezért olyan módszert is kerestünk, mely képes néhány példa alapján tanulni.

1. Táblázat: Viszonyítási alapul szolgáló gépi tanuló algoritmusok eredményei

Modell	Random Forest	Naive Bayes	Logistic Regression	SVM
Validációs pontosság	71,64%	76,36%	77,94%	83,65%

3 Alkalmazott módszerek

A megoldás kidolgozásához a BERT magyar változatára, a huBERT-re esett a választás mivel az ügyfélszolgálatokra beérkező e-mailek a jellemzően egyszerű hétköznapi nyelvezetet használnak. Az új modellek optimalizálása során a huBERT-hez tartozó súlyok változtatása le lett tiltva. Emellett olyan megoldásra volt még szükség, amely alkalmas arra, hogy már néhány példából is tanuljon és képes legyen azonnal osztályozni a beérkező emaileket. Így esett a választás a sziámi hálózat megoldásra, melyet az ún. 'egy-lövéses-tanulás' (One Shot Learning – OSL) megközelítés alatt szoktak említeni. A gyorsabb tanulást úgy éri el, hogy az eredeti multiklasszifikációs problémát áttranszformálja párok összehasonlítására úgy, hogy az eredeti feladatot megoldó már betanított neurális hálózatból indul ki és azt a modellt duplikálva beágyazza egy

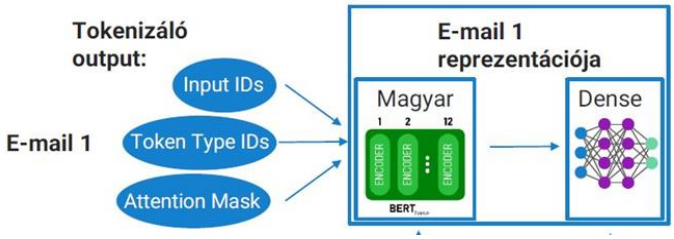
összehasonlító hálózati architektúrába (lásd lent). Ennek eredményeként a beágyazott hálózat által kialakított reprezentáció oly módon változik, hogy az egyes osztályok egymástól eltérő tulajdonságait próbálja detektálni és minél jobban elkülöníteni.

Programozásra a Python programnyelvet használtuk skálázhatósága, teljesítménye és a vállalati folyamatokba való integrálhatósága okán. A fejlesztés a Jupyter Notebook (v. 5.0), illetve az Anaconda (v. 3.8) környezetek segítségével történt. Általában az UTF-8 kódolás volt a jellemző. Egy e-mail lehet egy részből álló egyszerű szöveg (text/plain) vagy több részből álló (multipart). A HTML részeket kiszűrtük és egyszerű szöveggé konvertáltuk. Dekódoltuk a feladó, tárgy és törzsszöveg tartalmát. A törzs esetében megvizsgáltuk a részeket és több rész esetén megkerestük az első egyszerű szövegrészt, majd betöltöttük egy adattáblába. A tisztított törzsszöveget adtuk át a nyelvi modellnek. A BERT kétféle kimenettel dolgozik: ezek a teljes szekvenciát egyetlen 768 hosszú vektorral reprezentáló pooled output és az előzővel megegyező dimenziószámú vektorokat tokenenként legeneráló sequenced output.

A fenti környezetben a már előzetes tesztek után négy modell kiépítése történt meg, melyek mindegyikén több paraméter értékét is megvizsgálta a kutatás. A már betanított és működő de átlagos eredményt elérő osztályozót kellett első lépésben felépíteni, melyet később a sziámi beágyazott hálózatává alakítottunk. Az első kimenet kipróbálása után áttértünk a szekvenciális kimenet felhasználására és további rejtett rétegek hozzáadásával javítottunk az eredményen. Először magának az alapmodellnek a kialakításánál több lépésben került kiválasztásra egy LSTM-mel bővített megoldás, majd a sziámi architektúra alkalmazására került sor, a kész alapmodellre építve. A sziámi struktúrában egyrészt a belső hálózatok méretét (neuron számát) lehetett változtatni, másrészt a speciálisan kialakított batchek elemszámát, valamint a tanulási rátát optimalizálni. A kimeneti (osztályozásnál, illetve összehasonlításnál két aktivációs függvény vizsgálatára került sor: a két alapmodellnél softmax, a one-shot rendszernél pedig szigmoid.

3.1 huBERT + sűrű (Dense) multiklasszifikációs osztályozó

Az első modell, amely már huBERT megoldást használt (a tesztelés egyszerű gépi tanulási technikái helyett) egy bemeneti tokenizálóból és a huBERT utáni egyetlen sűrű (Dense) neurális háló rétegből állt (lásd 1. ábra).



1. ábra: huBERT + sűrű (Dense) multiklasszifikációs osztályozó

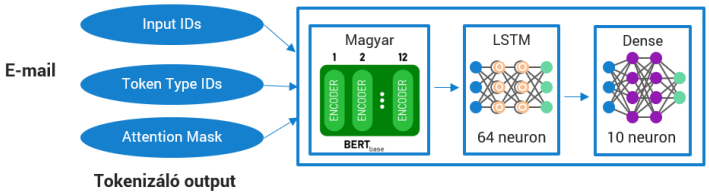
Az emailek feldolgozása során 64 tokenben maximalizáltuk a szekvencia hosszát. Az ennél rövidebb emaileket 0-kal kiegészítettük (padding) a maximális hosszig és attention mask jelezte, hogy ezeket a részeket ki kell hagyni a jelentés kialakítása szempontjából. Ezzel sikerült kiküszöbölni, hogy az E-Menza által küldött figyelmeztető

email is bekerüljön az értelmezett szövegbe. Az adattáblánk email törzsszöveg mezője alapján előre elkészítettük a BERT használatához szükséges 3 vektort: a bemeneti azonosítókat, típus azonosítókat és a figyelem maszkot. A huBERT szekvenciális kimenetét kifeszítés után egy 64 neuronból álló sűrű (Dense) réteghez kötöttük, amelynek rendeltetése egy 64 dimenziós reprezentáció kialakítása volt a huBERT 64 token x 768 vektor = 49 152 elemű kimenetéből. Ezzel a leszűkítéssel a neurális háló a legfontosabb jellemzőket kiemelve megfelelő, tömör reprezentációt alakít ki, amelyet már könnyebb osztályozni. Végül egy, a 10 kategóriának megfelelően 10 neuronból álló sűrű (Dense) réteg következik softmax aktivációs függvénnyel, amely egy valószínűségi eloszlást ad vissza. Az eloszlás legnagyobb értéke a megjósolt kategóriának megfelelő neuron aktivitásában mutatkozik meg.

A tokenizált bemeneteket betanító halmazra és validációs halmazra osztottuk. Fontos, hogy a validációs halmazba kiválasztott 10% megtartsa ugyanazt az eloszlást az osztályok közt, ami a teljes adathalmazra jellemző. 457 elem a betanító, 51 elem pedig a validációs halmazba került. A címkék szerinti eloszlást (stratify) megtartva a minimális elemszámbeli eltéréseket is sikeresen kezeltük. Az eloszlás megtartásától függetlenül a választás véletlenszerű, így a későbbi ellenőrzés, visszatöltés céljából a tanító és validációs halmazokat lementettük. Ez biztosítja, hogy a modell pontosan ugyanazt a tanító és teszt halmazt alkalmazza.

A modell paraméterezésekor az Adam optimalizálót adtuk meg, amely a momentum és az adaptív optimalizálás elvén működik. Az alapértelmezett tanulási rátát (0.001) nem módosítottuk. A loss értékét csak a pontosság mérésével egészítettük ki, a loss csökkentéséhez a multiklasszifikációs esetekben szokásos categorical_crossentropy függvényt adtuk meg. A batch méretét az adatok nagyságrendjéhez illeszkedően 64 hosszúságúra választottuk és az epochok számát többszöri futtatás után 20-ra állítottuk, mert a betanítási és validációs loss gyakorlatilag 15 epoch után ellaposodott. A modell összesen 113 764 554 paraméterrel rendelkezik, amely elsősorban a rétegek közti kapcsolatok súlyát jelenti. Mivel a huBERT rétegre letiltottuk a betanítást, ezért a modellnek csak 3 146 442 paramétere változtatható. A betanítás legvégén elmentjük a modell súlyait, hogy a későbbiekben bármikor visszatölthessük.

3.2 huBERT + LSTM + sűrű (Dense) multiklasszifikációs osztályozó



2. ábra: huBERT + LSTM + sűrű (Dense) modell felépítése

Az LSTM egy speciális RNN, ami a nagyobb távolságra elhelyezkedő, időben hosszú távú kontextust is képes elsajátítani. Különösen az információ hosszú távú megjegyzésére készítették őket. Egy szövegalapú NLP feladatban ez kritikus, hiszen akár több mondattal az aktuális előtt is lehet a szükséges információ. Ezért a jelentéstartalom

reprezentáció kiszámítására a sűrű hálózat helyett ezt alkalmaztuk. Az LSTM réteg 64 neuront tartalmazott és ezt követte egy 10 neuronból álló kimeneti Dense réteg a klaszszifikáció kiszámítása céljából (2. ábra). A modell összesen 110 832 010 paraméterrel rendelkezik, amelyből 213 898 paraméter tanítható.

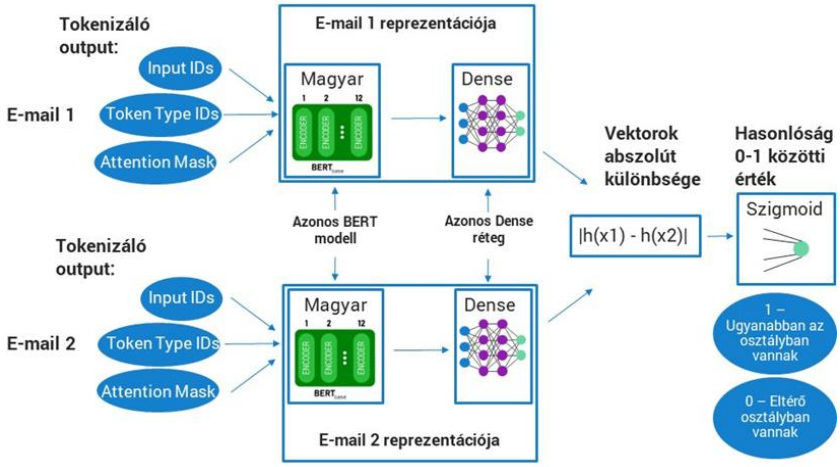
3.3 OSL sziámi hálózat két bemenettel és huBERT + sűrű (Dense) osztályozóval

Az előző architektúrák a multiklasszifikációs osztályozási feladatra jelentenek megoldást. Az egyes emailek összehasonlításán alapuló sziámi hálózatban a fenti architektúrákat beágyazott modellként használtuk fel. Ebben a modellben a huBERT + sűrű (Dense) hálózatot illesztettük be. A funkciója olyan reprezentáció kialakítása, amely az egy osztályba tartozó emaileket közelíti egymáshoz, míg a külön kategóriába tartozókat távolítja egymástól. A 2 bemenetre összeállított 16 hosszú reprezentációs vektorokat a dimenziónkénti abszolút különbségük kiszámításával a végső rétegben egy bináris döntéshez használjuk fel: a 2 vizsgált email ugyanabba vagy eltérő osztályba sorolható-e.

A huBERT-nek továbbított 3, tokenizált adatokat tartalmazó vektort kétszer tároljuk a bal, illetve jobb bemenetben egy tömbbé konkatenált formátumban. A huBERT modell után egy reprezentációt szűkítő 16 neuronból álló sűrű réteg kerül. Ezután a megduplázott hálózat kimenetét egy speciális Keras Lambda réteg kapja meg. A szokásos matematikai függvényeket nem szabad használni, hanem a Keras Backend könyvtára segítségével kell megadni a számításokat, mert ellenkező esetben megakadályozzuk a hálózat tanulását (csak ugyanazon loss értékek körül fog ugrálni a betanítás során). A Lambda réteg a párba állított reprezentációs vektorok dimenziónkénti abszolút különbségét számítja ki. A 16 dimenzió különbségét végül az egyetlen neuronból álló sűrű réteg kapja meg, ahol a szigmoid aktivációs függvény kiszámítja a hasonlósági értéket, ami 0 és 1 közötti érték. Ez a hasonlósági érték: 1, ha azonos osztályban vannak és 0, ha eltérő osztályba tartoznak az e-mailek. A 3. ábrán látható a teljes hálózat felépítése.

A súlyokat és az eltolást (bias) a belső sűrű rétegnél speciálisan inicializáltuk. A súlyoknál 0 középpontú, 10^{-2} varianciájú normális eloszlással dolgoztunk. Az eltolás esetén pedig egy 0,5 átlagú, előzővel azonos varianciájú normális eloszlást használtunk. A véletlenszerű inicializálással szemben ez a módszer biztosítja, hogy a betanítás indulásakor a kezdeti loss értékek sokkal jobbak legyenek és ennélfogva gyorsabban haladjon a hálózat.

A párosítással létrejött összes kombináción futtatni a modellt nagyságrenddel nagyobb számú iterálást jelentett volna, ezért a batchek saját összeállításával tartottuk kordában a lefutási időt. A 64-es batch méret helyett 16 email párt tartalmazó kötegelést alkalmaztunk, amelyen belül egyenlő arányban mutattunk véletlenszerűen kiválasztott pozitív (egyező kategória) és negatív (eltérő kategória) példákat a hálózatnak. Ha nem lenne kiegyensúlyozott a betanító halmaz, akkor ezzel a módszerrel biztosítani tudjuk az egyenlő arányokat is. Vegyük észre, hogy N elemű halmaz esetén a sziámi módszer N^2 nagyságrendű input teret alakít ki és ezért működik kiválóan kis N-ekre is. Például 10 email esetén (10x9)/2 pozitív párt tudunk összeállítani, ha a felcseréléseket nem számoljuk. A módszer másik következménye, hogy a betanítás nem epoch alapon zajlott, hanem a train_on_batch eljárás felhasználásával a saját párosításokkal. Ezért az iterációk számát a 20 epoch helyett 2000 batch-re kellett növelni.



3. ábra: OSL hálózat huBERT + sűrű (Dense) beágyazott modellel

Az Adam optimalizálónál csökkentettük az alapértelmezett tanulási rátát 6 x 10^{-5}-re, amely elegendőnek bizonyult a konvergencia eléréséhez. A modell nem módosítható paramétereinek száma 110 618 112, amely elsősorban a BERT modell súlyait jelenti. A 786 465 optimalizálható súlyt a BERT és az utána kötött 16 neuronból álló sűrű réteg kapcsolatainak a száma (49 152 x 16 = 786 432) plusz a sűrű és Lambda réteg közötti kapcsolatok (16), végezetül a Lambda és a végső 1 neuronnal rendelkező sűrű réteg összekötése (16 súly + 1 eltolás) biztosítja.

Az eredeti multiklasszifikációs probléma megoldása érdekében generáltunk egy minden osztályt pontosan egyszer magába foglaló támogató halmazt véletlenszerűen kiválasztott 1-1 db e-maillel. A támogató halmaz minden elemét összehasonlítottuk az OSL segítségével a kategorizálásra váró e-maillel. A legnagyobb hasonlósági érték alapján döntöttük el, hogy az adott e-mail melyik osztályba tartozik.

3.4 OSL sziámi hálózat 2 bementtel és huBERT + LSTM + Dense osztályozóval

A legtöbb paraméterben és működésében megegyezik az előző sziámi hálózattal, kivéve a beágyazott modellt, amely bővült egy LSTM réteggel, felhasználva a már letesztelt 2. multiklasszifikációs modellt. A gyorsabb betanítás érdekében dinamikus tanulási rátát vezettünk be. A betanítás elején jobb, ha nagyobb a ráta, mert nagyobbakat lépünk a gradiens irányába, míg a végén a minimum megtalálása érdekében, már egyre kisebb értékek alkalmazása a célravezető. A kezdeti 0,006 az összes iteráció legvégére 0,006 x 0,1(2000/2000)= 0,0006 értékre csökken. A fix tanulási ráta esetén majdnem 20.000 iteráción keresztül kellett tanítani a hálózatot a megfelelő eredmény eléréséhez.

4 Eredmények és azok diszkussziója

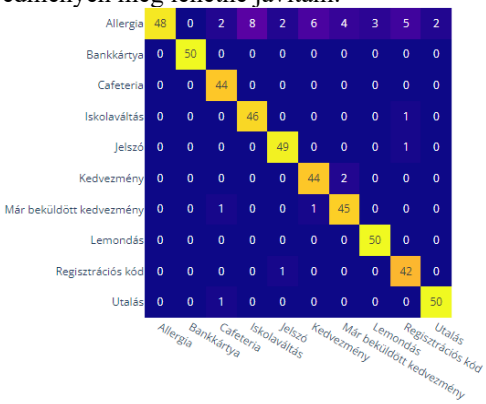
4.1 huBERT + sűrű (Dense) multiklasszifikációs osztályozó

A betanítási loss értéke a kezdeti 4 feletti értékről 0,008-0,007 szintekre csökkent, a pontosság 99,78%-on állt meg. De azért fontos az ellenőrzés a teszt halmazon az algoritmus által még nem látott emailekkel, hogy kiderüljön megfelelő általános szabályokat sajátított el a rendszer vagy sem. A validációs loss 1,41, míg a pontosság 68,6% körül alakult. A rendszer túlillesztette a betanító halmazt. A sűrű rejtett réteg esetén várható volt ez az eredmény, mivel ez a fajta réteg teljesen összekötött az előző réteg minden kimenetével és az adatok lineáris kombinációját képes csak előállítani, ami gyakorlatilag a klasszikus kulcsszó kiválasztási módszerekhez hasonlítható.

A teljes 508 számosságú halmaz tanulás utáni reprezentációs vektorait kinyertük. A TruncatedSVD és PCA függvények kétféle eljárással a 64 dimenziós vektorteret 3 dimenzióra redukálják. Az osztályok elkülönítése csak minimálisan sikerült, a legtöbb kategória feltűnően erős átfedésben volt.

4.2 huBERT + LSTM + sűrű (Dense) multiklasszifikációs osztályozó

A betanítási loss értéke a kezdeti 2 feletti értékről 0,36-ra csökkent, a pontosság 94%-on állt meg a betanító halmazon, de a validációs loss 0,76, míg a pontosság 80% körül alakult. A különbség csökkent a két görbe között és láthatóan megszűnt a túlillesztés problémája, de az eredményen még lehetne javítani.



4. ábra: huBERT + LSTM + sűrű multiklasszifikációs hálózat konfúziós mátrixa

A konfúziós mátrix (lásd 4. ábra) mind a 10 lehetséges kimenetre megmutatja hogyan viszonyulnak a modell által készített előrejelzések az email tényleges kategóriájához képest. Minden lefutás során tisztán látszott, hogy az „Allergia” kategória besorolása okozott problémát a modellnek. Az allergia osztály nehézségét az adja, hogy gyakran csak egy bizonyos összetevőre való érzékenység szerepel a törzsszövegben, az

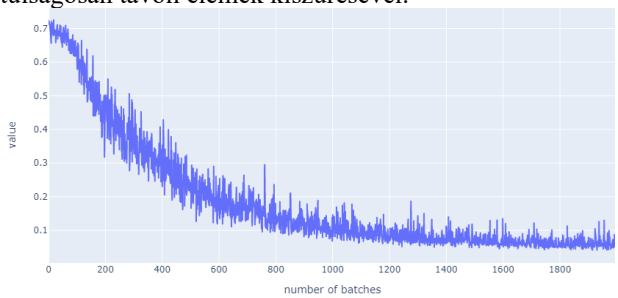
allergia szó maga nem fordul elő és ezzel összefüggésben új étrend kérése vagy a speciális étlap keresése, esetleg lemondás is fellelhető ezekben a levelekben.

4.3 OSL sziámi hálózat két bemenettel és huBERT + sűrű (Dense) osztályozóval

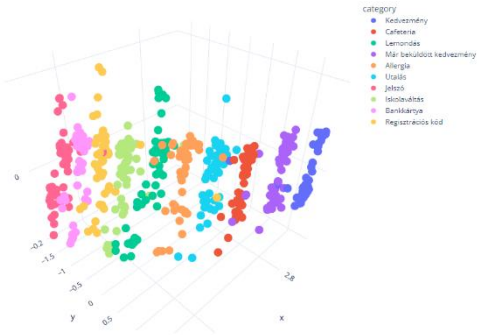
A betanítási loss értéke a kezdeti 0,8 körüli értékről 0,4-re csökkent, a legjobb lefutás pontossága 94%-on állt meg a betanító halmazon, miközben a validációs pontosság 82% volt. A huBERT + sűrű hálózat kombinációjából adódó túlillesztés megmaradt és jelentősen nőtt a véletlenszerű párok miatt a loss görbék volatilitása. A sűrű háló korlátozott képességei ellenére az osztályok rendezése elkezdődött, azonban a legtöbb kategóriának sok elszórt, a saját síkjától távoli pontja van.

4.4 OSL sziámi hálózat 2 bementtel és huBERT + LSTM + Dense osztályozóval

A betanítási loss értéke a kezdeti 0,7 körüli értékről 0,063-ra csökkent, a legjobb lefutás pontossága 96,88%-on állt meg a betanító halmazon, miközben a validációs pontosság 94% volt (5. ábra). A loss függvények volatilitása jelentősen csökkent és a betanítás során végig közel mozognak egymáshoz, azaz nem történt túlillesztés. Az OSL áttranszformálja az eredeti multiklasszifikációs problémát egy bináris páronkénti összehasonlítási feladattá. A hasonlósági értékekre támaszkodva generáltunk egy minden osztályt pontosan egyszer magába foglaló támogató halmazt véletlenszerűen kiválasztott 1-1 db e-maillel. A legnagyobb hasonlósági érték mutatja meg, hogy melyik kategóriába tartozik az adott e-mail. A teljes bemeneten 200 iterációval elvégzett támogató halmaz alapú kiértékelés pontossága 94,5% lett. A 'plotly' modul segítségével a vektorokat 3D interaktív grafikonon ábrázoltuk (6.ábra): mindegyik kategória jól látható módon elkülönül, síkokba rendeződik és viszonylag kevés olyan pont van, mely a saját osztályától távolabb helyezkedik el. A vizuális megjelenítés megerősíti a betanítás végén mért eredményeket. Néhány különálló, eltérő színű pontot még mindig látunk, amik akár egy másik kategória pontjai közé kerültek. Ezeket külön megvizsgáltuk és több esetben is kiderült, hogy emberi félrecímkézésről van szó vagy nem odaillő témakörű levélről. Ezt a fajta adattisztítást a jövőben automatizálni is lehet az adott kategória sűrűsödésétől túlságosan távoli elemek kiszűrésével.



5. ábra: OSL huBERT + LSTM + Dense betanítási loss lefutása



6. ábra: OSL huBERT + LSTM + sűrű háló belső reprezentáció 3D ábrázolása

4.5 Hálózatok eredményeinek összehasonlítása

Az eredmények jobb kiértékelése érdekében klasszikus, kevesebb erőforrást igénylő gépi tanuló eljárásokkal (SVM, Naive Bayes stb.) is elemeztük az adathalmazt. Tokenizálás, szótövezés, stop szavak kiszűrése után alkalmaztuk a TF-IDF módszert és kapott eredményhalmazon 10-szeres keresztvalidációval futtattuk az algoritmusokat. Az eredmények a korábban bemutatott 1. táblázatban láthatók.

Az eredmények igazolják a feladat megoldására egymás után elkészült modellek folyamatos javulását a legfontosabb metrikák, a validációs pontosság, a validációs loss és az F1 pontszám mentén (2. táblázat).

2. Táblázat: Az egyes hálózatok jellemzőinek és eredményeinek összehasonlítása

Modell	Iterációk	Tanítható paraméter	Tanítási pontosság	Validációs pontosság	F1 pontszám	Tanítási loss	Validációs loss
huBERT + Dense multiklasszif.	20 epoch	3 146 442	99,78%	68,6%	60,82%	0,007	1,41
huBERT + LSTM + Dense multiklasszif.	20 epoch	213 898	96,4%	80%	80,24%	0,36	0,76
OSL huBERT + Dense	2000 batch	786 465	Hasonlóság: 94%	Hasonlóság: 82% Támogató. halmaz: 88,5%	88,23%	0,42	0,61
OSL huBERT + LSTM + Dense	2000 batch	215 361	Hasonlóság: 96,88%	Hasonlóság: 94% Támogató. halmaz: 94,5%	98%	0,063	0,23

Az első modell túlillesztett betanítása jól látható: több, mint 30%-kal alacsonyabb a validációs pontosság a betanító halmazon mért értéknél. Az LSTM réteg beillesztése sikeresen orvosolta a hibát, a betanítási pontosság csak minimálisan esett vissza, ellenben 12%-ot nyert a rendszer a validációs metrikán úgy, hogy a validációs loss a felére csökkent. Az OSL modellek támogató halmazon történő kiértékelése a korábbiakkal összehasonlítható mérés, hiszen az eredeti feladatra adja vissza a teljesítményt. A lefutás során kapott értékek azonban az összehasonlítási feladatra vonatkoznak, amit a 2. táblázatban tüntettünk fel. Végül a legjobb eredmény 94,5% és 0,23 volt, ahol utóbbi érték az első loss érték kevesebb, mint ötöde. Az F1 pontszám a multiklasszifikációs modellek esetén kisebb vagy egyenlő a pontossággal, míg az OSL modellek esetén felülmúlja a hasonlósági pontosságot.

Az OSL architektúrára áttérés egy nagyobb volatilitást is behozott a rendszerbe, hiszen a véletlen hatás felerősödött a párba rendezés és a batchek összeállítása során. A fejlesztés előrehaladásával a tanulási rátát úgy állítottuk be, hogy minél kevesebb iteráció alatt eljusson a loss függvény az ellaposodott szakaszába. Az utolsó, LSTM réteget felhasználó modell látványosan jobban kordában tartotta az ingadozásokat ezen a szakaszon. Úgy véljük, az LSTM réteg hatékonysága és a tanulási ráta beállítása elégséges betanítási idővel kulcsfontosságú a hálózat által elért eredményben.

Az első két multiklasszifikációs modell bár osztályonként térségekre bontotta a teret, az ezek közti átfedés hibás kategorizáláshoz vezetett. Az „Allergia” kategória elkülönítése bizonyult a legnehezebb feladatnak. Az OSL architektúrák síkokat alakítottak ki és a síktól távol eső pontok száma, ezáltal a keveredés jelentősen csökkent. Érdekesség, hogy kiderült, a belső reprezentáció kinyerése és ábrázolása felhasználható az emberi pontatlanságok azonosítására is.

5 Összefoglalás és további kutatások

A cikkben bemutatott gyakorlati kutatás célja egy valós ügyfélszolgálati probléma, egy adott céghez beérkező nagyszámú emailek minél pontosabb kategorizálásának a megoldása volt. A kiindulási pont a cég által megadott 10 kategória és 508 előre besorolt email volt. A kutatás során számos különböző megoldás szisztematikus vizsgálatán volt a hangsúly. Nem csak különböző MI alapmodellek, hanem eltérő architekturális megoldások vizsgálatára is sor került – a modellek paramétereinek változtatása mellett. Mint az eredmények igazolták, egy ilyen lépcsőzetes építkezéssel a közbülső eredmények figyelembevételével jelentősen javíthatók a megoldás kimeneti paraméterei. Nemcsak a huBERT sokoldalú felhasználhatósága derült ki, de a Sziámi architektúra ’one-shot-learning’ ígérete is reálisnak bizonyult.

A fent bemutatott legjobb megoldás valós helyzetben való tesztelése (és ezzel a bevezetés előkészítése) jelenleg folyik, illetve további architektúrák kiépítése és az osztályozás több-címkés (multi-label) módban történő vizsgálata szerepel a tervekben.

Köszönetnyilvánítás

A cikk szerzői köszönetüket fejezik ki a Netkir Zrt., az E-Menza licencet birtokló cég irányába, valamint a Webra Kft. felé, amely az E-Menza rendszer fejlesztését végzi.

Bibliográfia

Borg, A., Boldt, M., Rosander, O., Ahlstrand, J. : E-mail classification with machine learning and word embeddings for improved customer support. Neural Computing and Applications, 33(6), 1881–1902 (2021).

Cassandra, C., Hartono, S., Karsen, M. Online Helpdesk Support System for Handling Complaints and Service. 2019 International Conference on Information Management and Technology (ICIMTech), pp. 314-319 (2019).

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. . BERT: Pre-training of deep bidirectional transformers for language understanding. ArXiv Preprint ArXiv:1810.04805 (2018).

Ferrario, A., Loi, M.: Algorithm, machine learning and artificial intelligence. Elgar Encyclopedia of Technology and Politics, 37(1), 135 (2022).

Finardi, P., Viegas, J.D., Ferreira, G.T., Mansano, A.F., Caridá, V.F.: BERTaת: Itaת BBERT for digital customer service. ArXiv Preprint ArXiv:2101.12015 (2021).

Hardalov, M., Koychev, I., Nakov, P.: Towards automated customer support. International Conference on Artificial Intelligence: Methodology, Systems, and Applications, pp. 48–59 (2018).

Hardalov, M., Koychev, I., Nakov, P.: Machine reading comprehension for answer re-ranking in customer support chatbots. Information, 10(3), 82 (2019).

Indurkhya, N., Damerau, F.J.: Handbook of natural language processing. Chapman and Hall/CRC (2010).

Lu, Z., Du, P., Nie, J.-Y.: VGCN-BERT: augmenting BERT with graph embedding for text classification. European Conference on Information Retrieval, pp. 369–382 (2020).

Mnasri, M.: Recent advances in conversational NLP: Towards the standardization of Chatbot building. ArXiv Preprint ArXiv:1903.09025 (2019).

Nemeskey, D.M.: Introducing huBERT. In Berend G., Gosztolya G. Vincze V. (Eds.), XVII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 29-36. Szegedi Tudományegyetem (2021)

Netkir Zrt.: Menza, E-Menza. www.e-menza.hu (2021).

Nguyen, T.: Potential effects of chatbot technology on customer support: A case study. Master's thesis. Http://urn.fi/URN:NBN:fi:aalto-201906233987, Aalto University (2019).

Saberi, M., Hussain, O.K., Chang, E.: Past, present and future of contact centers: A literature review. Business Process Management Journal, 23(3), 574-597 (2017).

Yang, Y., Cui, X.: Bert-enhanced text graph neural network for classification. Entropy, 23(11), 1536 (2021).

Zendesk: The Zendesk Benchmark-Your Prescription for Optimizing Customer Service. https://d16cvnquvjw7pr.cloudfront.net/resources/whitepapers/Zendesk_WP_benchmark.pdf (2020).

HuBERTUSz: Alacsony paraméterszámú transzformer modellek létrehozása és kiértékelése magyar nyelvre

Ficsor Tamás, Berend Gábor

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

Kivonat Hazánkban is megnőtt az érdeklődés a transzformer modellek alkalmazása iránt. Éppen ezért a modern előtanítási standardoknak megfelelően (pl. dinamikus maszkolás, mondatsorrend-predikció használata) közzétesszük az általunk előállított *tiny*, *small* és *medium* transzformer-variánsokat, amelyeket alapos kiértékelésnek vetettünk alá különböző feladatokon. Eredményeinkből kitűnik, hogy elsődlegesen a tokenosztályozási feladatokon a jóval kisebb paraméterszámmal bíró modellvariánsaink is képesek megközelíteni a nagyságrendekkel több paraméterrel bíró társaik eredményeit.
Kulcsszavak: Kompakt modellek, Előtanítás, Névelemfelismerés, Szófaji egyértelműsítés, Szentimentpredikció, Természetesnyelvi következtetés

1. Bevezetés

A transzformerek (Vaswani és mtsai, 2017) megjelenése nagy hatással volt a természetesnyelv-feldolgozásának folyamatára, amelyek kimagasló teljesítményt képesek elérni, azonban a paraméterek hatékony fölhasználásától még távol állnak. Az hogy pontosan mennyi paraméterrel oldható meg kellő minőségben egy bizonyos feladat, erősen függ annak komplexitásától, továbbá a feladattal kapcsolatban elérhető adat mennyiségétől és minőségétől is (McCoy és mtsai, 2019; Bhargava és mtsai, 2021).

A nyelvi modellek tipikusan több különböző méretkonfigurációban is elérhetők a felhasználók számára (Turc és mtsai, 2019). Egy adott konfiguráció a modell paramétereinek mennyiségét határozza meg, amely többek között a rétegek számát, fejek számosságát és a modellben használt vektorok dimenzióját foglalja magában. Egy-egy probléma megoldására igyekezhetünk az erőforrásaink és a szükségleteink szerint választani ezeket a paramétereket és egy egyedi konfigurációt kapunk. Az egyik legelterjedtebb konfiguráció a magyarra is elérhető ún. alap (*base*), amely ~ 110 millió paramétert tartalmaz.

Sok esetben *„ágyúval lövünk verébre”*, amikor a megoldani kívánt feladatot az elégségesnél – adott esetben nagyságrendekkel – nagyobb paraméterszámmal rendelkező modellel próbáljuk megoldani. Azonban ha rendelkezésre állnak kisebb paraméterszámú modellalternatívák, érdemes lehet először azok használatát feltérképezni. Amennyiben egy kisebb modell használatával is kielégítő eredményt

tudunk elérni, akkor az hosszútávon időbeli, anyagi és környezeti terhelést (CO2 emisszió) csökkentő megtérülésekkel jár.

Ebben a cikkben több, kisebb paraméterszámmal rendelkező magyar nyelvű BERT variáns előtanításával és változatos alkalmazásokban történő kiértékelésével foglalkozunk. Létrehozott modelljeinknek a *Hungarian BERT from University of Szeged* nevet adtuk, amit `HuBERTUSz`-ként rövidítünk. Az előtanítás folyamatát igyekeztünk a modern standardoknak megfelelően előkészíteni, létrehozott modelljeinket pedig különböző, diverz adathalmazokon nyújtott eredmények mentén vetjük össze több magyar nyelvet támogató és könnyen hozzáférhető transzformer modell használatával. Az általunk előtanított modelleket könnyen hozzáférhető módon, a HuggingFace felületén elérhetővé is tesszük[1].

2. Kapcsolódó irodalom

A transzformerek (Vaswani és mtsai, 2017; Devlin és mtsai, 2019) megjelenésével egy új hullám kezdődött számos kutatási területen: mostanra bevett gyakorlattá vált a transzformer-alapú modellek használata például kép-, hang-, és kódfeldolgozási (Dosovitskiy és mtsai, 2020; Hsu és mtsai, 2021; Feng és mtsai, 2020) feladatokon egyaránt. Ezen modellek előállítása rengeteg adatot és számítási erőforrást igényel, ezért a modelleket egyszer előtanítják, és ezt követően csak a kívánt feladatokon finomhangolják őket.

A magyar nyelvvel előtanított egy- és több nyelvet is támogató transzformer modellek is elkezdtek megjelenni az elmúlt évek folyamán. Az egynyelvű, kizárólag a magyar nyelv feldolgozására létrehozott maszkolt nyelvi modellezést végző transzformereket tekintve is számos lehetőség érhető el mostanra (Nemeskey, 2020, 2021; Feldmann és mtsai, 2021; Yang, 2022a; Yang és Váradi, 2021; Ficsor és mtsai, 2022; Orosz és mtsai, 2022; Yang és mtsai, 2022; Yang, 2022b).

Számos, több nyelvet támogató modell is megjelent, amelyeknek támogatott nyelvei között a magyar is szerepel. Ilyen modellek például az mBERT (Devlin és mtsai, 2018), XLM-RoBERTa (Conneau és mtsai, 2020) és RemBERT (Chung és mtsai, 2020). Az mBERT egy klasszikus BERT modell, ami 104 nyelven lett tanítva. Az XLM-RoBERTa, a RoBERTa (Liu és mtsai, 2019) előtanítási mechanizmusát követő, 100 nyelvre betanított modell. A RemBERT esetén a be- és kimeneti beágyazások szétválasztására tettek javaslatot (általában ezek a súlyok meg vannak osztva egymás között), és külön dimenzióméreteket rendelnek azokhoz, ezzel a generalizációját a modellnek tudják befolyásolni. Ez a modell rendelkezik azzal a jó tulajdonsággal, hogy a finomhangolás során a paraméterek száma jelentősen csökkenthető a kimeneti beágyazás eldobásával.

Az évek során számos arany és ezüst minőségű annotációval ellátott adathalmaz vált elérhetővé magyar nyelvre. Névelemfelismeréshez a SzegedNER (Szarvas és mtsai, 2006), NerKor (Simon és Vadász, 2021) és a NerKor 1.41e (Novák és Novák, 2022), szekvenciaosztályozáshoz egyebek mellett az OpinHuBank (Miháltz, 2013) egy gyakran használt adathalmaz, amelyben egyes személyekhez kapcsolódó szentimentkategóriát kell prediktálni, azok szövegkörnyezetének

[1] `https://huggingface.co/SzegedAI`

függvényében. További kiértékelési lehetőségeket biztosít a HuLU (Ligeti-Nagy és mtsai, 2022) (Hungarian Language Understanding Benchmark Kit), ami 6 adathalmazt tartalmaz.

A magyar nyelvi erőforrások kiértékelésére több munka is született már. Ács és mtsai (2021) a HuBERT modellt vetette össze 4 további többnyelvű modell használatával a névelemfelismerési, szófaji és morfológiai egyértelműsítési feladatokon. Az egyes transzformerek teljesítményét rétegekre bontva is megvizsgálták, ami elég nagy mélységet ad az eredményeknek. Simon és mtsai (2022) a NerKor és SzegedNER korpuszok alkalmazását járja körbe különböző ipari minőségű nyelvi keretrendszer (vagy modell) alkalmazása mellett.

3. HuBERTUSz előtanítása

A BERT-jellegű transzformer modellek hiperparamétereinek megválasztásával, valamint az előtanítás pontos menetével kapcsolatosan változatos – és időnként hiányosan dokumentált – megoldásokkal lehet találkozni az irodalomban. Munkánkban, a transzformer modelljeink létrehozása során elsődlegesen a Nemeskey (2020) által alkalmazott módszertant követtük.

Erőforráskorlátok miatt tanítókorpuszként a Hungarian WebCorpus 2.0 Wikipedia alkorpuszára támaszkodtunk. A számítási igény mérséklésére azt az általánosan alkalmazott stratégiát követtük, amely az előtanítást két fázisra osztja úgy, hogy a lépések első 90%-ában redukált, legfeljebb 128 szubtokennyi hosszal rendelkező inputok feldolgozása történik meg, míg a tanítás második, befejező fázisában az inputok hossza elérheti az 512 szubtokennyi hosszt is akár. A szöveg tokenizálására a HuBERT szótárát[2] használjuk fel.

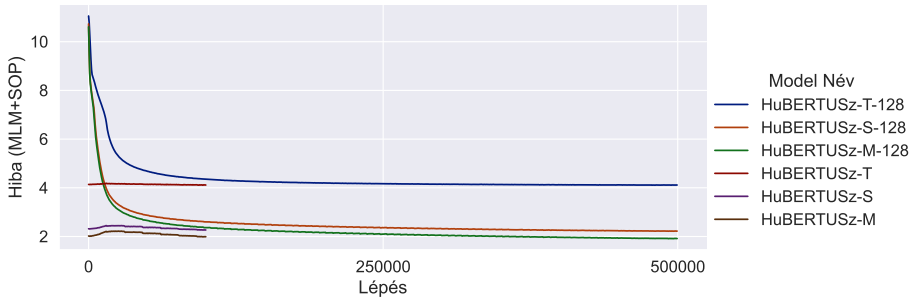
Liu és mtsai (2019) rámutattak, hogy a dinamikusan maszkolt nyelvimodellezés (MLM) – a BERT tanításánál is alkalmazott statikus párjával szemben – jobb általánosítóképességgel rendelkezik. Ennél fogva mi is ezt követjük, ami az alacsonyabb paraméterszámú modelljeink teljesítményét javíthatja, kisebb adathalmazon történő előtanításhoz mérve.

Lan és mtsai (2019) is rámutatott arra, hogy az eddig megszokott Next Sentence Prediction (NSP) feladat túl egyszerű a modell számára. Mivel az NSP során a feladat egy egyszerű témafelismerési feladatként viselkedik, helyette a Sentence Order Prediction-re (SOP) tesznek javaslatot, ami jobban képes a mondatok közötti koherenciára fókuszálni.

Az általunk tanított modellek dinamikus MLM-et és SOP-t oldanak meg, ahol a maszkolt nyelvi modellezés során alkalmazott maszkolási valószínűség megegyezik a BERT esetében alkalmazott értékkel ($p = 0,15$). A különböző modelljeink konfigurációi áttekinthetőek az 1. táblázat első 3 sorában. Három különböző méretű modellt (`tiny (T)`, `small (S)` és `medium (M)`) állítunk elő, amelyek checkpointjait megosztjuk az első és második fázis végéről is.

Az előtanítások során 64-es gradiens akkumuláció alkalmazásával 1024-es batchméretet értünk el, 10^{-4} tanulási rátával, 15000-es warmuppal, 500000 lépésen keresztül. A második fázisban a batch méretet 384-re állítottuk (batch =

[2] `https://huggingface.co/SZTAKI-HLT/hubert-base-cc`



1. ábra. A modellek előtanítása során a veszteségfüggvény alakulása. A `*-128` elnevezésű modellek az első fázis eredményeit, az ezen jelölést mellőző modellek pedig a második fázis modelleit jelölik.

16, gradiens akkumuláció = 24), és 100000 lépést hajtunk végre 512-es maximális szekvenciahossz mellett. You és mtsai (2019) ajánlása szerint a második fázisban ún. *re-warm-up*-ot alkalmazunk, vagyis a második fázis kezdetekor – az addigra alacsony értéket fölvevő – tanulási ráta értékét megemeljük. A tanításhoz használt kód elérhető GitHub-on[3], illetve a tanítás menete végigkövethető az 1. ábrán vagy ennél részletesebben a Weights&Biases[4] felületén.

4. Kísérletek

Az előállított modelljeinket számos feladaton finomhangoljuk. Az összes kísérlet részletes eredménye és konfigurációja megtekinthető a Weights&Biases[5] felületén, de a fontosabb paramétereket itt is megemlítjük. Továbbá mind az első (`HuBERTUSz-{T,S,M}-128`), mind a második (`HuBERTUSz-{T,S,M}`) előtanítási fázis által elkészült modelljeinkün is elvégezzük kiértékeléseinket.

4.1. Modellek

A lehetséges modellek kiválasztása során törekedtünk a lehető legjobban lefedni a könnyen hozzáférhető magyar nyelvet támogató modellek minél szélesebb spektrumát. Ami így tartalmaz már használt és újonnan megjelentetett modelleket is, amelyekről magyar nyelv kapcsán nem történtek eddig mérések. A választott modellek listája és a hozzájuk tartozó paraméterek áttekinthetőek az 1. táblázatban.

Egynyelvű modellek A már baseline-nak tekinthető HuBERT-et (Nemeskey, 2020, 2021) vesszük kísérleteink alapjául. Ehhez a *hubert-base-cc* modellt

[3] `https://github.com/ficstamas/hubertusz-pretraining`
[4] `https://wandb.ai/szegedai-semantics/hubertusz-pretraining`
[5] `https://wandb.ai/szegedai-semantics/hubertusz-finetuning`

Név	#Nyelvek	$\|\theta\|$	$\|\theta_{emb}\|$	Méret	$\|V\|$	#Rétegek	#Fejek	h_{hidden}	h_{inter}
HuBERTUSz-tiny	1	4,5 M	4,1 M	0,04x	32001	2	2	128	512
HuBERTUSz-small	1	29,5 M	16,6 M	0,26x	32001	4	8	512	2048
HuBERTUSz-medium	1	42,1 M	16,6 M	0,38x	32001	8	8	512	2048
HuBERT-base-wiki	1	109,4 M	23,8 M	0,99x	30501	12	12	768	3072
HuBERT	1	110,6 M	24,9 M	1,00x	32001	12	12	768	3072
XLM-RoBERTa-base	100	278,0 M	192,3 M	2,51x	250002	12	12	768	3072
XLM-RoBERTa-large	100	559,8 M	256,5 M	5,06x	250002	24	16	1024	4096
mBERT	104	177,8 M	92,2 M	1,60x	119547	12	12	768	3072
distil-mBERT	104	135,3 M	92,2 M	1,22x	119547	6	12	768	3072
RemBERT	110	575,9 M	64,2 M	5,20x	250300	32	18	1152	4608

1. táblázat. A használt modellek paramétereinek összevetése. θ a paramétereket, V a szótárat, h_{hidden} és h_{inter} a rejtett és köztes reprezentációk dimenzióját jelölik. A Méret oszlopban a modellek HuBERT-hez viszonyított relatív mérete található.

használjuk (`HuBERT`), amely HuggingFace-ről könnyedén elérhető, és a teljes Magyar Webcorpus 2.0-on lett tanítva. Ezen túl a ritkábban használt, kizárólag a Webcorpus 2.0 Wikipedia alkorpuszán tanított `HuBERT-wiki-cased` és `HuBERT-wiki-uncased` variánsokat is bevesszük a kísérleteinkbe a teljesség kedvéért, illetve mivel az általunk létrehozott modellek is ugyanezen szövegeken lettek előtanítva.

Többnyelvű modellek Az `mBERT` a legelső többnyelvű transzformerek egyike, amely a hagyományos BERT tanítási stratégiával lett előállítva 104 nyelven. A `distil-mBERT` az `mBERT` egy disztillált (Sanh és mtsai, 2019) variánsa, ahol a rétegek számát lefelezték. Az `XLM-RoBERTa` a RoBERTa (Liu és mtsai, 2019) által bemutatott tanítási stratégiával előállított többnyelvű transzformer, amelyet Conneau és mtsai (2020) mutatott be. A `RemBERT` (Chung és mtsai, 2020) létrehozása során szétválasztották a be- és kimeneti beágyazásokat. Ezek dimenzióinak megfelelő allokálásával a paraméterek számosságát tudták csökkenteni a finomhangolási lépéshez. Ez úgy valósulhatott meg, hogy kisebb dimenziószámot adtak a bemeneti beágyazási rétegnek, míg az MLM fejben található kimeneti beágyazásnak ennél több paraméter jutott. Mindennek köszönhetően több réteg és fej használatára nyílik lehetőség ennél a modellnél.

4.2. Adathalmazok

Modelljeink kiértékelésére diverz feladatokat választottunk, jól definiált adathalmazokon, amelyeken a modellek viselkedése jól összehasonlítható. Ezek a feladatok a névelemfelismerés, szófaji egyértelműsítés, szentimentosztályozás, valamint a természetesnyelvi következtetés különféle problémái.

Tokenklasszifikációs faladatok közül névelemfelismerésben és szófaji egyértelműsítésben vizsgáltuk meg a modelljeink viselkedését. A névelemfelismerés

terén két adathalmaz használata mellett döntöttünk, amelyeknek szöveges tartalma majnem teljesen átfed. Az egyik a NerKor (Simon és Vadász, 2021), ami hagyományos névelemfelismerési feladatot definiál, és a megszokott *PER*, *LOC*, *ORG* és *MISC* címkéket alkalmazza a fikciós (fiction), jogi (legal), hír (news), web és Wikipedia alkategóriák mentén. Novák és Novák (2022) a NerKor adathalmaz címkézésen javasolt módosításokat, ami eredményeképp a teljes adathalmazban így már 34 egyedi címke szerepel, amelyek a *OntoNotes 5.0* standardot követik. Ezen felül még egy autókról szóló részhalmazzal is bővítette a meglévő korpuszt. A szófaji egyértelműsítésre vonatkozó kiértékeléseinket a NerKor univerzális morfológiai információkkal ellátott alkorpuszain (hír, web és Wikipedia) hajtottuk végre.

Az OpinHuBank adathalmaz (Miháltz, 2013) személyek különböző kontextusban való előfordulásához rendel pozitív, negatív vagy neutrális címkéket. Az adathalmazt öten annotálták, ami alapján mi többségi döntés szerint hoztuk létre a végső címkét. Ha a többségi címke nem határozható meg egyértelműen holtverseny miatt, akkor az az adatpont neutrális címkét kapott.

A HuLU kiértékelő környezet (Ligeti-Nagy és mtsai, 2022) a már angolból is megszokott NLI (Natural Language Inference) feladatokat ülteti át magyar nyelvre, és az alábbi 6 adathalmaz alkotja:

i) a COLA, ami a mondatok nyelvhelyesség szerinti kategorizálását tűzi ki célul,
ii) a CoPA, ami egy feleletválasztós teszt,
iii) az SST2, ami egy szentimentosztályozási feladat,
iv) a WNLI, ami egy anaforafeloldási probléma,
v) a WS, ami egy anaforafeloldás, kérdésmegválaszolási alapon,
vi) az RC, ami pedig egy szövegértési problémát fogalmaz meg.

5. Eredmények

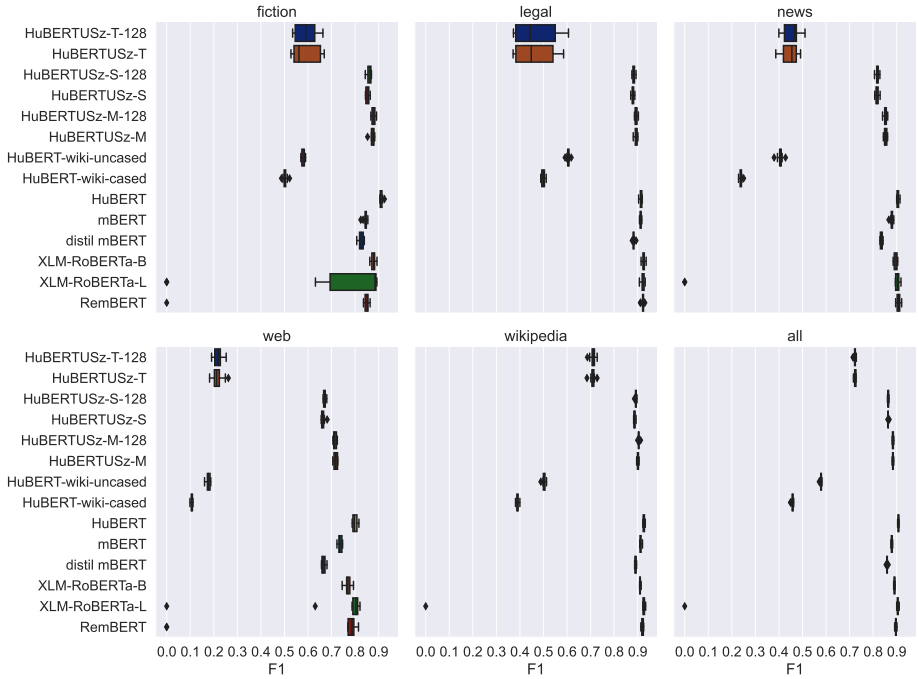
Az összes finomhangolásos kísérletet az osztályozó fejek 10 különböző random inicializálása mellett hajtottuk végre. Minden kísérlethez a következő hiperparamétereket alkalmaztuk: $5 \cdot 10^{-5}$ tanulási ráta, 32-es batch méret és 3 epoch. Minden más paraméter a `transformers` csomagban található *Trainer* alapvető paraméterének tekinthető, amelyek a már említett Weights&Biases oldalon is megtalálhatók.

A finomhangoláshoz és kiértékeléshez használt kódot a GitHubon[6] tettük elérhetővé, ami a használt adathalmazokat és modelleket HuggingFace kompatibilissá konvertálja futás közben. Ezenfelül a tokenszintű osztályozáshoz a metrikákat a *seqeval* könyvtár biztosította.

5.1. Finomhangolás – Névelemfelismerés

NerKor A hagyományos névelemfelismerési feladat eredményei alkorpuszonkénti lebontásban megtekinthetők a 2. ábrán. Az összes modell relatív teljesítménye

[6] `https://github.com/ficstamas/hu-eval/tree/mszny2023`



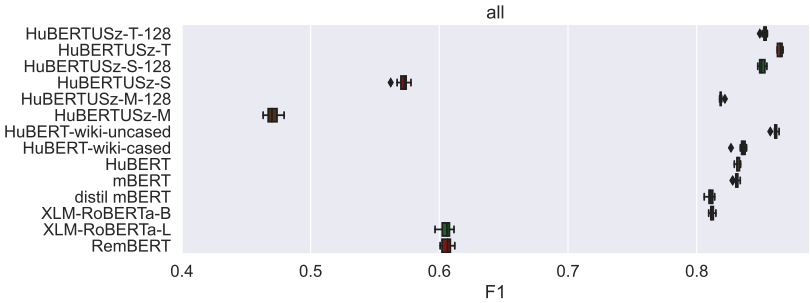
2. ábra. Finomhangolt modellek teljesítménye a NerKor egyes részhalmazain. Az *all* az egyes részhalmazok konkatenációját jelöli.

a különböző részhalmazokon lényegében azonos. Egyedül az `XLMRoBERTa-large` rendelkezik kiugró, outlier értékekkel. Az összes többi modell teljesítménye a megismételt kísérletek mentén stabilnak tekinthető. A legjobban eredményekkel rendelkező modellek között az F1-értékek különbsége szinte elhanyagolható, főleg ha figyelembe vesszük, hogy az egyes modellek hány paraméterrel rendelkeznek. Alacsony paraméterszámuk ellenére a `HuBERTUSz` modellek relatív teljesítménye jónak mondható, hiszen a `HuBERT`-ben lévő paramétereknek csupán a töredékével rendelkeznek (a `HuBERT` mérete kb. 2,6-szorosa a *medium* modellünknek).

NerKor 1.41e: Az adathalmazhoz tartozó összes alkorpusz aggregálása mentén kapott eredmények a 3. ábrán láthatók.

Megfigyelhetjük, hogy ezen feladat esetén nem feltétlen a paraméterek mennyisége számít, hiszen a *small* és *medium* konfigurációink a paraméterek mennyiségét tekintve relatív jó teljesítményt érnek el a többi modellhez képest. A végső sorrend a magyar nyelvű modellek esetén azonban a paraméterek mennyiségét követi.

A multilinguális modellek esetén hasonló viszonyokat nehéz leszűrni, mert mindegyik a paramétereit különböző módon osztja el. A végső sorrend sem ezek

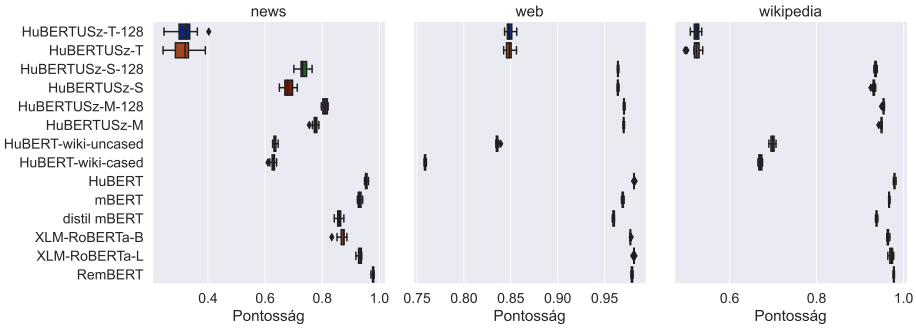


3. ábra. Finomhangolt modellek teljesítménye a NerKor 1.41e adathalmazon, ahol az *all* halmazt az egyes részhalmazok konkatenációjával kaptuk. A részhalmazonkénti eredmények megtekinthetők a Függelék 8. ábráján.

mennyiségével arányos. Azt viszont kijelenthetjük, hogy a `XLM-RoBERTa-large` modell teljesít a legjobban az összes modell közül.

5.2. Finomhangolás – Szófaji egyértelműsítés

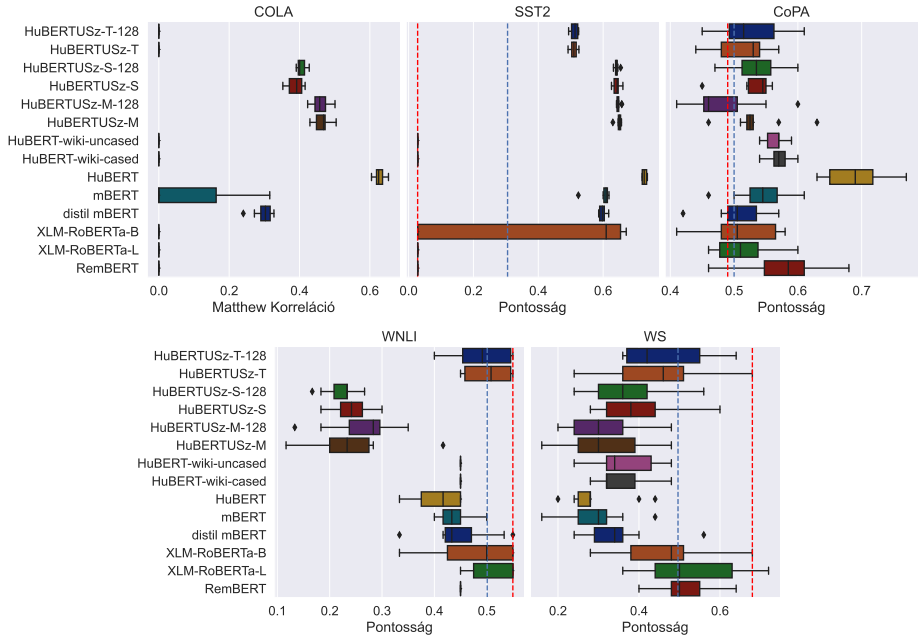
A 4. ábrán látható, hogy a legtöbb modell magas pontossággal oldotta meg a szófaji egyértelműsítési feladatot. A különböző modellek eredménye között a *news* adathalmazon figyelhető meg a legnagyobb szórás, ami valószínűleg az alkorpusz méretéből (is) adódik, mivel mindösszesen 9178 címkézett szóból áll (a vágások között összesen). Ezzel szemben a *web* és a *wikipedia* részhalmazok rendre 188250, illetve 26764 címkézett szóelemmel rendelkeznek.



4. ábra. A finomhangolt modellek teljesítménye a NerKor egyes részhalmazain a szófaji egyértelműsítési feladaton.

5.3. Finomhangolás – HuLU

A különböző modellek HuLU feladatokon nyújtott eredményességét az 5. ábrán láthatjuk. Piros szaggatott vonallal jelöltük annak a baseline modellnek a teljesítményét, ami minden esetben azonosan a tanítóhalmazon többségben lévő címkét prediktálja. A kék szaggatott vonal a véletlenszerű döntést hozó modell teljesítményét jelöli. Ezek a *COLA* esetén 0-nak tekinthetőek a Matthew korrelácóis együttható viselkedése miatt.



5. ábra. A HuLU kiértékelésre használt feladatain kapott eredmények. A szaggatott piros vonal a többségi, a szaggatott kék vonal pedig a véletlen baseline eredményét jelöli. A COLA feladaton ezek értéke azonosan 0-t vesznek fel a Matthew korrelációs együttható viselkedése okán.

Az egyes feladatokra vonatkozó vágások és az azokon belüli címkék eloszlásai megtekinthetők a 2. táblázatban. Minden eredménynél érdemes figyelembe venni az adathalmazok méretét és a címkék eloszlását, amelyek alapján a *COLA* tekinthető a legreprezentatívabb feladatnak a vizsgáltak közül.

A *COLA* esetén kapott eredmények alapján egyértelműen kijelenthető, hogy a többnyelvű modellek nem képesek kezelni ezt a feladatot. Még úgy sem, hogy némelyiknek 5-ször annyi paraméter áll rendelkezésére, mint a `HuBERT`-nek. Ez a viselkedés potenciálisan javítható jobb hiperparaméterválasztással, azonban ez kifejezetten erőforrásigényes lenne ennyi modell mentén. Érdekes módon a

	COLA	CoPA	SST2	WNLI	WS
# Tanító adathalmaz	7274	400	9328	562	170
# Kiértékelő adathalmaz	910	100	1165	60	25
Címkeeloszlás a tanító adathalmazon	22:78	50:50	31:29:40	51:49	51:49
Címkeeloszlás a kiértékelő adathalmazon	22:78	49:51	60:37:3	55:45	32:68

2. táblázat. A HuLU általunk használt feladatainak statisztikái.

kizárólag a Wikipedián tanult `HuBERT` modellek nem viselkednek túl jól se ezen, se a többi feladaton. A `HuBERT` és `HuBERTUSz` modellvariánsok a paramétereik számával arányosan követik egymást.

Az *SST2* esetében a *COLA* feladatnál látottakhoz hasonló viselkedéseket figyelhetünk meg. Ennél a feladatnál azonban a címkék eloszlása meglehetősen érdekesen alakul: a többségi döntés alapján a modellek $\approx 3\%$-ot tudnak elérni csupán, és ez is figyelhető meg azokban az esetekben, amikor 0 közelinek látszanak a felvett értékek. Ismételten a `HuBERT` teljesít a legjobban, ugyanakkor – a legkisebb *tiny* modellt leszámítva – az alacsonyabb paraméterszámmal rendelkező `HuBERTUSz` modellek eredményei sem sokkal maradnak el.

A CoPA esetén a `HuBERT` teljesítménye kiemelkedő, míg az összes többi modell a bennük található paraméterszámtól függetlenül hasonló, a véletlenszerű tippelés eredményét nem sokkal meghaladó eredménnyel rendelkezik.

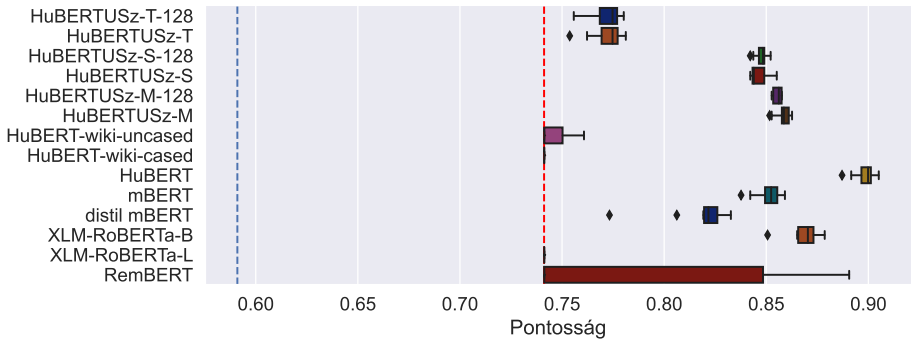
Az anaforafeloldáshoz kapcsolódó feladatokon (*WNLI* és *WS*) semelyik modell nem volt képest a baseline megoldások eredményét meghaladni. Mindez a modellek képességein és a feladatok nehézségén túl a rendelkezésre álló adatok méretével és minőségével is összefügghet, hiszen például a *WNLI* néhány tucatnyi tesztmondatai között angol nyelvűek is szerepeltek. Az előbbiek miatt, az anaforafeloldási feladatokon kapott eredmények inkább a teljesség igényét szolgálják, azokból messzemenő konklúziót nem vonnánk le.

Összességében a feladatok megoldására az egynyelvű modellek a vártak szerint viselkedtek (leszámítva a kizárólag a Wikipedián előtanított `HuBERT` modellek gyenge teljesítményét a *COLA* és *SST2* feladatokon), azonban meglepő módon a többnyelvű társaik nem voltak képesek használható szemantikai jellemzőket kinyerni.

5.4. Finomhangolás – Szentimentosztályozás

Az OpinHuBankon elért eredményeket a 6. ábra tartalmazza, amelyről leolvasható, hogy sem az `XLM-RoBERTa-large`, sem pedig az `HuBERT-wiki-*` modellek nem voltak képesek a többségi döntést hozó baseline eredményét érdemben meghaladni, így ezen feladaton az ezekre a modellekre építő modellalkotási kísérleteinket sikertelennek tekinthetjük.

Általánosságban a szentimentosztályozáson az egynyelvű modellek eredménye a paraméterek számával együtt növekszik (leszámítva a kizárólagosan Wikipedián előtanított `HuBERT` modelleket). A többnyelvű modellek esetén a `RemBERT`



6. ábra. Az OpinHuBankon elért eredmények.A függőleges szaggatott piros vonal a többségi, a kék a véletlen döntéssel rendelkező baseline eredményét mutatja.

és az `mBERT` eredményessége komoly eltérést mutat, a `RemBERT` modell eredményeit emellett nagyfokú instabilitás is jellemzi.

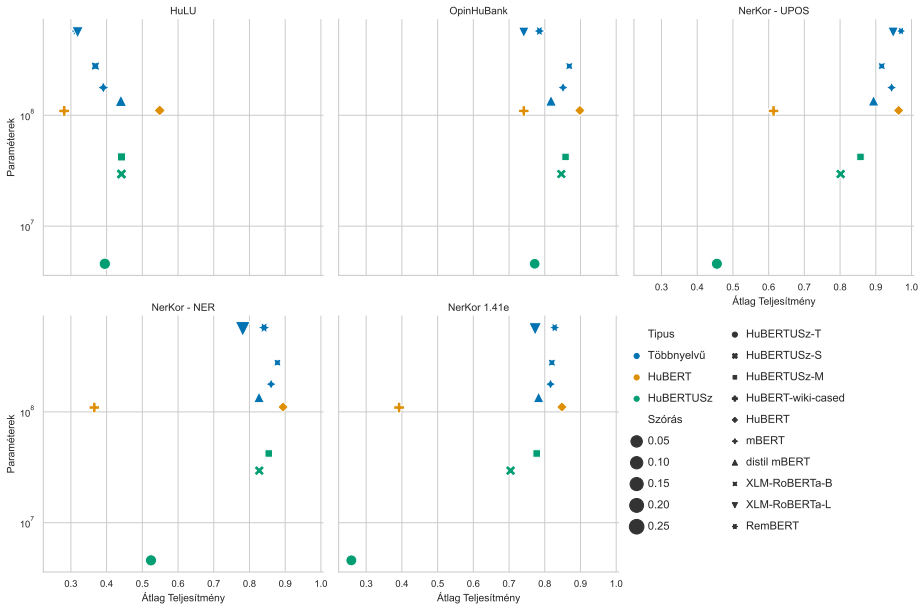
5.5. Paraméterek és teljesítmény viszonya

A 7. ábrán látható a paraméterek mennyiségének és az átlagos teljesítménynek a viszonya, ahol a szórást és átlagot az egyes részhalmazok között néztük. Az egyes részhalmazokon belül pedig a véletlen inicializációnkénti átlagot vettük.

Szekvenciaosztályozás esetén láthatjuk, hogy a *HuLU* megoldásához nem elég csak a paraméterek számának növelése, hanem ajánlott az egynyelvű modellekre támaszkodni. A *HuLU* esetén a `HuBERT` használata indokoltnak tűnik. Az *OpinHuBank* tekintetében is hasonló tendenciák figyelhetők meg, azonban ezen a feladaton már a `HuBERTUSz` modellek használata is érdemi alternatívaként tekintendő, hiszen a `HuBERT` úgy teljesít $\sim 4\%$-kal jobban a *medium* modellünkhöz képest, hogy közben több, mint 2,6-szer annyi paramétert használ. Megjegyzendő továbbá, hogy a – `HuBERTUSz` modellekhez hasonlóan – csak Wikipedia szövegeken előtanított `HuBERT` modellek teljesítményei jóval elmaradnak a `HuBERTUSz` modellek által kapottaktól.

A *NerKor* univerzális szófaj egyértelműsítési feladatán észrevehetjük, hogy a többnyelvű modellek is felzárkóztak. Ennek hátterében az állhat, hogy a feladat megoldásához általában elegendő a korai rétegek rejtett beágyazásai alapján döntést hoznunk. Más szóval élve a feladat a statikus szóreprezentációkhoz közel áll, amelyek a nyelv egyedi karakterkészlete miatt, a szótöredékek szinte csak a magyar nyelvet kódolják. Így nincsenek információval túlterhelve, mint a szekvenciaosztályozási token. A szófaji egyértelműsítési feladatoknál hasonló eredményeket fedezhetünk fel, amelynek hátterében hasonló okokat feltételezünk.

Összességében elmondható, hogy egy kisebb paraméterszámú egynyelvű modell jobb választásnak bizonyulhat egy többnyelvű modellhez képest. A `HuBERTUSz` modell pedig a `HuBERT`-hez képest elég közel kerül teljesítmény terén, figyelembe véve, hogy jóval kevesebb paraméterrel is rendelkezik.



7. ábra. A paraméterek és átlagos teljesítmény viszonya az egyes adathalmazokon. A szórást az egyes részhalmazok menti átlagos teljesítményből kaptuk.

6. Összegzés

Cikkünkben bemutattuk a legújabb előtanítással kapcsolatos ajánlásokat (dinamikus maszkolás, mondatsorrend-predikció) alkalmazó, eltérő méretekben előtanított `HuBERTUSz` modelljeinket, amelyeket a `HuBERT` esetében használtakhoz képest jóval kevesebb nyers adat és paraméter felhasználásával hoztunk létre. Az új modellek segítségével lehetőségünk nyílt a magyar nyelv vonatkozásában a különböző kapacitással rendelkező modellek skálázódásának összehasonlító vizsgálatára.

Kísérleteink a magyar nyelvet támogató, jóval nagyobb paraméterszámmal rendelkező többnyelvű modellek kiértékelésére is kiterjedtek. Összességében elmondható, hogy az általunk vizsgált feladatokon még a nagyságrendekkel kisebb kapacitással létrehozott egynyelvű modelljeink használata is kifizetődőbbnek bizonyult a többnyelvű modellek használatához képest, mind az elért eredmények, mind pedig az erőforrásigény szempontjából.

Az előállított modellek számos feladaton alkalmazhatónak tekinthetőek. Elsősorban szó szintű osztályozási feladatok megoldására ajánljuk, azonban itt is figyelni kell a megfelelő mennyiségű adatra.

Köszönetnyilvánítás

A kutatás az Emberi Erőforrások Minisztériuma ÚNKP-22-3 kódszámú Új Nemzeti Kiválóság Programjának támogatásával, valamint az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében. A MILAB SZTE nyelvtechnológia projekt nevében köszönetet mondunk az ELKH Cloud (lásd: Héder és mtsai (2022); https://science-cloud.hu/) használatáért, ami hozzájárult a publikált eredmények eléréséhez.

Hivatkozások

Ács, J., Lévai, D., Nemeskey, D.M., Kornai, A.: Evaluating contextualized language models for hungarian. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–28. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021)

Bhargava, P., Drozd, A., Rogers, A.: Generalization in NLI: ways (not) to go beyond simple heuristics. CoRR abs/2110.01518 (2021), https://arxiv.org/abs/2110.01518

Chung, H.W., Févry, T., Tsai, H., Johnson, M., Ruder, S.: Rethinking embedding coupling in pre-trained language models. CoRR abs/2010.12821 (2020), https://arxiv.org/abs/2010.12821

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020), https://www.aclweb.org/anthology/2020.acl-main.747

Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018), http://arxiv.org/abs/1810.04805

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1423

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR abs/2010.11929 (2020), https://arxiv.org/abs/2010.11929

Feldmann, Á., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.Gy., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 29–36. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021), http://real.mtak.hu/120856/1/feldmann21.pdf

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., Zhou, M.: Codebert: A pre-trained model for programming and natural languages. CoRR abs/2002.08155 (2020), https://arxiv.org/abs/2002.08155

Ficsor, T., Cserháti, R., Novák, A., Mihajlik, P., Zainkó, Cs., Berend, G.: Charmen electra - tokenizációmentes diszkriminatív nyelvi modellezés. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 45–58. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Héder, M., Rigó, E., Medgyesi, D., Lovas, R., Tenczer, S., Török, F., Farkas, A., Emődi, M., Kadlecsik, J., Mező, Gy., Pintér, Á., Kacsuk, P.: The past, present and future of the ELKH cloud. Információs Társadalom 22(2), 128 (aug 2022), https://doi.org/10.22503/inftars.xxii.2022.2.8

Hsu, W., Bolte, B., Tsai, Y.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: Hubert: Self-supervised speech representation learning by masked prediction of hidden units. CoRR abs/2106.07447 (2021), https://arxiv.org/abs/2106.07447

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. CoRR abs/1909.11942 (2019), http://arxiv.org/abs/1909.11942

Ligeti-Nagy, N., Ferenczi, G., Héja, E., Jelencsik-Mátyus, K., Laki, L.J., Vadász, N., Yang, Z.Gy., Váradi, T.: HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 431–446. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692 (2019), http://arxiv.org/abs/1907.11692

McCoy, R.T., Pavlick, E., Linzen, T.: Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. CoRR abs/1902.01007 (2019), http://arxiv.org/abs/1902.01007

Miháltz, M.: OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez. In: IX. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–345. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2013), http://acta.bibl.u-szeged.hu/58859/1/msznykonf_009_343-345.pdf

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020), https://hlt.bme.hu/en/publ/nemeskey_2020

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). pp. 3–14. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2021), http://acta.bibl.u-szeged.hu/73353/1/msznykonf_017_003-014.pdf

Novák, A., Novák, B.: NerKor 1.41e. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 389–412. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Orosz, Gy., Szántó, Zs., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 59–73. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108 (2019), `http://arxiv.org/abs/1910.01108`

Simon, E., Vadász, N.: Introducing nytk-nerkor, A gold standard hungarian named entity annotated corpus. In: Ekstein, K., Pártl, F., Konopík, M. (szerk.) Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12848, pp. 222–234. Springer (2021)

Simon, E., Vadász, N., Lévai, D., Nemeskey, D., Orosz, Gy., Szántó, Zs.: Az NYTK-NerKor több szempontú kiértékelése. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 403–416. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

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. Genova (Italy) (2006)

Turc, I., Chang, M.W., Lee, K., Toutanova, K.: Well-read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962v2 (2019)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017), `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`

Yang, Z.Gy.: BARTerezzünk! messze, messze, messze a világtól, BART kísérleti modellek magyar nyelvre. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 15–29. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022a)

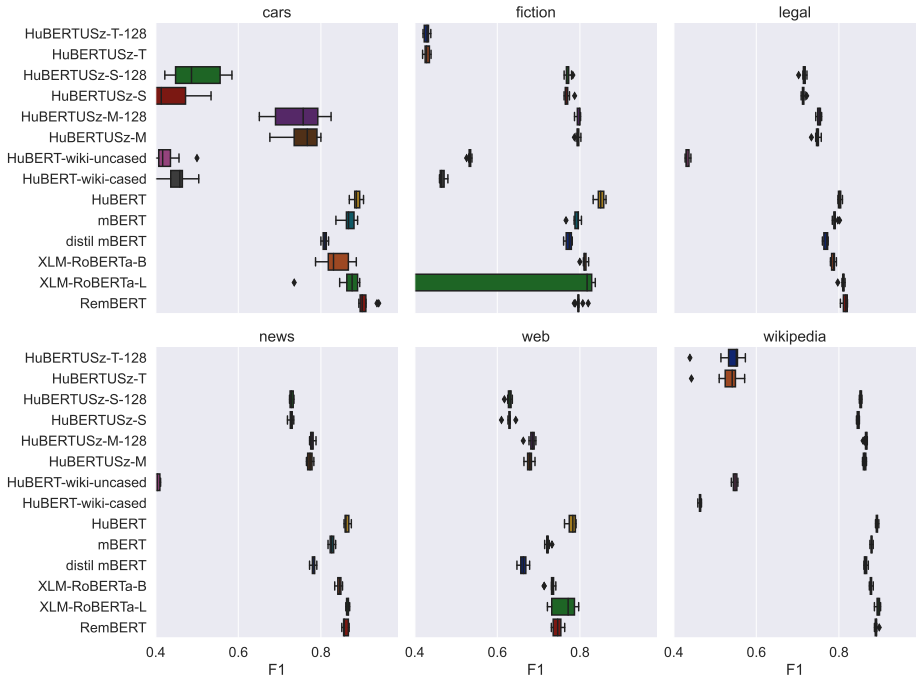
Yang, Z.Gy., Feldmann, Á., Váradi, T.: HILBERT, magyar nyelvű BERT-large modell tanítása felhő örnyezetben. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, pp. 603–617. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022)

Yang, Z.Gy.: "Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 463–476. Szegedi Tudományegyetem, Informatikai Intézet, Szeged (2022b)

Yang, Z.Gy., Váradi, T.: Training language models with low resources: RoBERTa, BART and ELECTRA experimental models for hungarian. In: Proceedings of 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021). pp. 279–285. IEEE, Online (2021)

You, Y., Li, J., Hseu, J., Song, X., Demmel, J., Hsieh, C.: Reducing BERT pre-training time from 3 days to 76 minutes. CoRR abs/1904.00962 (2019), `http://arxiv.org/abs/1904.00962`

A. További eredmények



8. ábra. Finomhangolt modellek teljesítménye a NerKor 1.41e részhalmazain.

A new ParlaMint corpus for Hungarian
30m tokens of annotated parliamentary data

Noémi Ligeti-Nagy[1], Réka Dodé[1], Kinga Jelencsik-Mátyus[1], Zsófia Varga[1,2], Enikő Héja[1], Tamás Váradi[1]

[1]Hungarian Research Centre for Linguistics
H-1068 Budapest, Benczúr u. 33.
{vezeteknev.keresztnev}@nytud.hu
[2]University College London
2 Wakefield St, London, WC1N 1PF, UK
zsofia.varga.20@ucl.ac.uk

Abstract. Parliamentary data constitute a rich source for research for academic fields in the social sciences and humanities (SSH). To facilitate such research, comparable, high-quality parliamentary corpora are needed. The ParlaMint project, funded by CLARIN-ERIC, aims to create such corpora for languages spoken in European parliaments in a shared framework consisting of uniform encoding schemas, metadata structure, and Universal Dependencies-type linguistic annotation. The newly built Hungarian corpus of ParlaMint II focuses on the minutes of the Hungarian National Assembly between May 2014 and June 2022 and can be considered a major improvement from the Hungarian corpus of ParlaMint I. It has a wider time frame, more extensive metadata on speakers and their affiliations, and more sophisticated linguistic analysis than what was available in ParlaMint I. The Hungarian ParlaMint II corpus is openly available, just as all the ParlaMint corpora for other languages. Some potential applications of ParlaMint corpora in SSH research are also discussed.

Keywords: language data, parliamentary data, corpus, linguistic annotation, metadata, TEI

1 Introduction

In the fields of SSH there is an ever-growing need for language data as a source of research. Parliamentary debates as a special kind of spoken language data are especially valuable as they provide opportunities for numerous branches of SSH (e.g., political science, sociology, history, discourse analysis or sociolinguistics) to study various aspects of the transcripts of the verbal exchanges. At the same time, parliamentary language data are also valuable as they are not subject to copyright, and thus can be easily used in building datasets (Erjavec et al., 2022).

In 2019, CLARIN (clarin.eu) – the pan-European research infrastructure funded to support researchers in SSH – launched ParlaMint, a large-scale project

with the aim of building corpora of parliamentary texts for the participating languages. In the first phase of the project, ParlaMint I, corpora for parliamentary debates of 17 national parliaments were created, spanning almost 500 m words in total. The project's main objective was to produce comparable, highly useful language resources of contemporary parliamentary data for SSH research and education. The language data in each corpus of ParlaMint I were analysed using the Universal Dependencies framework. As such, the 17 corpora of ParlaMint I provide comparable datasets, following the same encoding schema, with uniform linguistic annotation and rich metadata. These corpora are freely available at CLARIN.SI repository[1] to download, and at the NoSketch Engine[2] and KonText[3] platforms for analysis (Erjavec et al., 2021).

In December 2021 CLARIN ERIC decided to create ParlaMint II. As part of this second phase, new languages (and regional varieties) were added to the project, and a decision was made to extend participating languages' corpora up to July 2022 or further. In this new phase, the schema is updated and the validation processes are informed by the experiences of the first phase. ParlaMint II aims to include parliamentary texts in 31 languages. The project also incorporates data from regional parliaments such as Basque country or Galicia.

In the present paper, we introduce the Hungarian corpus of ParlaMint II created by the Language Technology Research Group of the Hungarian Research Centre for Linguistics. First, we review ParlaMint I and its Hungarian corpus. Then, we discuss the objectives of ParlaMint II and the details of the completed corpus, focusing on (meta)data collection and encoding and the linguistic annotation process. Lastly, we overview some potential applications of the new corpus.

2 ParlaMint I

Although there are several datasets of parliamentary data in numerous languages, they follow different encoding schemas and contain different kinds of information, making it difficult to conduct comparative studies. Following many preparatory events organised by CLARIN (see Erjavec et al., 2022), the ParlaMint project started in 2019 with the aim of creating a framework for parliamentary corpora for the four participating countries (Bulgaria, Croatia, Poland, Slovenia). Shortly later, 13 additional members joined the project.

2.1 Encoding the corpus

All 17 corpora in ParlaMint I have the same metadata and data structure, were created using the same encoding schema and follow the same type and degree of linguistic annotation. This framework was initially laid out in the Parla-CLARIN

[1] https://www.clarin.si/repository/xmlui/handle/11356/1432
[2] http://www.clarin.si/noske/
[3] https://www.clarin.si/kontext/corpora/corplist

recommendations (Erjavec and Pančur, 2019) and is based on the Text Encoding Initiative Guidelines (TEI, 2017). To make the corpora of the project interoperable, RelaxNG schemas were created and XSLT scripts were added to help the validation of the corpora. (Erjavec et al., 2022) Prior to the launch of the ParlaMint project, the 200-million-word siParl corpus was encoded using this framework, and all the lessons learned were later built into the newer version of Parla-CLARIN.

As a result, ParlaMint I contains transcriptions of speeches made in 17 European national parliaments running to half a billion words. The final corpora contain metadata of about 11 thousand speakers and are linguistically annotated following the Universal Dependencies formalism.[4]

2.2 The Hungarian corpus in ParlaMint I

The Hungarian parliamentary corpus in ParlaMint I was assembled by the Centre for Social Sciences (CSS), Hungary (Üveges and Ring, 2022). It contained two types of speeches (interpellations and urgent questions) from the plenary sessions of the National Assembly of Hungary, terms 7 and 8 (May 2014 – December 2020), amounting to 870,000 tokens in total. The authors used three separate tools for linguistic annotation and a fourth Java program to merge the output of the three tools. For morphosyntactic annotation, an old version of the magyarlanc linguistic toolkit was used (Zsibrita et al., 2013). Syntactic analysis was performed by UDPipe (Straka and Straková, 2017). Named entity recognition was done by a tool created by the MTA-SZTE Research Group on Artificial Intelligence (Szarvas et al., 2006). This approach has several drawbacks as the output of each module must be adjusted to provide a suitable input for the next module. This of course also means that additional bugs could appear.

For metadata, the political party, gender and date of birth were gathered for the 194 speakers of the corpus. Given the type of speeches collected, all speeches have an MP as speaker, and none of them are given by the chair of the session.[5]

3 ParlaMint II

ParlaMint II, while also upgrading the XML schema and validation, aims to feature new languages and extend the existing corpora to cover data to at least mid-2022. Originally, 30 languages were planned to be featured in this second phase of ParlaMint II, with Ukrainian being added to the project later on. Ukrainian parliamentary speeches are being processed at the moment.

[4] Samples of the corpora and conversion scripts are available from the project's GitHub repository (https://github.com/clarin-eric/ParlaMint), and the complete set of corpora is openly available at the platforms mentioned in the Intorduction of the present paper.

[5] The proportion of speeches given by MPs in the corpora of ParlaMint I are almost over 90%. The proportion of technical speeches varies widely between countries in PMI, although most are about half of all the speeches.

As the large number of partners led to a wide variety of parliamentary systems being featured, with different kinds of structures, positions, and speakers, encoding was very much like an iterative process both in ParlaMint I and II. In the second phase of the ParlaMint project, our institution took on the task to expand the Hungarian ParlaMint corpus both in terms of the time period covered and the range of text types covered. As a result of our work, this second ParlaMint-HU corpus now contains the minutes of the National Assembly from 2014 to 2022, comprising terms 7, 8, and the spring session of term 9 of the Third Republic of Hungary. The documents of the corpus thus range from 6 May 2014 to 14 June 2022 and contain the official textual transcriptions of 514 sitting days in this period.

The National Assembly is the unicameral legislative body of Hungary, currently consisting of 199 Members and advocates for nationalities that could not reach the threshold to elect any MPs (between 1990 and 2014 the number of MPs was 386). Currently, there are 12 advocates. Members and advocates are elected to 4-year terms, and a chair is chosen from MPs at the beginning of each parliamentary term. The National Assembly has 25 standing committees dedicated to areas of parliamentary activity that oversee ministers' activities and discuss and report on introduced bills.

While the first version of ParlaMint-HU was confined to interpellations, the PMII corpus includes all types of speeches and verbal exchanges made in the National Assembly. These equal 104,115 speeches, comprising 1,540,325 sentences and 32,353,437 tokens (27,533,236 words and 4,820,201 punctuation marks).

We detail our contribution to ParlaMint II in the following sections.

3.1 Data source and acquisition

The parliamentary texts were downloaded from the official website of the Hungarian National Assembly (https://www.parlament.hu/).

Metadata were gathered manually from official sources, primarily the website of the National Assembly. Other sources include *Magyar Közlöny* (the official journal of Hungary that publishes new laws, appointments, etc.), and various newspapers (e.g., for dates of resignations).

In total, we collected metadata for all 426 speakers appearing in the Hungarian National Assembly between May 2014 and June 2022, representing 91 organisations. Speakers include Members of Parliament, advocates of recognised minorities living in Hungary, secretaries of state, Hungarian members of the European Parliament, ombudsmans, visiting politicians from other countries, and (deputy) heads of offices and organisations such as the Hungarian National Bank, the State Audit Office, etc.

Among the 91 organisations, there are 17 parties, 21 parliamentary groups (the same party has a separate parliamentary group in each term), 17 ministries, and 36 other organisations. The last category includes parliamentary committees, relevant institutions of the European Union, etc. We decided not to indicate speakers who are commissioners of the state and/or members of parliamentary subcommittees.

Normalisation Before processing, the texts were manually normalised and – where applicable, – typos and other small errors were corrected. To avoid having false duplicates due to name changes or forms that are official for the persons but are not publicly used, speakers' names were unified and are included in their current form used for public appearances. (For example, Katalin Novák has some utterances as *Veresné Novák Katalin*, or Szilárd Németh appears as *Németh Szilárd István* in some cases. These were uniformly changed to *Novák Katalin* and *Németh Szilárd*, respectively. 14 names were changed in the corpus.)

Non-breaking spaces, non-breaking hyphens and soft hyphens were replaced by their regular counterpart. Quotation marks were left unmarked in texts. Occurrences of double opening or closing parentheses were uniformly changed to single parentheses.

Language usage Some speeches between 2014-2022 were held in languages other than Hungarian. These are the languages used by the representative of the German nationality in Hungary, and by advocates of other nationalities (Slovakian, Slovenian, Serbian, Bulgarian, Ukrainian, etc.). We used langdetect[6] to identify these segments. After the manual correction of langdetect's output, we had 213 segments detected as non-Hungarian. These are distributed as follows: de (85), pl (27), sk (14), sr (11), bg (10), sl (9), uk (5), cs (3), ro (3), el (2), hr(1), rom (1), ru (1). 42 segments were identified as multilingual, i.e. they contain chunks in two different languages.

3.2 Data encoding process in brief

Data processing and encoding were carried out using Python scripts. Each source txt already covered one day of a sitting, so we kept this structure. For extraordinary sittings, a separate file was created. We split the text into speeches and the speeches into segments and notes. For segments, the paragraphs of the original texts on the official National Assembly website were used. Speakers and transcription notes were detected, and the latter were categorised into TEI note types using regular expressions primarily. Segments were given a unique id, and id-segment pairs served as input for HuSpacy (Orosz et al., 2022), the linguistic analyser (more details on the linguistic analysis of the corpus in section 3.4). HuSpacy's output was converted into the required XML structure using another Python script.[7]

As the code creating the XML format is built to work on the quite rigid original format of the source files, input files had to be formatted accordingly.

[6] https://pypi.org/project/langdetect/

[7] All the scripts used to create the corpus are available at https://github.com/nytud/HuParlaMintII along with the metadata we collected. However, it is important to note that our primary aim was to create a corpus of good quality – hence the manual corrections –, so a reusable pipeline is just a side product of our project and by no means the main goal.

The presiding chair's technical remarks have *ELNÖK* ('president') as the speaker in the original transcription. The paraticular person acting in this role could not always be identified from the text alone, therefore, we needed to collect the chairs for all sessions manually. To do this, we used the data available on the official website of the National Assembly. Parallel to this, we manually collected all metadata.

3.3 Metadata

We collected metadata for the 426 speakers appearing in the Hungarian National Assembly between May 2014 and June 2022, and the 91 organisations they represent. Every organisation and person received a unique id, and we always included start and end dates for specific roles (indicating the length of the 'tenure' of a person in a role) and the duration of operation for organisations. For organisations, the start and end dates should be understood as the founding date and the date of disassembly/renaming.

If an organisation was renamed, the date of the renaming is featured as the end date of its entry on its original name, and a separate entry was created with the new name. People with multiple names in the corpus were only included under their current name, to allow for more traceability.

For each sitting day in each separate file, we also provided links to the video and the text of that sitting day available on the National Assembly website. We also annotated transcriber notes according to the ParlaMint TEI XML guidelines.[8]

Whilst some of the roles and categories we used in the metadata for organisations and speakers are not necessarily intuitive, these were dictated by the rigid metadata structure of ParlaMint.

Metadata of organisations We collected metadata on organisations according to the following categorisation:

1. *political parties*;
2. *parliamentary groups*;
3. *ministries*;
4. *other organisations*;
5. *opposition and coalition.*

We collected 17 parties, 9 parliamentary groups, 17 ministries, and 41 other organisations. The last category includes 24 committees, as well as the republic, some NGOs, and several European institutions (e.g., the European Parliament, the European Commission, and 'europeanInstitution' for additional ones).

Following a requirement of the XML structure, we marked each party as coalition or opposition in each term, depending on whether they were governing or not. Below, the entry for the Párbeszéd parliamentary group is shown as an example of the final XML structure.

[8] https://clarin-eric.github.io/ParlaMint/#chp-intro

```
<org xml:id="parliamentaryGroup.Parbeszed"
role="parliamentaryGroup">
   <orgName full="yes" xml:lang="hu">A Párbeszéd Magyarországért
   parlamenti frakciója</orgName>
   <orgName full="yes" xml:lang="en">Parliamentary group of the
   Dialogue for Hungary</orgName>
   <orgName full="abb">Párbeszéd-frakció</orgName>
   <listEvent>
      <event xml:id="parliamentaryGroup.Parbeszed.8"
            from="2018-05-08"
            to="2022-05-01">
         <label xml:lang="hu">A Párbeszéd Magyarországért
         parlamenti frakciója a 8. parlamenti ciklusban
         (2018-05-08 - 2022-05-01)</label>
         <label xml:lang="en">Parliamentary group of the Dialogue
         for Hungary in the 8th parliamentary term (2018-05-08 -
         2022-05-01)</label>
      </event>
      <event xml:id="parliamentaryGroup.Parbeszed.9"
      from="2022-05-02">
         <label xml:lang="hu">A Párbeszéd Magyarországért
         parlamenti frakciója a 9. parlamenti ciklusban
         (2022-05-02 - )</label>
         <label xml:lang="en">Parliamentary group of the Dialogue
         for Hungary in the 9th parliamentary term (2022-05-02 -
         )</label>
      </event>
   </listEvent>
</org>
```

XML snippet 1. A snippet of the XML file showing the encoded metadata of the Párbeszéd parliamentary group in the Hungarian corpus of ParliaMint II

Metadata of persons We collected the metadata on speakers appearing in the corpus as follows:

1. *basic personal data*: the name of the person (with any additional prefixes, such as *dr.*), their gender, and date of birth[9];
2. *membership of political parties and parliamentary groups:* start and end date of speakers' membership were included, and the specific role the person had in that entity[10];

[9] Out of the 426 speakers in the corpus, the speaker's date of birth could not be established in 37 instances

[10] We treated head and deputy head as roles taken on by some speakers in addition to being a member of a party. Being party-independent was treated as not having any roles in any party rather than as an additional label. MPs can be members of parliamentary groups of parties not their own. This happens, for example, if they are members of a party without parliamentary seats but they choose to join the parliamentary group of a different party.

3. *parliamentary membership*: the start and end date of the person's mandate in each parliamentary term;
4. *other roles*: any other important political role held by a person. For example, being in parliamentary committees, serving as a secretary of state, etc. We decided not to indicate speakers who are commissioners of the state and members of parliamentary subcommittees.

We collected each MP's committee positions for the period covered by the corpus, and 'party joining dates' in case of ongoing party affiliations that started before 2014. For judges and individuals operating irrespective of parliamentary terms, we added the start dates of their relevant appointments. For secretaries of state, whenever possible, we included a detailed version of their role names in English and Hungarian, as these are highly specific and variable across parliamentary terms and even under the same ministry.

The ParlaMint TEI XML schema[11] is quite rigid in terms of the possible roles a person or an organisation may have. The @role attribute has only a set number of valid values, so sometimes, we had to follow this schema instead of our intuition. This means that presidents of the republic (János Áder and Katalin Novák in the time period our corpus covers) are, for example, annotated as heads of the republic. A minister is encoded with four roles in the corpus: they are heads *and* members of a ministry (if not without portfolio), and ministers *and* members of the government. As the ParlaMint TEI XML does not have `notary` as a valid value for the @role attribute, notaries of the Hungarian Assembly are annotated as secretaries, and their exact role (*notary*, *junior notary* or *senior notary*) are listed as a natural language description of their role. A similar solution was used for the senior chair ('korelnök') which was merged into the role of chairperson with an additional note. Thus, the 426 speakers in the corpus have 3273 affiliations altogether. Figure 1 shows the number of roles assigned to the speakers. There are five speakers without any affiliation (but with basic personal data); they are representatives of foreign countries (e.g. Stanislaw Tillich), or presidential candidates giving a speech in the Assembly only once (László Majtényi and Péter Zsolt Róna). The ParlaMint TEI XML schema does not support the encoding of this kind of person metadata. Their role is encoded in the corpus as a `note` at the beginning of their speech. XML snippet 2 shows the corpus entry for MP Ágh Péter with all his encoded affiliations between 2014 and 2022.

[11] https://github.com/clarin-eric/ParlaMint/blob/main/Schema/README.md

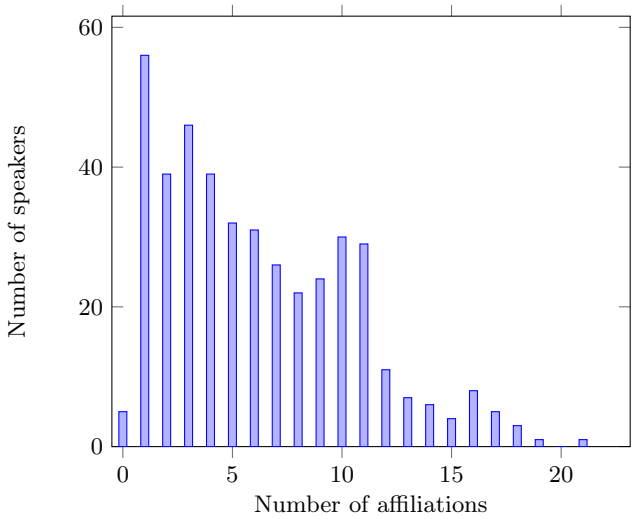


Fig. 1: Bar plot showing the number of affiliations the speakers in our corpus have. As can be seen, most speakers are affiliated with at least one role, and the majority of them have 2-10 roles. The recorder is Csaba Hende with 21 roles. Speakers without any role are guest speakers, representatives of other countries.

```
<person xml:id="AghPeter">
      <persName>
         <surname>Ágh</surname>
         <forename>Péter</forename>
      </persName>
      <sex value="M"/>
      <birth when="1982-01-30"/>
      <affiliation role="member"
                   ref="#OGY"
                   from="2014-06-06"
                   to="2018-05-07"
                   ana="#OGY.7"/>
      <affiliation role="member"
                   ref="#OGY"
                   from="2018-05-08"
                   to="2022-05-01"
                   ana="#OGY.8"/>
      <affiliation role="member" ref="#OGY" from="2022-05-02"
      ana="#OGY.9"/>
      <affiliation role="member" ref="#party.Fidesz" from="1999"/>
      <affiliation role="deputyHead"
                   ref="#parliamentaryGroup.Fidesz"
                   ana="#parliamentaryGroup.Fidesz.7"
                   from="2014-07-04"
                   to="2015-11-01">
```

```
        <roleName xml:lang="en">Deputy Head</roleName>
      </affiliation>
      <affiliation role="member"
                   ref="#parliamentaryGroup.Fidesz"
                   ana="#parliamentaryGroup.Fidesz.7"
                   from="2014-05-06"
                   to="2018-05-07"/>
      <affiliation role="member"
                   ref="#parliamentaryGroup.Fidesz"
                   ana="#parliamentaryGroup.Fidesz.8"
                   from="2018-05-08"
                   to="2022-05-01"/>
      <affiliation role="member"
                   ref="#parliamentaryGroup.Fidesz"
                   ana="#parliamentaryGroup.Fidesz.9"
                   from="2022-05-02"/>
      <affiliation role="deputyHead"
                   from="2018-10-15"
                   to="2020-05-05"
                   ref="#org.HRB">
      </affiliation>
      <affiliation role="member" from="2014-06-06" to="2018-05-07"
      ref="#org.HRB"/>
      <affiliation role="member" from="2018-05-18" to="2020-05-05"
      ref="#org.HRB"/>
      <affiliation role="member" from="2014-06-06" to="2015-09-22"
      ref="#org.TB"/>
      <affiliation role="member" from="2020-05-05" to="2022-05-01"
      ref="#org.TB"/>
      <affiliation role="member" from="2022-05-02" ref="#org.TB"/>
    </person>
```

XML snippet 2. A snippet of the XML file showing the rich metadata of Ágh Péter, a member of parliament featured in the Hungarian ParlaMint II corpus

Other metadata In addition to metadata about speakers and organisations, we have further data on each session day. These include the type of session (regular, special or ceremonial), links to the text and the video recording of the session, the number of speeches made, and the relative place of a sitting day situated in a sitting that is a specific, often several-day-long session of that parliamentary term.

Transcriber comments Transcriber comments are also noted in the corpus, containing information about the speaker, time, voting results, or, more importantly, incidents in the assembly hall. Such an element is encoded as a <note>, where *incidents* can be further specified as vocal, kinesic and incident.

Vocals are vocalised – but not necessarily lexical – happenings, with further subtypes of laughter, speaking, interruption, shouting or murmuring. Sometimes,

there are multiple speakers in an interruption. The further processing of such notes is part of our future work. Kinesic describes communicative – though not necessarily vocalised – events, with subtypes such as applause, ringing, snapping or any gestures.

Incidents can be very versatile, describing not necessarily communicative events. Subtypes of this category are break, pause, entering or leaving.

Coding the transcriber comments was not always easy as the categories were not very distinct. Transcriber comments represent very rich metadata in our corpus – more than 37 thousand occurrences in total. The processing of these data was partly automatic but lots of additional manual analysis was needed due to the detailed framework.

3.4 Linguistic annotation

ParlaMint II requires all corpora to be linguistically annotated as follows:

- tokenisation with preserving information on inter-token spaces while separately tagging words and punctuation marks
- sentence splitting
- lemmatisation
- part-of-speech and morphological features using the UD tagset, while possibly also providing part-of-speech tags of a different (local) tagset
- named entity tagging, using the four standard named entity classes: `PER, LOC, ORG, MISC`
- UD dependency syntactic parse of the sentences

Only the text content of the speech segments was linguistically annotated. Transcriber notes and speeches held in languages other than Hungarian were excluded from the linguistic analysis.

Hungarian is in the fortunate position of having several processing pipelines to choose from for carrying out the linguistic analysis required by ParlaMint II. e-magyar (Váradi et al., 2018) and HuSpaCy (Orosz et al., 2022) both provide the necessary morphological and syntactic analysis in addition to named entity recognition. emDep (the dependency parser of e-magyar) does not have the UD format as an available output, and no converter is available to create that either. The output of HuSpaCy fitted perfectly well with ParlaMint II's requirements, so we chose this pipeline over e-magyar.[12] For the linguistic annotation of the corpus, we used HuSpaCy and its hu_core_news_lg model. The lower-level analyses, such as sentence splitting, tokenisation, and lemmatisation resulted in an accurate and easily validatable annotation. The named entity recognition, on the other hand, generated erroneous labels in some cases that popped up during the automatic central validation of the XML files, causing the break of

[12] We are aware that the Stanza and UDPipe parsers are available as alternative dependency parsers in e-magyar – as emStanza and emUDPipe –, but to avoid conversion problems and to stick with the more compact option, we chose HuSpaCy.

the script generating the CoNLL-U format from the XML, among others. We manually corrected the named entity tags of these tokens.[13]

4 Potential applications

Numerous fields of SSH and especially digital humanities may benefit from well-structured, linguistically annotated parliamentary data with rich metadata. There is extensive research on political debates in Hungary as well as worldwide, for example in political science (Kiss, 1998; van Dijk, 2010), discourse analysis (Schirm, 2021), translation studies (Kovács, 2012) or history (Pančur and Šorn, 2016). By increasing the level of processing, more in-depth research is made possible on an ever larger scale of resources. For example, researchers can look at a longer time span, include more speeches, or better identify the ideal targets of their analysis. The importance of research into parliamentary data is further borne out by the fact that there were Parla-CLARIN workshops on creating corpora of parliamentary data at each of the 2018, 2020, and 2022 International Conferences on Language Resources and Evaluation (LREC). The keynote speaker of the 2022 workshop[14] was Luke Blaxill, a historian of British politics and monarchy from 1750 to present, who talked about the possible ways parliamentary data could be used in history and political science. Some specific use cases of the Danish and Basque ParlaMint corpora are demonstrated by Navarretta and Haltrup Hansen (2022) and Escribano et al. (2022), respectively. We welcome enquiries from researchers working in the social sciences and humanities regarding the use of the Hungarian ParlaMint II corpus.

5 Summary

We presented the newly built Hungarian corpus of the second phase of the ParlaMint project, funded by CLARIN-ERIC. The Hungarian corpus of ParlaMint II – alongside all other corpora of European parliamentary data in the participating languages – is a comparable, high-quality parliamentary corpus, which was built with encoding schemas, metadata structure, and Universal Dependencies-type linguistic annotation that are uniform across the corpora of ParlaMint II. Our corpus focuses on the minutes of the Hungarian National Assembly between May 2014 and June 2022. It has a wider time frame, more extensive metadata on speakers and their affiliations, and higher quality linguistic analysis than what was available before for Hungarian. The corpus will be released and be openly available online alongside with the rest of the ParlaMint II corpora at the end of the CLARIN ParlaMint II project (March 2023). Some potential applications of ParlaMint corpora in SSH research are also discussed.

[13] The – detected – false named entity tagging was mainly in connection with punctuation marks. For example, *... képviselő úr*[, *Jobbik*]$_{ORG}$. or *„Bal kéz*[, *jobb kéz”*]$_{MISC}$ *címmel.* both created errors.

[14] For more information on the workshop see https://www.clarin.eu/ParlaCLARIN-III

References

TEI P5: Guidelines for electronic text encoding and interchange. (2017), http://www.tei-c.org/Guidelines/P5/

van Dijk, T.A.: Political identities in parliamentary debates. In: Ilie, C. (ed.) European Parliaments under Scrutiny, chap. 1, pp. 29–56. John Benjamins Publishing Company, Amsterdam (2010)

Erjavec, T., Ogrodniczuk, M., Osenova, P., Ljubešić, N., Simov, K., Pančur, A., Rudolf, M., Kopp, M., Barkarson, S., Steingrímsson, S., Çöltekin, c., de Does, J., Depuydt, K., Agnoloni, T., Venturi, G., Pérez, M.C., de Macedo, L.D., Navarretta, C., Luxardo, G., Coole, M., Rayson, P., Morkevičius, V., Krilavičius, T., Darǵis, R., Ring, O., van Heusden, R., Marx, M., Fišer, D.: The ParlaMint Corpora of Parliamentary Proceedings. Language Resources and Evaluation (2022), https://link.springer.com/article/10.1007/s10579-021-09574-0

Erjavec, T., Ogrodniczuk, M., Osenova, P., Pančur, A., Ljubešić, N., Agnoloni, T., Barkarson, S., Pérez, M.C., Çağrı Çöltekin, Coole, M., Darġis, R., de Macedo, L., de Does, J., Depuydt, K., Diwersy, S., Hansen, D.H., Kopp, M., Krilavičius, T., Luxardo, G., Marx, M., Morkevičius, V., Navarretta, C., Rayson, P., Ring, O., Rudolf, M., Simov, K., Steingrímsson, S., Üveges, I., van Heusden, R., Venturi, G.: ParlaMint: Comparable Corpora of European Parliamentary Data. In: CLARIN Annual Conference Proceedings 2021. pp. 20–25. CLARIN ERIC, Utrecht, The Netherlands (2021)

Erjavec, T., Pančur, A.: Parla-CLARIN: TEI guidelines for corpora of parliamentary proceeding (2019), https://doi.org/10.5281/zenodo.3446164

Escribano, N., Gonzalez, J.A., Orbegozo-Terradillos, J., Larrondo-Ureta, A., Peña-Fernández, S., Perez-de Viñaspre, O., Agerri, R.: BasqueParl: A Bilingual Corpus of Basque Parliamentary Transcriptions. In: Proceedings of the Language Resources and Evaluation Conference. pp. 3382–3390. European Language Resources Association, Marseille, France (June 2022), https://aclanthology.org/2022.lrec-1.361

Kiss, J.: A pártok szimbolikus arculata és érvelési sajátosságai a parlamenti vitákban. Politikatudományi Szemle 7/2, 27–60 (1998)

Kovács, M.: Az európai frazeológiai univerzálék konceptualizációja és fordítási megfeleltetései. Fordítástudomány XIV/1., 48–68 (2012)

Navarretta, C., Haltrup Hansen, D.: The Subject Annotations of the Danish Parliament Corpus (2009-2017) - Evaluated with Automatic Multi-label Classification. In: Proceedings of the Language Resources and Evaluation Conference. pp. 1428–1436. European Language Resources Association, Marseille, France (June 2022), https://aclanthology.org/2022.lrec-1.153

Orosz, Gy., Szántó, Zs., Berkecz, P., Szabó, G., Farkas, R.: HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)

Pančur, A., Šorn, M.: Smart big data: use of Slovenian parliamentary papers in digital history 56(3), 130–146 (2016)

Schirm, A.: Diskurzusjelölők szövegeken innen és túl. Loisir Könyvkiadó, Budapest (2021)

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. pp. 88–99. Association for Computational Linguistics, Vancouver, Canada (Aug 2017), https://aclanthology.org/K17-3009

Szarvas, G., Farkas, R., Kocsor, A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) Discovery Science. pp. 267–278. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

Váradi, T., Simon, E., Sass, B., Mittelholcz, I., Novák, A., Indig, B., Farkas, R., Vincze, V.: E-magyar – A Digital Language Processing System. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), https://aclanthology.org/L18-1208

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In: Proceedings of RANLP. pp. 763–771 (2013)

Üveges, I., Ring, O.: A CLARIN ParlaMint magyar korpusza. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)

Korpuszépítés és -feldolgozás learatott webes tartalomból

Kalcsó Gyula[1], Mihály Eszter[1], Szűcs Kata Ágnes[1]

[1]Országos Széchényi Könyvtár, Digitális Bölcsészeti Központ, Digitális Filológiai és Webarchiválási Osztály, 1014 Budapest, Szent György tér 4-5-6.
{kalcso.gyula, mihaly.eszter, szucs.kata}@oszk.hu

Kivonat: A cikk az Országos Széchényi Könyvtár webaratási tevékenységének eredményeképpen létrejött webarchívum korpuszépítési hasznosulási lehetőségeit mutatja be. A fókusz a tematikus gyűjtések által felépített archívumrészekből történő tematikus korpuszépítésen van. Ismerteti a szövegkinyerés eljárását, amelynek eredményeképpen a learatott WARC-fájlokból a WARCIO Python-könyvtár felhasználásával elsőként HTML, majd az ún. boilerplate-nek a jusText nevű, Pythonban írt eszköz segítségével történő eltávolításával nyers szöveg jön létre. Ismertet egy példaprojektet is, amelynek során az emtsv-vel elemzett szövegekből kinyert adatok vizualizációja történt.

1 Az OSZK webaratási tevékenysége

Az Országos Széchényi Könyvtár a 2000-es évektől foglalkozik born digital dokumentumokkal, előbb a Magyar Elektronikus Könyvtár, majd az időszaki kiadványok számait archiváló Elektronikus Periodika Archívum és Adatbázis és a képi dokumentumokat gyűjtő Digitális Képarchívum keretében. A webarchívum létrehozásának az igénye már korábban is felmerült, de csak a 2015-ös gyűjtőköri szabályzatában rögzítette feladatként a nemzeti könyvtár. A szükséges informatikai és személyi feltételek végül 2017-ben teremtődtek meg, és egy két és fél éves tesztidőszak után elkezdődött a magyar webtérben közzétett tartalmak egy részének időszakos lementése és gyűjteménybe szervezése. Az OSZK webarchívuma azzal a céllal jött létre, hogy reprezentatív képet nyújtson az egy adott időszakban nyilvánosan elérhető, a magyar közönségnek szánt és a kulturális örökség részét képező online tartalomkínálatról, a hungarikumok körébe tartozó elektronikus dokumentumokról. Az Országos Széchényi Könyvtárban folyó webarchiválást a Digitális Bölcsészeti Központ Digitális Filológiai és Webarchiválási Csoportja végzi az 1997. évi CXL. törvény 59/A. paragrafusa és a Kormány 626/2020. (XII. 22.) számú rendelete alapján, az OSZK Gyűjtőköri Szabályzatában definiált elvek szerint. Az OSZK Webarchívum gyűjtőkörébe tartozik a magyar webtérben létező vagy valaha létezett, nyilvánosan közzétett digitális tartalmak összessége, beleértve tehát azokat is, amelyek már az élő weben nem elérhetők, de valahol még megőrződtek.

A dokumentumok gyűjtése háromféle módon történik: válogatva a legfontosabb magyar webhelyekről, kiemelt eseményekhez kötődve a főbb hírforrásokból, illetve általános jelleggel a magyar webtérről. Szelektíven kerül gyűjtésre a tudományos,

kulturális, oktatási, közéleti jellegű tartalmak meghatározott köre. Az általános gyűjtés a .hu domén alatt regisztrált vagy egyéb doménhez tartozó, de magyar közönséget megcélzó nyilvános webhelyekre terjed ki. A webaratás (crawling) csupán azon szervereket érinti, ahonnan technikailag biztosítható a tartalom automatikus lementése. Az aratás során a könyvtár figyelembe veszi a begyűjtő szoftver számára az adott webhely tulajdonosa által beállított korlátozásokat.

Az archivált webtartalom esetében a nemzeti könyvtár elsősorban annak hosszú távú megőrzésére törekszik. A szerzői és személyiségi jogok tiszteletben tartása érdekében a gyűjteménynek csak egy kis része tekinthető meg nyilvánosan, olyan webhelyek, amelyeknél a tartalom tulajdonosa erre engedélyt adott. Az archívum többi része csak a könyvtár zárt szolgáltatási felületén elérhető – elsősorban kutatási célokra.

A nemzeti könyvtár a webtérszintű és a tematikus aratások mellett eseményalapú gyűjtéseket is végez a jelentősebb kulturális, politikai és sportesemények ekről. 2022. február 21. óta elkezdte gyűjteni az orosz–ukrán konfliktussal majd később háborúval kapcsolatos híreket, 75 magyarországi és határon túli hírportálról. A hírek gyűjtése alapvetően a portálokon használt címkék vagy kategóriák alapján történik (ez 445 seed-URL-t jelent). Ezek a mentések hetente egyszer futnak. A gyűjtemény nem nyilvános, azonban a SolrWayback nevű szoftver segítségével készült hozzá egy nyilvános kereső, mellyel a hírek nem nézhetők meg szerzői jogi okok miatt, de a teljes szövere lehet keresni, a metaadatok illetve a szövegkörnyezet megjeleníthető. Március elejétől egy másik tematikus gyűjtemény is létrejött olyan webhelyekről és közösségimédia-oldalakról, melyek a Kárpátalján élő magyarok számára fontosak, beleértve a segítségnyújtással foglalkozó Facebook-csoportokat is. Ez több mint 1000 seed URL-t tartalmaz, s hetente egy aratás gyakorisággal archiválódik. A Digitális Bölcsészeti Központ a hírportálok anyagából épített tematikus korpuszt, a cikk ennek a munkálatait ismerteti a továbbiakban.

2 Előzmények: magyar webkorpuszok

Bár a tematikus korpuszépítés nemzetközi szinten is bevett gyakorlat (l. Barbaresi 2019), az ún. boilerplate removing (l. 3.2) nyelvspecifikussága miatt itt csak a magyar előzményeket tekintjük át. A korábban épült, legalább részben webes forrásokon alapuló korpuszokat Indig (2018) ismerteti (Indig, 2018: 127). Ebben öt korpuszról történik említés, amelyek valójában három projekt keretében készültek el (Webkorpusz, Magyar nemzeti szövegtár, Pázmány-korpusz), az első és a második két verzióban. Ezek módszertana bizonyos mértékig tekinthető mintának, azonban a webaratás (crawling) technikai feltételeinek a változásai, valamint az újabb webes technológiák megjelenése miatt csak részben használható fel újabb korpuszok építésekor.

Maga Indig (2018) is egy módszertant ismertet, amely arra irányul, „hogy a szabadon hozzáférhető anyagból milyen mennyiségű és minőségű magyar nyelvű korpusz gyártható” (Indig 2018: 129). A módszer az alábbi lépésekben nyer ki szöveget a CommonCrawl archívumából: a WARC-indexből kinyeri a letöltendő oldalak adatait (előzetesen kiszűrve a korpuszépítés szempontjából irreleváns tartalmakat), letölti az

oldalt, és azonnal megpróbálja belőle kinyerni a szöveget a jusText Python-eszköz segítségével.

A legfrissebb webkorpusz Nemeskey Dávid Webkorpusz 2.0 projektje keretében készült el (Nemeskey 2020). Ennek mérete messze meghaladja a korábbiakét (több mint 9 milliárd token). Nemeskey a korpusz letöltéséhez és feldolgozásához saját Python-eszközt is készített, amely szabadon elérhető a Githubon[1]. A szöveg kinyeréséhez ez a projekt is a jusTextet használta. További érdekessége, hogy a letöltött szövegeket utólag szűrte (pl. az 1500 szónál rövidebb vagy nem magyar nyelvű tartalmakat eldobta). Ugyancsak fontos sajátossága, hogy többféle duplikátumszűrést is végzett. Dokumentumszinten az ún. minhash algoritmus segítségével távolította el az egyező tartalmakat. Ezen túl azonban a gyakori egyező szövegrészeket (bekezdéseket) is kiszűrte korpusz- és dokumentumszinten is.

3 A szöveg kibontása a learatott webes tartalomból

Az OSZK által szelektíven learatott webes tartalomból más jellegű korpuszok építhetők, mint a korábbi projektek főként CommonCrawl alapú korpuszai. A szelektív aratás módszertana lehetővé teszi, hogy az archivált tartalomból tematikus korpuszok épüljenek.

3.1 A HTML-tartalom kinyerése a WARC-okból

A korpuszépítés első lépéseként a learatott WARC-fájlokból kellett kinyerni a HTML-részeket, amelyek a szöveget is tartalmazzák. Ehhez a WARCIO nevű Python-könyvtárat használtuk, amely a Webrecorder szoftver része[2]. Ennek segítségével a WARC-okban tárolt bármely MIME-típusú tartalom kibontható. Az alábbi táblázat mutatja, hogy a február 21. és június 6. között learatott tartalomból mennyi HTML-t bontott ki a script:

1. táblázat: A korpusz alapját képező learatott webes tartalom főbb adatai

Dátum	WARC-ok száma (mérete)	HTML-ek száma (mérete)
2022-02-21	12 (8,79 GB)	82906 (8,94 GB)
2022-02-28	11 (6,73 GB)	77913 (8,66 GB)
2022-03-07	12 (6,19 GB)	78737 (8,69 GB)
2022-03-14	12 (5,74 GB)	80500 (8,86 GB)
2022-03-21	12 (5,2 GB)	80806 (9,25 GB)
2022-03-28	13 (7,12 GB)	102096 (11,14 GB)
2022-04-05	8 (7,43 GB)	97427 (11,12 GB)
2022-04-11	15 (6,45 GB)	95666 (11,18 GB)

[1] https://github.com/DavidNemeskey/cc_corpus
[2] https://github.com/webrecorder/warcio

2022-04-18	13 (6,11 GB)	94260 (11,09 GB)
2022-04-25	13 (6,32 GB)	100361 (12,09 GB)
2022-05-02	14 (7,14 GB)	98847 (12,11 GB)
2022-05-09	13 (6,7 GB)	100667 (12,1 GB)
2022-05-16	14 (6,63 GB)	105468 (12,67 GB)
2022-05-23	13 (6,34 GB)	102542 (12,27 GB)
2022-05-30	13 (6,34 GB)	101333 (12,1 GB)
2022-06-06	17 (7,19 GB)	105505 (12,9 GB)

3.2 A jusText algoritmus

A következő lépésben a HTML-fájlokból kellett kinyeri a szöveget. Az ún. boilerplate removal a jusText algoritmus (Pomikálek 2011) segítségével történt[3]. Az algoritmus a HTML egyszerű szegmentálása alapján működik. Egyes HTML-tagek tartalmát a webböngészők (alapértelmezés szerint) vizuálisan blokkokként formázzák. Az ötlet lényege, hogy a script szöveges blokkokra szegmentálja a tartalmat a tagek alapján. A használt blokkszintű tagek teljes listája a következő: `blockquote, caption, center, col, colgroup, dd, div, dl, dt, fieldset, form, h1, h2, h3, h4, h5, h6, legend, li, optgroup, option, p, pre, table, td, textarea, tfoot, th, thead, tr, ul`. A blokkokat két vagy több `br` tagből álló rész is elválaszthatja egymástól.

Bár az ilyen blokkok némelyike tartalmazhatja a releváns tartalom és boilerplate keverékét, ez meglehetősen ritka. A legtöbb blokk ebből a szempontból homogén.

Az ilyen blokkokkal kapcsolatban több megfigyelés is tehető:

- A linket tartalmazó rövid blokkok szinte mindig boilerplate jellegűek.
- Minden olyan blokk, amely sok linket tartalmaz, majdnem mindig boilerplate.
- A nyelvileg magas szinten strukturált (pl. mondatokból álló) szöveget tartalmazó hosszú blokkok szinte mindig jók, míg az összes többi hosszú blokk szinte mindig boilerplate.
- Mind a jó (fő tartalom), mind a boilerplate blokkok hajlamosak tömbökben előfordulni, azaz egy boilerplate blokkot általában más boilerplate blokkok vesznek körül, és fordítva.

A jusText azt, hogy egy szöveg nyelvileg magas szinten strukturált vagy nem, egyszerű heurisztikával, a funkciószavak (stopszavak) mennyisége alapján állapítja meg. Míg egy nyelvileg magas szinten strukturált szöveg jellemzően tartalmaz bizonyos arányban funkciószavakat, addig az olyan egyszerű tartalmakban, mint a listák és felsorolások, kevés funkciószó lesz. A script beépítetten tartalmazza a használható nyelvek stopszólistáit (többek között a magyarét is).

[3] Bár létezik a publikált adatok alapján hatékonyabb, ráadásul magyar fejlesztésű algoritmus a problémára (Endrédy–Novák 2013), a forráskód hozzáférhetetlensége miatt le kellett mondanunk a használatáról.

Az algoritmus kulcsgondolata, hogy a hosszú blokkokat, valamint bizonyos rövid blokkokat nagyon nagy megbízhatósággal lehet osztályozni. Az összes többi rövid blokkot ezután a környező blokkok vizsgálatával lehet osztályozni.

3.2.1 Előfeldolgozás

Az előfeldolgozás fázisában a `<header>`, `<style>` és `<script>` tagek tartalma törlésre kerül. A `<select>` tagek tartalmát boilerplate-ként jelöli meg a script. Ugyanez igaz a copyright szimbólumát (©) tartalmazó blokkokra.

3.2.2 Kontextusfüggetlen osztályozás

A szegmentálás és az előfeldolgozás után kontextusfüggetlen osztályozás történik, amelynek során minden blokk besorolódik az alábbi négy osztály valamelyikébe:

rossz (bad) – boilerplate blokk
jó (good) – releváns tartalommal bíró blokk
rövid (short) – túl rövid ahhoz, hogy megbízhatóan oszályozható legyen
majdnem jó (near-good) – a két utóbbi között

Az osztályozás az alábbi algoritmus alapján történik:

```
if link_density > MAX_LINK_DENSITY:
    return 'bad'

# short blocks
if length < LENGTH_LOW:
    if link_density > 0:
        return 'bad'
    else:
        return 'short'

# medium and long blocks
if stopwords_density > STOPWORDS_HIGH:
    if length > LENGTH_HIGH:
        return 'good'
    else:
        return 'near-good'
if stopwords_density > STOPWORDS_LOW:
    return 'near-good'
else:
    return 'bad'
```

A hosszúság karakterszámban van mérve. A linksűrűség az <a> tagekben szereplő karakterek arányával van meghatározva. A stopszavak sűrűsége a stopszólistán szereplő szavak arányával van meghatározva.

Az algoritmus két egész számot (LENGHT_LOW és LENGTH_HIGH), valamint három tizedestörtet (MAX_LINK_DENSITY, STOPWORDS_LOW és STOPWORDS_HIGH) használ paraméterekként. Az első kettő használatos a blokkok rövid, közepes és hosszú kategóriákba sorolásához. Az utolsó három pedig a stopszósűrűség alacsony, közepes és magas mértékének a megállapításához használatos. Az alapértelmezett beállítások a következők:

MAX_LINK_DENSITY = 0,2
LENGTH_LOW = 70
LENGTH_HIGH = 200
STOPWORDS_LOW = 0,30
STOPWORDS_HIGH = 0,32

A közepes és hosszú blokkok osztályozását a blokkméret és a stopszósűrűség alapján a következő táblázat foglalja össze:

2. táblázat: A közepes és hosszú blokkok osztályozása a blokkméret és a stopszósűrűség alapján

Blokkméret (szószám)	Stopszósűrűség	Osztály
közepes	alacsony	rossz
hosszú	alacsony	rossz
közepes	közepes	majdnem jó
hosszú	közepes	majdnem jó
közepes	magas	majdnem jó
hosszú	magas	jó

3.2.3 Kontextusfüggő osztályozás

Az algoritmus kontextusérzékeny részének célja, hogy a rövid és majdnem jó blokkokat a környező blokkok osztályai alapján jónak vagy rossznak minősítse át. A már jónak vagy rossznak minősített blokkok ebben a szakaszban viszonyítási pontként szolgálnak.

Az előzetesen besorolt blokkok rövid és majdnem jó blokkok jó és rossz blokkokkal határolt tömbjeinek tekinthetők. Minden ilyen sorozatot két jó blokk, két rossz blokk vagy egyik oldalon egy jó blokk, a másikon pedig egy rossz blokk övezhet. Az előbbi két eset könnyen kezelhető. A szekvencia minden blokkja jónak, illetve rossznak minősül. Az utóbbi esetben a rossz blokkhoz legközelebbi, majdnem jó blokk szolgál a jó és rossz terület elhatárolására. A rossz és a közel jó blokk közötti összes blokk rossznak minősül. Az összes többi jónak minősül. Ha a tömbben az összes blokk rövid (nincs közel jó blokk), akkor a blokkok mindegyike a jó blokkok közé sorolódik.

A kibontott szövegekben összességében elenyészően kis arányban maradt boilerplate jellegű tartalom. A paraméterezéssel és a magyar stopszólista felülvizsgálatával a script teljesítménye a későbbiekben talán még javítható. Az egyes aratásokból kibontott nyers szöveges tartalom főbb adatai a következők:

3. táblázat: Az egyes aratásokból kibontott szövegek főbb adatai

Dátum	TXT mérete	Tokenszám (kerekítve)
2022-02-21	199 MB	26 millió
2022-02-28	204 MB	26 millió
2022-03-07	206 MB	27 millió
2022-03-14	206 MB	27 millió
2022-03-21	209 MB	26 millió
2022-03-28	260 MB	31 millió
2022-04-05	241 MB	29 millió
2022-04-11	253 MB	31 millió
2022-04-18	257 MB	31 millió
2022-04-25	269 MB	33 millió
2022-05-02	261 MB	34 millió
2022-05-09	269 MB	35 millió
2022-05-16	285 MB	35 millió
2022-05-23	280 MB	34 millió
2022-05-30	269 MB	33 millió
2022-06-06	271 MB	33 millió

4 Adatvizualizáció a korpuszból

Jelen cikknek nem célja, hogy a korpuszból kibányászott adatokat elemezze, csupán azt mutatja be, hogyan tettük lehetővé a szövegek feldolgozását. A korpusz első adatvizualizációs projektje egy interaktív felület elkészítése volt, amely a magyar internetes sajtó orosz–ukrán háborúval kapcsolatos szóhasználatának a változását volt hivatott illusztrálni. Ennek elkészítéséhez először nyelvi elemzés készült az emtsv (Váradi et al. 2017) segítségével. A pipeline-ból a tok, morph, spell, pos modulokra volt szükség, azaz tokenizáltunk, morfológiailag elemeztünk (lemmatizálással), és POS-tagginget végeztünk. Az elemzésből elkészült a szótövek gyakorisági listája, amelynek a felső 2000 elemét automatikus és kézi eszközökkel megtisztítottunk (pl. automatikusan eltávolítottunk bizonyos, a szófelhő szempontjából lényegtelen szófajokat, mint pl. a névutók, valamint kézzel kivettünk a szövegekben maradt boilerplate jellegű tartalomból származó szavakat, „szemetes” tartalmat)[4].

[4] A szűrt szólisták letölthetők az oldal alján található linkekről: https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet

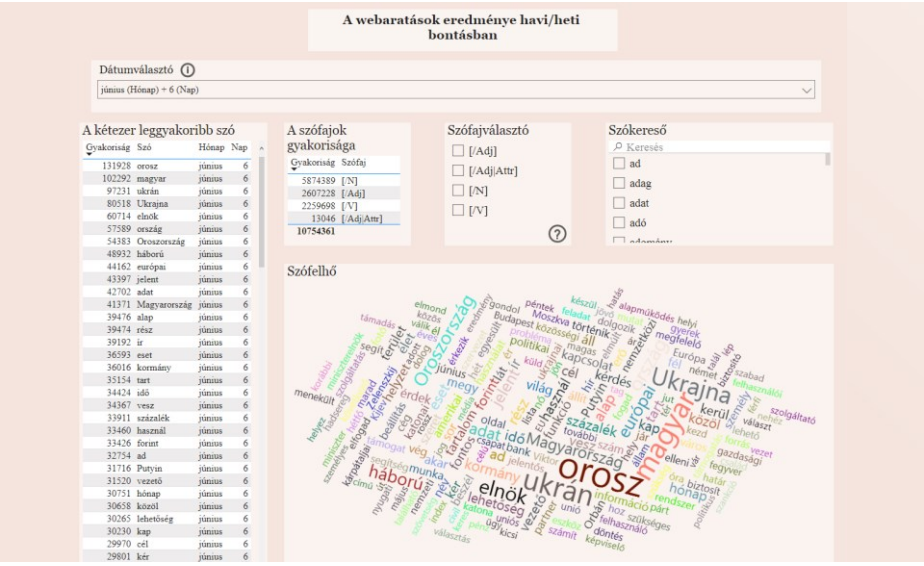


Fig. 1. Az orosz–ukrán háború cikkanyagából épített korpusz adatvizualizációja a Power BI felületén (https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet-interaktiv).

Az interaktív vizualizációt a magyar hírportálok orosz–ukrán háborúról szóló cikkeinek szóhasználatáról a dhupla.hu kreatív oldalán tettük közzé böngészhető, lekérdezhető, adatokat testre szabható formában[5]. Az adatok megjelenítéséhez a Microsoft által üzemletetett, ingyenesen is hozzáférhető Power BI szolgáltatást használtuk. Az alapvető koncepció az volt, hogy a szólisták manuális tisztítása után a kétezer leggyakoribb szót a webaratások heti bontásában egyesével és összesítve is meg tudjuk jeleníteni. A nyitóoldalon találhatóak az Összesített adatok. A Szófelhő az összesített adatok alapján méretezi a szavakat: minél nagyobb, annál több előfordulással szerepel. Az Idővonal segítségével heti vagy havi bontásban lehet szűrni, illetve az egymás melletti mezőket összevonva is lehet egyéni szűréseket végezni. Mellette táblázatos formában tudjuk nyomon követni a szavak alakulását: gyakoriság, hónap-nap és szófaj szerint. Az összesített adatok másik szűrési lehetőségét egy szókereső mező adja, ahol a listában szereplő szavakra egyesével is rá lehet keresni. A szavakat gyakoriság szerint teszi sorba egy diagram az idővonal alatt, melyet Egy statisztika és egy információs panel követ. A második oldalon egyesével tudjuk elemezni, vizsgálni a heti és a havi webaratások szókészletét. A dátumválasztó segítségével egy teljes hónapra vagy egy-egy hétre lehet szűrni. A szavak táblázatos formában gyakoriság és szófaj szerint rendezhetők, illetve szó és szófaj szerint is külön szűrhetők. Az oldal közepén lévő szófelhő interaktivitását az adja, hogy a táblázatok és a vizualizációk egymásra hivatkoznak. Így, ha valamelyik táblázaton szűrést végzünk, annak eredménye automatikusan látszik az ábrákon is. Háromféle megjelenítést ábrázolja a szavakat; a szófelhőn kívül egy sávdiagram mutatja a szavak gyakorisági sorrendjét és a fatérkép pedig a szófajok alakulását teszi láthatóvá az egyes webaratások adatain. Az oldalakat

[5] https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet-interaktiv

elláttuk olyan plusz információkkal is, amelyek a használatban és az eligazodásban segítik a felhasználókat.

5 Összegzés, kitekintés

A cikk az Országos Széchényi Könyvtár webarchívumának módszertanát mutatta be, amelynek segítségével a learatott webes tartalomból (WARC-fájlokból) nyelvileg elemzett tematikus korpuszt lehet építeni. Az eljárás további finomításával lehetséges tetszőleges, a webarchívumban mentett szövegegyüttesből korpuszok építése, valamint a nyelvileg elemzett szövegek adat-/szövegbányászati feldolgozása. A rugalmas korpuszépítéshez bizonyos problémákat még meg kell oldani: terveink közt szerepel a webarchívum hatalmas szövegmennyiségének automatikus tárgyszavazása (topic modeling), valamint fontos metaadatok (pl. a megjelenési dátum) automatikus szöveghez rendelése. Mindezek alapján lehetségessé válhat különböző témájú részkorpuszok építése akár egy bizonyos időszakra vonatkozóan is, ami a webarchívum bölcsészet- és társadalomtudományi kutatási hasznosulását nagyban elősegítené. Problémát jelent ugyanakkor, hogy a jogvédett szöveges tartalom (pl. a hírportálokról aratott szöveg) nem publikálható, megfelelő felhasználói felület létrehozásával biztosítani kell, hogy az adat-/szövegbányászat elvégezhető legyen.

Bibliográfia

Barbaresi, A.: The Vast and the Focused: On the need for thematic web and blog corpora. 7th Workshop on Challenges in the Management of Large Corpora (CMLC-7), Jul 2019, Cardiff, United Kingdom. pp. 29–32. (2019)

Indig B., Sass B., Simon E., Mittelholcz I., Kundráth P., Vadász N.: emtsv – Egy formátum mind felett. In: Berend Gábor, Gosztolya Gábor, Vincze Veronika (szerk.): MSZNY 2019, XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 235–247. (2019)

Endrédy, I., Novák, A.: More Effective Boilerplate Removal—The GoldMiner Algorithm. Polibits – Research Journal on Computer Science and Computer Engineering with Applications, 48, 79–83. (2013)

Endrédy, I., Prószéky, G.: A Pázmány Korpusz. Nyelvtudományi Közlemények, 112. pp. 191–205. (2016)

Indig, B.: Közös crawlnak is egy korpusz a vége – Korpuszépítés a CommonCrawl .hu domainjából. In: XIV. Magyar Számítógépes Nyelvészeti Konferencia : MSZNY 2018. pp. 125–134. Szegedi Tudományegyetem, Szeged (2018)

Indig, B., Kákonyi, T., Novák, A.: Crawling in Reverse – Lightweight Targeted Crawling of News Portals. In: Kubis, M. (szerk.): Proceedings of the 9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. pp. 81–87. (2019).

Indig, B., Knap, Á., Sárközi-Lindner, Zs., Timári, M., Palkó, G.: The ELTE.DH Pilot Corpus. In: Barbaresi, A. (szerk.): Proceedings of the LREC 2020 12th Web as Corpus Workshop (ACL SIGWAC) pp. 33–41. (2020)

Nemeskey, D. M.: Natural Language Processing methods for Language Modeling. PhD thesis. Eötvös Loránd University (2020)

Pomikálek, J.: Removing Boilerplate and Duplicate Content from Web Corpora. PhD thesis. Masaryk University, Brno (2011)

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: MSZNY 2017, XIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, pp. 49–60. (2017)

Tagmondatokra bontás és NP-chunking függőségi alapon

Dömötör Andrea[1,2,3], Nemeskey Dávid[1,2]

[1]ELTE BTK TI Digitális Bölcsészet Tanszék
1088 Budapest Múzeum krt. 6-8.
[2]Digitális Örökség Nemzeti Laboratórium
[3]PPKE BTK Nyelvtudományi Doktori Iskola
1088 Budapest, Mikszáth Kálmán tér 1.
{domotor.andrea,nemeskey.david}@btk.elte.hu

Kivonat Ebben a cikkben a tagmondatokat és a köztük lévő kapcsolat típusát a függőségi elemzés mintázataiból kíséreljük meg meghatározni. Mivel ennek a feladatnak a teszteléséhez még nincs gold sztenderd adatunk, a módszerünket kipróbáltuk egy másik feladaton, az NP-chunkingon is. Ez utóbbi kiértékelésénél nehézséget okozott, hogy az elvben gold sztenderd korpuszok több hibát is tartalmaztak, mind a függőségi elemzésben, mind az NP-chunkingban. Mindezekkel együtt 89%-os f-score-t értünk el, ami ugyan elmarad a state-ot-the-arttól, de abból a szempontból mégis ígéretes, hogy ezt az eredményt egy egyszerű szabályrendszerrel értük el. Ez alapján a függőségi elemzés mintaillesztése további kutatásra érdemes módszer lehet a hasonló feladatokban. **Kulcsszavak:** NP-chunking, függőségi elemzés, tagmondatok, mintaillesztés

1. Bevezetés

A legtöbb korpusz és nyelvfeldolgozó eszköz a szöveget mondatok, azon belül pedig tokenek összességeként kezeli. A mondat és a token szintje között lehetnek még többszavas kifejezések (chunkok), azonban a tagmondat mint nyelvi szint jellemzően nem jelenik meg a számítógépes szövegfeldolgozásban.

Az összetett mondatok tagmondatai teljes értékű mondatoknak tekinthetők, így elkülönítésük hasznos lehet olyan feladatok esetén, ahol fontosak a (tag)mondathatárok, ilyen lehet például az elváló igekötők és igéik összekapcsolása. Ezen kívül a tagmondatok közötti viszonyok meghatározóak a szövegértelmezés szempontjából: egy kötőszón múlhat, hogy egy szövegrész tartalma egy másikéból következik-e vagy fordítva.

Ebben a cikkben a tagmondatokat és a köztük lévő kapcsolat típusát a függőségi elemzés mintázataiból kíséreljük meg meghatározni. Mivel ennek a feladatnak a teszteléséhez még nincs gold sztenderd adatunk, a módszerünket kipróbáltuk egy másik feladaton, a főnévi csoportok felderítésén (*NP-chunkingon*) is. Bár itt természetesen két eltérő feladatról van szó, látni fogjuk, hogy mindkettő megoldható függőségi mintaillesztéssel.

Magasabb szintű nyelvi feladatok megoldása függőségi, vagy frázisstruktúra-nyelvtan segítségével természetesen nem új gondolat. Csak példaképp, Recski és mtsai (2009) a Szeged Treebank (Csendes és mtsai, 2005) elemzési fáiból nyerték ki az NP chunking korpuszt; Metheniti és mtsai (2019) pedig függőségi elemzés alapján azonosítottak nyelvtani szabályokat. Tudomásunk szerint azonban az ebben a cikkben leírt feladatokat magyar nyelvre még nem próbálták korábban függőségi elemzés alapján megoldani.

2. Mintaillesztés függőségi fákon

Szabály alapú felismerési feladatokban — akár az összetett mondatokról, akár a (maximális) főnévi csoportok felismeréséről beszélünk — meghatározott nyelvtani konfigurációk meglétét ellenőrizzük a mondatokban. A függőségi nyelvtanok esetén ez egyszerűen átfogalmazható gráfelméleti alapokra.

A függőségi elemzés egy csúcs- és élcímkézett fát ad eredményül, ahol minden csúcs egy tokennek, minden él pedig egy nyelvtani függőségnek felel meg. Hasonlóan, a keresett nyelvtani minták is egy fával írhatók le, hiszen általában mind a csúcsok adatai (szó, lemma, szófaj, stb.), mind a közöttük lévő kapcsolatok fontosak (példa összetett mondatok esetén: két ige, amik egy kötőszón keresztül kapcsolódnak). A feladat ekkor a mintafával izomorf részgráf keresése az elemzési fában.

A mintakeresés fa gráfokban jól kutatott terület, nyelvi vonatkozásban is (Kilpeläinen és mtsai, 1992). Több rendszert is javasoltak függőségi fák mintaillesztéséhez (Wallis, 2008; Nichols és Reisert, 2014; Luotolahti és mtsai, 2015), azonban ezek többsége nem elérhető. Az utolsó az egyedüli, amelyik fellelhető, de csak offline keresésre használható, és nem specifikálja, hogy a minta hol fordul elő a mondatban. Ezért egy saját mintaillesztő írása mellett döntöttünk.

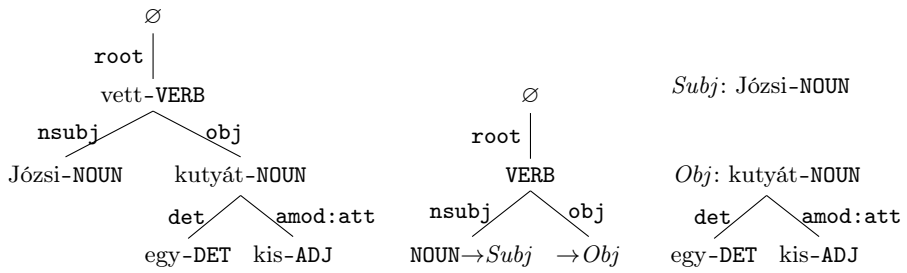
Az illesztés alapvetően a Kilpeläinen és mtsai (1992)-ben leírt *unordered path inclusion* (rendezetlen út tartalmazás) algoritmust követi. Két pontban bővítettük ki: egyrészt a csomópontok (esetünkben tokenek) több tulajdonságára (lemma, szófaj) is lehet keresni regex alapon, másrészt a minta élein megadott függőségi kapcsolatokkal is illesztünk. Az illesztés lehet pozitív, illetve negatív is; negatív feltétel teljesülése megakadályozza az illesztést.

Ha egy minta előfordul a mondatban, a rendszer kivágja az annak megfelelő részt a mondatfából. A minta csomópontjait meg lehet jelölni változókkal, amik megjelennek az eredményfában is. A jelölt szavak részfái, illetve a hozzájuk tartozó mondatrészek ezután egyszerűen lekérhetők.

Egy mintaillesztés példa látható az 1. ábrán. 1a. a *„Józsi vett egy kis kutyát”* mondat függőségi elemzése[1], 1b. pedig az illeszteni kívánt minta, ami igei (`VERB`) állítmánnyal (`root`) rendelkező mondatokban keresi a főnévi (`NOUN`) alanyt és a (bármilyen) tárgyat. Az előbbit *Subj*, az utóbbit *Obj* változóval jelöltük. 1c. mutatja a rendszer kimenetét: az egyes változókhoz tartozó részfákat.

Érdemes megemlíteni, hogy bár az algoritmus bármilyen függőségi címkekészlettel működik, a leírt rendszer a mára már sztenderdnek számító Universal

[1] A szemléltetés kedvéért a megszokottól eltérően „álló” fával.



(a) A mondat (b) A minta (c) Az eredmény

1. ábra. Példa mintaillesztésre

Dependencies (UD)[2] címkéit használja – ahogy a legtöbb magyar nyelvre elérhető szintaktikai elemző eszköz (Stanza, UDPipe, HuSpaCy) is. Hasonlóan, a módszer bizonyára adaptálható lehetne frázisstruktúra-nyelvtanok elemzési fáira is; ezek szerepét azonban a természetesnyelv-feldolgozásban szinte teljesen átvette a függőségi elemzés. A cikkben ezért mi is ez utóbbira koncentrálunk.

3. Tagmondatokra bontás

3.1. Kapcsolódó irodalom

A tagmondatkapcsolatok automatikus felismerésére Szemes (2021) tett kísérletet a közelmúltban. Módszere a mondatközi írásjelek, valamint a kötőszavak és vonatkozó névmások mintázatain alapul. Eszerint a tagmondathatárok egy írásjel és egy kötőszó vagy vonatkozó névmás szekvenciájával azonosíthatók, a tagmondatok közötti kapcsolat típusát pedig a kötőszók és vonatkozó névmások (kézi) klasszifikációjával határozta meg a szerző. A tanulmány elsősorban a mondattípusok gyakoriságának stilisztikai és irodalomtörténeti vonatkozásaira fókuszál, így magának a tagmondatkapcsolat-meghatározó algoritmusnak a kiértékelésére nem tér ki. Megemlíti azonban a módszer azon hiányosságát, miszerint az csak azokra az összetett mondatokra alkalmazható, ahol van mondatközi írásjel és kötőszó (vagy vonatkozó névmás) is. Ez azonban nem minden esetben igaz, a *hogy* kötőszó például gyakran elhagyható a tagmondathatárokról, és bár a helyesírás valóban megköveteli a tagmondatok közötti vesszőt, ezt a szabályt a szövegeket létrehozó nyelvhasználók gyakran nem tartják be. Másrészről a tagmondathatár-keresés azért sem mindig elégséges módja a tagmondatok meghatározásának, mert a tagmondatok nem feltétlenül folytonosak, lehetnek bennük megszakítások, közbeékelések. Ugyanakkor a kötőszók lexikális vizsgálata a mi módszerünknél jóval részletesebb tagmondatkapcsolat-kategorizálást tesz lehetővé.

[2] `https://universaldependencies.org/`

A tagmondatok meghatározására függőségi elemzésre alapozó módszert használ Gyulai (2021), azonban a miénkkel ellentétben bottom-up megközelítéssel. Ennek lényege, hogy olyan összefüggő kifejezéseket keres, aminek az első és utolsó eleméből rekurzívan el lehet jutni a tagmondat fejéig, egy V POS taggel ellátott összetevőig vagy kopuláig. Ez a módszer jól alkalmazható a Szeged Treebanken, ahol valóban minden (tag)mondat feje igei elem, hiszen az elliptált igék és zéró kopulák is megjelennek a függőségi elemzésben üres igei csomópontokként. Más szövegekben azonban nem feltétlenül tartalmaz minden tagmondat igét. Ezen kívül, bár a módszernek kétségtelenül előnye, hogy nem függ a függőségi elemzés címkekészletétől, az elemzési sémától azonban így is függ: például a UD elvei szerint a kopulás mondatok esetén nem a kopula a mondat feje, hanem a predikatív névszó, így ezeknél a mondatoknál ezzel a módszerrel nem jutunk el a kopuláig. Végül, Gyulai (2021) módszere csak a tagmondatokat állapítja meg, a miénk azonban kísérletet tesz a tagmondatok közötti kapcsolatok típusának meghatározására is.

3.2. Tagmondatkapcsolat-típusok

A Magyar grammatika (Lengyel, 2000) alapján megkülönböztetünk alárendelő és mellérendelő összetett mondatokat. Utóbbi alcsoportjai (kapcsolatos, ellentétes, megengedő, magayrázó, következtető) között leginkább csak a kötőszó alapján lehet különbséget tenni, szerkezeti eltérés nincs közöttük, ezért a mellérendelő összetételek altípusaival nem foglalkoztunk. Az alárendelő monatoknál a mellékmondat által kifejtett mondatrész szintaktikai szerepe alapján határoztuk meg az altípusokat (akárcsak a Magyar grammatika). Külön csoportba soroltuk a vonatkozó mellékmondatokat, ahol a kötőszó szerepét egy vonatkozó névmás tölti be.

A tagmondatkapcsolatok típusait az 1. táblázat foglalja össze. A példák forrása a Szeged Treebank.

Főtípus	**Altípus**	**Példa**
Mellérendelő		*Már 5.05 volt, azonban a Zoli bácsi és a felesége még seholsem volt.*
Alárendelő	Alanyi	*A könyvben részletesen le volt írva, hogy ez a szelidítés hogyan történt.*
	Tárgyi	*Mindenki nagyon izgatottan várta, hogy elinduljon a busz.*
	Határozós	*S úgy láttam, mintha kezdene megynugodni.*
	Jelzős	*Akkora pizzát adtak, hogy el kellett hoznunk.*
	Vonatkozó	*Zoli bácsi hozott fényképeket, amelyeket a felesége készített.*

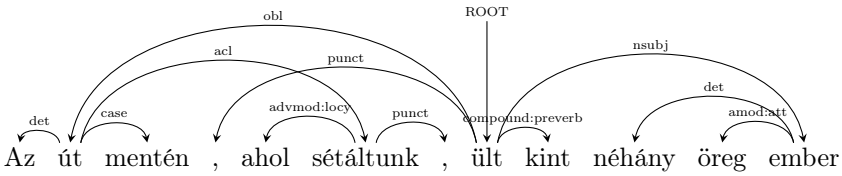
1. táblázat. Az összetett mondatok típusai

3.3. Tagmondatok meghatározása függőségi elemzés alapján

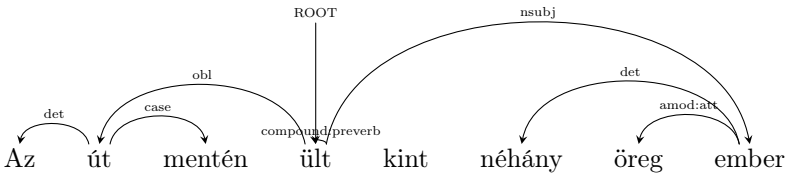
A tagmondatok meghatározásánál a függőségi elemzésből indultunk ki. A tagmondatokra úgy tekintünk, mint a teljes mondat függőségi elemzési fájának rész-

fái, így a tagmondatok meghatározása lényegében a részfák gyökércsomópontjainak meghatározását jelenti. Ez a megközelítés előnyösebb a szekvenciális (pl. IOB) címkézésnél, a tagmondatok ugyanis nem feltétlenül folytonosak. A gyökércsomópontok meghatározásával azonban a szétszakított tagmondatok részei is "egymásra találnak" a függőségi elemzésben (amennyiben persze az helyes). az 1–3. példák egy nem folytonos tagmondatot tartalmazó mondat függőségi elemzés szerinti felbontását mutatják. A mondat gyökere (ROOT) a főmondat gyökércsomópontja, a vonatkozó mellékmondat (3) gyökerét pedig az `acl` függőség alapján határozzuk meg. A gyökércsomópontokból kiindulva a függőségekből megkapjuk a tagmondatokat, azok sorrendjétől és elhelyezkedésétől (és általában a szórendtől) függetlenül, azaz a módszerünk nem érzékeny arra, hogy az *ahol sétáltunk* vonatkozó mellékmondat beékelődik-e a főmondatba, vagy követi azt.

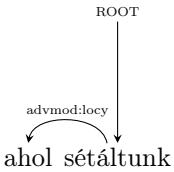
(1)



(2)

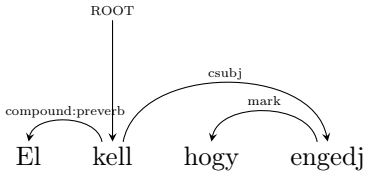


(3)

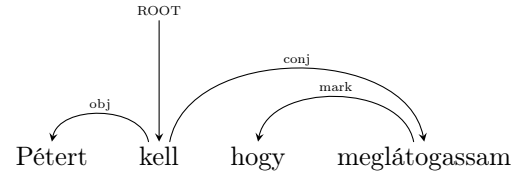


Problémásak azonban az úgynevezett mondatátszövődések (*El kell hogy engedj., Pétert kell hogy meglátogassam.*), ezeknél ugyanis a függőségi elemző számára nehéz az elváló igekötő vagy igei bővítmény fejének megtalálása. (4) és (5) példamondatokat Stanzával (Qi és mtsai, 2020) elemeztük, és mindkét esetben hibás elemzést kaptunk: (4) igekötője és (5) tárgya is a főmondathoz (*kell*) kapcsolódik a mellékmondatok megfelelői igéi helyett.

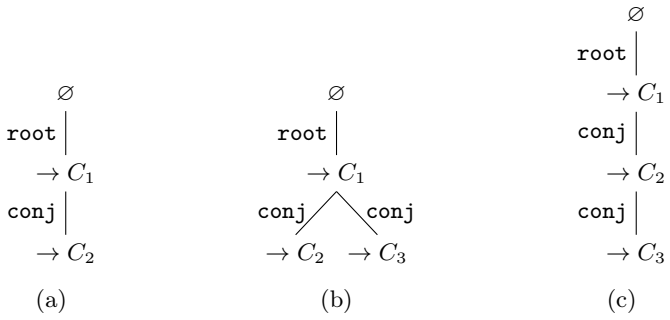
(4)



(5)



A tagmondatok gyökércsomópontjaira illeszkedő mintázatok meghatározásánál értelemszerűen az ideális esetből, azaz a helyes függőségi elemzésből indultunk ki. A mintázatok leírásához a Universal Dependencies címkekészletet használtuk, mert a legtöbb magyar nyelvre elérhető szintaktikai elemző eszköz (Stanza, UDPipe, HuSpaCy) ezt a sémát használja. A tagmondatokat meghatározó függőségi kapcsolatok a következők:

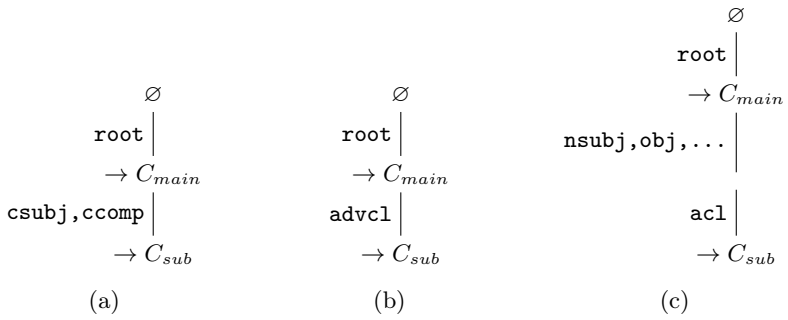


2. ábra. Minták mellérendelt tagmondatok azonosításához

- `root`: a főmondat, vagy mellérendelés esetén az első tagmondat gyökere
- `conj` (conjunct): mellérendelt tagmondat gyökere
- `csubj` (clausal subject): alanyi mellékmondat gyökere
- `ccomp:obj` (clausal complement: object): tárgyi mellékmondat gyökere
- `advcl` (adverbial close modifier): határozói mellékmondat gyökere
- `acl` (clausal modifier of noun)
  - jelzői mellékmondat, ha a fej egy másik tagmondat gyökércsomópontja
  - vonatkozó mellékmondat, ha a fej egy névszó

A vonatkozó mellékmondatok kivételével a tagmondatok gyökerének feje egy másik tagmondat (vagy a teljes mondat) gyökere. Ezért a függőségi elemzésben jellemzően láncszerű mintákat keresünk. A 2. ábrán néhány mellérendelő összetett mondat függőségi mintázatai láthatók. Többszörös összetételeknél a tagmondatok bizonyos esetekben láncszerűen kapcsolódnak egymáshoz (2c), más esetekben viszont egy tagmondat több mellérendelt tagmondat gyökereként van elemezve a függőségi elemzésben (2b).

Az alárendelő összetett mondatok esetén (3. ábra) a függőségi viszonyok tükrözik a tagmondatkapcsolatok struktúráját.



3. ábra. Minták alárendelt tagmondatok azonosításához

3.4. A tagmondatkapcsolatok jelölése CoNLL-ben

A tagmondatok közötti viszonyok nem reprezentálhatók fa struktúrában, egy tagmondat ugyanis több másik tagmondathoz is kapcsolódhat: lehet egyszerre egy tagmondat alárendeltje és egy másik mellérendeltje. Ezért a tagmondatkapcsolatok CoNLL-ben való megjelenítéséhez a CoNLL-U formátum továbbfejlesztett függőségi gráf (DEPS) mezőjéből indultunk ki. A továbbfejlesztett függőségi gráf lehetővé teszi a mellérendelések megjelenítését azáltal, hogy az elsődleges függőségi viszonyok mellett megadhatók további fej-reláció párok is (4. ábra).

1	They	they	PRON	PRP	Case=Nom|Number=Plur	2	nsubj	2:nsubj|4:nsubj
2	buy	buy	VERB	VBP	Number=Plur|Person=3|Tense=Pres	0	root	0:root
3	and	and	CONJ	CC	_	4	cc	4:cc
4	sell	sell	VERB	VBP	Number=Plur|Person=3|Tense=Pres	2	conj	0:root|2:conj
5	books	book	NOUN	NNS	Number=Plur	2	obj	2:obj|4:obj
6	.	.	PUNCT	.	_	2	punct	2:punct

4. ábra. Példa az UD továbbfejlesztett függőségi gráfjára

A tagmondatkapcsolatok jelöléséhez a CoNLL-U+ formátumot kiegészítettük egy `COMP:RELS` mezővel, ahol a tagmondatok gyökércsomópontjai kapnak

annotációt (a többi token esetén a mező _ jelet kap). A mondat gyökéreleme minden esetben `0:root` címkét kap, a belőle származó tagmondatot tekintjük a főmondatnak[3]. A többi tagmondat gyökerének annotációjában megadjuk annak a tagmondatnak a gyökerét (token id-ját), amellyel a tagmondat viszonyban áll, majd kettősponttal elválasztva meghatározzuk a tagmondatkapcsolat típusát. Ha egy tagmondat több másik tagmondattal is viszonyban áll, akkor mindegyiket jelöljük | jellel elválasztva.

Az egyes tagmondattípusok jelölései a következők:

- Mellérendelés: `coord`
- Alanyi alárendelés: `subj`
- Tárgyi alárendelés: `obj`
- Határozói alárendelés: `obl`
- Jelzői alárendelés: `att`
- Vonatkozó mellékmondat: **rel**

```
# global.columns = FORM WSAFTER FEATS UPOS XPOS LEMMA ID DEPREL HEAD COMP:RELS

Úgy " " PronType=Dem ADV _ úgy 1 advmod:mode 2 _
volt " " Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act VERB _ van 2 root 0 0:root
megbeszélve " " VerbForm=Conv ADV _ megbeszélve 3 advmod:mode 2 _
, " " _ PUNCT _ , 4 punct 2 _
hogy " " _ SCONJ _ hogy 5 mark 8 _
hajnali " " Case=Nom|Degree=Pos|Number=Sing|Number[psed]=None|Number[psor]=None|Person[psor]=None ADJ _ hajnali 6 amod:att 7 _
ötkor " " _ ADV _ ötkor 7 advmod:tlocy 8 _
indulunk " " Definite=Ind|Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act VERB _ indul 8 advcl 2 2:obl
és " " _ CCONJ _ és 9 cc 10 _
találkozunk " " Definite=Ind|Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act VERB _ találkozik 10 conj 8 8:coord|2:obl
a " " Definite=Def|PronType=Art DET _ a 11 det 12 _
Postánál " " Case=Ins|Number=Sing|Number[psed]=None|Number[psor]=None|Person[psor]=None PROPN _ Posta 12 nmod:obl 10 _
. "\n" _ PUNCT _ . 13 punct 2 _
```

5. ábra. Példa egy többszörösen összetett mondat CoNLL-U+ annotációjára

Az 5. ábrán látható egy példamondat tagmondatkapcsolatokat is tartalmazó CoNLL-U+ annotációja. Jelenleg zajlik egy gold standard tesztkorpusz annotációja, amely alkalmas lesz majd a tagmondat-azonosító módszerünk kiértékelésére.

4. A módszer tesztelése egy másik feladaton: NP-chunking

4.1. NP-chunking

Az NP-chunking feladat célja a főnévi (névszói) csoportok (*noun phrase*-ek, *NP*-k) azonosítása a mondatban. A főnévi csoport egy olyan szerkezet (frázis), amely főnévi funkciót tölt be.

A főnévi csoportok „rekurzívak”, mivel tartalmazhatnak más főnévi csoportokat is: a „Piroska könyve” kifejezés egy NP, de a „Piroska” szó maga is az. Emiatt

[3] Bár nyelvészeti szempontból a főmondat fogalma nem értelmezhető mellérendelés esetén, a jelölésrendszerünkben akkor is meghatározunk egy "főmondatot", ha csak mellérendelő tagmondatkapcsolatok fordulnak elő a mondatban. Főmondat alatt itt azt a tagmondatot értjük, amelyikből a gráf kiindul. Ez a tagmondatkapcsolatok meghatározásának viszonyítási alapja.

olyan lineáris címkézést, ami az összes csoportot jelöli, nem lehet – de legalábbis nem praktikus – készíteni. Az NP chunking feladat során ezért tipikusan a két végletet címkézik (Ramshaw és Marcus (1995) base NP-definíciója alapján):

- a *minimális NP* nem tartalmaz más NP-ket;
- a *maximális NP* épp ellenkezőleg, nem része magasabb szintű főnévi csoportnak.

A főnévi csoport a frázisstruktúra-nyelvtan eleme, és nem képezhető le triviálisan függőségi nyelvtani terminológiára. Az NP-k jól megfoghatók bottom-up konstrukciós szabályokkal (Recski, 2014), a minimális NP-k pedig akár véges nyelvtannal is leírhatók. Azonban mindkét konstrukció jelentős mértékben támaszkodik a szavak sorrendiségére, ami a függőségi nyelvtanban nem jelenik meg. Hasonló bottom-up megközelítés ezért tisztán a függőségi gráf alapján nem lehetséges.

A függőségi elemzésnek azonban megvan az az előnye a frázisstruktúrával szemben, hogy a különböző nyelvtani szerepek (mint például igei argumentumok: `nsubj`, `obj`, `nmod:obl`) megjelennek függőségként a fában. Ezt használva egy top-down algoritmus megkeresheti a főnévi csoportok fejeit a mondatbeli szerepeik alapján. A teljes NP ezek után kinyerhető a fejtől függő szavak felsorolásával.

A fentiek miatt az algoritmus számára az NP belseje egy fekete doboz, így csak maximális NP-k megtalálására alkalmazható. A továbbiakban ezért a maximális főnévi csoportokra koncentrálunk.

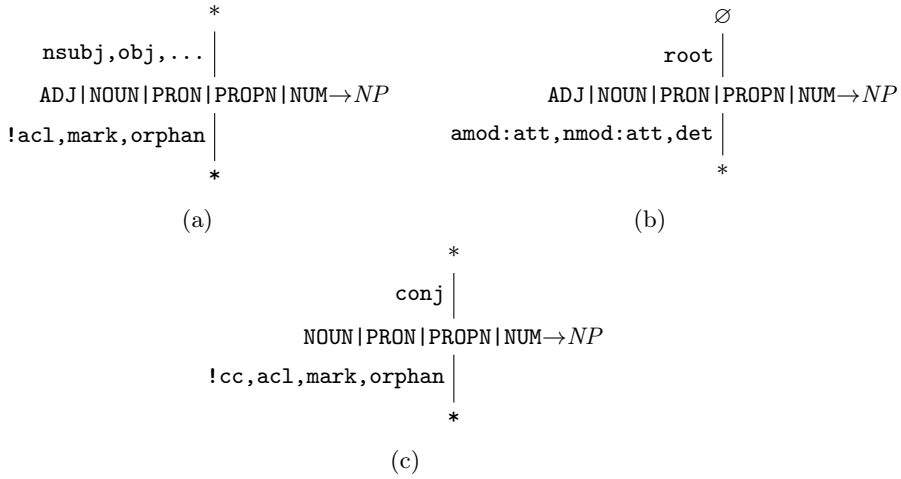
4.2. Az adat

A kísérlet során amennyire tudtunk, gold sztenderd adatokat használtunk (de lásd 4.5). Ennek megfelelően az annotációkat a Szeged korpusz különböző változataiból vettük: a morfológiai és függőségi elemzést a Szeged UD treebankből[4], az NP-ket pedig az NP-chunking korpuszból (Recski és mtsai, 2009). Mivel előbbi a Szeged korpusznak csak egy apró szeletét tartalmazza, egy egyszerű heurisztika alapján kiválogattuk az összes olyan mondatot, ami mindkettőben megtalálható. Ez 812–107–98 mondatot eredményezett a train–valid–test halmazokban.

4.3. A minták

A tanítókorpusz alapján három fő mintát azonosítottunk, ezeket a 6. ábra mutatja.

[4] `https://universaldependencies.org/treebanks/hu_szeged/index.html`



6. ábra. Minták főnévi csoportok azonosításához. A `!` a negatív feltételt jelöli

Az első minta olyan, névszói paradigmához tartozó szavakat keres, amelyek a `nsubj,obj,obj:lvc,iobj,nmod,nmod:obl,obl,obl:lvc,orphan` függőségei valaminek. A második a névszói állítmányokat fedi le, míg a harmadik a konjunkcióban álló főnévi csoportokat. Utóbbiban az `ADJ` szófaj hiánya nem tévedés: az `ADJP`-k tévesen `NP`-ként való címkézését kerüljük el vele.

Mivel a főnévi csoportok és a nyelvtani függőségek nem fedik pontosan egymást, az `NP` alatti részfára is kellett megkötéseket tennünk. Mindegyik mintánál tiltjuk a `acl,mark,orphan` függőségeket, az 6c-ben pedig a `cc`-t is. Ezen felül a visszaadott részfából kivesszük az összes, vonzat típusú élet (`nsubj,obj,...`), hiszen azok külön NP-t képeznek.

Mivel a mintaillesztő algoritmusunk nem veszi figyelembe a szavak sorrendjét, az alaprendszert ki kellett egészítenünk még egy NP-specifikus lépéssel: a fejtől jobbra minden `case,advmod,cc,punct` token törlünk, illetve a csoport elejéről az esetlegesen odakerült vesszőket is.

4.4. Eredmények

A chunkert lefuttattuk a teljes korpuszon, és a `seqeval`[5] csomaggal értékeltük ki. Az eredményeket a 2. táblázat listázza.

Mint látható, az eredmény elmarad a „state of the art”-tól: a 89%-os f-score nem mérhető a `emBERT` 97 százalékához (Nemeskey, 2021); gyakorlatilag a `hunchunk` szintjét hozza. Viszont a módszer erejét mutatja, hogy ezt három egyszerű szabály alkalmazásával értük el. A rendszer szabályalapúságának megfelelően a pontszámok függetlenek attól, hogy a korpusz melyik részén futtatjuk le.

[5] `https://github.com/chakki-works/seqeval`

	Accuracy	f-score
train	93,53%	0.89
valid	93,84%	0.88
test	93,86%	0.89

2. táblázat. NP-chunking kiértékelése

4.5. Értékelés és hibaelemzés

A hibásan címkézett szavak kézi átnézése során három hibaosztályt sikerült elkülöníteni. Ezeknek csak egyike származik a rendszerből; a másik kettő a felhasznált adatok problémáira mutat rá.

A Szeged UD Treebank sajnos számos szisztematikus hibás elemzést tartalmaz. A legfeltűnőbb a melléknévből képzett határozószók (*tisztán, remekül*, stb.) `ADJ`-ként címkézése. Ez magát a függőségi elemzőt is megtéveszthette, ami sokszor `obl`-ot társított ezekhez a szavakhoz, így ezeket mi is tévesen `NP`-nek jelöltük. Egyéb hibaként előfordultak `nmod:att` függőségek igei fejjel, rosszul kötött `conj`-ok, stb.[6]

Az NP-chunking korpusz sem hibamentes. Több esetben is előfordul, hogy a birtokos NP-összetételeket (pl. *a kormány egészének ellenőrzését*) nem vonja össze egy magasabb szintű főnévi csoportba, annak ellenére, hogy mind Kornai (1985), mind Recski (2014) listázza ezt a szabályt. Hasonló inkonzisztencia figyelhető meg az adpozíciók címkézésében: *Budapestre, a csepeli szabadkikötőbe* két külön NP, míg *a fővárosban a XVIII. kerületben* egy. Végül, vannak határozói „NP”-k, mint a *még egyszer* (`ADV ADV`).

Ezek a hibák sajnos megnehezítik a rendszer teljesítményének pontos kiértékelését, illetve továbbfejlesztését. Jó hír, hogy a fentiek fényében a 89%-os f-score egy konzervatív becslésnek tűnik; a „nettó” teljesítmény valószínűleg így 90% felett lehet. Viszont ahhoz, hogy a rendszert javítsuk, és újabb mintákat tudjunk hozzáadni, először a tanítókorpuszt kell kijavítani, vagy egy jobb minőségű tanítókorpuszt előállítani. A szerzők ezért a jövőben először erre terveznek koncentrálni.

Egy másik kísérlet a megközelítés más gyengeségére is rámutatott. Eleinte egy, a Szeged UD Treebanken feltanított Stanza modellel (Qi és mtsai, 2020) elemeztük a mondatokat. A függőségi elemzés rossz minősége miatt azonban az eredmények nagyon elmaradtak a fent közölttől. Ez két okra vezethető vissza: egyrészt a Szeged UD Treebank nagyon kicsi, használható modellt tanítani rajta nem igazán lehet. Másrészt egy szabályalapú megközelítés igényli a pontos bemenetet, ezért rendszerünk nem lesz olyan robosztus, mint egy gépi tanuláson alapuló modell. Azonban ez a hátrány csökkenthető lenne egy megfelelő méretű és minőségű függőségi nyelvtan tanítókorpusszal.

Végezetül, az NP-chunking feladat rámutatott a felhasznált gráfos leírás elégtelen voltára: ahhoz, hogy a le tudjuk vágni a felesleges függőségeket az NP körül,

[6] A szerzők ezeket a hibákat a korpusz Github repozitóriumában is jelentették.

illetve, hogy minimális NP-ket le tudjunk írni vele, sorrendezett reprezentációra van szükség. A jövőben ebbe az irányba szeretnénk továbbfejleszteni a rendszert.

5. Összegzés

Kutatásunk célja a függőségi elemzés mintaillesztésének alkalmazása volt a tagmondatra bontás és az NP-chunking feladatokban. Bár előbbihez még nem áll rendelkezésre tesztelésre használható gold sztenderd adat, az előzetes tapasztalataink alapján elmondható, hogy a tagmondatok és a köztük lévő viszonyok elvileg jól meghatározhatók a függőségi elemzés alapján. Kérdéses azonban, hogy mennyire számíthatunk a függőségi elemzés helyességére.

Az NP-chunking módszerünk kiértékelésénél is nehézséget okozott, hogy az elvben gold sztenderd korpuszok több hibát is tartalmaztak, mind a függőségi elemzésben, mind az NP-chunkingban. Mindezekkel együtt 89%-os f-score-t értünk el, ami ugyan elmarad a state-ot-the-arttól, de abból a szempontból mégis ígéretes, hogy ezt az eredményt egy egyszerű szabályrendszerrel értük el. Ez alapján a függőségi elemzés mintaillesztése további kutatásra érdemes módszer lehet a hasonló feladatokban.

Hivatkozások

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (szerk.) Text, Speech and Dialogue. pp. 123–131. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

Gyulai, L.: Az igekötők legjellemzőbb argumentumszerkezetváltoztató hatásainak korpuszalapú vizsgálata. In: Gráczi, T.E., Ludányi, Z. (szerk.) Alknyelvdok15. Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből. pp. 176–198. Nyelvtudományi Kutatóközpont (2021)

Kilpeläinen, P., és mtsai: Tree matching problems with applications to structured text databases. Ph.D.-értekezés, Department of Computer Science, University of Helsinki (11 1992)

Kornai, A.: The internal structure of noun phrases. In: István, K. (szerk.) Approaches to Hungarian, pp. 79–92. JATE (1985), `https://eprints.sztaki.hu/7848/`

Lengyel, K.: Magyar grammatika. Nemzeti Tankönyvkiadó, Budapest (2000)

Luotolahti, J., Kanerva, J., Pyysalo, S., Ginter, F.: SETS: Scalable and efficient tree search in dependency graphs. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. pp. 51–55. Association for Computational Linguistics, Denver, Colorado (Jun 2015), `https://aclanthology.org/N15-3011`

Metheniti, E., Park, P., Kolesova, K., Neumann, G.: Identifying grammar rules for language education with dependency parsing in German. In: Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019). pp. 100–111. Association for Computational Linguistics, Paris, France (Aug 2019), `https://aclanthology.org/W19-7712`

Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). p. TBA. Szeged (2021)

Nichols, E., Reisert, P.: Dgrep: A pattern-matching tool for dependency trees. In: Proceedings of the Twentieth Annual Meeting of the Association for Natural Language Processing.(to appear) (2014)

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020), https://nlp.stanford.edu/pubs/qi2020stanza.pdf

Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In: Third Workshop on Very Large Corpora (1995), https://aclanthology.org/W95-0107

Recski, G.: Hungarian noun phrase extraction using rule-based and hybrid methods. Acta Cybernetica 21(3), 461–479 (2014)

Recski, G., Varga, D., Zséder, A., Kornai, A.: Főnévi csoportok azonosítása magyar-angol párhuzamos korpuszban [Identifying noun phrases in a parallel corpus of English and Hungarian]. VI. Magyar Számítógépes Nyelvészeti Konferencia [6th Hungarian Conference on Computational Linguistics] (2009)

Szemes, B.: Kidolgozott viszonyok. a tagmondatkapcsolatok automatikus azonosításának hasznosíthatósága a stilisztikában és az irodalomtörténet-írásban. Digitális Bölcsészet 2021(4), 31–77 (2021)

Wallis, S.: Searching treebanks and other structured corpora. In: Corpus linguistics: An international handbook, pp. 738–759. Handbücher zur Sprache und Kommunikationswissenschaft, De Gruyter Mouton, Berlin (2008)

Szerzői index, névmutató

Abed, Mohammed Hamzah, 159
Ács, Judit, 91, 231
Arthur, Frigyes Viktor, 357

Babarczy, Anna, 45
Bagi, Anita, 17
Bajári, Lúcia, 77
Barta, Botond, 91, 231
Bartalis, Mátyás, 145
Berend, Gábor, 263, 417
Berkecz, Péter, 305, 319
Boncz, Ádám, 357
Bóna, Judit, 33

Csapó, Tamás Gábor, 145, 357
Csáki, Csaba, 405
Csányi, Gergely Márk, 77

Dam, Bernadett, 17
Dobsinszki, Gergely, 135
Dodé, Réka, 247, 433
Dömsödi, L Bíborka, 275
Dömötör, Andrea, 341, 457

Egas-López, José Vicente, 33
Egri, Erika, 77

Farkas, Richárd, 305, 319
Felletár, Fanni, 45
Ferenczi, Gergő, 247
Ficsor, Tamás, 417
Fülöp, Anna, 77

Gadó, Krisztián, 77
Gosztolya, Gábor, 33
Guba, Csenge, 17, 187

Héja, Enikő, 63, 217, 247, 433

Hoffmann, Ildikó, 33

Jelencsik-Mátyus, Kinga, 247, 433

K. Molnár, Emese, 341
Kádár, Máté Soma, 135
Kalcsó, Gyula, 447
Kicsi, András, 3
Kmetty, Zoltán, 275
Knap, Árpád, 275
Kőrös, Ádám, 247

Lakatos, Dorina, 91, 231
Laki, László János, 107, 217, 247, 331, 369
Ligeti-Nagy, Noémi, 63, 217, 247, 331, 433
Lipp, Veronika, 291

Mády, Katalin, 135
Megyeri, Andrea, 77
Mihajlik, Péter, 135
Mihály, Eszter, 447
Mogyorósi, Pálma, 275

Nagy, Attila, 91, 231
Nagy, Dániel, 77
Nagy, Péter, 357
Nanys, Patrick, 91
Nemeskey, Dávid, 457
Németh, Géza, 145
Németh, Norbert, 145
Novák, Attila, 203
Novák, Borbála, 203
Nyéki, Bence, 119
Nyist, Milán Konor, 231

Orosz, György, 305, 319

Pethő, Gergely, 291
Pusztai, Ágnes, 3

Ring, Orsolya, 187

Salih, Aziz Dosti Ali Hama, 171
Sass, Bálint, 291
Simon, László, 291
Solymos, Adrienn, 17
Svindt, Veronika, 33
Szabó Ledenyi, Klaudia, 3
Szabó, Gergő, 305, 319
Szabó, Martina Katalin, 17
Szántó, Zsolt, 305, 319
Szász, Gábor Krisztián, 145
Szendi, István, 17
Szigeti, Péter, 275
Sztahó, Dávid, 159, 171
Szűcs, Kata Ágnes, 447
Szviridov, István, 145

Takács, Dávid, 217
Tóth, Andor, 275

Üveges, István, 77, 187

Vadász, János Pál, 77
Vadász, Noémi, 119, 247, 331
Vági, Renátó, 77
Vándor, Péter, 405
Váradi, Tamás, 217, 247, 433
Varga, Zsófia, 433
Vidács, László, 3
Vincze, Veronika, 17, 187
Virág, Attila, 275

Yang, Zijian Győző, 107, 217, 247, 331, 369, 381, 395

Zainkó, Csaba, 145