Egy emBERT próbáló feladat

Nemeskey Dávid Márk[1]

[1] Számítástechnikai és Automatizálási Kutatóintézet
nemeskey.david@sztaki.hu

Kivonat Az utóbbi egy-két évben a mély, kontextuális szóbeágyazások kiszorították a hagyományos, kézzel összeállított feature halmazokat a legtöbb nyelvi feladatban. Ennek ellenére a magyar nyelvfeldolgozó rendszerek (`e-magyar`, `magyarlanc`) még mindig a hagyományos, kézi feature-ökkel dolgoznak. A cikkben bemutatjuk az `emBERT` modult, amely a `transformers` könyvtár segítségével lehetővé teszi kontextuális szóbeágyazás-alapú osztályozók integrálását az `e-magyar` rendszerbe. A modult főnévi csoport- és névelemfelismerésre tanítottuk fel. A modellek mindkét feladaton javítanak az eddigi legjobb eredményeken.
Kulcsszavak: BERT, `e-magyar`, névelem, chunking

1. Bevezetés

A gépi tanulásos nyelvi elemző rendszerek az utóbbi években drasztikus átalakuláson mentek keresztül. A hagyományos paradigma szerint minden szóhoz kézzel állítanak elő jellemzőket (*feature*). Ezek tipikusan nyelvi és írásképbeli jegyek, amiket általában nyelvészek hoznak létre. E jellemzők szolgálnak utána egy egyszerűbb, tipikusan off-the-shelf osztályozó (logisztikus regresszió, CRF) bemenetéül. A legtöbb szekvencia- vagy tokenklasszifikációs feladatot (szófajcímkézés, névelemfelismerés, szentimentelemzés) ilyen rendszerekkel oldották meg.

A mélytanulás elterjedésével a kézzel kiválasztott jellemzők fokozatosan a háttérbe szorultak. Helyüket a vektoriális szemantika világából ismert szóbeágyazás (*word embedding*) (Mikolov és mtsai, 2013; Pennington és mtsai, 2014) kezdte átvenni. Egy beágyazás minden szóhoz egy sokdimenziós, folytonos vektort rendel. Ezek a vektorok egy szemantikus teret feszítenek ki, ahol a hasonló jelentésű szavak vektorai egymáshoz közel esnek.

A beágyazások azonban nem csak szemantikai tartalommal bírnak, hanem implicit kódolják a szavak szintaktikus tulajdonságait is. Ez különösen alkalmassá teszi őket a gépi tanulók bemeneti jellemzőinek szerepére. A nagyobb szövegelemző láncok közül elsőként a Stanford CoreNLP szintaktikus elemzője egészült ki beágyazásokkal (Socher és mtsai, 2013). Mára a szóvektorok a legtöbb nyelvi elemző szoftverben megtalálhatóak.

A statikus beágyazások hátránya azonban, hogy egy szót minden környezetben ugyanaz a vektor reprezentál. Ez különösen a többjelentésű (pl. *körte*, *zebra*), vagy azonos alakú (pl. *dob*, *szív*) szavak esetén jelent problémát, mivel a szóvektor szükségszerűen a jelentések egyfajta amalgámja lesz, és nem fogja tükrözni a szó szintaktikai és szemantikai szerepét az aktuálisan elemzett mondatban.

A kontextualizált beágyazásoknál, mint az ELMo (Peters és mtsai, 2018) vagy a BERT (Devlin és mtsai, 2019), a szó vektora függ annak közvetlen környezetétől is. Ebből következik, hogy egy szó minden egyes előfordulásához más–más vektor tartozik. Ezek a vektorok implicit módon kódolják a szó szerepét a mondaton belül, teljesen kiváltva ezzel a kézileg összeállított feature-vektorokat. A beágyazásokat tipikusan nyelvmodellezéssel „*tanítják elő*".

Kontextuális beágyazáson alapuló rendszerek több nyelvi feladaton is felülmúlták hagyományos társaikat. A BERT, illetve követői, az XLNet (Yang és mtsai, 2019) és a RoBERTa (Liu és mtsai, 2019) főleg olyan, magasabb szintű feladatokban produkáltak erős eredményeket, mint a kérdésmegválaszolás, vagy GLUE (Wang és mtsai, 2018) teszt nyelvi megértést vizsgáló feladatai. Az ELMo és a Flair (Akbik és mtsai, 2018, 2019b) pedig névelemfelismerésben utasította maga mögé a korábbi rendszereket.

Ezek az eredmények a szövegelemző programokban is visszaköszönnek. A Flair rendszer[1] egy teljes nyelvi elemzőlánc, amelynek alapja a beágyazások szabad variálhatósága. Jelenleg ez nyújtja a legjobb teljesítményt névelemfelismerés mellett a főnévi csoport- és szófajcímkézésben is (Akbik és mtsai, 2019a).

A fenti eredmények természetesen angol nyelvre vonatkoznak. Ebben a cikkben megvizsgáljuk, hogy a kontextuális embeddingek képesek-e magyar nyelven is hasonlóan kimagasló teljesítményt nyújtani. Tesztfeladatnak a főnévi csoport- (*chunking*) és a névelemfelismerést (*named entity recognition, NER*) választottuk, mivel ezekre létezik angol precedens. Az elkészült modelleket egy új modulként integráljuk az `e-magyar` szövegelemző rendszerbe.

2. BERT

2.1. Miért a BERT?

Az előző fejezet végén felsorolt beágyazások közül a BERT-öt választottuk vizsgálatunk tárgyául. Ennek fő oka az, hogy a legtöbb beágyazás kizárólag angolul (esetleg kínaiul) elérhető. Tanításuk sok adatot és nagy számítási kapacitást igényel, ami a cikk írásakor nem állt rendelkezésünkre. A két kivétel az ELMo és a BERT, ahol elérhetőek előtanított többnyelvű modellek.

A kettő közül a BERT egyik előnye az ELMo-val szemben, hogy ún. *finomhangolásos* módszer (Devlin és mtsai, 2019): az előtanított modell könnyen finomhangolható a célfeladatra. Az ELMo ezzel szemben egy beágyazást ad, amit jellemzően feladatspecifikus architektúra bemenetén használnak. Mivel mi különálló modulban gondolkodtunk, meglévő rendszerek átalakítása nem jött szóba. A BERT másik előnye, hogy a magasabb szintű feladatokban jobb eredményeket ért el, mint az ELMo. A főnévi csoport- és névelemfelismerésre ez pont nem áll, ezért egy lehetséges további kutatási irány lehet az `emChunk` és `emNer` „ELMosítása".

[1] `https://github.com/zalandoresearch/flair`

2.2. A BERT bemutatása

A BERT egy többszintű, kétirányú Transformer kódoló (*encoder*) (Vaswani és mtsai, 2017). A modellt két nyelvmodellezési feladaton (Cloze teszt, következő mondat megjóslása) tanítják elő. A bemenetek a feladat jellegétől függően lehetnek mondatok, vagy mondatpárok. A szótár méretének korlátozása érdekében egy mondat nem szavak, hanem szóelemek (*wordpiece*) (Schuster és Nakajima, 2012) sorozata. A szótár a modellel együtt letölthető.

Az előtanított modellt minden célfeladathoz külön finomhangolják. Egy egyrétegű, előrecsatolt osztályozó hálót adnak hozzá, majd a BERTet és az osztályozót együtt tanítják.

Az angol BERT modellek két méretben hozzáférhetők: a `Base` modell 110 millió, a `Large` 340 millió paraméteres. A többnyelvű modell csak a kisebb, `Base` konfigurációban elérhető. Ezt 104 nyelvre tanították elő, és a szótára megközelítőleg 120 ezer szóelemet tartalmaz. A modellnek van nyers (`cased`) és kisbetűsített–ékezettelenített (`uncased`) változata is. Az e cikkben leírt kísérletek az előbbit használják, mivel egyrészt az angoltól eltérően a magyarban az ékezetek jelentésmegkülönböztető szereppel bírnak, másrészt névelemek azonosításakor fontos információ, hogy nagybetűvel kezdődik-e a szó.

2.3. Mennyire tud magyarul?

Mivel az általunk használt BERT 104 nyelven lett tanítva, felmerül a kérdés, hogy mennyire modellezi jól a magyar nyelvet. Kicsit pontosabban két kérdést fogalmazhatunk meg:

1. Mennyire tükrözik a szóelemek a magyar morfémákat?
2. Helyes szemantikai tartalommal bír-e egy-egy szóelem vektora, különös tekintettel a több nyelvben is előforduló homográf szóelemekre (pl. „*leg*”, „*old*”, stb.)?

Az első kérdés megválaszolásához szóelemekre bontottuk a Szeged NER korpusz összes szavát a többnyelvű modell tokenizálója, illetve egy több milliárd szavas magyar korpuszon tanított, 30 000[2] szavas BPE (Sennrich és mtsai, 2016) szótárral. Néhány kiragadott példát mutat be a 1. táblázat.

Mint látható, a szavak három csoportra oszthatók. Az első csoportba azok tartoznak, amiket a két tokenizáló hasonlóan kezel. Vagy azért, mert mindkét szótárban szerepelnek (és ezért maguk is szóelemek), vagy azért, mert egyik sem tudja értelmes egységekre bontani: utóbbira példa a „*zambiai*”.

A második csoport esetén a magyar szótár kevesebb, morfológiailag indokolt részre bontja a szavakat, míg a BERT szerinti tokenizálásban feltűnnek szemantika nélküli n-gramok is. Ennek megfelelően a többnyelvű változat mindig több szóelemből áll.

A harmadik csoportban az olló tovább nyílik: a magyar BPE tokenizálás változatlanul szemantikus, míg a BERT szóelemei véletlenszerű n-gramok. A

[2] Ez megegyezik az angol BERT szótárának méretével.

Szó	Többnyelvű	Magyar
Nemzeti	Nemzeti	Nemzeti
Andersen	Andersen	Andersen
labdarúgó	labdarúgó	labdarúgó
zambiai	za mbia i	z amb iai
megmaradt	meg maradt	megmaradt
hétfő	hét f ő	hétfő
keddtől	ke dd től	kedd től
edényben	ed ény ben	edény ben
Hétfőn	H ét f ő n	Hétfőn
tájékoztatták	tá j ék ozta tták	tájékoztat ták
leggazdagabb	leg ga zda gab b	leggazdagabb
elpártolt	el pá rto lt	el párt olt

1. táblázat. Néhány szó szóelemekre bontva a többnyelvű BERT szótára és egy magyar korpuszon épített BPE szótár alapján

hosszabb szavak lefedéséhez a többnyelvű tokenizálónak akár 4-5 szóelemere is szüksége van (a magyar BPE-nek elég 1-2). A „*hétfő*” és a „*Hétfő*” eltérő felbontása pedig arra utal, hogy a mondatkező szavak és névelemek szóelemekké tokenizálása különösen problémás lehet.

A fenti megfigyeléseket a 2. táblázat is megerősíti. A többnyelvű BERT átlagosan 50%-kal több szóelemet állít elő, mint a magyar BPE. A jelenség azonos mértékben érvényes csak a szótípusokat vagy a teljes korpuszt tekintve is. Mivel a leggyakoribb funkciószavak („*a*”, „*az*”, „*és*”) és írásjelek részei mindkét szótárnak, ez arra utal, hogy a gyakori szavak is konzisztensen rosszabb reprezentációt kapnak a többnyelvű BERTben.

Érdekes módon a nagybetűs szavak felbontásában nincs jelentős (kvantitatív) különbség a két szótár között: mindkét szótár átlagosan 4–5 szóelemre osztja őket. Ez a kisbetűs szavakhoz képesti relatív ritkaságukkal magyarázható, ugyanakkor előrevetíti, hogy a BERT (többnyelvű vagy sem) nem feltétlenül optimális névelemfelismerésre.

A második kérdés részletes megtárgyalása meghaladja e cikk kereteit. Implicit választ a két nyelvi feladaton elért eredmények adnak az 5. fejezetben.

3. Az `emBERT` modul

Fontos szempont volt, hogy az elkészült modelleket a kutatók, illetve nyelvfeldolgozás iránt érdeklődők számára egyszerűen hozzáférhetővé tegyük. E célból döntöttünk a modellek `e-magyar` rendszerbe (Váradi és mtsai, 2017) integrálása mellett. Az `e-magyar` új verziója, az `emtsv`[3] (Indig és mtsai, 2019) jelentősen

[3] `https://github.com/dlt-rilmta/emtsv`

Szóalak	Többnyelvű BERT	Magyar BPE	Különbség
kisbetű	2.24	1.34	67%
nagybetű	1.86	1.75	6%
együtt	2.14	1.44	49%
kisbetű (típus)	3.97	2.41	65%
nagybetű (típus)	4.65	4.27	9%
együtt	4.12	2.83	45%

2. táblázat. Átlagos szóelemszám szavanként / típusonként

megkönnyítette új modulok hozzáadását az elemzőlánchoz. Így született meg az `emBERT` modul.

Az `emBERT` követi az `emtsv` modulok konvencióit. Egyfelől működik önálló Python modulként, másfelől (opcionális) része az `e-magyar` elemzőláncnak. Telepítése után elérhetővé válnak a `bert-base-chunk`, `bert-max-chunk`, és `bert-ner` eszközök. Ezek tokenizált szöveget várnak bemenetükön, ezért az `emToken` futtatása előfeltétele a működésüknek. A többi, magasabb szintű `e-magyar` modultól (mint pl. az `emChunk` és az `emNer`) eltérően azonban az `emBERT` morfológiai információt nem igényel, ezért a morfológiai elemző és a lemmatizáló futtatása nem szükséges.

Mivel a BERT modellek (még `Base` konfigurációban is) nagyok, a modul nem tartalmazza őket. Ehelyett mind a három eszköz első meghívásakor letölti a saját modelljét az `emBERT_models` GitHub repozitóriumból[4].

A BERT finomhangolásához és futtatásához a HuggingFace `transformers`[5] (Wolf és mtsai, 2019) programkönyvtárat használtuk. A csomag előnye, hogy a BERT mellett tartalmazza más Transformer-alapú beágyazások (XLNet, RoBERTa) implementációit is. Ez lehetővé teszi később más beágyazások kipróbálását és integrálását a modulba.

A többi `e-magyar` modullal szemben az `emBERT` tartalmazza mind a tanító, mind a modelleket futtató kódot. Két okból választottuk ezt a megoldást: egyrészt a kód bonyolultsága nem indokolta a két funkció kettéválasztását; másrészt így a felhasználók egy kész csomagot kapnak, amivel kedvükre kísérletezhetnek. A kód a többi `e-magyar` modulhoz hasonlóan GitHubon[6] érhető el.

4. Kísérletek

A modellek képességeit két feladaton: főnévi csoport- és névelemfelismerésen mértük. A modelleket korábbi eredményekkel való összehasonlíthatóság érdeké-

[4] `https://github.com/DavidNemeskey/emBERT_models`
[5] `https://github.com/huggingface/transformers`
[6] `https://github.com/DavidNemeskey/emBERT`

ben a vonatkozó szakirodalomban használt korpuszokon tanítottuk és értékeltük ki.

A magyar statisztikai NP-felismerők (A `hunchunk` (Recski, 2010) és utódai) mindegyikét a Szeged Treebank 2.0 (Csendes és mtsai, 2005) korpuszon tanították. Mi is hasonlóképpen jártunk el: a 82 099 mondatos korpuszt korpuszt véletlenszerűen, 80%-10%-10% arányban osztottuk fel tanító-, validációs és teszthalmazokra. Mind a két alfeladatot (minimális és maximális főnévi csoportok) ugyanúgy futtatuk: az alap BERT modellt 4 epochon keresztül finomhangoltuk, majd kiértékeltük a teszthalmazon. A validációs halmaz alapján *early stopping*ra nem volt szükség.

A névelemfelismerőt a Szeged NER korpuszon (Szarvas és mtsai, 2006), a Szeged Treebank részhalmazán finomhangoltuk. Mivel a NER korpusz jóval kisebb, mint a teljes Treebank (a három vágás 8172–502–900 mondatos), ezért a modellt több, különböző konfigurációval is feltanítottuk. A legjobb modell 30 epochon keresztül tanult 10^{-5}-ről lineárisan csökkenő tanulási rátával.

A kísérletekhez a korábban említett `transformers` könyvtár PyTorch (Paszke és mtsai, 2017) verzióját használtuk. A tanítást párhuzamosan futtattuk 3 db GeForce RTX 2080 Ti kártyán, 16-os batch size-zal. Ezzel a konfigurációval mind a legjobb NER modell, mind a (jóval kevesebb epochig tanított) chunking modellek 3 óra alatt tanulnak fel. A chunkinghoz a hiperparaméterek többségét az alapértelmezett értéken hagytuk. A NER esetében több hiperparaméter-beállítást is kipróbáltunk, de végül (az epochszám és a tanulási ráta kivételével) itt is az alapértelmezett értékek bizonyultak a legjobbnak.

A tanítás pontos paraméterei a letöltött modellekhez tartozó konfigurációs file-okban megtekinthetők.

5. Eredmények

5.1. Főnévi csoportok

Az `emBERT` és a `hunchunk` család eredményeit a 3. táblázat foglalja össze. Mint látható, az `emBERT` mindkét korábbi rendszernél jobban teljesít, és mind a minimális, mind a maximális NP-k azonosításában state-of-the-art eredményt ér el.

A különbség minimális NP-k esetében nem jelentős; a maximális csoportokon elért F1 érték viszont szignifikánsan, másfél százalékkal jobb, mint az `e-magyar`-ban jelenleg (`emChunk` néven) működő HunTag3.

5.2. Névelemek

Névelemfelismerésben a kép vegyesebb (4. táblázat). Az `emBERT` jelentősen, 2%-al magasabb F1-et ér el, mint Szarvas és mtsai (2006) és Varga és Simon (2007), de a HunTag3 eredményétől elmarad. A spaCy az összehasonlítás szempontjából nem releváns, mivel a tanítóadata ki lett bővítve a hunNERwiki korpusszal (Nemeskey és Simon, 2012); kizárólag a teljesség kedvéért szerepel a táblázatban.

Rendszer	Minimális	Maximális
`hunchunk`/HunTag (Recski, 2010)	95,48%	89,11%
HunTag3 (Endrédy és Indig, 2015)	–	93,59%
`emBERT`	**95,58%**	**95,05%**

3. táblázat. A magyar főnévi csoport-felismerők összehasonlítása

Rendszer	F1
(Szarvas és mtsai, 2006)	94,77%
`hunner` (Varga és Simon, 2007)	95.06%
HunTag3 (Endrédy és Indig, 2015)	**97.87%**
`emBERT`	97,08%
spaCy[7]	*93,95%*

4. táblázat. A magyar névelemfelismerők összehasonlítása

A NER tanítása alatt belefutottunk abba a problémába, ami minden gépi, de különösen mélytanuló rendszer rákfenéje: az eredmények erősen függenek a tanítás hiperparamétereitől, a megfelelő hiperparaméterek megtalálása azonban extrém módon erőforrásigényes. A Szeged NER-hez hasonló, apró korpuszok esetén ez a hatás hatványozottan jelentkezik, mivel a modell nagyságrendekkel több paraméterrel rendelkezik, mint ahány tanítópélda rendelkezésre áll. A megoldás egy, a jelenleginél nagyobb NER korpusz (például a hunNERwiki egy ellenőrzött minőségű részhalmaza) lehetne.

6. További kutatás

Az `emBERT`, bár javít a korábbi legjobb eredményen NP-felismerésben, több szempontból is proof-of-conceptnek tekinthető. Az alábbiakban sorra vesszük ezen szempontokat, és a kapcsolódó lehetséges kutatási irányokat.

Egyrészt láttuk, hogy a többnyelvű BERT használata mindenképpen szuboptimális: mind a szóelemek, mind a teljes modell kénytelen a (viszonylag szűkös, hiszen csak `Base` változat) kapacitását 104 nyelv között megosztani. Egy magyar korpuszon feltanított BERT, különösen a `Large` modell, minden bizonnyal további javulást érne el. A jövőben tervezzük ilyen modellek tanítását és nyilvánosságra hozását.

Másrészt a BERT csak a jéghegy csúcsa; számos egyéb kontextuális szóbeágyazás létezik, mint az ELMo, a RoBERTa, vagy a Flair. Ahogy láttuk, ezek bizonyos feladatokban – pl. névelemfelismerésben is – felülmúlják a BERT-öt. Reményeink szerint ezen beágyazások magyar változata is elkészülhet, mely esetben természetesen integráljuk őket az `emBERT`-be.

Harmadrészt, a névszói csoport- és névelemfelismerés mellett érdemes lenne megvizsgálni más nyelvfeldolgozási lépések BERT-ösíthetőségét. A nyilvánvaló jelölt a morfológiai elemzés, amire már létezik mélytanulásos megoldás (Ugray, 2019). Emellett – a GLUE-hoz (Wang és mtsai, 2018) vagy SQuAD-hoz (Rajpurkar és mtsai, 2016) hasonló magyar nyelvi erőforrások megléte esetén – olyan, magasabb szintű feladatokra is adaptálni lehetne a modult, mint a szentimentelemzés, parafrázisok felismerése, vagy kérdésmegválaszolás. Ezzel pedig az `emBERT` a meglévő funkciók javításán felül új képességekkel is fel tudná ruházni az `e-magyar`t.

7. Összegzés

A cikkben bemutattuk az `e-magyar` szövegelemző rendszer egy új modulját. Az `emBERT` lehetővé teszi kontextuális szóbeágyazás-alapú osztályozók integrálását az `e-magyar`ba. A többnyelvű BERT modellt névszói csoport- és névelemfelismerésre tanítottuk fel. A modellek összemérhetőek az eddigi legjobb eredményekkel, vagy javítanak is rajtuk.

Az `emBERT` számos továbbfejlesztési lehetőséggel rendelkezik. A modul könnyen kiterjeszthető más mély beágyazások, illetve nyelvi feladatok támogatására, amennyiben a vonatkozó erőforrások (maga a beágyazás, tanítókorpusz) elérhetővé válnak.

Köszönönetnyilvánítás

A kutatást részben a 2018-1.2.1-NKP-2018-00008 *A mesterséges intelligencia matematikai alapjai* és az NKFIH 120145-ös *Szószerkezet felismerése mélytanulással* projektek támogatták. A finomhangolási kísérletek egy részét az NVIDIA által adományozott grafikus kártyákon futtattuk.

Hivatkozások

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: FLAIR: An easy-to-use framework for state-of-the-art NLP. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 54–59. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019a), `https://www.aclweb.org/anthology/N19-4010`

Akbik, A., Bergmann, T., Vollgraf, R.: Pooled contextualized embeddings for named entity recognition. 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. 724–728. Association for Computational Linguistics, Minneapolis, Minnesota (06 2019b), `https://www.aclweb.org/anthology/N19-1078`

Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1638–1649. Association for Computational Linguistics, Santa Fe, New Mexico, USA (08 2018), https://www.aclweb.org/anthology/C18-1139

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Lecture Notes in Computer Science: Text, Speech and Dialogue. pp. 123–131. Springer (2005)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of NAACL (2019)

Endrédy, I., Indig, B.: HunTag3, a General-purpose, Modular Sequential Tagger – Chunking Phrases in English and Maximal NPs and NER for Hungarian, p. 213–218. Uniwersytet im. Adama Mickiewicza w Poznaniu, Poznan (2015)

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: emtsv – Egy formátum mind felett [emtsv – One format to rule them all]. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport (2019)

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (szerk.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013), https://bit.ly/39HikH8

Nemeskey, D.M., Simon, E.: Automatically generated ne tagged corpora for english and hungarian. In: Proceedings of the 4th Named Entity Workshop. pp. 38–46. Association for Computational Linguistics (2012)

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)

Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics (2014), http://www.aclweb.org/anthology/D14-1162

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics (2018), http://aclweb.org/anthology/N18-1202

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 (11 2016), https://www.aclweb.org/anthology/D16-1264

Recski, G.: Főnévi csoportok azonosítása szabályalapú és hibrid módszerekkel. In: Tanács, A., Vincze, V. (szerk.) VII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 333–341 (2010)

Schuster, M., Nakajima, K.: Japanese and korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5149–5152. IEEE (2012)

Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany (8 2016), https://www.aclweb.org/anthology/P16-1162

Socher, R., Bauer, J., Manning, C.D., Andrew Y., N.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013). pp. 455–465. Association for Computational Linguistics, Sofia, Bulgaria (2013)

Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In: Discovery Science, 9th International Conference, DS 2006, Barcelona, Spain, October 8-10, 2006, Proceedings. pp. 268–278 (2006)

Ugray, G.: Pos-tagging and lemmatization with a deep recurrent neural network. In: XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2019). pp. 215–224. Szeged (2019)

Varga, D., Simon, E.: Hungarian named entity recognition with a maximum entropy approach. Acta Cybern. 18(2), 293–301 (Feb 2007)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: e-magyar: digitális nyelvfeldolgozó rendszer. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017). Szeged (2017)

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multitask benchmark and analysis platform for natural language understanding (2018)

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing (2019)

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding (2019)