XV. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2019. január 24–25.

Témaspecifikus gépi fordítórendszer minőségének javítása domain adaptáció segítségével

Laki László János[1,2,3]

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

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

1. Bevezetés

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

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

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

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

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

2. Kapcsolódó irodalom

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

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

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

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

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

3. Adatszelekciós módszerek

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

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

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

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

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

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

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

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

4. Kísérleti környezet leírása

4.1. Tanítóanyag összetétele

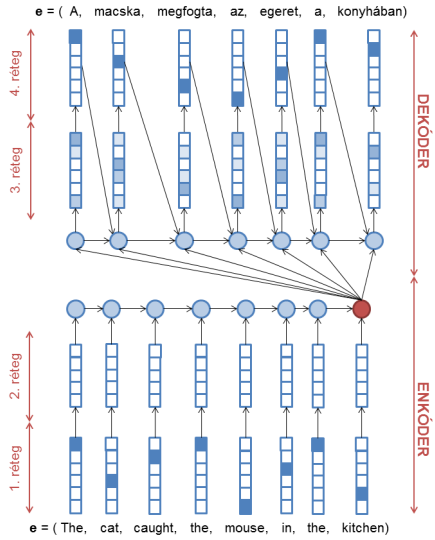
Mivel kutatásomban egy kereskedelmi fordítási környezet minőségének javítását tűztem ki célul, így a rendszerek betanításához fordítócégek témaspecifikus fordítómemóriáit használtam. A méréseket 3 különböző nyelvpáron végeztem el. Az angol-német és az angol-francia nyelpárok mellett a japán-angol nyelvpárral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

4.2. Gépi fordítórendszer bemutatása

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

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

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



1. ábra: Enkóder-dekóder architektúra vázlatos rajza

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

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

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

5. Eredmények és kiértékelés

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6. Összegzés

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

Köszönetnyilvánítás

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

Hivatkozások

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

3. Chatterjee, R., Negri, M., Turchi, M., Federico, M., Specia, L., Blain, F.: Guiding Neural Machine Translation Decoding with External Knowledge. In: Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers, Copenhagen, Denmark, Association for Computational Linguistics (2017) 157–168
4. Silva, C.C., Liu, C.H., Poncelas, A., Way, A.: Extracting In-domain Training Corpora for Neural Machine Translation Using Data Selection Methods. In: Proceedings of the Third Conference on Machine Translation, Belgium, Brussels, Association for Computational Linguistics (2018) 224–231
5. Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast Neural Machine Translation in C++. In: Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, Association for Computational Linguistics (2018) 116–121
6. Salton, G., Yang, C.S.: On the specification of term values in automatic indexing. Journal of Documentation **29**(4) (1973) 351–372
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. CoRR **abs/1607.01759** (2016)
9. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and Word2vec for text classification with semantic features. In: 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC). (2015) 136–140
10. Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P.: Scalable Modified Kneser-Ney Language Model Estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria (2013) 690–696
11. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010 **2** (2010) 1045–1048
12. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. ArXiv e-prints (2017)
13. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014)
14. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR **abs/1406.1078** (2014)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR **abs/1409.3215** (2014)
16. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. CoRR **abs/1508.07909** (2015)
17. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 311–318
18. Tantug, A.C., Oflazer, K., El-Kahlout, I.D.: BLEU+: a tool for fine-grained BLEU computation. In: LREC 2008. (2008)
19. Callison-Burch, C., Osborne, M., Koehn, P.: Re-evaluating the role of BLEU in machine translation research. In: In EACL. (2006) 249–256
20. Banerjee, S., Lavie, A.: METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop

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