Alkalmazott Nyelvészeti Közlemények, Miskolc, XII. évfolyam, 2. szám (2017), pp. 22–31.

SZÓBEÁGYAZÁS ÉS DISKURZUS

WORD EMBEDDING AND DISCOURSE

ERDEI TAMÁS[1]

Absztrakt: A szavak természetesnyelv-feldolgozási eljárások során a számítógép által is nagy hatékonysággal használható reprezentációjának létrehozása régi célkitűzés. A szóbeágyazás (Word Embedding) módszere 2013-as megjelenése óta az egyik legnépszerűbb eszközzé vált, és számos célra használták a szemantikai relációk kezelésétől a gépi fordításig. A szóbeágyazás alkalmas lehet olyan eljárások továbbfejlesztésére is, mint a tudásreprezentáció, témaazonosítás, témaszerkezet kinyerése, a szövegösszefüggés vizsgálata, diskurzusszegmentáció, illetve az explicit és implicit diskurzusjelölők feltárása, amely területek a diskurzuselemzésnek is vizsgálati területei.

Kulcsszavak: szóbeágyazás, szemantikai reprezentáció, természetesnyelv-feldolgozás, diskurzus, diskurzuselemzés

Abstract: Creating word representations which can be effectively used by computers during natural language processing tasks has been a goal for a long time. Since its appearance in 2013, Word Embedding has gained great popularity, and has been used for a wide variety of tasks ranging from handling semantic relations to machine translation. Word Embedding may also prove to be useful for such processes as knowledge representation, theme extraction, theme structure extraction, text coherence detection, discourse segmentation and detecting explicit and implicit discourse markers, which areas are researched by discourse analysis.

Keywords: Word Embedding, semantic representation, natural language processing, discourse, discourse analysis

BEVEZETÉS

A SZÓBEÁGYAZÁS ALAPELVEI

A szavak számítógépek által is használható szemantikai reprezentációjának létrehozása régi célkitűzés a természetesnyelv-feldolgozásban. A témában megszületett első jelentős cikk (KATZ–FODOR 1963) óta számos próbálkozás történt ezen a téren. Igazán hatékony megoldásra azonban a közelmúltig kellett várni. A szóbeágyazás (Word Embedding) megjelenése óta nagy népszerűségnek örvend. 2015-ben az

[1] ERDEI Tamás
tanársegéd
Szegedi Tudományegyetem
Juhász Gyula Pedagógusképző Kar
Magyar és Alkalmazott Nyelvészeti Tanszék
erdeit@jgypk.szte.hu

Empirical Methods in Natural Language Processing (Empirikus módszerek a természetesnyelv-feldolgozásban) című konferencián olyan sokan foglalkoztak a témával, hogy többen is úgy gondolták, az Embedding Methods in Natural Language Processing (Szóbeágyazásos módszerek a természetesnyelv-feldolgozásban) találóbb konferenciacím lett volna. A következő évben ugyanezen a konferencián két workshopot is szenteltek a témának (RUDER 2016).

A módszer alapja a disztribúciós hipotézis. Ezt az elvet legtöbben John Rupert Firth (1957) leírásában ismerik: ahogyan madarat tolláról és embert barátjáról, úgy Firth szerint egy szót a környezetéről lehet megismerni, vagyis arról, hogy milyen „társaságban" mutatkozik. Ha két szó hasonló környezetekben fordul elő, úgy a jelentésük is hasonló, és minél inkább hasonlít a két szó szokásos környezete, annál hasonlóbb lesz a jelentésük is. Firth előtt már Harris (1954) is megfogalmazott egy ilyen állítást: a hasonló kontextusban előforduló szavaknak a jelentése is közel áll egymáshoz. Ebből a hipotézisből ered a szóbeágyazásos módszer neve: azt használja föl, hogy az adott szó milyen más szavak közé van beágyazva.

Elsőre meglepőnek tűnhet, hogy habár a disztribúciós hipotézis már az ötvenes években megszületett, a gyakorlatban való tesztelésére a 21. századig kellett várni (BENGIO et al. 2003). Ha végiggondoljuk, hogy milyen eljárást igényel ez a módszer, érthetővé válik a mintegy ötven év. Szükség van egy nagyméretű korpuszra, amelyben minden egyes előforduló szóalak esetében statisztikát készítünk az összes előfordulása körül található szavak alapján, így kapva összesített adatokat annak jellemző környezetéről. Ez olyan teljesítményű számítógépet igényel, amilyenek csak az ezredforduló körül jelentek meg.

A szóbeágyazás másik alapvető eszköze a korpusz szavainak (a továbbiakban szóalakot értek alatta) vektoros reprezentációja (JURAFSKY–MARTIN 2017). Kezdetben ezek ún. one-hot vektorok voltak: minden szó saját vektorában annyi szám található, ahány szó a korpuszban található. Ezekből mindegyik nulla, kivéve egyet, ami annyiadik helyen található a vektorban, ahányadik leggyakoribb szóról van szó a korpuszban. Az ilyen vektorok alkalmasak a szó azonosítására, de diszkrét reprezentációk, azaz a szavakat egymástól függetlenül reprezentálják; a szavak egymással való viszonyáról, hasonlóságáról, különbözőségéről stb. semmit sem árulnak el. Ezen felül rendkívül nagyok: ha N db szóalakunk van egy korpuszban, akkor a vektorok N dimenziósak, azaz N db számból állnak, amiből ráadásul majdnem mind nulla. Az ilyen vektorok dimenziószáma több százezer is lehet.

1. TOMAS MIKOLOV ET AL. MODELLJE, A WORD2VEC

A legismertebb, legtöbbet idézett szóbeágyazásos modell Tomas Mikolov (és munkatársai) nevéhez fűződik (MIKOLOV et al. 2013a, 2013b, 2013c, a matematikai rész magyarázatához lásd pl. GOLDBERG–LEVY 2014; RONG 2014, 2015), az ezen alapuló rendszer a word2vec (kb. „szavak vektorokká") nevet kapta. Ez volt az első olyan modell, ami rendkívül hatékonyan egyesítette a vektoros reprezentációt a disztribúciós hipotézissel, ráadásul egy, a gépi szemantikai reprezentációval foglalkozók körében meglepően egyszerű struktúrára alapozva, így széles érdeklődés övezte a rákövetkező

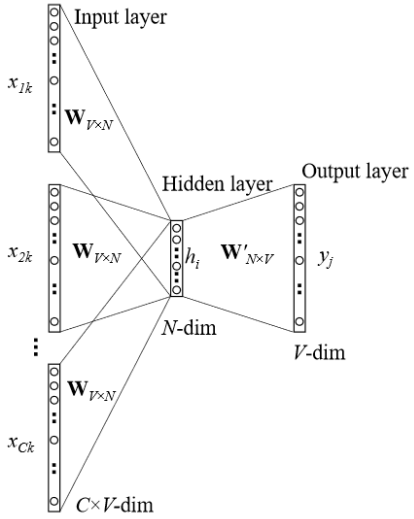
években. A fő gondolat az volt, hogy olyan vektorokat rendeljünk a szóalakokhoz, amelyeket a környezetükben előforduló szavak vektoraiból számoltunk ki.

A rendszer alapja egy neurális hálózat. A neurális hálózat a gépi tanulás jelenleg legnépszerűbb eszköze a természetesnyelv-feldolgozásban (lásd JURAFSKY–MARTIN 2017), szoftveres formában létező neuronsejtekből álló rétegekből épül fel, ahol minden réteg a következő felé küldi az „aktivációs energiát". Mikolov modellje egy bemeneti, egy közbülső és egy kimeneti rétegből áll. A bemeneti és a kimeneti rétegek one-hot vektor-alapúak, vagyis ha egy N szóalakos korpuszon tanítjuk a rendszert, akkor N neuronból állnak. A közbülső réteg mérete attól függ, hogy hány dimenziós vektorokat szeretnénk kapni a végén, Mikolov és munkatársai többnyire pár száz dimenzióval dolgoztak (MIKOLOV et al. 2013B).

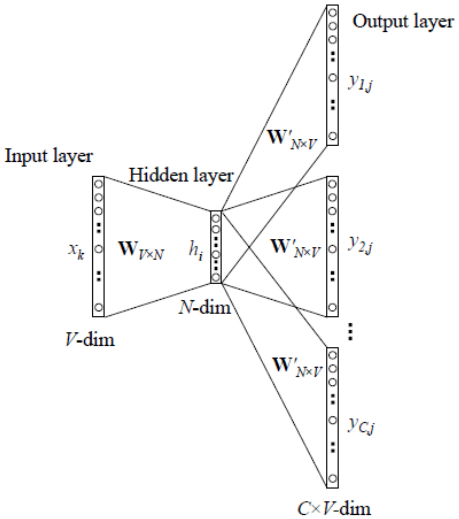
A bemeneti és a középső (rejtett), illetve a középső és a kimeneti rétegek között egy-egy mátrix közvetít a mátrixokkal való szorzás műveletének segítségével. A végső szóvektorokat az első, vagyis a bemeneti és a középső réteg közötti mátrixból fogjuk kapni. Ha a középső réteg K dimenziós, akkor ez egy N × K méretű mátrix, N sorral (ahány szóalakunk van, hiszen ennyi dimenziós a bemeneti réteg), és K oszloppal, hiszen ennyi dimenziós a középső réteg. A tanulási folyamat végére ezek a sorok lesznek a vektorok, a K db szám a vektorban pedig a vektor dimenzióinak értékeit jelenti: minden szó one-hot vektora (amiben egyetlen egyes van a nullákon kívül) megmutatja, hogy melyik sorban lévő vektor az övé. Más szóval, az adott szóhoz fog tartozni egy N (értéke több százezer) dimenziós one-hot vektor és egy K (értéke pár száz) dimenziós vektor, amelynek az értékei viszont nem nullák.

A tanulási folyamat a következő. Minden szópéldány esetében megnézzük a környezetét. Mikolov újítása, hogy az addigi szokástól eltérően nemcsak az adott szópéldány előtti szavakat tekintette a környezetének, hanem az utána levőket is. 4 sugarú környezettel dolgozott (az adott szópéldány előtti és utáni 4-4 szó). A környezetekre alapozva két módszert dolgoztak ki. A continuous bag-of-words (CBOW) modell veszi az adott szópéldány környezeteiben előforduló szavak vektorait, ebből kiszámol egy eredőt. Feltételezzük, hogy ez megegyezik annak a szónak a vektorával, ami valójában ott áll ezek között középen. A rendszer azt tapasztalja, hogy nem egyezik meg vele, ezért a backpropagation nevű hibajavító eljárást alkalmazza: egy függvény a kimeneti rétegből visszafelé kiindulva javít a mátrixokban található értékeken, hogy közelebb legyen az eredményül kapott vektor a valóban az adott kontextusban lévő szóalak vektorához. Minden egyes szóalak minden környezete alapján javítja az értékeket, akár több százszor újra földolgozva a teljes korpuszt. Egy idő után már nem történik érdemi változás az újraszámolások közben, a rendszer közel került egyfajta egyensúlyi állapothoz. A hasonló környezetben előforduló szavaknak hasonló lett a vektora, a különbözőeknek különböző. A skip-gram modell nagyon hasonlóan működik a CBOW-modellhez egy lényeges különbséggel: nem a kontextusból próbálja megjósolni a rendszer a középen álló szót, hanem fordítva, a középen álló szó alapján ad becslést annak környezetére, és ennek alapján javít *(1.* és *2. ábra)*. Természetesen a korpusz minden szavának minden környezetében található szavak vektorait több százszor újraszámolni túl sok gépidőt venne igénybe, ezért több olyan eljárást is bevezettek, amelyek a folyamatot gyorsítják: ennek köszönhetően egy

több milliárd szópéldányos korpusz feldolgozási ideje a kezdeti egy napról kb. egy órára csökkent, ennyi idő alatt ad ki a rendszer használható és pontos szóvektorokat.



1. ábra. A CBOW-modell, a felhasznált korpusz (egyben a bemeneti és kimeneti rétegek vektorai) mérete V-vel, a középső réteg (egyben a végén kapott szóvektorok) mérete N-nel jelölve, a kontextusba C db szót véve (RONG 2014: 6)



2. ábra. A skip-gram modell, a felhasznált korpusz (egyben a bemeneti és kimeneti rétegek vektorai) mérete V-vel, a középső réteg (egyben a végén kapott szóvektorok) mérete N-nel jelölve, a kontextusba C db szót véve (RONG 2014: 8)

2. FELHASZNÁLÁS

A CBOW- és a skip-gram modell felhasználásával kapott vektorok legnagyobb előnye, hogy (mivel a disztribúciós elven alapulnak) sokat elárulnak a szavak egymással való kapcsolatáról, vagyis összehasonlíthatóak. A korábban használt rendszerekben is meg lehetett vizsgálni, hogy egy adott szóhoz melyik a leginkább hasonló másik szó. Ezt úgy lehet megtenni, hogy megvizsgáljuk, melyik vektor zárja be vele a legkisebb szöget, azaz melyikkel a legkisebb a cosinus-távolsága. Mikolov és munkatársai (2013a, 2013b, 2013c) újítása az analógiatesztek bevezetése: *Rome* (Róma) olyan *Italy*-nak (Olaszország), mint *France*-nak (Franciaország) micsoda? Ezt így adhatjuk meg formálisan: *Italy* : *Rome* = *France* : _____. A megoldás a szóvektorokkal meglepően egyszerű: vonjuk ki a *Rome* vektorából az *Italy* vektorát és adjuk hozzá a *France* vektorát (ezt a számítógépen egyszerű vektorműveletekkel megtehetjük). Az eredményül kapott vektor jó eséllyel nem lesz benne a rendszerben, de ha megkeressük a hozzá (cosinus-távolságban) legközelebb álló vektort, akkor ez bizony a *Paris* vektora lesz. Formálisan: V*(Rome)* – V*(Italy)* + V*(France)* ≈ V*(Paris)*, ahol ≈ a cosinus-távolság alapján legközelebbi vektort jelöli. Mikolov híres példája ugyanez a *férfi : nő = király :* _____ „egyenlettel”: V*(king)* – V*(man)* + V*(woman)* ≈ V*(queen)*. Szemantikai téren nemcsak analógiás teszteken teljesített jól a szóbeágyazás: kakukktojás-teszteken (melyik vektor van a legtávolabb a többitől), mondatkiegészítési teszten (MIKOLOV et al. 2013b), és kompozícionális műveletek végrehajtásában [pl. V*(Germany)* + V*(capital)* ≈ V*(Berlin)*] is jól szerepelt, de össze tudott kötni pl. városokat sportcsapatokkal, újságokkal, folyókkal is (MIKOLOV et al. 2013c).

Fontos felismerés volt, hogy az analógiateszt nemcsak szemantikai műveletekre alkalmas, hanem morfológiai műveletekre is (MIKOLOV 2013b), pl. V*(great)* : V*(greater)* ≈ V*(tough)* : V*(tougher)*. Még a rendhagyó alakok, vagy a szótári forma hiánya sem okoztak nehézséget a rendszernek: V*(walking)* : V*(walked)* ≈ V*(swimming)* : V*(swam)*. Tehát szóbeágyazással a nyelvtani műveletek is kezelhetőek.

A szóbeágyazás felhasználási lehetőségei ennél is szerteágazóbbak, eddig a következő területek fejlesztéséhez járult hozzá (RONG 2014, 2015; RUDER 2016): hasonlóság, szavak csoportosítása, szöveg- és dokumentumcsoportosítás és -osztályozás, címkézés (PoS tagging), függőségi elemzés, szentimentelemzés, tulajdonnév-fölismerés, parafrázis-felismerés, gépi fordítás, információkinyerés, válaszadó rendszerek, tények helyességének ellenőrzése, illetve egyéb (nem szóalapú) hozzárendelések (felhasználó–tweet, vevő–termék stb.). Ahogy Mikolov et al. (2013b) – némileg szerénytelenül – megjegyzi: lehet, hogy olyan eljárásokhoz is fel lehet majd használni a módszert, amiket még csak ezután találnak majd föl.

3. ELŐNYÖK ÉS HÁTRÁNYOK

A fentiekből is látható a szóbeágyazás egyik legnagyobb előnye: az, hogy széleskörűen felhasználható a természetesnyelvű szövegek legkülönbözőbb típusú számítógépes feldolgozása során. Kiemelten alkalmasnak látszik szemantikai műveletekre,

és ezeken keresztül egyfajta tudásreprezentációra is, ami Katz és Fodor (1963) óta kitűzött cél.

Az eljárás, akár a CBOW-, akár a skip-gram modellt használjuk, alacsony műveletigényű (RONG 2014; MIKOLOV 2013b). Egy óra alatt egy többmilliárd szópéldányos korpuszból lehet vektorokat kinyerni. Kiemelendő továbbá, hogy ez a korpusz a gépi tanulás előtt nem megy át előfeldolgozáson, nem kell a szavakat előre semmilyen módon megcímkézni. Érzéketlen továbbá a rendszer a korpusz nyelvére, az tetszőleges nyelvű lehet. Ugyanis az eljárás csak azt nézi, hogy egy szóalak milyen más szóalakok között fordul elő. Ezeket a szóalakokat semmilyen módon nem értelmezi, csak együttes előfordulást vizsgál. Jól kezeli a rendszer a ritka, ismeretlen, esetleg hibás szóalakokat is, ami a gépi tanuláson alapuló természetesnyelv-feldolgozó rendszerek között sem gyakori (RONG 2014, 2015; RUDER 2016; SIKLÓSI–NOVÁK 2016a)

A szóbeágyazás fentebb leírt modelljével kapcsolatban három hátrány említhető (RONG 2015). A gépi tanuló rendszer érzéketlen arra, hogy a kontextusokban a szavak milyen sorrendben vannak. Mivel a cél szemantikai reprezentációk létrehozása volt, ezért ez nem volt probléma, de bizonyos típusú feladatoknál jól jöhet a sorrend számontartása, reprezentálása a modellben. Szintén nem megoldott a többszavas kifejezések kezelése, pl. egy több szóból álló tulajdonnév esetében minden szó külön vektort kap. Végül nem megoldott a több jelentéssel bíró szavak kezelése sem. Minden szóalakhoz egyetlen vektort rendel a rendszer, ami a különböző jelentések összesítéséből jön létre. Ez bizonyos feladatoknál problémát is okozhat, magyar nyelvre például a *férfi* : *nő* ≈ *király* : *királynő* analógia nem működött a *nő* szó több jelentése miatt (SIKLÓSI–NOVÁK 2016a). Megjegyzendő, hogy mind a három fenti probléma megoldására folynak kutatások, mind a három irányban van a módszernek „kiterjesztése", vagyis a fentiekkel kiegészített szóbeágyazásos rendszer (RUDDER 2016).

4. SZÓBEÁGYAZÁS ÉS DISKURZUS

A fentiekből látható, hogy az általunk használt szavakat, szóalakokat a szóbeágyazás módszere a gép számára is használható formában reprezentálja. Mivel ezek között a szavak között azok vektorainak segítségével sokféle kapcsolat megadható, illetve kimutatható, ezért felhasználhatónak tűnik a módszer számítógéppel végzett diskurzuselemzés céljára is. Különösen, ha belegondolunk abba, hogy a rendszer valós szövegeken tanul, azaz pontosan azt és úgy reprezentálja, amit és ahogyan mi a nyelvhasználat során használunk.

Először is remélhetjük tehát a módszertől, hogy valamiféle tudásreprezentációt szolgáltat. A diskurzuselemzők a formális nyelvészekkel ellentétben az elemzés tárgyává teszik a kontextus vizsgálatát is (BROWN–YULE 1993; JOHNSTONE 2008; WIDDOWSON 2011), illetve azt, hogy az hogyan jelenik meg a diskurzusban. A diskurzuselemzésben kontextuson az adott szöveg létrejöttében szerepet játszó körülményeket értjük, pontosabban ezek belső reprezentációját a szöveg létrehozójának fejében. Tehát a kontextus a fejben van, egyfajta hiedelem- és tudásrendszerként. Mint fentebb már láttuk, éppen egy tudásrendszerhez felhasználható vektorrendszert

kapunk a szóbeágyazás alkalmazásával. Jó példa egy ilyen sémarendszer felépítésére Siklósi és Novák (2016a, 2016b) szóbeágyazásos megoldása. Egy nagyméretű korpuszon tanították modelljüket, amelyben 5 sugarú környezeteket vizsgáltak, és 300 dimenziós vektorokkal dolgoztak. Először is megnézték, hogy egy adott szóhoz mik a leginkább hasonló szavak. A rendszer nemcsak meglepően jól azonosítja ezeket, de a toldalékolásokat is figyelembe veszi. A *kenyerek* szóra például a *kiflik*, *zsemlék*, *lepények*, *pogácsák* stb., a *pirosas* szóra a *lilás*, *rózsaszínes*, *barnás* stb. szavakat kapták. A fejünkben lévő sémák szempontjából érdekes eredmény például, hogy az *egerekkel* szóra nemcsak rágcsálók *(*pl. *patkányokkal)* jöttek ki, hanem olyanok is, mint pl. *férgekkel*, *hüllőkkel*, *pókokkal*. Nem nehéz összefüggést látni: ezektől félnek az emberek. Amikor szóbeágyazással szócsoportokat próbáltak kinyerni, az *autó* szó többször is a családtagokat tartalmazó csoportba került. Az autóra mint családtagra való utalás valóban nem szokatlan informális szövegekben. Jó eredmény, hogy számos szócsoportot sikerült a rendszerből kinyerni, így például foglalkozások, nyelvek vagy anyagnevek klasztereit, amelyeken belül újabb csoportokat képeztek annak megfelelően, hogy pl. mely foglalkozások hasonlítanak egymásra jobban. Úgy tűnik, a számítógép a szóbeágyazás módszerén keresztül egyfajta tükröt tart elénk: ti, emberek ezt és ezt így és így használjátok. Különösen figyelemreméltó ez, ha belegondolunk, hogy a számítógép nem tudja, hogy ezek a szavak micsodák, nem tudja, mi az a kifli vagy zsömle, csak azt tudja, hogy hasonló környezetben használják őket.

A diskurzusok témájának, témaszerkezetének kinyerését (lásd JURAFSKY–MARTIN 2009) megkönnyíti, ha megfelelő szemantikai reprezentációkkal dolgozunk. A kulcsszavak kinyerése során lehetőség van a kulcsszavak szóvektorainak klaszterezésére, így kirajzolódhat előttünk egy vagy több adott téma. Ha megnézzük, hogy a kulcsszavak vektoraihoz milyen más szavak vektora áll közel (esetleg figyelve, hogy vane olyan szóvektor, amely több kulcsszóvektorhoz is közel van), akkor kinyerhetőek olyan kulcsszavak is, amelyek a szövegben egyébként nem szerepelnek. Az információkinyerés (Information Extraction, IE) és az információ-visszakeresés (Information Retrieval, IR) egyik feladata a szövegben lévő kapcsolatok, relációk feltárása – a szóbeágyazás pedig éppen a szavak egymással való viszonyán alapul, így az ilyen típusú feladatokban nagy segítséget nyújthat. Könnyen felismerhetővé válnak a hasonló jelentéssel bíró szavak.

A szövegösszefüggés gépi vizsgálata régóta kitűzött cél, az eddig kifejlesztett megoldásokat többek között az esszék automatikus értékelésére használják (JURAFSKY–MARTIN 2009). A szövegösszefüggés mint diskurzustulajdonság (lásd BROWN–YULE 1993; JOHNSTONE 2008) könnyebben mérhetővé válik, ha a szövegben sorban előforduló fontosabb szavak (kulcsszavak) vektorainak távolságát mérjük. Ahol nagyobb távolságot tapasztalunk, ott témaváltásról lehet szó. Ennek érzékelésével diskurzusszegmentációt is meg tudunk valósítani. Fejleszthető a szóbeágyazással kapott vektorokon alapuló módszerrel az emberi felügyelet nélküli, automatikus diskurzusszegmentáció (pl. TextTiling-módszer [JURAFSKY–MARTIN 2009]), és az ember által valamilyen módon segített, felügyeletes gépi tanuláson alapuló szegmentáció is. Ez utóbbira több példa is akad. Pacheco és munkatársai (2016) a korpuszszövegekben eseményeket jelöltek be, ezek alapján eseményvektorokat hoztak létre.

Az eseményvektorok és a szóvektorok kombinált felhasználásával (a konkurens megoldásokhoz képest) olyan helyeken is nagy pontossággal tudtak diskurzusrelációkat és szegmentálási pontokat megállapítani, ahol a diskurzusjelölő nem volt explicit módon a szövegben. Mihaylov és Frank (2016) szintén eredményesek voltak diskurzusrelációk felismerésében úgy, hogy a diskurzusjelölők argumentumainak szemantikai hasonlóságát (azaz azok szóvektorainak közelségét) elemezték. Wu és munkatársai (2017) diskurzusspecifikus szóbeágyazási (DSWE) módszert fejlesztettek ki. A szóvektorokat használják a diskurzusrelációk felismerésére, és rendszerük azokat implicit diskurzusjelölő esetében is jó eredménnyel ismeri fel.

5. Összegzés

A szóbeágyazás tehát a részben vagy teljesen automatizált számítógépes diskurzuselemzés továbbfejlesztésére alkalmas eszköznek tűnik. A természetesnyelv-feldolgozás számos területén alkalmazták már eredményesen, de a diskurzusok gépi elemzésében való felhasználásáról eddig kevés munka született. Ennek oka többek között az ilyen rendszerek esetleges többszörös összetettsége, bonyolultsága. Diskurzusjelölő-érzékeny beágyazásoknál, vagy diskurzusszegmentációnál akár kétszer, két szinten is létre kell hozni a megfelelő vektorokat: az egyszeri szóbeágyazással kapott vektorokat egy következő körben egy összetett struktúrát jelképező vektor létrehozásához használjuk föl. Mivel mind a felhasznált módszerek, mind a számítógépek teljesítménye, képességei folyamatos fejlődés alatt állnak, ezért remélhető, hogy a közeljövőben újabb eredmények születnek a gépi diskurzuselemzés terén.

Irodalom

[1] Bengio, Y., Ducharme, R.,Vincent, P. & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research,* 3. évf., pp. 1137–1155.

[2] Brown, G. & Yule, G. (1993). *Discourse Analysis*. Cambridge: Cambridge University Press.

[3] Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*. Philological Society.

[4] Goldberg, Yoav & Levy, Omer (2014). *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method.* https://arxiv.org/abs/1402.3722 (Letöltve: 2018. 05. 19.)

[5] Harris, Z. S. (1954). Distributional structure. *Word*, Vol. 10, pp. 146–162.

[6] Johnstone, B. (2008). *Discourse Analysis*. Oxford: Blackwell Publishing.

[7] Jurafsky, D. & Martin, J. H. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River: Prentice Hall.

[8] Jurafsky, D. & Martin, J. H. (2017). *Speech and language processing. 3rd ed. draft.* https://web.stanford.edu/~jurafsky/slp3/ (Letöltve: 2018. 05. 19.)

[9] Katz, J. J. & Fodor, J. A. (1963). The structure of a semantic theory. *Language*, Vol. 39, 2, pp. 170–210. https://www.um.edu.mt/__data/assets/pdf_file/0015/137112/Week_1_-_Katz_J.J._Fodor_J.A._-_1963_-_The_Structure_of_Semantic_Theory.pdf (Letöltve: 2018. 05. 19.)

[10] Mihaylov, T. & Frank, A. (2016). Discourse Relation Sense Classification Using Cross-argument Semantic Similarity Based on Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.* Berlin: ACL. 100–107. http://www.aclweb.org/anthology/K16-2014 (Letöltve: 2018. 05. 19.)

[11] Mikolov, Tomas, Yih, Wen-Tau & Zweig, Geoffrey (2013a). *Linguistic regularities in continuous space word representations*. http://www.aclweb.org/anthology/N13-1090 (Letöltve: 2018. 05. 19.)

[12] Mikolov, Tomas, Chen, Kai, Corrado, Greg & Dean, Jeffrey (2013b). *Efficient estimation of word representations in vector space*. https://arxiv.org/abs/1301.3781v3 (Letöltve: 2018. 05. 19.)

[13] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg & Dean, Jeffrey (2013c). *Distributed representations of words and phrases and their compositionality*. https://arxiv.org/abs/1310.4546 (Letöltve: 2018. 05. 19.)

[14] Pacheco, M. L., Lee, I. T., Zhang, X., Zehady, A. K, Daga, P., Jin, D., Parolia, A. & Goldwasser, D. (2016). Adapting Event Embedding for Implicit Discourse Relation Recognition. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin: ACL. pp. 136–142. http://www.aclweb.org/anthology/K16-2019 (Letöltve: 2018. 05. 19.)

[15] Rong, Xin (2014). *word2vec parameter learning explained*. http://arXiv:1411.2738v4 (Letöltve: 2018. 05. 19.)

[16] Rong, Xin (2015). *Word embedding explained, and visualized. word2vec and wevi*. Ann Arbor: University of Michigan. https://www.youtube.com/watch?v=D-ekE-Wlcds (Letöltve: 2018. 05. 19.)

[17] Ruder, Sebastian (2016). *On word embeddings, part 1*. http://sebastianruder.com/word-embeddings-1/ (Letöltve: 2018. 05. 19.)

[18] Siklósi Borbála & Novák Attila (2016a). Beágyazási modellek alkalmazása lexikai kategorizációs feladatokra. In Tanács Attila, Varga Viktor, Vincze Veronika (szerk.). *XII. Magyar Számítógépes Nyelvészeti Konferencia.* Szeged: Szegedi Tudományegyetem, Informatikai Intézet.

[19] Siklósi Borbála, Novák Attila (2016b). Közeli rokonunk, az autó. In Tanács Attila, Varga Viktor, Vincze Veronika (szerk.). *XII. Magyar Számítógépes*

Nyelvészeti Konferencia. Szeged: Szegedi Tudományegyetem, Informatikai Intézet.

[20] Widdowson, H. G. (2011). *Discourse Analysis*. Oxford: Oxford University Press.

[21] Wu, C., Shi, X., Chen, Y., Su, J. & Wang, B. (2017). Improving Implicit Discourse Relation Recognition with Discourse-specificWord Embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*. Vancouver: ACL. pp. 269–274. http://www.aclweb.org/anthology/P17-2042 (Letöltve: 2018. 05. 19.)