



TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja

Text Analysis and Knowledge Engineering Lab

Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska



Zaštićeno licencijom

Creative Commons Imenovanje-Nekomercijalno-Bez prerada 3.0 Hrvatska

<https://creativecommons.org/licenses/by-nc-nd/3.0/hr/>

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1150

**Automatska izgradnja prijevodnih
rječnika temeljena na semantičkim
vektorskim prostorima**

Toni Antunović

Zagreb, srpanj 2015.

Zagreb, 6. ožujka 2015.

Predmet: **Analiza i pretraživanje teksta**

DIPLOMSKI ZADATAK br. 1150

Pristupnik: **Toni Antunović (0036482376)**

Studij: Računarstvo

Profil: Računarska znanost

Zadatak: **Automatska izgradnja prijevodnih rječnika temeljena na semantičkim vektorskim prostorima**

Opis zadatka:

Rječnici i prijevodne tablice izraza osnova su modernih sustava za statističko strojno prevođenje. Tradicionalni postupci za automatsku izgradnju prijevodnih rječnika oslanjaju se na usporedne dvojezične korpus. Kako je izgradnja usporednih korpusa iznimno naporan i skup postupak, noviji se postupci automatske izgradnje prijevodnih rječnika oslanjaju na usporedive korpus u kojima je uparivanje između dvaju jezika načinjeno tek na razini dokumenata. Izgradnja usporedivih korpusa značajno je manje zahtjevnost od izgradnje usporednih korpusa, a razvijeni su i pouzdani postupci za automatsku izgradnju takvih korpusa.

U okviru diplomskoga rada potrebno je proučiti postupke za automatsku izgradnju prijevodnih rječnika temeljene na usporedivim korpusima. Proučiti modele koji se temelje na semantičkoj reprezentaciji riječi u vektorskom prostoru obaju jezika, poput modela Mikolova i dr. (2013). Razraditi iterativni postupak za automatsku izgradnju prijevodnih rječnika za koji nisu potrebni ručno pripremljeni prijevodni parovi riječi. Razviti programsku implementaciju postupka, oslanjajući se na javno dostupne biblioteke za izgradnju semantičkih vektorskih prostora. Izgraditi usporediv hrvatsko-engleski web-korpus. Primjenom razvijenog postupka izgradnje prijevodnih rječnika nad usporedivim hrvatsko-engleskim korpusom izgraditi prijevodni hrvatsko-engleski rječnik. U okviru rada potrebno je provesti eksperimentalno vrednovanje postupka, usporedbu s odgovarajućim referentnim metodama te detaljnu analizu pogrešaka. Radu priložiti izvorni i izvršni kod razvijenog sustava, skupove podataka

Zadatak uručen pristupniku: 13. ožujka 2015.

Rok za predaju rada: 30. lipnja 2015.

Mentor:

Doc. dr. sc. Jan Šnajder

Djelovoda:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
diplomski rad profila:

Prof. dr. sc. Siniša Sribljčić

SADRŽAJ

1. Uvod	1
2. Izgradnja prijevodnih rječnika	3
3. Semantički vektorski prostori	5
3.1. Distribucijski modeli leksičke semantike	5
3.2. Neuronske mreže	6
3.2.1. Model neurona	6
3.2.2. Model umjetne neuronske mreže	7
3.2.3. Algoritmi učenja	8
3.3. Modeli neuronskih mreža	13
3.3.1. Jezični model unaprijedne neuronske mreže	13
3.3.2. Jezični model povratne neuronske mreže	14
3.4. Log-linearni modeli	15
3.4.1. Model CBOW	15
3.4.2. Model skip-gram	16
3.4.3. Metoda negativnog uzorkovanja	17
4. Modeli nenadziranog učenja prijevodnih rječnika na temelju usporedivih tekstnih zbirki	19
4.1. Usporedive tekstne zbirke	20
4.1.1. Wikipedija kao izvor za izradu usporedivih tekstnih zbirki	20
4.2. Model miješanja dokumenata različitih jezika	21
4.2.1. Osnovna inačica	21
4.2.2. Proširena inačica	22
4.2.3. Iterativni model	24
4.3. Mikolovljevo mapiranje	24
4.3.1. Linearni odnosi među jezicima	25

4.3.2. Translacijska matrica	25
4.4. Kombinacija Mikolovljevog mapiranja i modela miješanja dokumenata različitih jezika	26
5. Vrednovanje	27
5.1. Skupovi podataka	27
5.1.1. Korišteni usporedive tekstne zbirke	28
5.1.2. Format datoteka	28
5.1.3. Primjene	29
5.1.4. Izrada usporedive hrvatsko-engleske tekstne zbirke	30
5.2. Rezultati vrednovanja	31
5.2.1. Englesko-talijanski	31
5.2.2. Englesko-španjolski	35
5.2.3. Englesko-njemački	38
5.2.4. Englesko-francuski	41
5.3. Diskusija rezultata	44
5.4. Analiza pogrešaka	45
6. Zaključak	46
Literatura	48

1. Uvod

Rječnici i prijevodne tablice izraza osnova su modernih sustava za statističko strojno prevođenje. Tradicionalni postupci za automatsku izgradnju prijevodnih rječnika se oslanjaju na usporedne dvojezične tekstne zbirke. Kako je izgradnja usporednih zbirki iznimno naporan i skup postupak, noviji postupci automatske izgradnje prijevodnih rječnika oslanjaju se na usporedive u kojima je uparivanje između dvaju jezika načinjeno tek na razini dokumenta. Izgradnja usporedivih zbirki značajno je manje zahtjevna od izgradnje usporednih, a razvijeni su i pouzdani postupci za automatsku izgradnju takvih zbirki.

U okviru ovog diplomskog rada proučeni su postupci za automatsku izgradnju prijevodnih rječnika temeljeni na usporedivim tekstnim zbirkama (Vulić i Moens, 2015). Proučeni su modeli koji se temelje na semantičkom prikazu riječi u vektorskom prostoru obaju jezika, poput modela Mikolova i dr. (2013). Razrađen je postupak za automatsku izgradnju prijevodnih rječnika za koji nije potrebno imati ručno pripremljene prijevodne parove riječi. Oslanjajući se na javno dostupne biblioteke za izgradnju semantičkih vektorskih prostora razvijena je programska implementacija postupka. Primjenom razvijenog postupka izgradnje prijevodnih rječnika nad usporedivom hrvatsko-engleskom tekstnom zbirkom izgrađen je prijevodni hrvatsko-engleski rječnik. U okviru rada provedena je detaljna analiza pogrešaka, eksperimentalno vrednovanje postupka te usporedba s referentnim metodama. Radu su priloženi izvorni kod i skupovi podataka.

Rad je organiziran u šest poglavlja. Prvo poglavlje je uvod, a iza njega slijedi poglavlje koje govori o izgradnji prijevodnih rječnika. U njemu je detaljnije opisan taj zadatak i napravljen pregled najbitnijih radova koji su se bavili tom problematikom. U trećem poglavlju objašnjen je koncept semantičkih vektorskih prostora, opisano na koji način se razvijala ideja predstavljanja riječi vektorima realnih brojeva te pokazano na koji način rade algoritmi koji se koriste u ovom radu.

U četvrtom poglavlju se govori o svim modelima nenadziranog učenja prijevodnih rječnika na temelju usporedivih zbirki koji su korišteni u ovom radu. Nakon njega je opisano vrednovanje modela i detaljno prikazani rezultati. Također je napravljena i diskusija dobivenih rezultata i osnovna analiza pogrešaka. Posljednje poglavlje je zaključak rada.

2. Izgradnja prijevodnih rječnika

Prijevodni rječnici imaju bitnu ulogu u mnogim zadacima obrade prirodnog jezika. Na primjer, strojno prevođenje koristi dvojezične prijevodne rječnike za uparivanje riječi i fraza, a međujezično pronalaženje informacija za prevođenje upita. Izravni način izgradnje prijevodnih rječnika je korištenje usporednih tekstnih zbirki i izdvajanje najvjerojatnijih prijevodnih kandidata, ali takav pristup nije primjenjiv na sve jezike jer za mnoge od njih ne mogu se naći usporedne tekstne zbirke.

Pristup zasnovan na kontekstu najpopularniji je pristup za izgradnju prijevodnih rječnika iz usporedivih tekstnih zbirki. U tom slučaju se posmatra samo leksički kontekst riječi i sličan pristup zasnovan na semantičkim vektorskim prostorima je iskorišten u ovom radu.

U nastavku je dan pregled bitnih radova koji su se bavili izgradnjom vektorskih prikaza riječi na jednojezičnim i višejezičnim tekstnim zbirkama. Kod dijela s višejezičnim vektorskim prikazima riječi bitne metode podijeljene su na one koje zahtijevaju uparivanje na razini riječi, rečenica, zatim na one koji zahtijevaju unaprijed spremne rječnike i novi pristup koji zahtijeva samo usporedive tekstne zbirke (Vulić i Moens, 2015).

Jednojezični vektorski prikazi riječi

Ideja da se riječi predstavljaju kao vektori realnih brojeva nastala je još osamdesetih godina dvadesetog stoljeća (Rumelhart et al., 1988). U posljednje vrijeme ponovno je aktualizirana (Bengio et al., 2003) u obliku arhitekture neuronske mreže za statističko modeliranje jezika, koja je inspirirala niz novih pristupa za učenje prikaza riječi (Collobert i Weston, 2008; Collobert et al., 2011; Mnih i Hinton, 2007). Svi oni teže da što je moguće kvalitetnije predstave semantičke sličnosti riječi koristeći vektore realnih brojeva.

Modeli CBOW i skip-gram su se pojavili nedavno i pokazuju da za učenje kvalitetnih vektorskih prikaza riječi nije potrebna potpuna neuronska mreža (Mikolov et al., 2013a). Oba imaju jednoslojnu arhitekturu i za cilj imaju predviđanje konteksta trenutne riječi (skip-gram) ili predviđanje riječi na osnovu njezinog konteksta (CBOW). Pokazano je da model skip-gram u osnovi radi istu stvar kao i tradicionalni distribucijski modeli, samo što on to radi jako dobro (Baroni et al., 2014). U ovom radu za stvaranje vektorskih prikaza riječi će se koristiti upravo ova dva modela.

Višejezični vektorski prikazi riječi

Zbog uspjeha u raznim zadacima koji su postigli modeli za vektorski prikaz riječi iz jednog jezika, u posljednje vrijeme počelo se raditi na stvaranju višejezičnih vektorskih prikaza riječi koji će biti koherentni i upareni između jezika (Vulić i Moens, 2015). Gradnjom dijeljenih međujezičnih vektorskih prostora moguće je generalizirati razne semantičke zadatke.

Većina današnjih pristupa induciranju višejezičnih vektorskih prikaza riječi oslanja se na usporedne tekstne zbirke uparene na razini rečenica (Kočiský et al., 2014; Lauly et al., 2014; Gouws et al., 2014). Neki pristupi se oslanjaju na strogo uparivanje riječi dobiveno iz usporednih podataka (Klementiev et al., 2012; Zou et al., 2013), a neki na unaprijed spremne rječnike (Mikolov et al., 2013b).

Ovaj rad se temelji na pristupu koji se oslanja na usporedive zbirke za induciranje višejezičnih vektorskih prikaza riječi (Vulić i Moens, 2015). Također, koristit će se nadzirani model koji zahtijeva rječnik za učenje mapiranja između vektorskih prostora jezika (Mikolov et al., 2013b).

3. Semantički vektorski prostori

Mnogi današnji sustavi za obradu prirodnog jezika tretiraju riječi kao nedjeljive jedinice i ne obraćaju pažnju na pojam sličnosti između riječi. Ovakav odabir ima nekoliko logičnih razloga kao što su jednostavnost, robusnost, ali i spoznaja da jednostavniji modeli trenirani na velikim količinama podataka imaju bolje performanse od kompleksnih modela treniranih na manje podataka (Mikolov et al., 2013a).

Međutim, ovakvi jednostavni modeli imaju svoja ograničenja u mnogim zadacima. Na primjer, u mnogim primjenama količina dostupnih podataka je dovoljno mala da jednostavni model koji je treniran na njima nije u stanju ostvariti zadovoljavajuće rezultate. Prema tome, postoje situacije u kojima se moraju koristiti kompleksniji modeli, a napredak tehnika strojnog učenja u posljednjim godinama omogućio je treniranje kompleksnih modela na mnogo većim skupovima podataka nego prije i oni sada obično imaju bolje performanse od jednostavnih modela. Vjerojatno najuspješniji koncept koji se koristi u posljednje vrijeme su distribuirani vektorski prikazi riječi.

Predstavljanje riječi u obliku gustih vektora realnih brojeva visokih dimenzija koristi se već duže vrijeme (McClelland et al., 1986; Elman, 1990; Collobert i Weston, 2008). U ovom radu će se koristiti model od Mikolov et al. (2013a), a budući da je on dosta sličan modelima koji su u osnovi neuronske mreže u nastavku će biti opisane i neuronske mreže.

3.1. Distribucijski modeli leksičke semantike

Distribucijski modeli leksičke semantike koriste velike tekstne zbirke da bi odredili semantičke sličnosti između riječi. Osnova ovakvih modela je u hipotezi da se semantički slične riječi nalaze u sličnim kontekstima (Harris, 1954). Našli su široku primjenu u obradi prirodnog jezika. Mnogi znanstvenici skloni su tvrdnji da distribucijski modeli leksičke semantike odražavaju ljudske kognitivne procese (Bruni et al., 2011).

Tradicionalni modeli leksičke semantike zasnovani su na analizi tekstnih zbirki i predstavljaju semantičku sličnost između riječi kao funkciju stupnja preklapanja konteksta u kojima se one pojavljuju. Sličnost se računa u semantičkom prostoru predstavljenom uz pomoć matrice gdje su riječi redovi, a kontekstualni elementi kolone. Riječi se potom porede koristeći neku metriku udaljenosti, kao što je kosinus kuta između vektora kojima su predstavljeni. Primjeri ovakvih modela su LSA (*engl. Latent Semantic Analysis*) i LDA (*engl. Latent Dirichlet Allocation*).

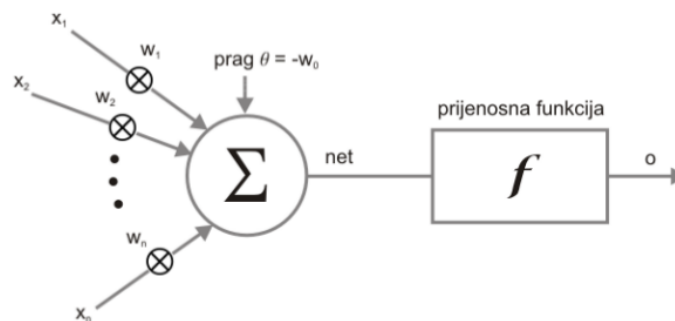
3.2. Neuronske mreže

Rad na umjetnim neuronskim mrežama je počeo kao pokušaj da se imitira način na koji ljudski mozak obrađuje informacije. Mozak je jako složeno, nelinearno i paralelno računalo koje obrađuje informacije na potpuno drukčiji način od konvencionalnog digitalnog računala.

3.2.1. Model neurona

Mozak je centralni dio živčanog sustava kod ljudi. U njemu se nalazi približno 10^{11} neurona koji se dijele na preko 100 različitih vrsta, a svaki od njih je povezan s 10^4 ostalih. Neuron predstavlja osnovnu procesnu jedinicu u neuronskim mrežama veličine otprilike $100 \mu\text{m}$.

Umjetni neuroni su građeni po uzoru na biološke. Kada se gradi neuronska mreža, ulazni sloj neurona se može smatrati receptorskim neuronima. Umjesto električnih impulsa koji se prenose unutar živčanog sustava, ovdje se radi o numeričkim vrijednostima, a jakost sinapse se opisuje težinskim faktorima veza između umjetnih neurona.



Slika 3.1: Umjetni neuron (Bašić et al., 2008)

Na slici se vide osnovni dijelovi umjetnog neurona. To su skup ulaznih veza, zbrajalo i aktivacijska funkcija. Svaka veza u skupu ulaznih veza okarakterizirana je težinom w . Svaki signal x_i se množi s težinom te veze w_i , nakon čega zbrajalo izračuna zbroj svih tih umnožaka i doda w_0 . Aktivacijska funkcija u većini slučajeva ograničava amplitudu izlaznog signala na neku konačnu vrijednost.

$$net = \sum_{i=1}^n w_i x_i + w_0 \quad (3.1)$$

$$o = f(net) \quad (3.2)$$

Postoji više tipova aktivacijskih funkcija koje se koriste u praksi, a najpopularnija je sigmoidalna funkcija. To je monotono rastuća nelinearna funkcija koja se na pojedinim dijelovima ponaša kao linearna. Ograničava amplitudu izlaznog signala na interval između 0 i 1 i deribabilna je na cijelom svom domenu, a to je važno svojstvo jer omogućuje primjenu gradijentnih metoda učenja.

$$f(net) = \frac{1}{1 + e^{-net}} \quad (3.3)$$

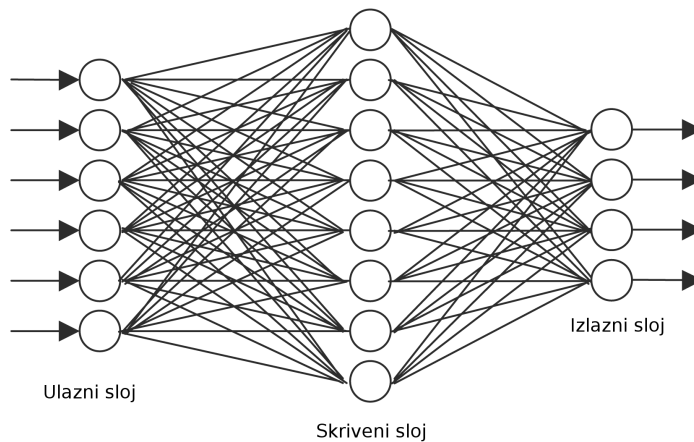
3.2.2. Model umjetne neuronske mreže

Budući da se prethodno opisani umjetni neuron može koristiti samo za najjednostavnije zadatke, oni se međusobno povezuju na različite načine u strukture koje se nazivaju umjetne neuronske mreže. Uobičajeno je da se one prikazuju u obliku usmjerenih grafova gdje su neuroni čvorovi, a veze između njih bridovi.

Način povezivanja neurona u neuronskoj mreži naziva se arhitektura mreže i usko je povezana s algoritmom koji će se koristiti za učenje mreže. Postoje tri osnovna tipa mrežnih arhitektura, a to su jednoslojne unaprijedne mreže, višeslojne unaprijedne mreže i povratne mreže.

Unaprijedne mreže nazivaju se još i aciklične što zapravo označava da rezultati izračuna idu isključivo u jednom smjeru, od ulaznog sloja prema izlaznom, bez ciklusa. Ulazni sloj ne obavlja nikakve transformacije nad ulaznim podacima nego ih samo prosljeđuje sljedećem sloju.

Jednoslojna unaprijedna mreža je ona koja poštuje prethodno navedeno pravilo i osim ulaznog sloja ima još točno jedan sloj koji je ujedno i izlazni.



Slika 3.2: Potpuno povezana višeslojna unaprijedna neuronska mreža

Višeslojne unaprijedne neuronske mreže osim ulaznog i izlaznog sloja imaju još minimalno jedan skriveni sloj između njih. Funkcija skrivenog sloja je da ulaze transformira na povoljan način i preda u izlazni sloj mreže. Ako se u skrivenom sloju nalazi više od jednog sloja neurona, onda je izlaz prvog sloja ulaz u drugi, izlaz drugog ulaz u treći i tako dalje do izlaznog sloja. Mreža je potpuno povezana kada je svaki čvor bilo kojeg sloja povezan sa svim čvorovima u susjednim slojevima.

Povratne neuronske mreže odlikuje postojanje barem jednog ciklusa. Mreža koja se sastoji od samo jednog sloja neurona je povratna ako barem jedan neuron šalje svoj izlaz barem jednom neuronu iz istog tog sloja. Moguće je, naravno, napraviti povratnu mrežu i sa skrivenim slojevima neurona pa i s ciklusima u kojim neuroni dobijaju vlastiti izlaz kao jedan od ulaza.

3.2.3. Algoritmi učenja

Jedna od najvažnijih osobina neuronskih mreža je mogućnost učenja. Kod njihovog učenja događa se sljedeći niz događaja:

1. Stimulacija mreže od strane okoline
2. Promjene parametara kao odgovor na stimulaciju okoline
3. Drukčiji odgovor na pobudu sljedeći put zbog prethodno izvršenih promjena

Algoritam učenja predstavlja skup unaprijed precizno utvrđenih pravila za rješenje problema učenja. Postoje dvije glavne paradigme vezane za algoritme učenja, a to su učenje s učiteljem i učenje bez učitelja.

Učenje s učiteljem često naziva se i nadzirano učenje (*engl. supervised learning*). Pod učiteljem podrazumijeva se nešto što ima znanje o okolini kojoj se neuronska mreža pokušava prilagoditi. Odvija se na način da se neuronskoj mreži daju trenirajući vektori na ulaz, ona daje određene izlaze, a učitelj uspoređuje željene izlaze s dobivenim. Nakon toga parametri mreže se prilagođavaju. Ovaj proces se ponavlja onoliko puta koliko je potrebno da neuronska mreža počne emulirati ponašanje svog učitelja (Haykin, 1994). Naravno, ovdje se treba voditi računa i o prenaučnosti, odnosno prevelikoj prilagođenosti modela primjerima za učenje, što najčešće povlači umanjenu sposobnost generalizacije. Pri učenju koristi se funkcija pogreške za određivanje promjene u parametrima nakon svake iteracije. Ako je ona derivabilna moguće je koristiti algoritme temeljene na gradijentu. Jedan od takvih je i algoritam s prostiranjem unazad koji će biti opisan poslije.

Kod učenja bez učitelja, kao što i samo ime kaže, ne postoji učitelj koji ima znanje o tome kako su označeni primjeri za učenje. Dijeli se na dvije vrste: pojačano učenje (*engl. reinforcement learning*) i nenadzirano učenje (*engl. unsupervised learning*).

Pojačano učenje je vrsta učenja u kojoj mreža uči iz interakcije s okolinom. Učenje se odvija na principu pokušaja i pogreške, a mreža uči na način da izvodi one akcije za koje dobija najveću nagradu od okoline (Sutton i Barto, 1998).

Nenadzirano učenje odvija se također bez učitelja na način da se kreira određena nezavisna mjera kvaliteta prikaza kojeg mreža treba naučiti, tako da se parametri mreže kroz proces učenja prilagođavaju toj mjeri. Mreža će se tako automatski prilagoditi određenim statističkim pravilnostima koje postoje u podacima za učenje i stvarati novi prikaz osobina ulaza.

Gradijentni spust

Ideja gradijentnog spusta koristi se u mnogim algoritmima nadziranog učenja, uključujući i algoritmu s prostiranjem unazad koji će biti opisan u nastavku, a temelji se na izračunu gradijenta u točki i pomjeranju pretrage u odgovarajućem smjeru.

Radi jednostavnosti za pojašnjenje ideje koristit će se jednostavna kvadratna funkcija jedne varijable $f(x) = x^2$ koju je potrebno minimizirati. Na početku nasumično se odabire početna točka pretraživanja x i na osnovu vrijednosti gradijenta, odnosno iznosu derivacije funkcije u njoj se vrijednost x povećava ili smanjuje (Čupić et al., 2013). Pravilo učenja koje se koristi je sljedeće:

$$x_{i+1} = x_i - \eta \cdot \left. \frac{df(x)}{dx} \right|_{x=x_i} \quad (3.4)$$

gdje je x_{i+1} nova točka pretraživanja, x_i stara, a η stopa učenja o kojoj ovisi ponašanje algoritma koje će biti opisano poslije.

Na slici je prikazana spomenuta funkcija $f(x) = x^2$ zajedno s tangentom na nju u točki $x = 2$. Budući da vrijednost prve derivacije funkcije $f(x)$ u nekoj točki predstavlja koeficijent smjera tangente u toj točki koji je jednak tangensu kuta koji tangenta zatvara s x-osi, zbog oblika kvadratne funkcije za sve točke desno od minimuma funkcije, koji je očito u točki $x^* = 0$, vrijednost prve derivacije će biti pozitivna, a za sve točke lijevo od minimuma negativna. Prema tome, za pomjeranje prema minimumu funkcije, od trenutne vrijednosti x će se oduzimati vrijednost prve derivacije u toj točki pomnožena sa stopom učenja. Ovakvo zaključivanje potvrđuje prethodno pravilo učenja navedeno u jednadžbi 3.4.

U ovisnosti o ranije spomenutoj stopi učenja η moguće su tri situacije prilikom provođenja gradijentnog spusta (Čupić et al., 2013). To su:

1. Monotona konvergencija. Ako je parametar η dovoljno malen, postupak će vrijednosti za x monotono približivati minimumu x^* . Granična vrijednost za η pri kojoj ponašanje postupka prelazi iz monotone u alternirajuću konvergenciju naziva se ograda monotonosti.
2. Alternirajuća konvergencija. Kada je η veća od ograde monotonosti, ali i dalje nedovoljno velika da postupak počne divergirati, x se približava optimumu u svakoj iteraciji alternirajući oko x^* . Granična vrijednost za η pri kojoj ponašanje postupka prelazi u divergenciju naziva se ograda divergencije.
3. Divergencija. Kada je η veća od ograde divergencije vrijednost x se ne približava optimumu nego u svakoj iteraciji alternira oko x^* , svaki put na većim udaljenostima.

Algoritam s prostiranjem unazad

Algoritam s prostiranjem unazad (*engl. backpropagation*) je algoritam nadziranog učenja neuronskih mreža otkriven 1986. godine (Rumelhart et al., 1988). Da bi se mogao iskoristiti za učenje neuronske mreže potrebno je da je ona unaprijedna i da su prijenosne funkcije neurona derivabilne.

Algoritam se odvija tako da se najprije desi unaprijedni prolaz kojim se na osnovu ulaza računaju aktivacije svakog od neurona, a zatim i parametar $\delta_i^{(k+1)}$ za svaki od njih. Parametar $\delta_i^{(k+1)}$ označava koliko je i -ti neuron k -tog sloja odgovoran za grešku na izlazu. Za izlazni sloj je ovaj parametar jednostavno odrediti jer je moguće izravno mjeriti razliku između aktivacije mreže i željenog rezultata. Za neurone u skrivenom sloju $\delta_i^{(k)}$ određuje se na osnovu težinskog prosjeka parametara $\delta_i^{(k+1)}$, tako da se vrijednosti izračunate u kasnijim slojevima propagiraju unatrag (Ng et al., 2012).

U nastavku će biti napisane formule za ažuriranje težina u slučaju da se koristeći algoritam s prostiranjem unazad realiziran kao algoritam grupnog učenja trenira unaprijedna višeslojna neuronska mreža s m izlaza, koristeći N primjera za učenje. Svi neuroni imaju sigmoidalnu aktivacijsku funkciju, a kao kriterijsku funkciju za mjerenje kvaliteta mreže koristi se polovina srednjeg kvadratnog odstupanja. Tada je izraz za računanje kriterijske funkcije sljedeći:

$$E = \frac{1}{2N} \sum_{s=1}^N \sum_{o=1}^m (t_{s,o} - y_{s,o}^{(k+1)})^2 \quad (3.5)$$

S $w_{i,j}^{(k)}$ će biti označena težina koja spaja i -ti neuron k -tog sloja i j -ti neuron u sloju $(k+1)$. Pravilo za ažuriranje težinskih faktora izlaznog sloja glasi:

$$w_{i,j}^{(k)} \leftarrow w_{i,j}^{(k)} - \eta \cdot \left(\sum_{s=1}^N [\delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)}] \right) \quad (3.6)$$

gdje je $y_{s,i}$ aktivacija i -tog neurona prethodnog sloja, η stopa učenja i $\delta_{s,j}^{(k+1)}$ pogreška j -tog neurona izlaznog sloja za s -ti uzorak za učenje, a računa se kao:

$$\delta_{s,j}^{(k+1)} = y_{s,j}^{(k+1)} \cdot (1 - y_{s,j}^{(k+1)}) \cdot (t_{s,j} - y_{s,j}^{(k+1)}). \quad (3.7)$$

U prethodnom izrazu $t_{s,j}$ je očekivani izlaz j -tog neurona izlaznog sloja za s -ti primjer, a $y_{s,j}^{(k+1)}$ dobiveni izlaz tog neurona.

Težine neurona skrivenih slojeva računaju se kao:

$$w_{i,j}^{(k-1)} \leftarrow w_{i,j}^{(k-1)} + \eta \cdot \left(\sum_{s=1}^N [\delta_{s,j}^{(k)} \cdot y_{s,i}^{(k-1)}] \right) \quad (3.8)$$

$$\delta_{s,j}^{(k)} = y_{s,j}^{(k)} \cdot (1 - y_{s,j}^{(k)}) \cdot \left(\sum_{o=1}^m [\delta_{s,o}^{(k+1)} \cdot w_{j,o}] \right) \quad (3.9)$$

3.3. Modeli neuronskih mreža

Postoje mnogi modeli za stvaranje vektorskih prikaza riječi, uključujući poznate LSA (*engl. Latent Semantic Analysis*) i LDA (*engl. Latent Dirichlet Allocation*), kao i modeli temeljeni na neuronskim mrežama koji postižu bolje performanse od LSA, dok je LDA računalno prezahtjevan da bi se koristio na velikim skupovima podataka.

Za sve modele opisane u nastavku složenost treniranja je proporcionalna:

$$O = E \times T \times Q, \quad (3.10)$$

gdje je E broj epoha treniranja, T broj riječi u skupu za treniranje i Q će biti u nastavku definiran za svaki pojedinačni model. Svi oni mogu se trenirati koristeći prethodno opisane stohastički gradijentni spust i algoritam s prostiranjem unazad.

3.3.1. Jezični model unaprijedne neuronske mreže

Jezični model unaprijedne neuronske mreže (*engl. Feedforward Neural Net Language Model*) sastoji se od ulaznog, projekcijskog i izlaznog sloja (Bengio et al., 2003). Na ulaznom sloju N prethodnih riječi kodira se korištenjem 1-od- V kodiranja, gdje je V veličina rječnika. Ulazni sloj se potom projicira na projekcijski sloj P koji ima dimenzije $N \times D$, korištenjem dijeljene projekcijske matrice. Projekcija je računalno relativno jeftina operacija, ali arhitektura modela postaje računalno složenija na prijelazu iz projekcijskog u skriveni sloj. Skriveni sloj se također koristi za računanje vjerojatnosne distribucije svih riječi u rječniku, što rezultira s izlaznim slojem koji ima dimenzionalnost V (Mikolov et al., 2013a). Prema tome, složenost treniranja svakog primjera za učenje je:

$$Q = N \times D + N \times D \times H + H \times V \quad (3.11)$$

Dominantni član u ovom slučaju je $H \times V$. Postoje načini da se izbjegne ovo računanje. Neki od njih su korištenje hijerarhijskih verzija softmaxa ili potpuno izbjegavanje korištenja ovakvih modela. Kada se rječnik predstavi pomoću binarog stabla, onda broj izlaznih stavki koje moraju biti izračunate se smanjuje na približno $\log_2(V)$ pa se većina složenosti sada nalazi u izrazu $N \times D \times H$.

U modelima koji su korišteni u ovom radu nalazi se hijerarhijski softmax, a rječnik je predstavljen Huffmanovim binarnim stablom. Huffmanovo binarno stablo češće riječi predstavlja kraćim binarnim kodovima što dodatno smanjuje broj izlaznih stavki koje treba izračunati i u odnosu na uravnoteženo binarno stablo.

3.3.2. Jezični model povratne neuronske mreže

Jezični model povratne neuronske mreže (*engl. Recurrent Neural Net Language Mode*) osmišljen je da bi se zaobišli određeni nedostaci NNLM modela kao što je potreba za određivanjem veličine konteksta, ali i zbog činjenice da teoretski rekurzivne neuronske mreže mogu efikasnije predstaviti kompleksnije uzorke nego plitke unaprijedne. One nemaju projekcijski sloj, nego samo ulazni, skriveni i izlazni. Izlazi iz skrivenog sloja su istovremeno i ulazi u skriveni sloj s vremenskom odgodom (Mikolov et al., 2013a).

Složenost treniranja jednog primjera za učenje u ovom slučaju iznosi:

$$Q = H \times H + H \times V, \quad (3.12)$$

gdje prikazi riječi D imaju istu dimenziju kao i skriveni sloj H , a član $H \times V$ može biti reduciran na $H \times \log_2(V)$ koristeći hijerarhijski softmax.

3.4. Log-linearni modeli

U ovom dijelu opisana su dva modela za učenje distribuiranih vektorskih prikaza riječi. Oba modela pokušavaju minimizirati računalnu složenost i iako nisu u mogućnosti predstaviti podatke precizno kao neuronske mreže, može ih se efikasno trenirati koristeći puno veći skup za učenje. Metoda negativnog uzorkovanja koja se koristi za učenje će biti pojašnjena na skip-gram modelu.

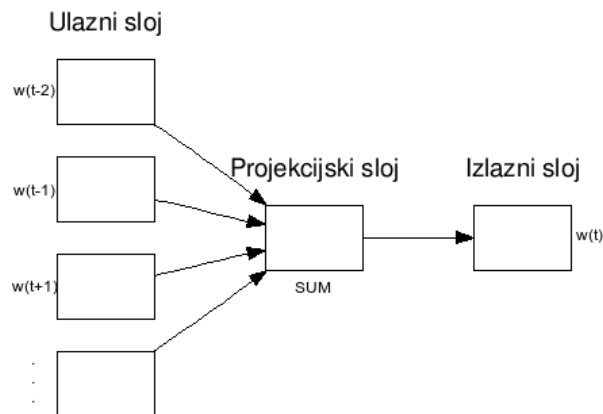
3.4.1. Model CBOW

Model CBOW (*engl. Continuous Bag-of-Word*) je sličan modelu NNLM. Razlika je u tome da je nelinearni skriveni sloj uklonjen i projekcijski sloj se dijeli između svih riječi. Ova arhitektura ima naziv koji u prijevodu znači vreća riječi jer redoslijed riječi ne utječe na projekciju. Koriste se riječi iz prošlosti, ali i riječi iz budućnosti, a kriterij za učenje je ispravno klasificiranje trenutne, odnosno riječi u sredini (Mikolov et al., 2013a).

Složenost treniranja je:

$$Q = N \times D + D \times \log_2(V). \quad (3.13)$$

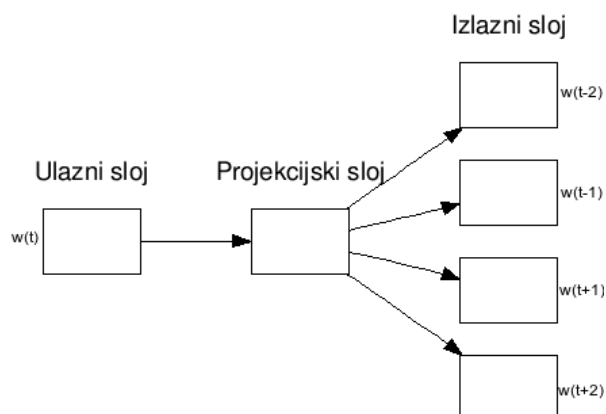
Težinska matrica između ulaznog i projekcijskog sloja je ista za sve pozicije trenutne riječi. Arhitektura modela prikazana je na sljedećoj slici.



Slika 3.3: Arhitektura modela CBOW

3.4.2. Model skip-gram

Druga arhitektura je slična CBOW arhitekturi, ali umjesto predviđanja trenutne riječi na osnovu njezinog konteksta, ona pokušava maksimizirati kvalitetu klasifikacije riječi na osnovu druge riječi u rečenici. Preciznije rečeno, trenutna riječ koristi se kao ulaz u log-linearni klasifikator i služi za predviđanje riječi u određenom rasponu ispred i iza nje.



Slika 3.4: Arhitektura modela skip-gram

Utvrđeno je da povećavanje veličine prozora poboljšava kvalitetu rezultirajućih vektora riječi, ali također i računalnu složenost. S obzirom da su udaljenije riječi obično manje vezane za trenutnu riječ nego one koje su joj blizu, njima se daju manje težine kroz uzorkovanje manje udaljenijih riječi u primjerima za učenje (Mikolov et al., 2013a).

Složenost treniranja ovog modela je:

$$Q = C \times (D + D \times \log_2(V)), \quad (3.14)$$

gdje je C maksimalna udaljenost između riječi, D dimenzija vektora riječi i V veličina vokabulara.

3.4.3. Metoda negativnog uzorkovanja

Kao što je prethodno rečeno, ova metoda će biti objašnjena na skip-gram modelu. Kod ovog modela dan je skup riječi $w \in V$ i njihovi konteksti $v \in V^c$. Veličina konteksta ovisi o veličini prozora cs koja je ulazni parametar algoritma.

Svaka riječ $w \in V$ ima svoj vektorski oblik $\vec{w} \in \mathbb{R}^d$, gdje je d veličina vektora koji predstavljaju riječi i također je ulazni parametar algoritma.

Glavna ideja algoritma je da prolazom kroz tekstnu zbirku riječ po riječ nauči vektorske oblike svih riječi. Cilj je maksimizirati mogućnost predviđanja riječi iz konteksta trenutnog pivota. Vjerojatnost neke riječi $\vec{v}_c$ iz konteksta pivota $\vec{w}$ je dana sa:

$$P(\vec{v}_c|\vec{w}) = \frac{1}{1 + \exp(-\vec{w} \cdot \vec{v}_c)} \quad (3.15)$$

Svaka riječ iz teksta u jednom trenutku postaje pivot w i svi parovi riječi s pivotom i riječima iz njihovih konteksta (w, v_c) postaju dio skupa za učenje D . Tako da je globalni cilj učenja J maksimizirati vjerojatnost posmatranja svih parova iz skupa za učenje D u tekstnoj zbirci:

$$J = \arg \max_{\theta} \sum_{(\vec{w}, \vec{v}) \in D} \log \frac{1}{1 + \exp(-\vec{w} \cdot \vec{v}_c)} \quad (3.16)$$

gdje θ prestavlja parametre modela, odnosno vektore riječi pivota i njezinog konteksta. Ovaj problem ima trivijalno rješenje u obliku $\vec{w} = \vec{v}_c$ i $\vec{w} \cdot \vec{v}_c = N$, gdje je N neka dovoljno velika vrijednost. U cilju izbjegavanja ovog trivijalnog rješenja koristi se procedura negativnog uzorkovanja.

Ukratko, ova metoda dodaje postojećem skupu za treniranje D novi skup negativnih primjera za učenje D' koji se sastoji od negativnih primjera za učenje, odnosno sadrži parove pivot-kontekst riječi (w, v'_c) koji se ne nalaze u tekstnoj zbirci. Nakon toga model pokušava maksimizirati i vjerojatnost da se ovi negativni parovi ne nalaze u tekstnoj zbirci.

$$J = \arg \max_{\theta} \sum_{(\vec{w}, \vec{v}) \in D} \log \frac{1}{1 + \exp(-\vec{w} \cdot \vec{v}_c)} + \sum_{(\vec{w}, \vec{v}') \in D'} \log \frac{1}{1 + \exp(\vec{w} \cdot \vec{v}'_c)} \quad (3.17)$$

Optimalni parametri θ računaju se koristeći stohastički gradijentni spust i algoritam s prostiranjem unazad. Na kraju će riječi koje se često pojavljuju u istom kontekstu

imati slične naučene vektorske prikaze, a ta činjenica je iskorištena kod stvaranja dvojezičnih rječnika.

Modeli CBOW i skip-gram daju jako dobre rezultate u stvaranju kvalitetnih vektorskih prikaza riječi. Međutim, ne postoji formalno objašnjenje zašto toliko dobro rade. Postoji tek intuitivno objašnjenje da riječi koje se pojavljuju u istom kontekstu imaju i slično značenje, a funkcija cilja treniranja modela koja je prethodno navedena očigledno pokušava povećati vrijednost $w \cdot v_c$ za ispravne parove pivot-kontekst i smanjiti za neispravne pa se taj cilj i postiže (Goldberg i Levy, 2014).

4. Modeli nenadziranog učenja prijevodnih rječnika na temelju usporedivih tekstnih zbirki

U ovom poglavlju će biti opisano nekoliko metoda za automatsko stvaranje prijevodnih rječnika temeljenih na usporedivim tekstnim zbirkama. Veliki dio radova koje se bave distribucijskom semantikom temelji se na distribucijskoj hipotezi koja govori da se slične riječi nalaze u sličnim kontekstima (Harris, 1954). Svi pristupi semantici koji se temelje na tekstnim zbirkama se oslanjaju na kontekste na jedan ili drugi način (Vulić i Moens, 2015).

U posljednje vrijeme pojavile su se metode za dobivanje kvalitetnih prikaza riječi u obliku gustih vektora realnih brojeva. Prvi takvi modeli su imali arhitekture zasnovane na neuronskim mrežama. Svi oni nude bogatije i koherentnije vektorske prikaze riječi nego tradicionalne metode poput LSA i LDA.

Rječnici i tabele fraza osnova su modernih statističkih sustava za strojno prevođenje. Takvi sustavi su razvijani godinama i jako su uspješni u praksi. Međutim, za stvaranje rječnika i tabela fraza je potrebno uložiti mnogo truda, a njihove performanse su još uvijek daleko ispod performansi ljudskih prevoditelja. U ovom radu je istraženo nekoliko metoda za automatsko stvaranje takvih rječnika temeljenih na semantičkim vektorskim prostorima (Mikolov et al., 2013b).

Model koji je isproban i nadograđen u ovom radu omogućava nenadzirano stvaranje dvojezičnih rječnika i zasnovan je na radu Vulić i Moens (2015). Osim toga, isprobano je i nekoliko drugih metoda, uključujući i rad Mikolov et al. (2013b).

4.1. Usporedive tekstne zbirke

Višejezični resursi u obradi prirodnog jezika najčešće se konstruiraju iz usporednih tekstnih zbirki. Međutim, do usporednih tekstnih zbirki se teško dolazi i zbog toga su često veličinom veoma ograničeni i postoje samo za relativno malen broj parova jezika. Ovakva situacija je učinila da su se istraživači sve više počeli zanimati za druge višejezične resurse kao što su usporedive tekstne zbirke. Usporedive tekstne zbirke su skupovi tekstova na različitim jezicima koji nisu međusobni prijevodi, ali su iz iste domene (Bowker i Pearson, 2002).

One imaju nekoliko očiglednih prednosti na usporednim. Dostupni su na internetu u velikim količinama, za mnoge jezike i mnoge domene. Također, višejezični rječnici su osnova svih međujezičnih primjena obrade prirodnog jezika, kao što su strojno prevođenje i međujezično pronalaženje informacija (Sellami et al., 2012).

Usporedne tekstne zbirke, iako najvažniji resurs koji se koristi u statističkom strojnom prevođenju, brojem su ograničeni, postoje samo za određene jezike i domene. Zbog toga usporedive tekstne zbirke omogućavaju poboljšanje kvalitete statističkih sustava strojnog prevođenja kroz veću pokrivenost različitih domena, posebno kod jezika za koje postoji malen broj usporednih tekstnih zbirki.

U ovom radu kao izvor usporedivih tekstnih zbirki korištena je Wikipedija, odnosno dokumenti su bili organizirani u parove koji na različitim jezicima govore o istoj temi. U nastavku teksta detaljnije je opisana Wikipedija kao izvor za izradu usporedivih tekstnih zbirki.

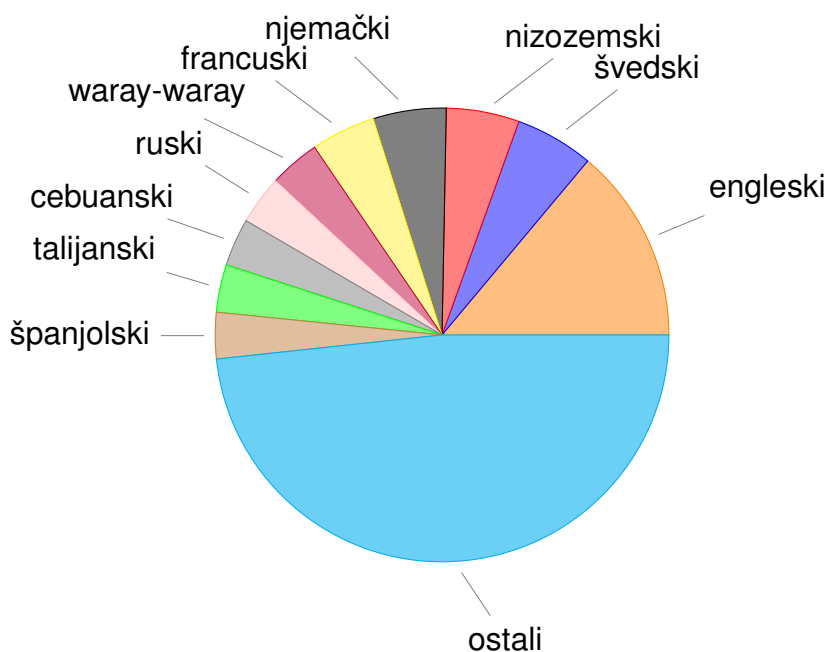
4.1.1. Wikipedija kao izvor za izradu usporedivih tekstnih zbirki

Wikipedija je besplatna online enciklopedija koja djeluje kao dio neprofitne Wikimedia fondacije. Zasnovana je na Wiki konceptu (Leuf i Cunningham, 2001) pa svatko može doprinijeti stvaranju novih članaka, uređivanju i poboljšavanju već postojećih. U početku su sve napravljene promjene bile odmah objavljivane, ali je zbog pojave lažnih informacija s vremenom uvedena zaštita i proces odobravanja promjena.

Trenutno postoji 288 Wikipedija izdanja na različitim jezicima, a najveća je engleska Wikipedija koja sadrži više od 4.8 milijuna članaka (wikipedia.org, 2015).

Svi članci su razvrstani u različite kategorije poput povijesti, umjetnosti, društva, znanosti i tehnologije. Također, brojni su članci o imenovanim entitetima, poput imena osoba i slično i njihov broj se stalno povećava.

Međujezične poveznice su poveznice između dva članka koja imaju istu temu, ali pisani su na različitim jezicima. Članci obično imaju po jednu takvu poveznicu za svaki jezik. Automatizirano prevođenje članaka nije omogućeno pa prevedeni članci predstavljaju samo manji dio ukupnog broja članaka kod većine izdanja.



Slika 4.1: Distribucija članaka izdanja na različitim jezicima

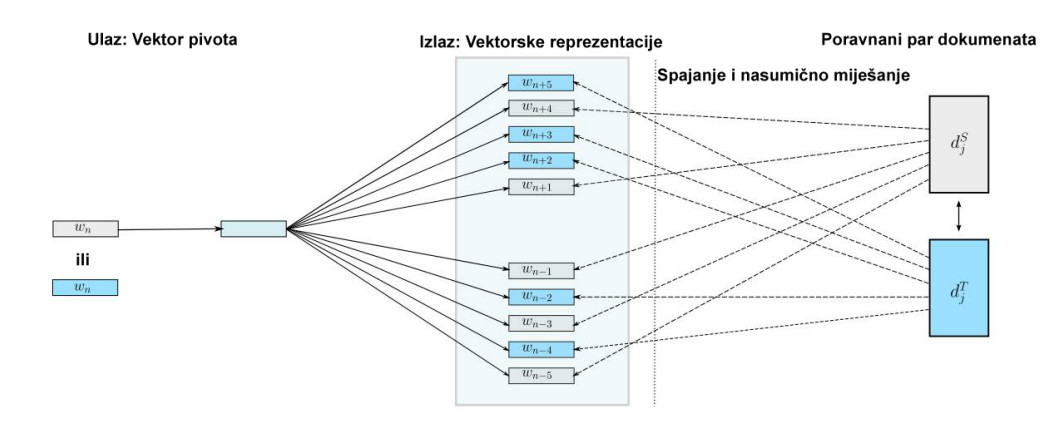
4.2. Model miješanja dokumenata različitih jezika

4.2.1. Osnovna inačica

Ovaj model je ustvari nadogradnja prethodno opisanog skip-gram modela primijenjena na višejezičnom skupu podataka za treniranje.

Za primjenu ovog modela potrebno je imati usporedivu tekstnu zbirku dokumenata na dva jezika koji su upareni prema temi. Kao prvi korak potrebno je spojiti sve parove dokumenata na način da se dokumenti koji su dio jednog para budu spojeni u jedan novi "pseudo-dvojezični" dokument. Nakon toga nasumično se izmiješaju sve riječi unutar

tog novog dokumenta s ciljem da svaka riječ ima susjede iz oba jezika. Ideja iza ovog koraka je dobiti višjezični kontekst za svaku riječ koja će biti pivot kod primjene skip-gram modela (Vulić i Moens, 2015).



Slika 4.2: Model BWESG (Vulić i Moens, 2015)

Primjena skip-gram modela na ovako generiranom dokumentu će stvoriti dijeljeni međujezični vektorski prostor. Intuitivno je pretpostaviti da će, budući da je tekstna zbirka uparena samo na razini dokumenta, veća veličina prozora koja se prosljeđuje kao parametar skip-gram modelu dati bolje rezultate. Ova pretpostavka će biti potvrđena u poglavlju u kojem su navedeni rezultati eksperimenata.

Prema tome, konačni model nazvan BWE skip-gram (BWESG) se oslanja na jednojezičnu varijantu skip-gram modela treniranu na nasumično izmiješanim pseudo-dvojezičnim dokumentima. Model uči vektorske prikaze riječi u dijeljenom vektorskom prostoru i pronalazak prijevoda za riječ se svodi na pronalazak riječi iz drugog jezika koja ima najsličniji vektor.

4.2.2. Proširena inačica

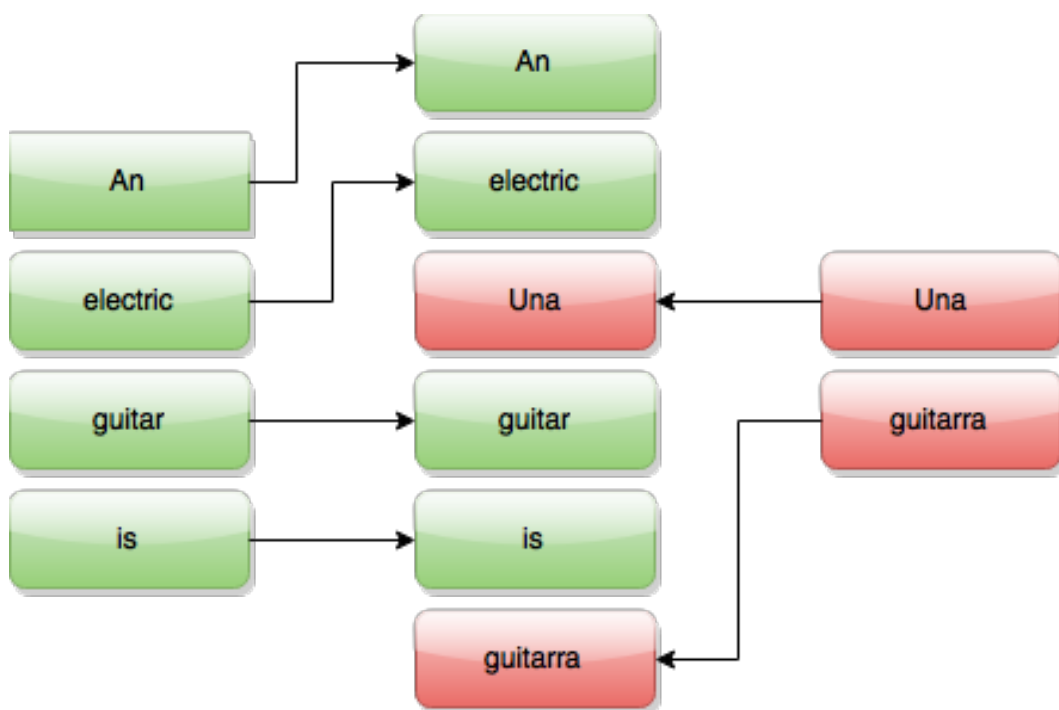
U cilju poboljšanja performansi osnovnog modela napravljena je promjena u njegovoj inicijalizaciji. Ona je, kako je prethodno opisano, bila nasumično miješanje riječi iz para dokumenata i njihovo spajanje u jedinstven dokument na kome se trenira model CBOW ili skip-gram.

Nova inicijalizacija podrazumijeva početno računanje broja riječi u svakom dokumentu, a potom raspoređivanje riječi u ovisnosti o njihovom omjeru. Tako da ako prvi

dokument na prvom jeziku ima dva puta više riječi od dokumenta na drugo jeziku, najprije će se u novi pseudodokument slijedno postaviti dvije riječi iz prvog dokumenta, potom jedna iz drugog, zatim opet dvije iz prvog i tako dalje dok se ne rasporede sve riječi iz oba dokumenta.

Ova metoda će biti detaljnije opisana na jednostavnom primjeru. Ako se koriste članci o električnoj gitari s engleske i španjolske Wikipedije, i ako radi jednostavnosti u primjeru smatramo da članak na engleskom sadrži samo četiri riječi, a članak na španjolskom samo dvije, onda će se najprije izračunati odnos broja riječi u oba članka. Jasno je da na jednu španjolsku riječ dolaze dvije engleske. U tom slučaju će miješanje članaka teći tako da se najprije uzmu dvije engleske riječi, dodaju u pseudodokument, zatim jedna španjolska riječ, potom opet dvije engleske i na kraju još jedna španjolska.

Ovakva jednostavna ideja daje značajno bolje rezultate od osnovne inačice što će se vidjeti u poglavlju s eksperimentima u kojem su navedeni svi rezultati.



Slika 4.3: Primjer stvaranja pseudodokumenta kod proširene inačice modela miješanja dokumenata

4.2.3. Iterativni model

Ideja iterativne izgradnje osnovnog modela je pretpostavljala da će umetanje vjerovatnih prijevoda u kontekste određenog broja najčešćih riječi dati bolje rezultate od osnovnog modela.

Umetanje se vršilo na način da se za neki broj n najčešćih riječi u oba jezika odabere njihovih k najvjerovatnijih prijevoda dobivenih primjenom osnovnog nenadziranog modela (n i k su parametri koji se zadaju prije pokretanja procedure). Zatim se u kontekst svake od n riječi koristeći slučajni težinski odabir odabire jedan od njezinih k najvjerovatnijih prijevoda. Tako najsličnije riječi, odnosno najvjerovatniji prijevodi imaju najveću šansu da budu umetnuti u konteksts.

Procedura se iterativno ponavlja sve dok se preciznost mjerena na evaluacijskom skupu povećava. U poglavlju koje govori o rezultatima eksperimenata se nalaze rezultati izvođenja.

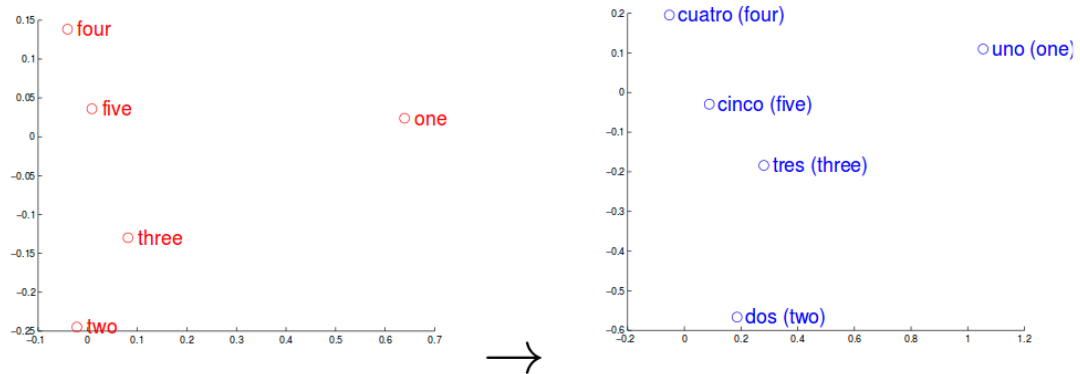
4.3. Mikolovljevo mapiranje

Kod ovog modela objavljenog u Mikolov et al. (2013b), stvaranje dvojezičnog rječnika se postiže učenjem linearne projekcije između vektorskih prostora koji predstavljaju svaki jezik. Metoda se sastoji od dva jednostavna koraka. Prvi je izrada jednojezičnih modela za oba jezika koristeći velike količine teksta, a drugi je korištenje malenog dvojezičnog rječnika da bi se naučila linearna projekcija između jezika. Nakon toga kao rezultat se dobije translacijska matrica koja može služiti da se svaka riječ iz vlastitog vektorskog prostora prenese u vektorski prostor ciljnog jezika. Zatim se u vektorskom prostoru ciljnog jezika traže najsličnije riječi i one se smatraju prijevodima.

Vektorski prikazi riječi se uče koristeći modele CBOW i skip-gram. Oba modela imaju jednostavne arhitekture temeljene na neuronskim mrežama i mogu efikasno biti trenirani na velikim skupovima podataka. Ova metoda daje uvijek neki prijevod za riječ, što može biti korisno kod proširivanja postojećih tabela fraza novim prijevodnim kandidatima.

Distribuirani vektorski prikazi riječi su predloženi još u McClelland et al. (1986) i njihova glavna prednost je da semantički slične riječi dobivaju i slične vektore. Nakon toga je logično bilo da se takvi modeli počnu primjenjivati u statističkom strojnom

prevođenju i otkriveno je da su sposobni prikazati iznenađujuće veliki broj jezičnih pravilnosti. Među tim pravilnostima je i mogućnost izražavanja riječi pomoću linearnih translacija.



Slika 4.4: Geometrijski raspored naučenih vektora brojeva u engleskom i španjolskom jeziku (Mikolov et al., 2013b)

4.3.1. Linearni odnosi među jezicima

Na slici 4.4 je prikazano da su vektori sličnih riječi u različitim jezicima vezani linearnom transformacijom. Na primjer, vektori za engleske i španjolske brojeve od jedan do pet imaju sličan geometrijski raspored. Prema tome, odnos vektora riječi koji su naučeni za ova dva jezika može se prikazati linearnim mapiranjem.

4.3.2. Translacijska matrica

Ako je dan skup parova riječi iz dva različita jezika i njihovi odgovarajući vektorski prikazi $(\vec{x}_i, \vec{z}_i)_{i=1}^n$, gdje je $\vec{x}_i \in \mathbb{R}^{d_1}$ vektorski prikaz riječi i u izvornom jeziku, a $\vec{z}_i \in \mathbb{R}^{d_2}$ je vektorski prikaz njezinog prijevoda.

Cilj učenja ovog modela je pronalazak transformacijske matrice W takve da Wx_i aproksimira z_i . W se može odrediti rješavanjem sljedećeg optimizacijskog problema koji može biti riješen stohastičkim gradijentnim spustom:

$$\min_W \sum_{i=1}^n \|W \cdot \vec{x}_i - \vec{z}_i\|^2 \quad (4.1)$$

U trenutku kada je potrebno napraviti predikciju prijevoda izvorne riječi, svaku novu riječ x i njen vektorski prikaz je moguće mapirati u vektorski prostor ciljnog

jezika koristeći translacijsku matricu W i izraz $\vec{z}_i = W \cdot \vec{x}_i$. Potom se nađe riječ iz ciljnog jezika čiji je vektorski prikaz najbliži dobivenom vektoru $\vec{z}_i$ koristeći kosinusnu sličnost kao metriku udaljenosti.

4.4. Kombinacija Mikolovljevog mapiranja i modela mi- ješanja dokumenata različitih jezika

Rezultati prethodno opisanog modela koji pomoću linearne transformacije pronalazi prijevode za riječi iz izvornog jezika prijavljeni u Mikolov et al. (2013b) su jako dobri, pa je zbog toga nastala ideja da se pokuša učiti transformacijska matrica pomoću parova riječi dobivenih iz nenadziranog modela Vulić i Moens (2015).

Skup za učenje transformacijske matrice se dobije uzimanjem najvjerojatnijih prijevoda za n najčešćih riječi izvornog jezika, uz dodatak da se uzimaju prijevodi koji imaju sličnost veću od nekog realnog broja između 0 i 1, koji je zadan kao hiperparametar. Potom se uči transformacijska matrica na način koji je opisan jednom od prethodnih potpoglavlja, a predikcija prijevoda novih riječi matričnim množenjem vektorskog prikaza izvorne riječi s transformacijskom matricom i pronalaskom najsličnijeg vektora iz vektorskog prostora ciljnog jezika.

5. Vrednovanje

Ovo poglavlje sadrži opis i rezultate svih provedenih eksperimenata. Najprije su opisani korišteni skupovi podataka, postupak predobrade i njihovog stvaranja. Nakon toga slijedi dio s rezultatima u kojem je za sve eksperimente i sve parove jezika najprije ukratko opisana postavka eksperimenta, a potom tablično navedeni rezultati.

Eksperimenti su provedeni na skupovima podataka za pet parova jezika: englesko-talijanski, englesko-španjolski, englesko-njemački, englesko-francuski i englesko-hrvatski. Kreirana tekstna zbirka za englesko-hrvatski par imala je 14000 parova članaka i mjerenje preciznosti su bile na proširenoj inačici modela jako niske (svega nekoliko posto), pa nisu tablično prikazane.

Za svaki od par jezika tablično su navedeni rezultati eksperimenata s različitim modelima koji su prethodno opisani, najprije osnovna inačica nenadzirani model, te njegova jednostavna nadgradnja s novom inicijalizacijom, zatim iterativna izgradnja tog modela i na kraju rezultati kombinacije Mikolovljevog mapiranja i modela miješanja dokumenata. Eksperimenti nad osnovnom inačicom nenadziranog modela miješanja dokumenata su vršeni tri puta i u tablicama su prikazani prosječni rezultati.

Nakon izlaganja rezultata, provedena je kratka diskusija istih, usporedba sa srodnim radovima i osnovna analiza pogrešaka.

5.1. Skupovi podataka

Kao osnova za stvaranje skupova za treniranje i testiranje poslužio je Wikipedia Comparable tekstna zbirka koja je prethodno opisana. Nad Wikipedia Comparable XML datotekama je vršena predobrada i izdvajanje određenog broja parova članaka za svaki par jezika. U nastavku će posebno biti opisani postupci koji su poslužili za stvaranje skupova za treniranje i testiranje.

5.1.1. Korišteni usporedive tekstne zbirke

Za eksperimente su korišteni usporedive tekstne zbirke iz Wikipedija usporedivih tekstnih zbirki (*engl. Wikipedia Comparable Corpora*) dostupnog na internetu (linguato-ols.org, 2014).

Ove usporedive tekstne zbirke su dvojezične i uparene na razini dokumenta. Generirane su iz jednojezičnih XML Wikipedija tekstnih zbirki koristeći međujezične linkove. Svaka usporediva zbirka sastoji se od parova dokumenata: članci iz prvog jezika su povezani s člancima iz drugog jezika na osnovu iste teme. Sve zajedno, tu se nalazi preko 41 milijun uparenih članaka za 253 para jezika.

5.1.2. Format datoteka

Datoteke su u XML formatu. Korijski XML element je *wikipediaComparable*. Njegov atribut *name* sadrži skraćenice para jezika. Zatim slijedi zaglavlje, odnosno element *header* koji ima dva čvora potomka, oba tipa *wikipediaSource*. Nihovi atributi sadrže skraćenice jezika i imena jednojezičnih Wikipedija tekstnih zbirki koje su korištene pri generiranju usporedivih.

Poslije zaglavlja nalazi se *n* elemenata tipa *articlePair*, koji imaju atribut *id* s jedinstvenim identifikacijskim brojem. Svaki element ovog tipa sadrži dva elementa tipa *article*, od kojih prvi sadrži članak na prvom jeziku, a drugi članak na koji pokazuje međujezični link na drugom jeziku. Slijedi primjer koji pokazuje kako izgleda jedna Wikipedia Comparable datoteka.

```
<?xml version="1.0" encoding="utf-8"?>
<wikipediaComparable name="nl-ro">
  <header>
    <wikipediaSource language="nl"
      name="nlwiki-20140804-corpus.xml"/>
    <wikipediaSource language="ro"
      name="rowiki-20140729-corpus.xml"/>
  </header>
  <articlePair id="1">
    <article lang="nl" name="Les_Fleurs_du_mal">
      <categories name="Dichtbundel_Franse_literatuur"/>
      <content>
```

```

    <p>Les Fleurs du mal (De bloemen van het kwaad) is
    de belangrijkste dichtbundel van de Franse dichter
    Charles Baudelaire.</p>
    ...
  </content>
</article>
<article lang="ro" name="Florile_raului">
  <categories name="Carti_aparute_in_1857"/>
  <content>
    <p>Florile raului este o culegere de poezii ale
    poetului francez Charles Baudelaire.</p>
    ...
  </content>
</article>
</articlePair>
...
</wikipediaComparable>

```

5.1.3. Primjene

Ovakva usporediva tekstna zbirka je jako vrijedan resurs za obradu prirodnog jezika. Neke od mogućih primjena su izgradnja dvojezičnih rječnika, izgradnja usporednih rečeničnih parova, pomaganje prevoditeljima u pronalasku dvojezične terminologije i sl. U nastavku će biti opisano na koji način se ova zbirka može koristiti kod nenadziranog generiranja dvojezičnih rječnika temeljenog na semantičkim vektorskim prostorima. Konkretno, parovi članaka će biti korišteni za generiranje tzv. pseudodokumenata, koji će također biti definirani u daljnjem tekstu.

5.1.4. Izrada usporedive hrvatsko-engleske tekstne zbirke

Za izradu usporedive hrvatsko-engleske tekstne zbirke korišten je Wikidata API (*engl. Application Programming Interface*). Wikidata je besplatna kolaborativno-uređivana višezjezična baza podataka koja može biti mijenjana od strane ljudi i strojeva. Ona ima ulogu centralnog repozitorija struktuiranih podataka svojih Wikimedia sestrinskih projekata Wikipedia, Wikivoyage, Wikisource i drugih.

Wikidata je centralna pohrana podataka kojoj pristupaju klijentske aplikacije povezane na repozitorij. Centralizirani pristup je pogodan za održavanje sadržaja jer se ne moraju održavati posebne verzije za svako Wikipedija izdanje. Osim toga, Wikidata je centralizirala sve Wikipedija međujezične poveznice.

Wikidata repozitorij se sastoji uglavnom od stavki (*engl. items*), od kojih svaka ima oznaku, opis i jedan ili više aliasa. Poveznice na stranice spajaju klijentska izdanja, a izjave (*engl. statements*) detaljno opisuju svaku stavku. Svaka izjava sastoji se od osobine i vrijednosti, tako da stavke koje se odnose na ljude se mogu povezati s njihovim mjestom rođenja, zanimanjem i slično. Sve ove informacije mogu se koristiti u bilo kojem jeziku za prikaz, iako su sve uzete iz drugog jezika. Na taj način, pristupajući ovim podacima klijentska aplikacija će uvijek imati najsvježije informacije (wikidata.org, 2015).

Hrvatsko-engleska usporediva tekstna zbirka izgrađena je tako da je najprije preuzeta indeks datoteka s izvatkom (*engl. dump file*) hrvatske Wikipedije koja sadrži naslove svih članaka hrvatskog izdanja. Zatim se uz pomoć Wikidata API poziva dohvaćao tekst odgovarajućeg članka na hrvatskom jeziku i odgovarajućeg članka na engleskom. Sve zajedno je ispisivano u XML datoteku u prethodno opisanom Wikipedia Comparable formatu.

Skupovi za treniranje

Stvaranje skupova za treniranje podrazumijevalo je predobradu dostupnih tekstnih zbirki u kojoj je najprije vršeno uklanjanje html oznaka, zatim tokenizacija, označavanje vrsta riječi, lematizacija, te ispisivanje rezultata u obliku pseudodokumenta. Za označavanje vrsta riječi korišten je TreeTagger (Schmid, 1995).

Označavanje vrsta riječi provedeno je jer je odlučeno da će u pseudodokumente ulaziti samo imenice, a lematizacija zbog prirode problema u kojoj je cilj samo pronaći najbolji mogući prijevod za osnovni oblik riječi i dodati unos u rječnik.

Kao rezultat predobrade stvarane su datoteke pseudodokumenta (nasumični ili izmiješani slijedni raspored riječi), datoteke obrađenih tekstova članaka koje mogu poslužiti za učenje mapiranja Mikolovljevog modela te rječnici koje sadrže lematizirane oblike svih riječi za oba jezika i broj njihovih pojavljivanja.

Skupovi za testiranje

Testiranje performansi modela je rađeno za svaki par jezika koristeći skupove za testiranje od po 1000 parova riječi izvornog jezika i njihovih prijevoda na ciljnj jezik.

Skupovi za testiranje stvarani su automatski koristeći rječnike dobivene kao rezultat predobrade i servis Google translate. Za svaki par jezika napravljen je prijevod nekoliko puta većeg broja riječi od željene veličine skupa za testiranje, a zatim je nasumično odabrano po 1000 parova riječi i prijevoda i oni su postajali skup za testiranje.

5.2. Rezultati vrednovanja

Za vrednovanje svih modela korištena je evaluacijska mjera preciznost na n (*engl. precision at n - $Prec@n$*). Ova mjera se računa tako da se odredi broj riječi izvornog jezika iz skupa za testiranje čija lista od prvih n najvjerojatnijih prijevoda sadrži riječ iz ciljnog jezika koja je njezin točan prijevod, a potom se taj broj podijeli s ukupnim brojem parova riječi iz skupa za testiranje.

5.2.1. Englesko-talijanski

Eksperimenti koji nad talijansko-engleskom tekstnom zbirkom provedeni su nad 100000 parova članaka s Wikipedije koji su upareni u Wikipedia Comparable XML datoteci. U procesu predobrade izdvajane su samo imenice koje se pojavljuju 5 ili više puta.

Osnovni nenadzirani model

Hiperparametri d i cs označavaju veličinu naučenih vektora i veličinu prozora, respektivno. Najbolji rezultati su podebljani.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$d=50, cs=5$	0.264	0.366	0.414	0.348	0.456	0.495
$d=50, cs=16$	0.297	0.366	0.408	0.318	0.435	0.495
$d=50, cs=48$	0.312	0.441	0.495	0.339	0.495	0.540
$d=100, cs=5$	0.300	0.393	0.429	0.330	0.435	0.471
$d=100, cs=16$	0.393	0.486	0.504	0.387	0.513	0.552
$d=100, cs=48$	0.420	0.525	0.561	0.468	0.567	0.594
$d=150, cs=5$	0.297	0.381	0.411	0.324	0.429	0.471
$d=150, cs=16$	0.372	0.474	0.519	0.426	0.522	0.561
$d=150, cs=48$	0.465	0.555	0.585	0.486	0.567	0.621
$d=200, cs=5$	0.258	0.366	0.384	0.294	0.420	0.438
$d=200, cs=16$	0.396	0.498	0.516	0.417	0.522	0.552
$d=200, cs=48$	0.462	0.558	0.588	0.495	0.579	0.630
$d=250, cs=5$	0.243	0.336	0.372	0.273	0.381	0.429
$d=250, cs=16$	0.372	0.465	0.507	0.408	0.525	0.567
$d=250, cs=48$	0.477	0.555	0.597	0.501	0.591	0.621
$d=300, cs=5$	0.225	0.321	0.348	0.264	0.375	0.414
$d=300, cs=16$	0.375	0.462	0.498	0.402	0.495	0.537
$d=300, cs=48$	0.477	0.552	0.597	0.516	0.582	0.624

Tablica 5.1: Rezultati testiranja nenadziranih modela treniranih skip-gram modelom

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
d=50, cs=5	0.246	0.378	0.420	0.315	0.438	0.486
d=50, cs=16	0.318	0.447	0.507	0.306	0.438	0.495
d=50, cs=48	0.369	0.495	0.558	0.327	0.477	0.519
d=100, cs=5	0.285	0.402	0.456	0.339	0.465	0.519
d=100, cs=16	0.375	0.486	0.540	0.330	0.480	0.543
d=100, cs=48	0.447	0.561	0.594	0.375	0.552	0.597
d=150, cs=5	0.282	0.396	0.447	0.321	0.480	0.528
d=150, cs=16	0.381	0.498	0.543	0.357	0.519	0.546
d=150, cs=48	0.435	0.543	0.582	0.408	0.546	0.597
d=200, cs=5	0.261	0.396	0.447	0.348	0.465	0.513
d=200, cs=16	0.390	0.510	0.552	0.357	0.501	0.546
d=200, cs=48	0.456	0.564	0.597	0.444	0.573	0.618
d=250, cs=5	0.267	0.390	0.441	0.294	0.462	0.516
d=250, cs=16	0.381	0.495	0.528	0.372	0.522	0.564
d=250, cs=48	0.447	0.561	0.591	0.441	0.597	0.636
d=300, cs=5	0.261	0.393	0.435	0.318	0.453	0.504
d=300, cs=16	0.369	0.486	0.543	0.366	0.510	0.582
d=300, cs=48	0.456	0.540	0.594	0.447	0.576	0.603

Tablica 5.2: Rezultati testiranja nenadziranih modela treniranih modelom CBOW

Iterativna izgradnja osnovnog modela

Hiperparametar n označava broj najčešćih riječi za koje se umeće jedan od k najvjerojatnijih prijevoda.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, k=0$	0.276	0.366	0.414	0.312	0.459	0.495
$n=100, k=5$	0.225	0.357	0.381	0.303	0.411	0.471
$n=100, k=10$	0.234	0.324	0.363	0.297	0.417	0.489
$n=1000, k=5$	0.150	0.273	0.330	0.234	0.399	0.453
$n=1000, k=10$	0.147	0.255	0.312	0.252	0.363	0.441

Tablica 5.3: Rezultati testiranja iterativne izgradnje nenadziranog modela treniranog pomoću modela skip-gram uz $d=50, cs=5$

Kombinacija Mikolovljevog mapiranja i nenadziranog modela miješanja riječi

Hiperparametar n označava broj najčešćih riječi između kojih se biraju one koje imaju sličnost s izvornom riječi veću od p .

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, p=0$	0.279	0.381	0.411	0.336	0.447	0.504
$n=1000, p=0.7$	0.0	0.018	0.021	0.009	0.018	0.027
$n=1000, p=0.8$	0.009	0.039	0.054	0.027	0.082	0.097
$n=1000, p=0.9$	0.006	0.012	0.012	0.021	0.042	0.051
$n=5000, p=0.7$	0.0	0.009	0.015	0.009	0.015	0.039
$n=5000, p=0.8$	0.009	0.027	0.033	0.027	0.076	0.109
$n=5000, p=0.9$	0.109	0.198	0.253	0.192	0.289	0.359
$n=7000, p=0.7$	0.0	0.009	0.015	0.006	0.015	0.036
$n=7000, p=0.8$	0.006	0.021	0.027	0.030	0.070	0.094
$n=7000, p=0.9$	0.091	0.149	0.201	0.210	0.314	0.368

Tablica 5.4: Rezultati testiranja kombinacije modela Mikolova i nenadziranog modela. Sve trenirano modelom skip-gram uz $d=50, cs=5$

5.2.2. Englesko-španjolski

Testiranje je vršeno na 100000 parova englesko-španjolskih Wikipedija članaka.

Osnovni nenadzirani model

Hiperparametri d i cs označavaju veličinu naučenih vektora i veličinu prozora, respektivno. Najbolji rezultati su podebljani.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$d=50, cs=5$	0.331	0.434	0.479	0.385	0.488	0.547
$d=50, cs=16$	0.322	0.448	0.484	0.394	0.493	0.569
$d=50, cs=48$	0.358	0.466	0.533	0.403	0.538	0.569
$d=100, cs=5$	0.358	0.443	0.470	0.354	0.470	0.538
$d=100, cs=16$	0.417	0.529	0.565	0.430	0.565	0.636
$d=100, cs=48$	0.461	0.560	0.609	0.506	0.605	0.650
$d=150, cs=5$	0.331	0.403	0.430	0.340	0.484	0.520
$d=150, cs=16$	0.421	0.515	0.542	0.461	0.556	0.623
$d=150, cs=48$	0.497	0.582	0.632	0.551	0.663	0.704
$d=200, cs=5$	0.309	0.390	0.417	0.318	0.439	0.493
$d=200, cs=16$	0.399	0.506	0.556	0.448	0.578	0.645
$d=200, cs=48$	0.502	0.614	0.632	0.573	0.672	0.713
$d=250, cs=5$	0.273	0.363	0.412	0.336	0.430	0.484
$d=250, cs=16$	0.403	0.493	0.511	0.439	0.538	0.600
$d=250, cs=48$	0.515	0.618	0.659	0.578	0.690	0.713
$d=300, cs=5$	0.255	0.363	0.385	0.300	0.421	0.457
$d=300, cs=16$	0.381	0.470	0.506	0.457	0.551	0.605
$d=300, cs=48$	0.524	0.614	0.672	0.573	0.690	0.717

Tablica 5.5: Rezultati testiranja nenadziranih modela treniranih modelom skip-gram

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
d=50, cs=5	0.291	0.430	0.488	0.349	0.488	0.542
d=50, cs=16	0.354	0.475	0.538	0.331	0.488	0.573
d=50, cs=48	0.354	0.506	0.569	0.336	0.524	0.582
d=100, cs=5	0.354	0.443	0.497	0.363	0.533	0.596
d=100, cs=16	0.412	0.533	0.578	0.399	0.551	0.596
d=100, cs=48	0.439	0.565	0.614	0.448	0.578	0.609
d=150, cs=5	0.313	0.479	0.520	0.358	0.520	0.582
d=150, cs=16	0.412	0.551	0.609	0.421	0.605	0.632
d=150, cs=48	0.484	0.582	0.636	0.452	0.596	0.632
d=200, cs=5	0.304	0.470	0.506	0.363	0.520	0.587
d=200, cs=16	0.381	0.538	0.582	0.470	0.609	0.636
d=200, cs=48	0.457	0.614	0.641	0.515	0.627	0.650
d=250, cs=5	0.300	0.457	0.497	0.385	0.524	0.582
d=250, cs=16	0.403	0.560	0.614	0.457	0.596	0.636
d=250, cs=48	0.443	0.609	0.636	0.466	0.641	0.681
d=300, cs=5	0.273	0.448	0.497	0.367	0.515	0.578
d=300, cs=16	0.385	0.542	0.614	0.484	0.609	0.645
d=300, cs=48	0.461	0.587	0.609	0.493	0.650	0.708

Tablica 5.6: Rezultati testiranja nenadziranih modela treniranih modelom CBOW

Iterativna izgradnja osnovnog modela

Hiperparametar n označava broj najčešćih riječi za koje se umeće jedan od k najvjerojatnijih prijevoda.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, k=0$	0.331	0.412	0.475	0.381	0.493	0.529
$n=100, k=5$	0.300	0.358	0.417	0.340	0.475	0.520
$n=100, k=10$	0.264	0.376	0.412	0.381	0.479	0.520
$n=1000, k=5$	0.188	0.340	0.390	0.269	0.421	0.497
$n=1000, k=10$	0.197	0.322	0.372	0.269	0.408	0.470

Tablica 5.7: Rezultati testiranja iterativne izgradnje osnovnog modela treniranog pomoću modela skip-gram uz $d=50$, $cs=5$

Kombinacija Mikolovljevog mapiranja i nenadziranog modela miješanja riječi

Hiperparametar n označava broj najčešćih riječi između kojih se biraju one koje imaju sličnost s izvornom riječi veću od p .

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, p=0$	0.331	0.430	0.488	0.403	0.506	0.533
$n=1000, p=0.7$	0.0	0.009	0.018	0.018	0.049	0.063
$n=1000, p=0.8$	0.0	0.022	0.045	0.045	0.113	0.140
$n=1000, p=0.9$	0.022	0.031	0.036	0.040	0.063	0.067
$n=5000, p=0.7$	0.0	0.027	0.036	0.022	0.049	0.054
$n=5000, p=0.8$	0.018	0.049	0.058	0.049	0.113	0.140
$n=5000, p=0.9$	0.131	0.230	0.271	0.226	0.375	0.420
$n=7000, p=0.7$	0.0	0.031	0.036	0.018	0.049	0.049
$n=7000, p=0.8$	0.009	0.049	0.058	0.049	0.090	0.131
$n=7000, p=0.9$	0.122	0.239	0.280	0.221	0.352	0.411

Tablica 5.8: Rezultati testiranja kombinacije modela Mikolova i nenadziranog modela. Sve trenirano modelom skip-gram uz $d=50$, $cs=5$

5.2.3. Englesko-njemački

Osnovni nenadzirani model

Hiperparametri d i cs označavaju veličinu naučenih vektora i veličinu prozora, respektivno. Najbolji rezultati su podebljani.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$d=50, cs=5$	0.156	0.233	0.282	0.190	0.293	0.336
$d=50, cs=16$	0.141	0.251	0.269	0.169	0.267	0.303
$d=50, cs=48$	0.151	0.244	0.300	0.179	0.295	0.344
$d=100, cs=5$	0.167	0.246	0.298	0.190	0.295	0.329
$d=100, cs=16$	0.205	0.303	0.341	0.197	0.290	0.329
$d=100, cs=48$	0.236	0.311	0.359	0.231	0.344	0.377
$d=150, cs=5$	0.146	0.239	0.275	0.174	0.280	0.311
$d=150, cs=16$	0.205	0.298	0.323	0.208	0.275	0.326
$d=150, cs=48$	0.264	0.329	0.375	0.264	0.362	0.411
$d=200, cs=5$	0.159	0.215	0.259	0.169	0.262	0.298
$d=200, cs=16$	0.190	0.264	0.316	0.197	0.295	0.321
$d=200, cs=48$	0.259	0.336	0.372	0.262	0.362	0.398
$d=250, cs=5$	0.131	0.218	0.241	0.143	0.249	0.287
$d=250, cs=16$	0.203	0.277	0.300	0.179	0.259	0.298
$d=250, cs=48$	0.251	0.347	0.388	0.249	0.367	0.398
$d=300, cs=5$	0.136	0.203	0.228	0.128	0.221	0.267
$d=300, cs=16$	0.192	0.259	0.293	0.179	0.257	0.282
$d=300, cs=48$	0.244	0.347	0.385	0.272	0.365	0.401

Tablica 5.9: Rezultati testiranja nenadziranih modela treniranih modelom skip-gram

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
d=50, cs=5	0.123	0.246	0.275	0.233	0.341	0.398
d=50, cs=16	0.164	0.290	0.334	0.185	0.298	0.354
d=50, cs=48	0.174	0.282	0.336	0.185	0.295	0.352
d=100, cs=5	0.154	0.259	0.298	0.239	0.372	0.408
d=100, cs=16	0.215	0.316	0.347	0.218	0.323	0.385
d=100, cs=48	0.228	0.326	0.390	0.210	0.336	0.383
d=150, cs=5	0.161	0.259	0.313	0.228	0.365	0.398
d=150, cs=16	0.210	0.316	0.385	0.231	0.331	0.403
d=150, cs=48	0.226	0.336	0.411	0.223	0.354	0.398
d=200, cs=5	0.156	0.262	0.318	0.223	0.357	0.398
d=200, cs=16	0.213	0.331	0.372	0.210	0.357	0.406
d=200, cs=48	0.228	0.341	0.398	0.236	0.367	0.426
d=250, cs=5	0.177	0.269	0.295	0.218	0.354	0.406
d=250, cs=16	0.226	0.339	0.377	0.239	0.352	0.408
d=250, cs=48	0.226	0.354	0.406	0.236	0.383	0.411
d=300, cs=5	0.167	0.269	0.311	0.215	0.357	0.401
d=300, cs=16	0.221	0.323	0.372	0.231	0.357	0.413
d=300, cs=48	0.231	0.339	0.408	0.226	0.354	0.416

Tablica 5.10: Rezultati testiranja nenadziranih modela treniranih modelom CBOW

Iterativna izgradnja osnovnog modela

Hiperparametar n označava broj najčešćih riječi za koje se umeće jedan od k najvjerojatnijih prijevoda.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, k=0$	0.159	0.233	0.269	0.182	0.293	0.344
$n=100, k=5$	0.105	0.192	0.215	0.164	0.272	0.308
$n=100, k=10$	0.092	0.172	0.213	0.172	0.254	0.305
$n=1000, k=5$	0.074	0.133	0.174	0.131	0.218	0.269
$n=1000, k=10$	0.079	0.136	0.167	0.125	0.213	0.259

Tablica 5.11: Rezultati testiranja iterativne izgradnje osnovnog modela treniranog pomoću modela skip-gram uz $d=50$, $cs=5$

Kombinacija Mikolovljevog mapiranja i nenadziranog modela miješanja riječi

Hiperparametar n označava broj najčešćih riječi između kojih se biraju one koje imaju sličnost s izvornom riječi veću od p .

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, p=0$	0.146	0.239	0.277	0.172	0.316	0.365
$n=1000, p=0.7$	0.0	0.007	0.010	0.0	0.013	0.020
$n=1000, p=0.8$	0.005	0.010	0.013	0.002	0.026	0.031
$n=1000, p=0.9$	0.013	0.020	0.026	0.013	0.028	0.041
$n=5000, p=0.7$	0.002	0.010	0.010	0.002	0.010	0.013
$n=5000, p=0.8$	0.005	0.010	0.010	0.002	0.013	0.013
$n=5000, p=0.9$	0.039	0.057	0.067	0.088	0.145	0.177
$n=7000, p=0.7$	0.0	0.005	0.010	0.002	0.007	0.013
$n=7000, p=0.8$	0.0	0.007	0.010	0.002	0.010	0.013
$n=7000, p=0.9$	0.026	0.044	0.057	0.054	0.104	0.140

Tablica 5.12: Rezultati testiranja kombinacije modela Mikolova i nenadziranog modela. Sve trenirano modelom skip-gram uz $d=50$, $cs=5$

5.2.4. Englesko-francuski

Kao i kod prethodnih jezičnih parova, testiranje je također vršeno na 100000 parova članaka s Wikipedije i korištene su samo imenice koje se pojavljuju 5 ili više puta u tekstnoj zbirci.

Osnovni nenadzirani model

Hiperparametri d i cs označavaju veličinu naučenih vektora i veličinu prozora, respektivno. Najbolji rezultati su podebljani.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$d=50, cs=5$	0.204	0.336	0.351	0.292	0.419	0.458
$d=50, cs=16$	0.239	0.370	0.4	0.282	0.409	0.443
$d=50, cs=48$	0.224	0.360	0.419	0.297	0.414	0.434
$d=100, cs=5$	0.229	0.341	0.365	0.287	0.414	0.453
$d=100, cs=16$	0.307	0.419	0.434	0.356	0.458	0.487
$d=100, cs=48$	0.341	0.429	0.458	0.390	0.482	0.531
$d=150, cs=5$	0.243	0.336	0.375	0.292	0.419	0.443
$d=150, cs=16$	0.317	0.404	0.419	0.360	0.448	0.502
$d=150, cs=48$	0.365	0.453	0.497	0.390	0.492	0.526
$d=200, cs=5$	0.214	0.307	0.341	0.282	0.409	0.439
$d=200, cs=16$	0.326	0.390	0.404	0.336	0.443	0.502
$d=200, cs=48$	0.385	0.482	0.512	0.409	0.487	0.551
$d=250, cs=5$	0.219	0.307	0.346	0.278	0.390	0.429
$d=250, cs=16$	0.292	0.365	0.419	0.351	0.434	0.473
$d=250, cs=48$	0.395	0.478	0.512	0.409	0.502	0.546
$d=300, cs=5$	0.195	0.292	0.331	0.273	0.380	0.424
$d=300, cs=16$	0.312	0.380	0.409	0.356	0.458	0.492
$d=300, cs=48$	0.375	0.463	0.492	0.419	0.512	0.541

Tablica 5.13: Rezultati testiranja nenadziranih modela treniranih modelom skip-gram

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
d=50, cs=5	0.195	0.351	0.395	0.287	0.439	0.478
d=50, cs=16	0.278	0.395	0.429	0.278	0.424	0.478
d=50, cs=48	0.307	0.395	0.443	0.273	0.4	0.448
d=100, cs=5	0.268	0.380	0.424	0.287	0.458	0.487
d=100, cs=16	0.341	0.429	0.453	0.326	0.478	0.512
d=100, cs=48	0.336	0.439	0.468	0.351	0.468	0.497
d=150, cs=5	0.278	0.375	0.424	0.292	0.448	0.512
d=150, cs=16	0.351	0.434	0.458	0.317	0.478	0.526
d=150, cs=48	0.346	0.458	0.497	0.341	0.453	0.512
d=200, cs=5	0.258	0.375	0.429	0.321	0.448	0.502
d=200, cs=16	0.360	0.434	0.478	0.370	0.478	0.526
d=200, cs=48	0.390	0.458	0.487	0.395	0.478	0.536
d=250, cs=5	0.282	0.390	0.414	0.331	0.453	0.517
d=250, cs=16	0.375	0.443	0.458	0.380	0.512	0.536
d=250, cs=48	0.380	0.492	0.531	0.360	0.492	0.531
d=300, cs=5	0.248	0.385	0.443	0.336	0.458	0.507
d=300, cs=16	0.351	0.429	0.473	0.365	0.473	0.512
d=300, cs=48	0.390	0.482	0.526	0.370	0.531	0.570

Tablica 5.14: Rezultati testiranja nenadziranih modela treniranih modelom CBOW

Iterativna izgradnja osnovnog modela

Hiperparametar n označava broj najčešćih riječi za koje se umeće jedan od k najvjerojatnijih prijevoda.

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, k=0$	0.190	0.346	0.375	0.321	0.434	0.468
$n=100, k=5$	0.185	0.273	0.307	0.273	0.390	0.453
$n=100, k=10$	0.180	0.273	0.336	0.282	0.385	0.448
$n=1000, k=5$	0.136	0.258	0.273	0.219	0.346	0.390
$n=1000, k=10$	0.136	0.229	0.282	0.224	0.307	0.380

Tablica 5.15: Rezultati testiranja iterativne izgradnje osnovnog modela treniranog pomoću modela skip-gram uz $d=50$, $cs=5$

Kombinacija Mikolovljevog mapiranja i nenadziranog modela miješanja riječi

Hiperparametar n označava broj najčešćih riječi između kojih se biraju one koje imaju sličnost s izvornom riječi veću od p .

Model	Osnovna inačica miješanja			Proširena inačica miješanja		
Hiperparametri	Prec@1	Prec@5	Prec@10	Prec@1	Prec@5	Prec@10
$n=0, p=0$	0.209	0.307	0.346	0.312	0.429	0.463
$n=1000, p=0.7$	0.0	0.0	0.0	0.010	0.027	0.070
$n=1000, p=0.8$	0.0	0.010	0.021	0.027	0.086	0.118
$n=1000, p=0.9$	0.0	0.0	0.0	0.0	0.0	0.010
$n=5000, p=0.7$	0.0	0.005	0.010	0.010	0.043	0.064
$n=5000, p=0.8$	0.010	0.021	0.043	0.043	0.108	0.135
$n=5000, p=0.9$	0.054	0.091	0.108	0.102	0.216	0.237
$n=7000, p=0.7$	0.0	0.005	0.010	0.005	0.037	0.048
$n=7000, p=0.8$	0.005	0.021	0.043	0.032	0.086	0.129
$n=7000, p=0.9$	0.070	0.097	0.118	0.118	0.210	0.264

Tablica 5.16: Rezultati testiranja kombinacije modela Mikolova i nenadziranog modela. Sve trenirano modelom skip-gram uz $d=50$, $cs=5$

5.3. Diskusija rezultata

Iz prethodnog dijela vidljivo je da su najbolji rezultati postignuti upotrebom nenadziranog modela iz Vulić i Moens (2015) nad kojim je primijenjena nova inicijalizacija. To podrazumijeva da su pseudodokumenti stvarani tako da su spajana dva dokumenta na različitim jezicima, ali koja imaju istu temu, slijednim naizmjeničnim umetanjem riječi iz oba dokumenta.

Model koji je referentni u ovom slučaju je nenadzirani model s nasumičnom inicijalizacijom (Vulić i Moens, 2015). Pokazano je da u prosječnom slučaju taj model daje lošije rezultate od onog s novom inicijalizacijom. Svi rezultati jako variraju u ovisnosti o vrijednosti odabranih hiperparametara za treniranje, ali i o veličini skupa podataka nad kojim se trenira. Kao što je i očekivano, s obzirom da su dokumenti upareni samo na osnovu teme, povećanjem veličine prozora modela skip-gram i CBOW povećava se preciznost mjerena na skupu za testiranje.

Testirana je i iterativna izgradnja obje navedene varijante nenadziranog modela. Rezultati su pokazali da preciznost opada što se više mijenja sadržaj originalnog pseudodokumenta. Tako da s porastom broja riječi za koje se umeću najvjerojatniji prijevodi (n) i s porastom broja riječi između kojih se bira prijevod koji će biti umetnut (k), preciznost na skupu za testiranje opada. Pretpostavka je da se ovo dešava jer ovakvim načinom iterativne izgradnje se ne unosi nikakva bitna informacija u pseudodokument. Ono što se dešava je tek malena promjena odnosa riječi koje se već nalaze u kontekstu date riječi.

Na kraju rezultata za svaki par jezika navedeni su rezultati tzv. nenadziranog nadziranog učenja, odnosno korištenja izlaza nenadziranog modela (Vulić i Moens, 2015) kao skup za treniranje nadziranog modela (Mikolov et al., 2013b). Hiperparametri u ovom slučaju su bili broj najčešćih riječi iz prvog jezika za koje se određuje najvjerojatniji prijevod (n) i prag sličnosti tih prijevoda (p) koji je određivao koje riječi ulaze u skup za treniranje nadziranog modela. Intuicija iza ovakvog postupka je bila da su se najčešće riječi pojavile u najviše konteksta i da su za njih naučeni najkvalitetniji vektorski prikazi. Međutim, pokazalo se da unatoč veoma visokim performansama ovog nadziranog modela prijavljenim u Mikolov et al. (2013b), u ovom slučaju preciznost je bila manja nego kod osnovnog nenadziranog modela. Objašnjenje za to se može nalaziti u činjenici da parovi za treniranje nadziranog modela jednostavno nisu

bili dovoljno precizni da bi se naučila ispravna linearna transformacija između vektorskih prostora dvaju jezika. Pretpostavka je da bi se uz nešto elaboriraniji način odabira prijevodnih parova za učenje modela Mikolova rezultati mogli popraviti,

5.4. Analiza pogrešaka

U ovom dijelu će biti napravljena osnovna analiza pogrešaka. Svi modeli i postavke svih eksperimenata opisane su u prethodnim poglavljima, a uz te opise bitno je napomenuti još i da je pri testiranju iz ciljnog jezika vršeno uklanjanje svih riječi koje su se nalazile u izvornom. Ovo je rađeno iz razloga što su tekstovi iz tekstne zbirke na jednom jeziku sadržavali brojne riječi iz drugog jezika, a to je bilo posebno naglašeno kod brojnih pojava engleskih riječi u ostalim jezicima. Tako da su takve riječi imale maksimalni iznos mjere sličnosti, a zapravo nisu bile ispravan prijevod. Spomenuto uklanjanje riječi izvornog jezika iz ciljnog posebno je utjecalo na slučaje kada su se prevodili imenovani entiteti. Tako je engleska riječ *Serbia*, koja bi trebala imati identičan oblik i na španjolskom jeziku, bila prevođena sa španjolskom riječi *Croacia*.

U općem slučaju prijevodi su bili semantički slične riječi koje su se pojavile mnogo puta u istom kontekstu s izvornom riječi. Bitno je napomenuti da je treniranje vršeno na automatski i nasumično odabranim parovima članaka s Wikipedije, a testiranje je također obavljano nad automatski generiranim rječnikom. Moguće je da su bolji rezultati mogli biti ostvareni da se koristio pogodniji skup za treniranje, odnosno da su po nekim kriterijima odabirani parovi članaka nad kojim bi se učili vektorski prikazi riječi, ali to bi učinilo da se rad dodatno odmakne od svog osnovnog cilja, a to je bilo potpuno automatizirano stvaranje dvojezičnih rječnika.

6. Zaključak

Tradicionalni postupci za automatsku izgradnju prijevodnih rječnika se oslanjaju na usporedne dvojezične tekstne zbirke. Kako je izgradnja usporednih tekstnih zbirki iznimno naporan i skup postupak, noviji postupci automatske izgradnje prijevodnih rječnika oslanjaju se na usporedive tekstne zbirke u kojima je uparivanje između dvaju jezika načinjeno tek na razini dokumenta. Izgradnja usporedivih tekstnih zbirki značajno je manje zahtjevna od izgradnje usporednih, a razvijeni su i pouzdani postupci za automatsku izgradnju takvih tekstnih zbirki.

U okviru ovog rada proučeni su postupci za automatsku izgradnju prijevodnih rječnika temeljeni na usporedivim tekstnim zbirkama. Proučeni su modeli koji se temelje na semantičkom prikazu riječi u vektorskom prostoru obaju jezika. Razrađen je postupak za automatsku izgradnju prijevodnih rječnika za koji nije potrebno imati ručno pripremljene prijevodne parove riječi.

Isprobano je nekoliko novih metoda kojima se pokušalo doći do pobošljanja rezultata osnovnog nenadziranog modela. Te metode su opisane u poglavlju koje govori o izgradnji prijevodnih rječnika, a detaljni rezultati i opis postavke svih provedenih eksperimenata su navedeni u poglavlju koje govori o eksperimentima. Radu su priloženi izvorni kod i skupovi podataka.

Opisane metode predstavljaju još jedan pokušaj da se primjenom računala olakšaju zadaci koje inače obavljaju ljudski stručnjaci iz tog područja. Cilj je bilo automatizirano stvaranje dvojezičnih rječnika koje će uz što se poslije, pod uvjetom da su stvoreni rječnici dovoljno precizni, može koristiti u raznim drugim zadacima. To je u nekoj mjeri i ostvareno - rječnici se generiraju na nenadzirani način, a s određenim vrijednostima hiperparametara ostvaruje se i prilično visoka preciznost. Međutim, treba imati na umu da su takvi rezultati ostvareni na skupu koji je sadržavao samo imenice, i to njihove lematizirane oblike koji su dobiveni koristeći TreeTagger (Schmid, 1995),

koji je treniran na nadzirani način i zahtijeva označeni skup za učenje.

U prethodnim poglavljima je na raznim mjestima predloženo na koji način bi se prikazani rezultati mogli popraviti. Tako da je kod učenja nadziranog modela Mikolova izlazom iz nenadziranog modela vjerojatno moguće napraviti nešto elaboriraniji način odabira prijevodnih parova za učenje koji će dati nešto bolju preciznost na skupu za testiranje. Također, moguće je primijeniti ovakav nenadzirani model nad tekstnom zbirkom koja je uparena na razini rečenica.

Za značajnija poboljšanja performansi na ovom zadatku bit će potrebno na neki način u novije modele ugraditi razumijevanje konteksta riječi i njezine funkcije unutar njega, odnosno ne posmatrati samo leksički kontekst unutar prozora neke veličine, nego također vršiti i sintaksnu analizu.

LITERATURA

- Marco Baroni, Georgiana Dinu, i Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. U *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, svezak 1, stranice 238–247, 2014.
- Bojana Dalbelo Bašić, Marko Čupić, i Jan Šnajder. Umjetne neuronske mreže. *Fakultet elektrotehnike i računarstva*, 2008.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, i Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.
- Lynne Bowker i Jennifer Pearson. *Working with specialized language: a practical guide to using corpora*. Routledge, 2002.
- Elia Bruni, Giang Binh Tran, i Marco Baroni. Distributional semantics from text and images. U *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, stranice 22–32. Association for Computational Linguistics, 2011.
- Ronan Collobert i Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. U *Proceedings of the 25th international conference on Machine learning*, stranice 160–167. ACM, 2008.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, i Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.
- Marko Čupić, Bojana Dalbelo Bašić, i Marin Golub. *Neizrazito, evolucijsko i neuro-računarstvo*. Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2013.
- Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

- Yoav Goldberg i Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
- Stephan Gouws, Yoshua Bengio, i Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. *arXiv preprint arXiv:1410.2455*, 2014.
- Zellig S Harris. Distributional structure. *Word*, 1954.
- Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- Alexandre Klementiev, Ivan Titov, i Binod Bhattarai. Inducing crosslingual distributed representations of words. 2012.
- Tomáš Kočiský, Karl Moritz Hermann, i Phil Blunsom. Learning bilingual word representations by marginalizing alignments. *arXiv preprint arXiv:1405.0947*, 2014.
- Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, i Amrita Saha. An autoencoder approach to learning bilingual word representations. U *Advances in Neural Information Processing Systems*, stranice 1853–1861, 2014.
- Bo Leuf i Ward Cunningham. The wiki way: collaboration and sharing on the internet. 2001.
- linguatools.org. Wikipedia comparable corpora, 2014.
URL <http://linguatools.org/tools/corpora/wikipedia-comparable-corpora/>.
- James L McClelland, David E Rumelhart, PDP Research Group, et al. Parallel distributed processing. *Explorations in the microstructure of cognition*, 2:216–271, 1986.
- Tomas Mikolov, Kai Chen, Greg Corrado, i Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Tomas Mikolov, Quoc V Le, i Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.
- Andriy Mnih i Geoffrey Hinton. Three new graphical models for statistical language modelling. U *Proceedings of the 24th international conference on Machine learning*, stranice 641–648. ACM, 2007.

- Andrew Ng, Jiquan Ngiam, Chuan Y Foo, Yifan Mai, i Caroline Suen. Ufdl tutorial, 2012.
- David E Rumelhart, Geoffrey E Hinton, i Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5:3, 1988.
- Helmut Schmid. Treetaggerl a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43:28, 1995.
- Rahma Sellami, Fatiha Sadat, i Lamia Hadrich Belguith. Exploiting wikipedia as a knowledge base for the extraction of linguistic resources: Application on arabic-french comparable corpora and bilingual lexicons. U *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*, stranica 72, 2012.
- Richard S Sutton i Andrew G Barto. *Introduction to reinforcement learning*. MIT Press, 1998.
- Ivan Vulić i Marie-Francine Moens. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. *Proceedings of 53rd Annual Meeting of Association for Computational Linguistics. Beijing, China. ACL*, 2015.
- wikidata.org. Wikidata introduction, 2015. URL <https://www.wikidata.org/wiki/Wikidata:Introduction/>.
- wikipedia.org. Wikipedia - the free encyclopedia, 2015. URL <http://en.wikipedia.org/w/index.php?title=Wikipedia&oldid=665164226>.
- Will Y Zou, Richard Socher, Daniel M Cer, i Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. U *EMNLP*, stranice 1393–1398, 2013.

Automatska izgradnja prijevodnih rječnika temeljena na semantičkim vektorskim prostorima

Sažetak

Rječnici i prijevodne tablice izraza osnova su modernih sustava za statističko strojno prevođenje. Tradicionalni postupci za automatsku izgradnju prijevodnih rječnika oslanjaju se na usporedne dvojezične tekstne zbirke. Kako je izgradnja usporednih tekstnih zbirki iznimno naporan i skup postupak, noviji se postupci automatske izgradnje prijevodnih rječnika oslanjaju na usporedive zbirke u kojima je uparivanje između dvaju jezika načinjeno tek na razini dokumenta. Izgradnja usporedivih zbirki značajno je manje zahtjevna od izgradnje usporednih, a razvijeni su i pouzdani postupci za automatsku izgradnju. U okviru ovog diplomskog rada proučeni su postupci za automatsku izgradnju prijevodnih rječnika temeljeni na usporedivim tekstnim zbirkama.

Ključne riječi: dvojezični, rječnici, automatska, izgradnja, usporedivi, korpus, nenadzirano

Automatic generation of bilingual dictionaries based on semantic vector spaces

Abstract

Dictionaries and phrase tables are the basis of modern statistical machine translation systems. Traditional methods for automatic generation of bilingual dictionaries depend on parallel bilingual corpora. Since parallel corpora is hard to acquire, newer methods only require comparable corpora in order to work. This paper develops a method that can automate the process of generating and extending dictionaries and phrase tables in an unsupervised way from comparable corpora.

Keywords: bilingual, dictionaries, automatic, generation, comparable, corpora, unsupervised