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

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4693

Primjena modela distribucijske semantike u igri asocijacija riječi

Mihael Nikić

Zagreb, lipanj 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 11. ožujka 2016.

ZAVRŠNI ZADATAK br. 4693

Pristupnik: **Mihael Nikić (0036480506)**
Studij: Računarstvo
Modul: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Primjena modela distribucijske semantike u igri asocijacija riječi**

Opis zadatka:

Računalna leksička semantika bavi se prikazom značenja riječi te ima važnu ulogu u sustavima za obradu i razumijevanje prirodnoga jezika. Distribucijski semantički modeli značenje riječi prikazuju visokodimenzijskim kontekstnim vektorima, ekstrahiranima na temelju supojavljivanja riječi u korpusu. Takvi su se modeli pokazali vrlo korisnima na nizu zadataka obrade prirodnoga jezika.
Tema završnoga rada jest primjena modela distribucijske semantike u igri asocijacija riječi. U toj je igri potrebno pogoditi zadani ciljni pojam na temelju njegove asocijativne povezanosti s drugim ciljnim pojmovima, i to u što manje koraka. U okviru završnoga rada potrebno je proučiti modele distribucijske semantike, kao i statističke pristupe za modeliranje leksičke povezanosti između riječi. Izgraditi nekoliko modela distribucijske semantike za hrvatski jezik, koristeći postojeće korpuse. Razraditi i implementirati postupak za generiranje igre asocijacije riječi, kao i njemu odgovarajući postupak za rješavanje igre. Izgraditi prikladan ispitni skup podataka te provesti eksperimentalno vrednovanje postupka. Radu priložiti izvorni i izvršni kod razvijenog sustava, skupove podataka i programsku dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 18. ožujka 2016.
Rok za predaju rada: 17. lipnja 2016.

Mentor:

Doc. dr. sc. Jan Šnajder

Djelovođa:

Doc. dr. sc. Ivica Botički

Predsjednik odbora za završni rad modula:

Prof. dr. sc. Krešimir Fertalj

SADRŽAJ

1. Uvod

Tijekom 80-ih godina prošlog stoljeća, na Radioteleviziji Zagreb (današnjoj Hrvatskoj radioteleviziji) emitirao se popularni kviz općeg znanja pod imenom *Kviskoteka*. Kviz bio je organiziran u više igara, koje su se s vremenom izmjenjivale. Neke od igara su bile : *ABCD-pitalice*, *Bliski susreti*, *Igra asocijacija*, *Igra detekcije* i dr. *Igra asocijacija* je upravo tema ovog rada. Cilj igre je pogoditi riječ na temelju četiri riječi koje asociraju na nju. Nakon što se taj proces ponovi četiri puta, peta riječ se pogađa na temelju tih prethodno pogođenih četiriju riječi.

Za tu igru smo izgradili jedan model koji ju generira. Izgradnja modela je ostvarena uz pomoć obrade prirodnog jezika (engl. *Natural language processing*) – multidisciplinarnog područja istraživanja i razvoja čiji je cilj produkcija aplikacija za svakodnevni život poput aplikacija namijenjenih za pretraživanje informacija, inteligentnog pretraživanja interneta, strojnog sažimanja teksta i dr. te uz pomoć tehnike često korištene u teoriji informacija – točkaste procjene uzajamne informacije (engl. *Pointwise mutual information*).

Obrada prirodnog jezika je hijerarhijski podijeljena, a jedna od grana koja nas posebno zanima je semantika – grana koja se bavi značenjem riječi i kako ta značenja složena u rečenicu tvore smisao rečenice. Razlog zbog kojeg nam je ta grana interesantna je taj što ona nudi distribucijsku[1] tehniku analize srodstva kolekcije dokumenata i pojmova koji su sadržani u tim dokumentima stvaranjem skupa koncepata povezanih s dokumentima i pojmovima, poznatu pod nazivom latentna semantička analiza (engl. *latent semantic analysis*). Za ostvarivanje cilja izgradnje modela koristila nam je i leksičko-semantička mreža *WordNet*, iz koje smo izvadili ono što nam je potrebno – odnose višeg i nižeg pojma, odnosno hiperonima i hiponima.

Tehnikom izračuna točkaste procjene uzajamne informacije računamo mjeru zajedničkog pojavljivanja dva događaja (u našem slučaju susjednih riječi), x i y; te na

[1] Distribucijska semantika je područje istraživanja koja razvija i proučava teorije i metode za kvantificiranje i kategoriziranje semantičke sličnosti između jezičnih stavki na temelju distribucijskih svojstava.

temelju izračunate mjere otkrivamo sličnost između tih riječi.

Ostatak rada strukturiran je na sljedeći način. Drugo poglavlje daje pregled opisa problema i sličnih radova. Treće poglavlje daje pregled teorijskog dijela rada, u kojem se opisuju postupci za određivanje sličnosti među riječima. U četvrtom poglavlju opisana je primjena tih postupaka za gradnju modela koji generira primjerak igre asocijacije riječi. Peto poglavlje daje kratak pregled strukture programskog rješenja i detaljan opise baze podataka, koja se koristi za pohranu rezultatu različitih postupaka. U šestom poglavlju opisani su svi provedeni eksperimenti. Zaključak rada je u sedmom poglavlju.

2. Analiza problema

2.1. Opis problema

Računalna leksička semantika bavi se prikazom značenja riječi te ima važnu ulogu u sustavima za obradu i razumijevanje prirodnog jezika. Distribucijski semantički modeli značenje riječi prikazuju visokodimenzijskim kontekstnim vektorima, ekstrahiranima na temelju supojavljivanja riječi u korpusu. Takvi su se modeli pokazali vrlo korisnima na nizu zadataka obrade prirodnog jezika. U okviru ovog rada izgrađen je model koji generira primjerke igre asocijacije. Primjerak jedne takve igre prikazan je tablicom 2.1. U poglavlju 3 opisani su postupci namijenjeni za izgradnju jednog

uzeti	kuća	zrinjevac	jezgra
platiti	socijalan	uvlačenje	zaprešić
preglednik	rađen	postupan	knjižnica
copy	pet	nogomet	aleksandar
taksi	stanovi	dinamo	grad
zagreb			

Tablica 2.1: Primjerak jedne izgrađene igre asocijacija riječi

takvog modela, poput distribucijskih semantičkih postupaka, postupaka temeljenih na bazama znanja te također i statističkih postupaka za modeliranje leksičke povezanosti između riječi, dok su u poglavlju 4 opisane primjene tih postupaka za izgradnju modela. Nad izgrađenim skupom podataka provedeno je eksperimentalno vrednovanje postupaka.

2.2. Slični radovi

Jedan od primjera radova je nizozemska baza podataka sa asocijativnim riječima (engl. *Dutch Word Association Database*), koja je napravljena u sklopu projekta *Dutch Word Asso-*

ciations[1] koji je dio istraživanja semantičkih mreža. Trenutno je u bazi pohranjeno 8995 pojmova i preko 100 000 različitih asocijacija. U ovom projektu je za pronalaženje sličnosti među riječima korištena tehnika točkaste procjene uzajamne informacije, koja će se također koristiti u ovom radu iz istog razloga. Osim navedenog primjera postoji još nekolicina srodnih radova koji se pretežito fokusiraju na semantičku sličnost riječi općenito.

Jedan od tih radova je *Modeling Information Scent: A Comparison of LSA, PMI and GLSA Similarity Measures on Common Tests and Corpora* (Budiu i dr. , 2007), koji opisuje tri različite tehnike za procjenu sličnosti između riječi: latentnu semantičku analizu (Landauer i Dumais, 1997), točkastu procjenu uzajamne informacije (Turney, 2001) i generaliziranu latentnu semantičku analizu (Matveeva i dr. , 2005). Za usporedbu svih triju tehnika koristi se jedinstveni korpus (TASA) te je za točkastu procjenu uzajamne informacije i generaliziranu latentnu semantičku analizu priložen izvještaj evaluacije nad većim web-korpusom.

Još jedan sličan rad je *Exploring ESA to Improve Word Relatedness* (Aggarwal i dr. , 2014), koji opisuje tehniku eksplicitne semantičke analize (engl. *Explicit Semantic Analysis – ESA*) čija je svrha izračunati semantičku povezanost između dvije riječi ili tekstova napisanih prirodnim jezikom uz pomoć koncepata temeljenih na ljudskim spoznajama.

[1] `http://www.kuleuven.be/semlab/interface/index.php`

3. Postupci za izgradnju modela

3.1. Točkasta procjena uzajamne informacije

Točkasta procjena uzajamne informacije (engl. *Pointwise mutual information*) (Budiu i dr. , 2007) između dvije riječi A i B bilježi kolika je vjerojatnost pronalaska riječi B u danom tekstu za koji je poznato da sadrži riječ A. To je mjera zajedničkog pojavljivanja, utoliko da normalizira vjerojatnost zajedničkog pojavljivanja dviju riječi s njihovim individualnim vjerojatnostima pojavljivanja. Točkastu procjenu uzajamne informacije između riječi A i B računamo prema sljedećoj formuli:

$$PMI(A,B) = log\frac{p(A,B)}{p(A)p(B)} = log\frac{C(A,B) \times N}{C(A)C(B)} \tag{3.1}$$

gdje je $P(A,B)$ vjerojatnost da će se riječi A i B zajedno pojaviti u istom dokumentu; $p(X)$ vjerojatnost pojavljivanja riječi X; $C(A,B)$ je broj dokumenata u kojima se riječi A i B zajedno pojavljuju; $C(X)$ je broj dokumenata u kojem se riječ X pojavljuje, i N je broj dokumenata korpusa.

3.1.1. Primjer korištenja

Jedan od dobrih primjera korištenja ove tehnike može se pronaći na Wikipediji.[1] Sljedeća tablica prikazuje parove riječi koji su ostvarili najbolji rezultat. U obzir se uzimaju prvih 50 milijuna riječi engleske Wikipedije, pri čemu se zanemaruju parovi čija je vrijednost zajedničkog pojavljivanja manja od 1000. Prvih nekoliko najbolje ostvarenih rezultata, koji su dobiveni primjenom ove tehnike, prikazani su u tablici 3.1.

[1] `https://en.wikipedia.org/wiki/Pointwise_mutual_information`

A	B	C(A)	C(B)	C(A,B)	P(A,B)
puerto	rico	1938	1311	1159	10.0349081703
hong	kong	2438	2694	2205	9.72831972408
los	angles	3501	2808	2791	9.560677615065
carbon	dioxide	4265	1353	1032	9.09852946116
prize	laureate	5131	1676	1210	8.85870710982
san	fransicso	5237	2477	1779	8.83305176711
nobel	prize	4098	5131	2498	8.68948811416
ice	hockey	5607	3002	1933	8.6555759741
star	trek	8264	1594	1489	8.639746776575
car	driver	5578	2749	1384	8.41470768304
it	the	283891	3293296	3347	-1.72037278119
are	of	1938	1311	1159	-2.09254205335
this	the	199882	3293296	1211	-2.38612756961

Tablica 3.1: Prvih nekoliko najsličnijih riječi na Wikipediji

3.2. Postupci temeljeni na bazama znanja – WordNet

WordNet (Fellbaum, 2010) je računalno pohranjena leksičko-semantička mreža koja je razvijena za engleski jezik na na Sveučilištu *Princeton* u Sjedinjenim Državama tijekom ’90. Prinstonski WordNet je zapravo računalni leksikon u kojem riječi nisu poredane abecedno, već su prema svojim značenjima podijeljene u skupove sinonima. Svaki sinonimski skup (engl. *Synset*) sadrži jednu ili više riječi za koje se smatra da se međusobno mogu zamijeniti u barem jednom kontekstu. Svaki se sinonimski skup nalazi u barem još jednom semantičkom odnosu s nekim drugim sinonimskim skupom. Sinonimski se skupovi u prinstonskom WordNetu sastoje od iste vrste riječi, od kojih se u tom leksikonu obrađuju imenice, glagoli, pridjevi i prilozi. Određena se riječ, s obzirom na svoju potencijalnu višeznačnost, može pojaviti u nekoliko različitih skupova. Svaki je sinonimski skup popraćen definicijom njegova značenja i rečeničnim primjerima kojima se želi ilustrirati tipična uporaba njegovih članova.

Najznačajniji semantički odnosi među sinonimskim skupovima za imenice i glagole, koji su u tom leksikonu ujedno i najbrojnije, jesu hiponimija/hiperonimija, antonimija i meronimija. Od navedenih semantičkih odnosa za izgradnju modela jedino se promatra hiponimija/hiperonimija.

3.2.1. Hrvatski WordNet

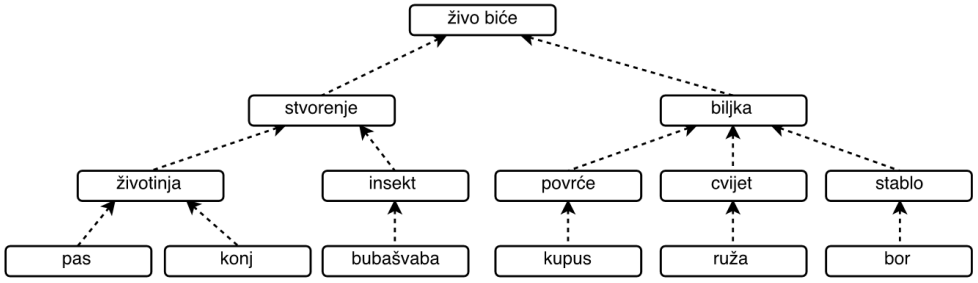
Hrvatski WordNet[2] (K. Šojat, 2010) je hrvatska verzija leksičko-semantičke mreže WordNet i ona je iskorištena za izgradnju modela. Iako je Hrvatski WordNet tek u začetku, sadašnji sadržaj koji on nudi bio je dovoljan za gradnju modela generatora.

3.2.2. Računanje sličnosti

Prema (Karan, 2012), mjere sličnosti riječi koje se temelje na WordNetu analiziraju strukturu mreže. Primjer strukture prikazan je na slici 3.1. Većina mjera pri izračunu koristi najspecifičniji zajednički hiperonim (engl. *least common subsumer – LCS)*. Najspecifičniji zajednički hiperonim dvaju koncepata, c_1 i c_2, definiran je kao koncept koji je najniži u hijerarhiji hiperonim-hiponim, a koji je istovremeno hiperonim i od c_1 i od c_2.

Najspecifičniji zajednički hiperonim možemo shvatiti kao najmanji zajednički višekratnik zadanih koncepata. Jedna od metoda za računanje sličnosti je metoda temeljena na duljini puta. Sličnost se računa na temelju duljine puta po relacijama hiperonim-hiponim. Intuitivno, sličnije riječi bit će bliže u mreži, pa će put među njima biti kraći. Sličnost je definirana formulom:

$$slicnost(c_1, c_2) = \frac{1}{duljinaPuta(c_1, c_2)} \tag{3.2}$$



Slika 3.1: Primjer hijerarhijske strukture koju definira odnos hiperonim-hiponim u WordNetu.

[2]Hrvatski WordNet izrađuje se u okviru projekta *Leksička semantika u izradi Hrvatskog WordNeta*, kao dijela programa *Računalnolingvistički modeli i jezičke tehnologije za hrvatski jezik.*

3.3. Postupci temeljeni na distribucijskim semantičkim modelima

Postupci temeljeni na bazama znanja pokazuju se korisnima u semantičkom modeliranju, ali nisu idealni jer imaju neke važne nedostatke. Jedan od njih je taj što u bazama znanja najčešće nedostaju pojedine veze između riječi koje u stvarnosti postoje, npr. u WordNetu veza između riječi *automobil* i *voziti*. Također još jedan nedostatak je taj što se sama izgradnja baze znanja obavlja ručno. Shodno tome, za svaki jezik ne postoje takvi resursi.

Osim postupaka temeljenih na bazama znanja postoje još i postupci temeljeni na distribucijskim semantičkim modelima (engl. *distributional semantic models – DSM*). Glavna intuicija na kojoj počiva rad DSM-a jest opažanje postojanosti veza između značenja riječi i njenog konteksta (susjedstva u tekstu). Stoga je dovljno modelirati značenje riječi modelirajući samo njenog susjedstvo u tekstu. Za modeliranje koristimo DSM čije parametre određujemo na temelju velikog korpusa. Sličnost riječi tada možemo odrediti uspoređujući tako dobivene distribucijske vektore riječi.

3.3.1. Računanje sličnosti

Nakon što postupak izgradnje DSM-a (Karan, 2012) odredi distribucijske vektore koji modeliraju distribuciju riječi po svim mogućim kontekstima, potrebno je odrediti koliko su oni slični. U tu svrhu može se koristiti bilo koja od klasičnih mjera usporedbe vektora, kao što je npr. euklidska udaljenost. Ipak, u računalnoj leksičkoj semantici najčešće se koristi kosinusna sličnost koja je za vektore $\vec{x}$ i $\vec{y}$ sa N definirana formulom:

$$sim_{cosine}(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}|\,|\vec{y}|} = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}} \tag{3.3}$$

Ova mjera može poprimati vrijednosti iz intervala $[-1, 1]$, a vektori se smatraju to sličnijim što je kut između njih manji.

3.3.2. Latentna semantička analiza

Latentna semantička analiza (Landauer i Dumais, 1997; Landauer, Foltz i Laham, 1998) je tehnika računanja semantičke udaljenosti između riječi i koristi se kao mjera semantičke sličnosti. Bazirana je na kolekciji više dokumenata. LSA reprezentira riječi kao vektore u prostoru gdje su pojmovi elementi tih dokumenata. Za riječi sličnog

značenja pretpostavlja se da se pojavljuju u sličnim dokumentima. Mjera semantičke udaljenosti između riječi računa se kosinusom kuta dvaju vektora kao što je objašnjeno u prethodnom potpoglavlju.

4. Gradnja modela

Kao što je već u uvodu napisano, gradnja modela je ostvarena je uz pomoć tehnika latentne semantičke analize i točkaste procjene uzajamne informacije te uz pomoć pohranjene leksičko-semantičke mreže *WordNet*, iz koje smo izvadili ono što nam je potrebno – odnose višeg i nižeg pojma, odnosno hiperonima i hiponima. Prije nego što opišemo kako se gradi model, opisat ćemo skupove podataka (engl. *datasets*) koji nam pružaju potrebne podatke za izgradnju modela.

4.1. Priprema skupova

Za izgradnju modela pomoću tehnika točkaste procjene uzajamne informacije i latentne semantičke analize koristili smo korpus[1] fHrWaC-parsed (Šnajder, Padó i Agić, 2013) – filtriranu verziju hrvatskog web korpusa HrWaC (Ljubešić i Erjavec, 2011). Korpus je pogodan za obavljanje zadaća vezanih uz obradu prirodnog jezika u kojima je jezična kvaliteta važnija od pokrivenosti (na primjer, za parsiranje).[2]

Korpus je pohranjen u formatu CoNLL,[3] u kojem su riječi iz rečenica zapisane u zasebnim redovima zajedno sa svojstvima vezanim uz njih. Svojstva su razdvojena tabulatorom, a rečenice praznim retkom. Jedan primjer rečenice iz korpusa prikazan je tablicom 4.1. Svojstva koja su bitna za gradnju modela:

- Id – indeks u rečenici, počinje od 1;
- Form – pojavnica (engl. *token*);
- Lemma – lematizirani[4] (engl. *lemmatized*) oblik riječi;

[1] Korpus je skupina označenih ili neoznačenih tekstova nad kojom se primjenjuju metode obrade prirodnog jezika u cilju otkrivanja novih teorija o prirodnom jeziku.

[2] `http://takelab.fer.hr/data/fhrwac/`

[3] Conference on Natural Language Learning.

[4] Lematizacija u enciklopedistici označava postupak dodavanja naslova članaka (natuknica, lema, deskriptora), odnosno riječi ili skupa riječi koji opisuje osnovni sadržaj onoga o čemu se govori u enciklopedijskom članku.

– Pos – oznaka vrste riječi (engl. *part of speech tag*).

Id	Form	Lemma	Pos	Ppos	Feat	Head	Deprel	Deps	Misc
1	Osobe	osoba	N	N-fpn	-	2	Sb	-	-
2	mogu	moći	V	Vmr3p	-	0	Pred	-	-
3	normalno	normalno	R	Rgp	-	4	Adv	-	-
4	razgovarati	razgovarati	V	Vmn	-	2	Sb	-	-
5	u	u	S	Sl	-	4	Prep	-	-
6	realnom	realan	A	Agpnsl	-	7	Atr	-	-
7	vremenu	vrijeme	N	N-nsl	-	5	Obj	-	-
8	jer	jer	C	Cs	-	4	Sub	-	-
9	je	biti	V	Vcr3s	-	11	Aux	-	-
10	u	u	S	Sl	-	11	Prep	-	-
11	biti	biti	V	Vcn	-	8	Pred	-	-
12	GSMSPYEAR	gsmspyear	N	N-msn	-	16	Ap	-	-
13	bežični	bežičan	A	Agpmsn	-	14	Atr	-	-
14	hands	hands	N	N-msn	-	16	Ap	-	-
15	free	free	N	N-fsg	-	14	Atr	-	-
16	uređaj	uređaj	N	N-msn	-	11	Pnom	-	-
17	.	.	Z	Z	-	0	Punc	-	-

Tablica 4.1: Rečenica *Osobe mogu normalno razgovarati u realnom vremenu jer je u biti GSMSPYEAR bežični hands free uređaj.* zapisana u formatu CoNLL.

Prilikom gradnje modela bilo tehnikom točkaste procjene uzajamne informacije ili tehnikom latentne semantičke analize koristit će se samo lematizirani oblik riječi, osim u slučaju kada se pojavi broj (riječ sa oznakom vrste riječi *M*) – tada se uzima sama pojavnica zbog toga što je lematizirani oblik svakog broja zapisan u neodgovarajućem obliku, tj. zapisan kao <*num*>. Prilikom odabira riječi u obzir se uzimaju samo imenice, brojevi, pridjevi, glagoli i prijedlozi.

Za izgradnju modela pomoću leksičko-semantičke mreže *WordNet*, učitavanje i obavljanje različitih zadaća nad korpusom neće biti potrebno, već je dovoljno učitati samo sadržaj WordNeta.

4.2. Izgradnja modela uz pomoć točkaste procjene uzajamne informacije

Prvi problem koji se javlja prilikom izgradnje modela je bio odabir izvora podataka iz kojeg je potrebno za svaku riječ u pojedinoj rečenici bilježiti koliko često se ta riječ pojavljivala sa svojim susjednim riječima i bilježiti koliko se općenito ta riječ pojavila u različitim rečenicama, tj. potrebno je pripremiti podatke kako bi se mogla primijeniti točkasta procjena uzajamne informacije na razini dokumenta (engl. *Document-based PMI*), pri čemu se svaka rečenica promatra kao jedan dokument. Rezultat bilježenja potrebno je pohraniti u bazu podataka. Primjer rezultata koji se dobije priložen je u dodatku A.

Nakon što se razriješi prethodno navedeni problem potrebno je primjenom formule (3.1) izgenerirati jednu igru asocijacije na sljedeći način:

1. Nasumice odabrati jednu riječ iz korpusa kao glavni pojam asocijacije, po mogućnosti onu čiji je broj pojavljivanja u različitim rečenicama korpusa veći od prosjeka;

2. Na temelju odabrane riječi potrebno je nasumice pronaći još četiri različite riječi čiji je rezultat točkaste procjene uzajamne informacije s glavnom riječju velik, pri čemu treba paziti da nakon svake generirane riječi iduća riječ koju je potrebno generirati mora imati što manji rezultat točkaste procjene uzajamne informacije s prethodno generiranim riječima, i naravno, što veći sa glavnom riječju – time nastojimo dobiti četiri riječi koje nisu međusobno slične, budući da nije u cilju izgraditi igru asocijacije koja će igračima ponuditi riječi koje praktički imaju isto značenje. Algoritam pronalaska tih četiriju različitih riječi detaljnije je objašnjen u potpoglavlju 4.2.1;

3. Ponavljati taj postupak za svaku od četiri nove riječi, pri čemu treba paziti da se svaka od generiranih riječi ne ponovi na bilo kojem drugom mjestu u igri.

Ako smo ispravno pratili prethodna tri koraka, trebali bismo dobiti jedan primjerak igre asocijacije, pri čemu naravno treba uzeti u obzir da točkasta procjena uzajamne informacije nije jedina tehnika s kojom gradimo model, tako da će u konačnici samo pojedine riječi biti generirane ovom tehnikom, dok će preostale biti generirane drugim tehnikama (LSA i WordNet).

4.2.1. Algoritam pronalaska četiriju različitih asocijativnih riječi

Algoritam pronalaska četiriju različitih asocijativnih riječi ilustriran je Java-pseudokodom i sastoji se od dvije metode: `stvoriListu(rijec x)` i `oduzmiPmiSaPostojecim(rijec y, pmiXY, iskoristeneRijeci)`. U nastavku objašnjenja algoritma za rezultat točkaste procjene uzajamne informacije između riječi `x` i `y` koristit će se oznaka `pmi(x,y)`. Java-pseudokod algoritma dan je u nastavku:

```
stvoriListu(rijec x) {
        asocijativneRijeci =
            NasumicnoIzmjesaj(dohvatiAsocijativneRijeciIzBazeZa(x));
        sortirajPoPmiRezultatima(asocijativneRijeci);
        lista = stvoriListu();
        for(i = 0; i < 4; i++) {
                sljedecaRijec = NULL;
                maksPmi = dohvatiMinimalanRealniBroj();
                fore(rijec y : asocijativneRijeci) {
                    if(lista.sadrziRijec(y)) continue;
                        pmiXY = dohvatiPmi(x,y);
                        nPmiXY = oduzmiPmiSaPostojecima(y, pmiXY,
                            lista);
                        if(pmiXY == nPmiXY && (sljedecaRijec ==
                            NULL || maksPmi < max)) {
                                sljedecaRijec = y;
                                break;
                        }
                        else if(nPmiXY > maksPmi) {
                                maksPmi = nPmiXY;
                                sljedecaRijec = y;
                        }
                }

                if(sljedecaRijec  == NULL) {
                        throw exception();
                }
                lista.dodajRijec(sljedecaRijec);
        }
        return lista;
}

oduzmiPmiSaPostojecim(rijec y, pmiXY, iskoristeneRijeci) {
        nPmiXY = pmiXY;
        for(rijec z : iskoristeneRijeci) {
```

```
                nPmiXY = nPmiXY - dohvatiPmi(y,z);
            }
            return nPmiXY;
}
```

Metoda `stvoriListu(rijec x)` dohvaća sve riječi iz baze koje su se pojavljivale zajedno s onom koja je predana kao argument metode te ih sortira po rezultatima koji se dobiju točkastom procjenom uzajamne informacije. Prilikom dohvata riječi iz baze, njihov redoslijed u listi nasumično se izmiješa, kako bi se izbjeglo generiranje uvijek istih riječi za zadanu. Nakon što se lista riječi dohvati, potrebno je stvoriti drugu listu u kojoj će se pohraniti četiri različite riječi čiji je rezultat točkaste procjene uzajamne informacije s glavnom riječju velik. Generiranje riječi svodi se na iteraciju liste u kojoj se nalaze iz baze dohvaćene riječi, pri čemu se u svakom koraku iteracije bira riječ koja ima što veći rezultat s glavnom riječi i što manji s prethodno generiranim. Kako bi se navedeni problem riješio, potrebno je pozvati metodu `oduzmiPmiSaPostojecima(rijec y, pmiXY, iskoristeneRijeci)`, čija je zadaća oduzeti `pmi(x,y)` sa svim `pmi(y,z)`, pri čemu je `z` jedna od riječi koja je prethodno generirana.

Ako se uspostavi da je `pmi(x,y)` jednak prije i nakon oduzimanja i ako je to prva riječ u iteraciji ili riječ čiji je `pmi(x,y)` s riječi x veći od prethodnih, tada se riječ dodaje u listu. Ako uvjet nije zadovoljen, iteriranje se nastavlja sve dok još ima riječi ili dok se ne pojavi riječ čiji `pmi(x,y)` ostane nepromijenjen nakon iteracije. Ako je došlo do slučaja da više nema riječi, riječ čiji je `pmi(x,y)` nakon oduzimanja bio najveći dodaje se u listu, ili, ako takva ne postoji, dojavljuje se greška (najčešći uzrok pojave greške je nedostatak riječi u bazi), koja bi trebala signalizirati pozivatelju da proba pozvati metodu s nekom drugom zadanom riječi. Ako je pak došlo do slučaja da `pmi(x,y)` ostane nepromijenjen, onda se u listu dodaje riječ čiji je `pmi(x,y)` nakon oduzimanja bio najveći.

Prilikom iteracije, ako se pojavi riječ koja je već dodana u stupac, onda se ona jednostavno ignorira i nastavlja se dalje s iteracijom.

4.3. Izgradnja modela uz pomoć latentne semantičke analize

Prvi problem koji se javlja prilikom izgradnje modela je sličan onom s kojim smo se susreli kada smo gradili model tehnikom točkaste procjene uzajamne informacije, s

	Osoba	Mogu	Razgovarati	Realan	Vrijeme	Bežićan	Hands	Free	Uređaj	Cijena	Jedan	Relativan	Nizak
normalno	1	1	1	1	1	0	0	0	0	0	0	0	0
GSMSPYEAR	0	0	0	0	1	1	1	1	2	1	1	1	1

Slika 4.1: Primjer izgrađene matrice

jedinom razlikom da ćemo sad pod terminom susjedna riječ smatrati sve one riječi koje su relativno za pet mjesta udaljene od neke zadane riječi.

Rješenje prvog problema koji se javlja možemo prikazati matricom u kojoj redovi predstavljaju riječi, a stupci predstavljaju sve susjedne riječi. Ipak s obzirom da gradimo matricu koja bi u konačnici mogla biti prevelika, nećemo razmatrati baš svaku riječ, već samo npr. najučestalijih 50 tisuća riječi (pri čemu će stupci biti sve riječi koje su bile susjedne tim najučestalijim riječima) i za njih ćemo bilježiti koliko često se pojavljuju sa svojim susjednim riječima. Npr. ako imamo zadane rečenice:

- *Osobe mogu normalno razgovarati u realnom vremenu jer je u biti GSMSPYEAR bežični hands free uređaj*;
- *Cijena jednog GSMSPYEAR uređaja je relativno niska.*

i ako pretpostavimo da su u skupinu 50 tisuća najučestalijih riječi ušle riječi *GSMSPYEAR* i *normalno*, dobit ćemo matricu kao na slici 4.1.
Ono što je bitno uočiti je to da su riječi koje nisu susjedne najučestalijim riječima u matrici svejedno upisane kao par, ali je njihova vrijednost zajedničkog pojavljivanja jednaka 0.

Sada kada imamo izgrađenu matricu mogli bismo odmah računati kosinusnu sličnost pojedinih redaka prikazanih kao vektora, ali budući da je u ovoj matrici puno elemenata s vrijednostima 0, za lakše računanje kosinusne sličnosti matricu ćemo reducirati tehnikom singularne dekompozicije matrice[5] (engl. *SVD – singular value decomposition*). Format u kojem matrica treba biti zapisana kako bi se redukcija mogla izvršiti zove se *compressed sparse column*.[6] Nakon što se razriješi prethodno navedeni problem potrebno je primjenom formule 3.2 izračunati kosinusnu sličnost za svaki par

[5]SVD (Karan, 2012) se može shvatiti kao mehanizam kojim se iz koreliranog skupa konteksta generira novi skup latentnih konteksta koji su međusobno nezavisni. Pri tome smanjenje dimenzionalnosti odbacuje manje važne latentne kontekste i tako naglašava temeljnu semantiku riječi.

[6]`https://en.wikipedia.org/wiki/Sparse_matrix#Compressed_sparse_column_.28CSC_or_CCS.29`

riječi iz redaka te rezultat zabilježiti u bazu podataka. Primjer rezultata, koji se dobije prije pohrane u bazu podataka, priložen je u dodatku A. Sljedeći korak je izgenerirati jednu igru asocijacije na sljedeći način:

1. Nasumice odabrati jednu od 50 tisuća najučestalijih riječi iz korpusa kao glavni pojam asocijacije, po mogućnosti onu čiji je broj pojavljivanja u različitim rečenicama korpusa veći od prosječnog pojavljivanja;

2. Na temelju odabrane riječi potrebno je nasumice pronaći još četiri različite riječi čija je kosinusna sličnost s glavnom riječju velika (tj. bliža broju 1), pri čemu treba paziti da nakon svake generirane riječi iduća riječ koju je potrebno generirati mora imati što manju kosinusnu sličnost s prethodno generiranim riječima i, naravno, što veću sa glavnom riječju – iz istog razloga kao i u potpoglavlju 4.2. Algoritam pronalaska tih četiriju različitih riječi vrlo je sličan onom u potpoglavlju 4.2.1, s jedinom razlikom, da se umjesto točkaste procjene uzajamne informacije dviju riječi uzima u obzir kosinusna sličnost između tih riječi;

3. Ponavljati taj postupak za svaku od te četiri nove riječi, pri čemu treba paziti da se svaka od generiranih riječi ne ponovi na bilo kojem drugom mjestu u igri.

Ako smo ispravno pratili prethodna tri koraka, trebali bismo dobiti jedan primjerak igre asocijacije, pri čemu treba uzeti u obzir, kao što je već rečeno u prethodnom potpoglavlju 4.2, da ovo nije jedina tehnika s kojom ćemo u konačnici izgraditi model.

4.4. Izgradnja modela uz pomoć hrvatskog WordNeta

Prva stvar koju je potrebno napraviti je učitati sadržaj hrvatskog WordNeta, koji se može preuzeti sa web-stranice Filozofskog fakulteta u Zagrebu.[7] Prilikom učitavanja potrebno je jedino pročitati imenice i glagole za koje postoje semantički odnos hiponimima i hiperonima, dok se ostale riječi u potpunosti zanemaruju.

Budući da se u obzir uzimaju samo semantički odnosi hiponima i hiperonima, ova tehnika neće moći samostalno izgraditi kvalitetan model, već će se prilikom izgradnje također koristiti tehnika točkaste procjene uzajamne informacije i latentna semantička analiza. Ova tehnika će zapravo samo poslužiti kao pomoćna tehnika za generiranje ciljnog modela.

Jedini korak koji ćemo koristiti prilikom generiranja modela je taj da ćemo za neku nasumice odabranu riječ vratiti njezin hiperonim ili hiponim (ako postoji) i taj postupak

[7] `http://crown.ffzg.hr/`

ponavljati dok se ne sagradi model. Iako postoje bolji načini za generiranje modela uz pomoć WordNeta, u ovom radu neće biti razmatrani.

5. Implementacija

5.1. Implementacija programskog rješenja

Programsko rješenje implementirano je u Javi 1.8.92. Osim Java SE[1] korištene su i dvije vanjske biblioteke :

* JDBC Driver for MySQL (Connector/J) – biblioteka služi za omogućavanje pristupa bazi podataka MySQL[2] u kojoj su pohranjene riječi iz korpusa te podaci potrebni za generiranje igre asocijacije pomoću tehnika navedenih u prethodnim poglavljima;
* JSON In Java – biblioteka služi za učitavanje datoteka json, konkretno u ovom radu za učitavanje hrvatskog WordNeta koji je pohranjen u obliku json.

Implementacija programskog rješenja načinjena je od nekoliko nezavisnih komponenti koje obavljaju određene zadaće:

1. Dohvaćanje riječi i podataka vezanih uz njih iz korpusa i serijalizacija izlaza u formatu pogodnom za pohranjivanje odgovarajućih stavki koje su potrebne kako bi se model mogao izgraditi tehnikom točkaste procjene uzajamne informacije – izvedeno u paketu:

   `hr.fer.zemris.zr.wagame.pmi`;

2. Dohvaćanje riječi i podataka vezanih uz njih iz korpusa i serijalizacija izlaza u formatu pogodnom za pohranjivanje odgovarajućih stavki koje su potrebne kako bi se model mogao izgraditi tehnikom latentne semantičke analize – izvedeno u paketu:

   `hr.fer.zemris.zr.wagame.lsa`;

[1] Java Platform, Standard Edition ili Java SE je naširoko rasprostranjena platforma za razvoj i implementaciju prijenosnog koda za osobna i poslužiteljska računala.

[2] MySQL je besplatan, sustav otvorenog koda za upravljanje bazom podataka. Uz PostgreSQL, MySQL je čest izbor za projekte otvorenog koda te se distribuira kao sastavni dio serverskih Linux distribucija.

3. Dohvaćanje riječi i podataka vezanih uz njih iz hrvatskog WordNeta, pohranjenog u obliku datoteke json i serijalizacija izlaza u formatu pogodnom za pohranjivanje odgovarajućih stavki koje su potrebne kako bi se model mogao izgraditi pomoću odnosa hiperonima i hiponima – izvedeno u paketu :

   `hr.fer.zemris.zr.wagame.wordnet`;

4. Pristupanje bazi podataka, učitavanje podataka iz serijaliziranih izlaza, dobivenih prethodno navedenim tehnikama te konačno pohranjivanje tih podataka u bazu – izvedeno u paketu:

   `hr.fer.zemris.zr.wagame.database`;

5. Dohvaćanje podataka iz baze te generiranje jednog primjerka igre asocijacije na temelju tih podataka – izvedeno u paketu:

   `hr.fer.zemris.zr.wagame.game`;

6. Pokretanje igre i vizualizacija rješenja – izvedena u paketu:

   `hr.fer.zemris.zr.wagame.main`.

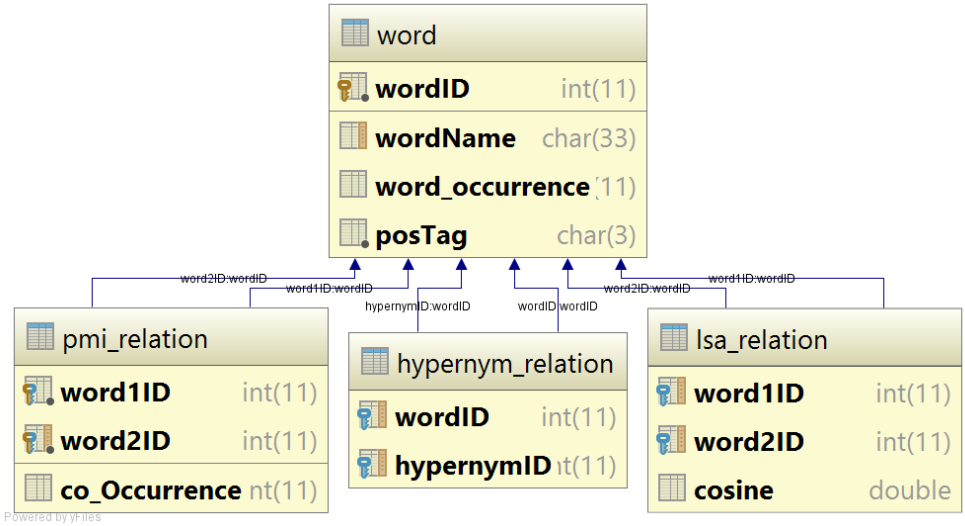
5.2. Implementacija baze podataka

Prilikom gradnje modela koji generira igru asocijacija, potrebno je i pohraniti rezultate obrade kako bi se kasnije mogli koristiti. Zbog velikog broja riječi u korpusu bilo bi nepraktično i neefikasno svaki puta učitavati te rezultate, pa je stoga izgrađena jednostavna baza podataka u kojoj se pohranjuju potrebni podaci.

Baza ima strukturu tablica svedenih na 3NF kako bi se izbjegla redundancija. Entiteti koji ju čine su **word**, **pmi_relation**, **lsa_relation** i **hypernym_relation**.

Za svaku riječ iz korpusa, koja je predstavljena entitetom **word**, evidentirana je šifra, naziv, broj pojavljivanja u različitim rečenicama i oznaka vrste riječi. Za potrebe generiranja modela pomoću tehnike točkaste procjene uzajamne informacije evidentirane su šifre dviju riječi, koje se referenciraju na dvije različite riječi iz entiteta **word** i broj njihovog zajedničkog pojavljivanja u različitim rečenicama. Za potrebe generiranja igre pomoću latentne semantičke analize također su evidentirane šifre dviju riječi, koje se referenciraju na dvije različite riječi iz entiteta **word** te je za njih potrebno izračunati kosinus kuta između tih dviju riječi (nakon što se predstave kao vektori). Da bi se izgradio model uz pomoć pohranjene leksičko-semantičke mreže *WordNet*, u tablici **hypernym_relation** evidentirana je šifra jedne riječi iz korpusa te njegovog hiperonima, koji je također riječ iz korpusa.

Izgled baze podataka prikazan je na slici 5.1, ER-modelom. Odgovarajući opisi atributa i primarnih ključeva u relacijama i njihovim vezama napisani su unutar slike.



Slika 5.1: ER-model baze podataka

6. Eksperimenti

Cilj ovog rada bio je izgraditi model koji generira igru asocijacija riječi. Cilj je ostvaren uz pomoć tehnike točkaste procjene uzajamne informacije, latentne semantičke analize i uz pomoć pohranjene leksičko-semantičke mreže WordNet.

Za potrebe eksperimentiranja izrađen je program komandne linije koji implementira jedan takav model. Program radi na takav način da svaku riječ generira jednom od navedenih tehnika, ovisno o parametru koji je zadan prilikom pokretanja programa za tu tehniku. Taj parametar utvđuje kolika je vjerojatnost da će se iduća riječ, koju je potrebno generirati, generirati baš tom tehnikom. Npr. ako su zadani parametri :

- 0% za WordNet
- 0% za latentnu semantičku analizu
- 100% za točkastu procjenu uzajamne informacije

tada će svaka riječ koju je potrebno generirati biti generirana tehnikom točkaste procjene uzajmne informacije. Cilj programa je omogućiti korisnicima da utvrde s kojim vjerojatnostima odabira tehnike, koja će generirati iduću riječ, se ostvare najbolji rezultati. Osim navedenih parametara, prilikom pokretanja programa potrebno je i navesti riječ za koju se želi izgraditi primjerak igre asocijacija.

U eksperimentu sudjeluju ispitanici koji na temelju svog mišljenja donose subjektivnu odluku o tome s kojim parametrima se postižu najbolji rezultati, tj. s kojim parametrima se generira, po njihovom mišljenju, najbolja igra asocijacija riječi. U konačnici, cilj eksperimenta je na temelju dobivenih odgovora od ispitanika, utvrditi da li je bolje preferirati tehniku točkaste procjene uzajmane informacije ili latentnu semantičku analizu za generiranje pojedine riječi u igri.

Kako je leksičko-semantička mreža WordNet gotova kolekcija riječi i njihovih odnosa, kvalitetu generiranja riječi putem WordNeta nemamo u cilju posebno razmatrati, tako da ćemo prilikom eksperimentiranja ograničiti ispitanicima koju maksimalnu vrijednost parametra, koji predstavlja vjerojatnost generiranje sljedeće riječi uz pomoć

nje, smiju zadati prilikom pokretanja programa.

6.1. Provedba eskperimenta

U provedbi eskperimenta sudjeluju tri ispitanika. Riječ, za koju se generira jedan primjerak igre asocijacija, je riječ *grad*.

Svaki ispitanik mora odrediti s kojom kombinacijom ulaznih parametara se postižu najbolji rezultati. Prilikom zadavanja ulaznih parametara, ispitanik smije postaviti vjerojatnost generiranja iduće riječi pomoću leksičko-semantičke mreže WordNet na vrijednost od maksimalno 7%. Cilj toga, kao što je već prije navedeno, je taj da se ispitanik fokusira na kvalitetu generiranja riječi pomoću točkaste procjene uzajamne inforamcije i latentne semantičke analize.

Svaki ispitanik pokreće program 10 puta sa svaki put različitim zadanim parametrima te na kraju odabire najbolje izgeneriranu igru, na temelju subjektivne procjene.

6.1.1. Rezultati eksperimenta

Rezultat prvog ispitanika

Ulazni parametri koje je ispitanik zadao prilikom pokretanja programa prikazani su u tablici 6.1.

	WordNet	LSA	PMI
1	90%	10%	0%
2	7%	64%	29%
3	7%	22%	71%
4	0%	50%	50%
5	0%	0%	100%
6	7%	7%	86%
7	5%	13%	82%
8	2%	23%	75%
9	5%	5%	90%
10	3%	41%	56%

Tablica 6.1: Ulazi koje je zadao pvi ispitanik prilikom pokretanja programa

Rezultat :

– ispitanik je ocijenio da su najbolji rezultati postignuti uz parametre 5, 5, 90 (9).

Rezultat drugog ispitanika

Ulazni parametri koje je ispitanik zadao prilikom pokretanja programa prikazani su u tablici 6.2.

	WordNet	LSA	PMI
1	0%	40%	60%
2	0%	5%	95%
3	0%	75%	25%
4	0%	50%	50%
5	1%	98%	1%
6	0%	0%	100%
7	2%	3%	95%
8	7%	50%	43%
9	1%	49%	50%
10	5%	5%	90%

Tablica 6.2: Ulazi koje je zadao drugi ispitanik prilikom pokretanja programa

Rezultat :

– ispitanik je ocijenio da su najbolji rezultati postignuti uz parametre 0, 5, 55 (2).

Rezultat trećeg ispitanika

Ulazni parametri koje je ispitanik zadao prilikom pokretanja programa prikazani su u tablici 6.3.
Rezultat :

– ispitanik je ocijenio da su najbolji rezultati postignuti uz parametre 0, 19, 81 (1).

6.2. Analiza rezultata

Prema ocjenama ispitanika može se uočiti da se točkasta procjena uzajamne informacije pokazala učinkovitijom od latentne semantičke analize. Iako je u cilju bilo ostvariti bolje rezultate uz pomoć latentne semantičke analize, oni su ostvareni pomoću tehničke točkaste procjene uzajamne informacije. Postoji mnogo mogućih razloga, poput:

– prilikom gradnje modela pomoću latentne semantičke analize, u obzir se uzimalo samo 50 tisuća najučestalijih riječi – najvjerojatniji uzrok,

	WordNet	LSA	PMI
1	0%	19%	81%
2	0%	1%	99%
3	0%	35%	65%
4	7%	27%	66%
5	1%	18%	81%
6	0%	66%	34%
7	5%	55%	40%
8	2%	35%	63%
9	0%	91%	9%
10	7%	81%	12%

Tablica 6.3: Ulazi koje je zadao treći ispitanik prilikom pokretanja program

– dogodila se pogreška prilikom reduciranja matrice,

– i dr.

Prvi sljedeći korak bi svakako bio rješavanje prvog problema – uzeti veći broj riječi u obzir prilikom generiranja modela pomoću latentne semantičke analize.

Ono što je također zanimljivo primijetiti je to da se korisnicima nije svidio trenutno izgrađeni model za generiranje riječi uz pomoć leksičko-semantičke mreže WordNet. Razlog tome je taj što su se prilikom gradnje modela u obzir uzimali samo odnosi višeg i nižeg pojma, odnosno hiperonima i hiponima. Naknadne izmjene modela, u okviru završnog rada, zbog nedostatka vremena nisu moguće.

U dodatku B priloženi su primjerci igre asocijacije koje su ispitanici, prema svojim subjektivnim procjenama, ocijenili najboljima.

7. Zaključak

Računalna leksička semantika bavi se prikazom značenja riječi te ima važnu ulogu u sustavima za obradu i razumijevanje prirodnog jezika. Distribucijski semantički modeli značenje riječi prikazuju visokodimenzijskim kontekstnim vektorima, ekstrahiranima na temelju supojavljivanja riječi u korpusu. Takvi su se modeli pokazali vrlo korisnima na nizu zadataka obrade prirodnog jezika. U okviru ovog rada izgrađen je model koji generira jednu igru asocijacija riječi.

Cilj ovog rada bio je izgraditi model koji generira igru asocijacija riječi uz postupak temeljen na distribucijskim semantičkim modelima, temeljen na bazi znanja i uz točkastu procjenu uzajamne informacije.

U sklopu rada obavljen je eksperiment čiji je cilj bio usporediti rezultate koje generiraju tehnika točkaste procjene uzajamne informacije i latentna semantička analiza. Iako je u cilju bilo ostvariti bolje rezultate generiranjem igre uz pomoć latentne semantičke analize, oni su ostvareni pomoću tehničke točkaste procjene uzajamne informacije. Razlog tome je vjerojatno taj, što se prilikom gradnje modela pomoću latentne semantičke analize, u obzir uzimalo samo 50 tisuća najučestalijih riječi.

Buduća nadogradnja bila bi svakako uzimanje većeg broja riječi u obzir prilikom generiranja modela pomoću latentne semantičke analize, proučavanje i korištenje nekih drugih tehnika za geniranje ciljnog modela te također provedba sustavnije evaluacije postupaka.

LITERATURA

Nitish Aggarwal, Kartik Asooja, i Paul Buitelaar. Exploring esa to improve word relatedness. *Lexical and Computational Semantics (* SEM 2014)*, 51, 2014.

Raluca Budiu, Christiaan Royer, i Peter Pirolli. Modeling information scent: A comparison of lsa, pmi and glsa similarity measures on common tests and corpora. stranice 314–332, 2007.

Mladen Karan. Računalni modeli leksičke semantike. 2012. URL `https://www.fer.unizg.hr/_download/repository/KDI_Mladen_Karan.pdf`.

Thomas K Landauer i Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

Levow G. Farahat A. Royer C. Matveeva, I. Terms representation with generalized latent semantic analysis. 2005.

Peter Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. 2001.

K. Šojat. *Struktura glagolskog dijela Hrvatskog WordNeta. Filologija, (59)*. 2010. URL `http://hrcak.srce.hr/98091`.

DODATAK A

U ovom poglavlju nalaze se primjeri rezultata obrade korpusa nastalih primjenama tehnika latentne semantičke analize i točkaste procjene uzajamne inforamcije.

X	Y	$sim_{cosine}(X, Y)$
buzet	cottbus	0.7278740371730686
buzet	diskusija	0.8737446709359946
buzet	zauzdati	0.7620921681974484
buzet	perica	0.7349687419536657
buzet	muskulatura	0.713773269814445
buzet	utemeljiteljica	0.7647197998780499
buzet	telefona	0.7310171862163284
buzet	staja	0.7891651705444264
rukopis	milat	0.8379946199966718
rukopis	dovati	0.7096349930684125
rukopis	blackout	0.7626595256694907
južnoslavenski	presudno	0.7533382021297494
južnoslavenski	makarska	0.726932412880097
lijepo	trojanac	0.7525562025197204
lijepo	grubišan	0.8386971496137942
lijepo	nominacija	0.8155801870332068
iskušenja	muenchen	0.7180140281516876
iskušenja	limen	0.7608286201606103
iskušenja	promptno	0.7551356658189518
iskušenja	meksik	0.7565105107771798
radioaktivnost	lijepo	0.766702748976084
radioaktivnost	postupke	0.704092083148378
radioaktivnost	strmi	0.7146735790309776

radioaktivnost	nerazuman	0.8254574212718214
radioaktivnost	slijetanje	0.7699283716896684
radioaktivnost	spomenuti	0.7436259303243802
radioaktivnost	loeb	0.7999018184979307
radioaktivnost	vrt	0.8308603716615776
radioaktivnost	operator	0.7894799062146473
porcija	igrica	0.9552210370855480
porcija	hamburger	0.7010497968455037
porcija	logitech	0.7511133034119051
porcija	rumunjska	0.7827328197763735
porcija	pjeval	0.7513255257662402
porcija	posrtanje	0.7278656295671699
naramenica	svindal	0.8500941222841583
naramenica	kalogjera	0.7817236829027278
naramenica	nielsen	0.7087047866662702
naramenica	postajemo	0.7464365818414358
naramenica	poletiti	0.7809229207862806
naramenica	ground	0.8103259751966786
doživljavamo	kositi	0.7520480599744563
doživljavamo	mobilizacija	0.8957392070402873
doživljavamo	ubrzavati	0.7358633876948454
kompjuter	bend	0.8603353443073087
najužiti	lost	0.7502788507464822
najzanimljivije	najužiti	0.8352075809275686
najzanimljivije	užitka	0.7253028789282513
najzanimljivije	rimljan	0.7300996411911668
najzanimljivije	lost	0.7358635233482518
nadoplata	intranet	0.7712072743683013
nadoplata	uvriježen	0.7844131520377622
nadoplata	kamp	0.8051894818553311
nadoplata	anglikanski	0.7582262462647898
nadoplata	humanistički	0.7679094294783000

Tablica 7.1: Primjer rezultata obrade korpusa uz pomoć latentne semantičke analize

X	Y	$C(X)$	$C(Y)$	$C(X,Y)$	$PMI(X,Y)$
prtljažnik	automobil	530	19736	56	2,06958
prtljažnik	aut	530	9012	26	2,10429
prtljažnik	staviti	530	8331	24	2,101201
prtljažnik	otvoriti	530	10143	20	1,438192
suradnja	međunarodan	8104	7775	252	1,546073
suradnja	poslovan	8104	7821	165	1,006356
suradnja	policija	8104	66480	158	0,113369
suradnja	gospodarski	8104	4694	123	1,249948
suradnja	uspješan	8104	5030	92	0,872468
suradnja	haaški	8104	1033	91	4,202144
suradnja	sporazum	8104	2748	89	1,54491
suradnja	prekid	8104	3609	86	1,136688
suradnja	nastavak	8104	3941	80	0,968308
suradnja	grad	8104	34280	78	0,108538
zgodan	žena	996	37210	74	0,771866
zgodan	jako	996	22954	60	1,014525
zgodan	muškarac	996	28798	42	0,566053
zgodan	visok	996	16495	28	0,658833
zgodan	djevojka	996	14844	24	0,627524
zgodan	cura	996	2018	22	4,231277
zgodan	kazati	996	114606	20	0,067732
upravljanje	sustav	2272	10959	248	3,850363
upravljanje	ured	2272	7213	126	2,972183
upravljanje	zabrana	2272	3012	106	5,987863
upravljanje	vozilo	2272	14467	97	1,140812
upravljanje	hitan	2272	10279	72	1,191798
upravljanje	daljinski	2272	447	68	25,88346
upravljanje	motoran	2272	839	60	12,16775
upravljanje	zaštićen	2272	1424	48	5,735248
upravljanje	nadzor	2272	3906	47	2,047324
upravljanje	ustanova	2272	1622	46	4,82534

Tablica 7.2: Primjer rezultata obrade korpusa uz pomoć točkaste procjene uzajamne informacije

DODATAK B

U ovom poglavlju nalaze se primjeri izgeneriranih igri asocijacija, koje su ispitanici tijekom eksperimenta ocijenili, prema subjektivnoj procjeni, ocijenili najboljima.

jos	gradnja	boksač	starogradski
naselje	izvođač	momčad	povijestan
staklo	pogodovati	plinovod	činiti
muzej	inicijator	reprezentacija	dubrovački
antički	invenstitor	ukrajinski	jezgra
grad			

Tablica 7.3: Primjer koji je prvi korisnik ocijenio najboljim

znanstven	intera	odlazeći	poslovan
čitaonica	kraj	lbp	međuinstitucionalan
nacionalan	zagreb	središte	organiziran
školski	inter	otok	raskid
knjižnica	zaprešić	rovinj	suradnja
grad			

Tablica 7.4: Primjer koji je drugi korisnik ocijenio najboljim

rimski	usuglasiti	kralj	novigrad
mitologija	upotrijebljen	euforija	nedaleko
panathinaikos	hleb	panika	pazin
prvenstvo	grdić	svijet	igrati
grčki	aleksandar	zavladati	poreč
grad			

Tablica 7.5: Primjer koji je treći korisnik ocijenio najboljim

Primjena modela distribucijske semantike u igri asocijacija riječi

Sažetak

Računalna leksička semantika bavi se prikazom značenja riječi te ima važnu ulogu u sustavima za obradu i razumijevanje prirodnog jezika. Distribucijski semantički modeli značenje riječi prikazuju visokodimenzijskim kontekstnim vektorima, ekstrahiranima na temelju supojavljivanja riječi u korpusu. Takvi su se modeli pokazali vrlo korisnima na nizu zadataka obrade prirodnog jezika. U okviru ovog rada izgrađen je model koji generira igru asocijacija riječi. Izgradnja modela je ostvarena uz pomoć obrade prirodnog jezika i uz pomoć tehnike točkaste procjene uzajamne informacije. Izgradnja je ostvarena uz korištenje postojećih korpusa. Nad izgrađenim modelom provedeno je eksperimentalno vrednovanje postupaka.

Ključne riječi: točkasta procjena uzajamne informacije, latentna semantička analiza, hrvatski WordNet, obrada prirodnog jezika, asocijacija riječi

Use of Distributional Semantic Models in the Word Association Game

Abstract

Computer lexical semantics deals with the representation of word meanings and has an important role in systems for processing and understanding of the natural language. Distributional semantic models represent word meanings with high-dimensional contextual vectors, which are extracted based on the co-occurrence of words in the corpus. Such models proved very useful with a series of natural language processing tasks. In this thesis, we have constructed a model that generates a word association games. Model construction was accomplished with the use of existing corpus. The constructed models underwent an experimental process evaluation.

Keywords: pointwise mutual information, latent semantic analysis, Croatian WordNet, natural language processing, word association