UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Kaja Vidmar

Vizualizacija konceptualnega prostora besedilnih zbirk

DIPLOMSKO DELO
NA INTERDISCIPLINARNEM UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Matija Marolt

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00021/2010
Datum: 01.09.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Kandidat: **KAJA VIDMAR**

Naslov: **VIZUALIZACIJA KONCEPTUALNEGA PROSTORA BESEDILNIH ZBIRK**

VISUALISATION OF TEXT DOCUMENTS BASED ON CONCEPTUAL SPACES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V diplomski nalogi preučite kako lahko za vizualizacijo besedilnih zbirk uporabimo konceptualne prostore. Preučite tehnike semantične obdelave besedilnih zbirk, predvsem iskanja tem v besedilih. Raziščite tudi tehnike vizualizacije, ki omogočajo učinkovito in razumljivo vizualizacijo koncenptualnih prostorov, ki iz tovrstnih semantičnih obdelav izhajajo. Za demonstracijo razvijte aplikacijo, ki bo omogočala vizualizacijo in iskanje po konceptualnem prostoru besedilnih zbirk.

Mentor:

doc. dr. Matija Marolt

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan Fakultete za računalništvo in informatiko:

prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:

UNIVERZA V LJUBLJANI • Fakulteta za matematiko in fiziko • 6

prof. dr. Andrej Likar

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Kaja Vidmar,

z vpisno številko 63050221,

sem avtor/-ica diplomskega dela z naslovom:

Naslov diplomskega dela

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
  prof. [doc.] dr. Matije Marolt
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15.12.2010 Podpis avtorja/-ice:

Zahvala

Za vse napotke, vodenje, usmerjanje in pomoč pri nalogi bi se rada iskreno zahvalila svojemu mentorju doc. dr. Matiji Maroltu. Zahvaljujem se tudi mag. Gregorju Strletu za idejo in pomoč pri izdelavi diplomske naloge. Posebno zahvalo namenjam učiteljicama iz Osnovne šole Spodnja Šiška, Veri Fujs in Nini Bradič, ki sta mi pomagali pri lektoriranju diplomske naloge.

Zahvalila bi se tudi mojim prijateljem, ki so me vzpodbujali in mi pomagali ob težkih trenutkih. Seveda pa ne smem pozabiti na mamo in očeta, ki sta mi stala ob strani in mi dajala moralno podporo.

Z diplomsko nalogo se zaključuje verjetno moje najlepše obdobje študijskega življenja.

Kazalo

Seznam uporabljenih kratic

Seznam kratic:

- SOM - samoorganizirajoča mreža
- LSA - latentna semantična analiza
- LSI - latentno semantično indeksiranje
- PLSA - verjetnostna latentna semantična analiza
- PLSI - verjetnostno latentno semantično indeksiranje
- LDA - latentna Dirichletova alokacija
- TV - televizija
- BOW - bag of words, vreča besed

Povzetek

V diplomskem delu je opisano, kako lahko za vizualizacijo besedilnih zbirk uporabimo konceptualne prostore. Razdeljeno je na dva dela, prvi del je namenjen analizi besedilnih zbirk, drugi pa vizualizaciji dobljenih rezultatov.

Zaradi vse večjega števila elektronskih podatkov težimo k samostojni analizi in organizaciji le teh podatkov v različne, v naprej neznane skupine. Opisani so nekateri algoritmi (latentna semantična analiza, verjetnostna semantična analiza in latentna Dirichletova alokacija), s katerimi lahko to storimo.

V diplomski nalogi iščemo neznane teme, ki se pojavljajo v zbirki besedil. Z izbranim algoritmom analiziramo zbirko besedil in jo nato predstavimo v konceptualnem prostoru. Konceptualni prostor je zgrajen iz geometričnih struktur. Točka predstavlja semantiko (pomen) besede. Povezave in regije pa predstavljajo relacije med koncepti. Semantika besede je zgrajena iz konceptov, ki so predstavljeni kot regije v prostoru.

Za vizualizacijo konceptualnega prostora besedilnih zbirk sem se odločila uporabiti tridimenzionalen prikaz s samoorganizirajočimi mrežami in dvodimenzionalen prikaz z Voronoijevim diagramom. Oba prikaza omogočata prostorsko interakcijo, s katero si lahko konceptualni prostor še lažje predstavljamo.

Ključne besede:

latentna semantična analiza, verjetnostna latentna semantična analiza, latentna Dirichletova alokacija, konceptualni prostor, semantika besede, vizualizacija, SOM, Voronoijev diagram

Abstract

In my thesis I am presenting an approach of conceptual spaces for vizulalization of text corpora. Thesis is divided into two parts. First part is overview of methods for text corpora analysis and the second one presents some ways for result vizualization.

Due to increasing number of eletronic data, we tend to automatic analisys and organisation of this data into various, pre-unknown groups. Some algorithms, that are providing us ways to do this, are presented (such as latent semantic analysis, probabilistic latent semantic analysis, latend Dirichlet Allocartion) further on in thesis.

We are looking for unknown topics, that arise in the text corpora. Text corpora is then analyzed with selected algorithm and presented in conceptual space. Conceptual space represents information by geometric structures: semantics of words are represented by points and relations between them are represented with regions. This suggests that word semantics is generated from concepts, that are represented as regions in conceptual space.

For vizualization of conceptual space of text corpora, I decided to use three dimensional representation with self-organizing maps and two dimensional representation with Voronoi diagram. Both representations allow spatial interaction, which can offer us easier way to imagine the conceptual space.

Key words: latent semantic analysis, probabilistic latent semantic analysis, latent Dirichlet allocation, conceptual space, word semantics, vizualization, selforganizing map, Voronoi diagram

Poglavje 1

UVOD

Hitrejši tempo življenja zahteva od nas, da si hitreje zapomnimo in učinkoviteje arhiviramo vse več informacij in podatkov. V industriji je stroj zaradi podobnih zahtev (hitreje, učinkoviteje) že zamenjal večino delavcev. Priznajmo si, da je človeško oko površno, roke počasne in telo utrudljivo. Vse to teži k uporabi sodobne elektronike - računalnikov, ki danes vse bolj in bolj krmilijo naše življenje.

Kot stroj zamenja delavca v industriji, bo tudi elektronska informacija počasi zamenjala tisto zapisano na papirju. Z večanjem števila elektronskih podatkov in informacij, pa se veča tudi želja po samodejni organiziranosti le - teh v tematske skupine. Za to pa potrebujemo primerne algoritme, ki znajo poiskati povezave med podatki.

V svojem diplomskem delu se posvečam osnovnim algoritmom za iskanje relacij med podatki in pa tudi načinom, kako dobljene rezultate prikazati.

V jezikoslovju, filozofiji in umetni inteligenci, je semantika besede (torej pomen besede) obravnavana kot povezava med jezikom in svetom ter predpostavlja, da beseda dobi svoj pomen šele glede na objekte in dogodke v svetu. Taka simbolična predstavitev za nas ni primerna, saj objektov in dogodkov v svetu ne poznamo vnaprej.

Rešitev, ki jo predlagata Gärdenfors [6] in Strle [15] je uporaba konceptualnega pogleda, v katerem so pomeni besed, predstavljeni kot preslikave besed na konceptualne strukture. To pomeni, da je pomen besede zgrajen iz konceptov, ki so predstavljeni kot regije v prostoru.

Konceptualni prostor je zgrajen iz geometričnih struktur v večdimenzionalnem prostoru. Cilj je poiskati in prepoznati dobre lastnosti elementov (na primer: barvo, rotacijo, temperaturo, težo, ...) in iz njih zgraditi domene, razrede za predstavitev konceptov. Taka struktura konceptov je metrična, kar pomeni, da lahko govorimo o razdaljah med dimenzijami. Obstaja povezava med razdaljo v prostoru in podobnostjo dveh elementov. Predpostavljamo, da manjša kot je razdalja, bolj sta si elementa podobna. To nam omogoča, da lahko s konceptualnimi prostori na naraven način predstavimo podobnosti med elementi. Naravni koncept je predstavljen kot množica regij v več utežnih domenah z informacijo, kako so različne domene medseboj povezane [5].

Problem konceptualnega iskalnika je v postavljanju mostu med podkonceptualnim nivojem, ki uporablja verjetnostne modele za prepoznavanje [1], [3] ,[8], [14] in konceptualnim nivojem.

1.1 Terminologija

Podatki uporabljeni v diplomi so besede vzete iz različnih besedil. Tako predstavlja:

- beseda - osnovno enoto diskretnih podatkov, element v slovarju;
- slovar - zbirka besed;
- dokument ali besedilo - zaporedje N-tih besed in
- korpus - zbirko M-tih dokumentov.

V splošnem temu ni tako. Element, kot osnovna enota diskretnih podatkov, je izbran tako, da ustreza raziskovalnemu področju.

1.2 Namen diplomske naloge

Namen naloge je izdelava osnovnega konceptualnega iskalnika, ki predstavlja in povezuje podkonceptualni nivo, v katerem zgradimo večdimenzionalni verjetnostni model za prepoznavo, s konceptualnim nivojem, s katerim nato predstavimo dimenzije (na primer z Voronoijevim diagramom). Vektorji z najboljšimi vrednostmi predstavljajo prototipe - to so centri regij, točke pa predstavljajo besede / objekte, regije pa teme in koncepte.

Za izdelavo podkonceptualnega modela in vektorskega prostora je uporabljeno orodje Matlab. Sama vizualizacija pa je realizirana v Javi s pomočjo knjižnic Processing in OpenGL.

1.3 Povzetek glavnih funkcionalnosti naloge

Cilj naloge je poiskati primerno vizualizacijo konceptualnega prostora dobljenega iz Matlaba. Nastali program vizualizira zbirko besedil z izrisom konceptualnih prostorov, hkrati pa omogoča iskanje po sami zbirki besedil tako po besedah, kot tudi po dokumentih. Za vsako iskanje nam prikaže, kam v konceptualni prostor spada.

Uporabila sem dve osnovni predstavitvi konceptualnega prostora: tridimenzionalno predstavitev s samoorganizirajočimi mrežami in dvodimenzionalno predstavitev z Voronoijevim diagramom.

V Voronoijevem diagramu se besede prikazujejo glede na projekcijo iz tri v dvodimenzionalen prostor in ne le v središče regije. Klik na posamezno regijo Voronoijevega diagrama povzroči, da se ta regija v tridimenzionalnem prikazu (torej na samoorganizirajoči mreži), prikaže v središču. Ker je Voronoijev diagram dvodimenzionalen prikaz, so tudi rotacije dvodimenzionalne, medtem ko je prikaz samoorganizirajoče mreže tridimenzionalen in se zato vrti v vseh treh dimenzijah okoli svojega središča. Voronoijev diagram se prikazuje ločeno, ali pa skupaj s samoorganizirajočo mrežo.

Iskanje besed oziroma dokumentov sem ločila od prikazovanja. Tako je nadzorni del ločen od prikaznega. Iščemo lahko tako besede, kot tudi dokumente. Za vsakega od njih, lahko določimo, na koliko točkah v prostoru naj se izpiše. Če želimo videti še verjetnosti besed ali dokumentov v posameznih regijah, si lahko le-te prikažemo s tortnim diagramom.

Omogočeno je tudi iskanje asociacij besed ali dokumentov. Z drsnikom določimo mero podobnosti, ki se nato uporabi pri iskanju asociacij.

Z iskanjem prvih nekaj besed ali dokumentov v posamezni regiji, si lahko olajšamo preimenovanje samodejno poimenovanih tem.

V pomoč iskanju je tudi samopadajoči meni, ki vsebuje slovar besed. Iskanja so lahko v prikazu ločena po barvah, velikostih in po različnih pisavah. Lahko si jih tudi shranimo ali zbrišemo.

Sledi pregled področja (poglavje 2). V poglavju 3.1 so opisane metode, s katerimi lahko opazujemo semantiko besed. Opisanih je nekaj možnih načinov vizualizacije besedil (poglavje 3.2). Sledi poglavje 4, ki je namenjeno opisu poteka izdelave programa, nabora dokumentov in samemu opisu programa. Zadnje poglavje 5 je namenjeno idejam s katerimi bi diplomsko delo še lahko razširili in nadgradili.

Poglavje 2

Pregled področja

Vse več podatkov in vse večja potreba po njihovem elektronskem arhiviranju, težijo k samodejnemu postopku označevanja in grupiranja le-teh. Da lahko to dosežemo, je potrebno algoritme naučiti prepoznavati razlike in podobnosti med podatki. Pri analiziranju besedil in dokumentov, nam podatki predstavljajo besede, le te pa se med seboj razlikujejo po pomenu.

Besede, ki jih avtorji besedil izberejo, da nam z njimi nekaj povedo, si tematsko ustrezajo (če ne upoštevamo t.i. "stop"besed[1]). Na primer, v člankih iz področja računalništva, se pojavljajo besede, ki so tako ali drugače povezane z računalništvom. Tematika besedila lahko predstavlja področje, v katerega spada besedilo, ali pa kakšna druga skupna točka besed v besedilu.

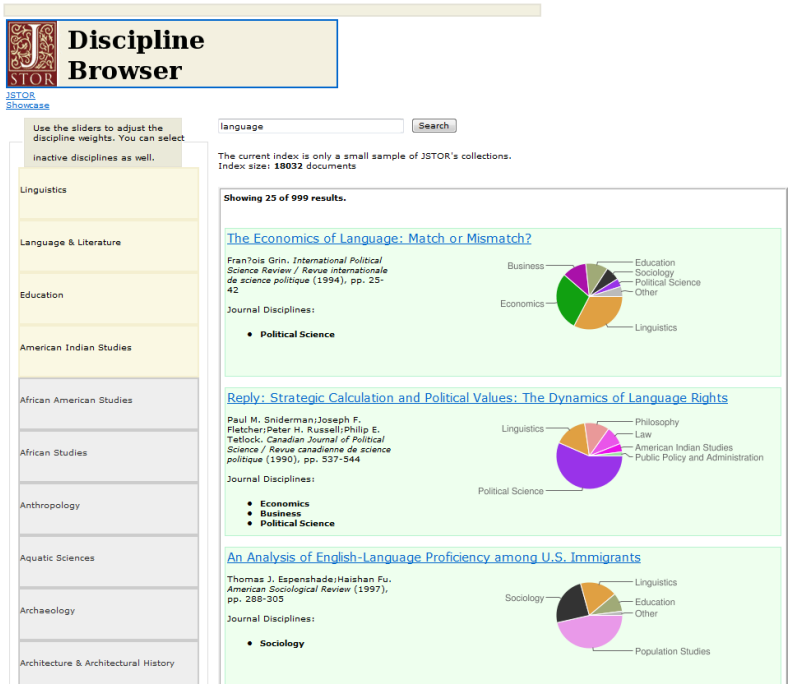
Za lažje predstavljanje, vzemimo sedaj tri tematsko različna besedila, za katera ne vemo, v katera področja znanosti spadajo. Da področja ugotovimo, moramo besedila prebrati. Ko jih preberemo, vemo, kateremu tematskemu področju pripadajo in jih lahko pravilno razvrstimo. Podobno počnejo tudi algoritmi. Le, da algoritmi nimajo "naše pameti"v razpoznavanju pomena besed, in ne morejo kar tako ugotoviti tematike besedila. Znano je, da se podobnost besed meri v podobnem obnašanju besed v besedilih. Naša tri besedila, bi tako podali algoritmu, hkrati pa bi mu povedali, da gre za tri različne tematike. Algoritem bi preštel število pojavitev vsake besede v posameznem besedilu in izračunal porazdelitev, kolikokrat se katera beseda v katerem od besedil pojavi. Tako nam algoritem lahko vrne le tipične besede (tiste, ki se največkrat ponovijo v njem) za posamezno besedilo. Mi lahko pregledamo le te besede (in ne celotnega besedila) in določimo tematiko.

[1] "stop"besede so besede, ki jih uporabljamo za povezovanje povedi v zaključeno obliko, na primer: glagol biti, brezosebne oblike,...

Algoritmi, ki jih uporabljamo za iskanje po pomenu podobnih si besed, so se v preteklih letih začeli hitro in uspešno razvijati. Posledica tega je, da se danes ti algoritmi uporabljajo na zelo različnih področjih, na primer za iskanje podobnih člankov [9], [11], [4], za časovni pregled tem v reviji Science [1], za predvidevanje in predlaganje ustrezne TV vsebine [12].

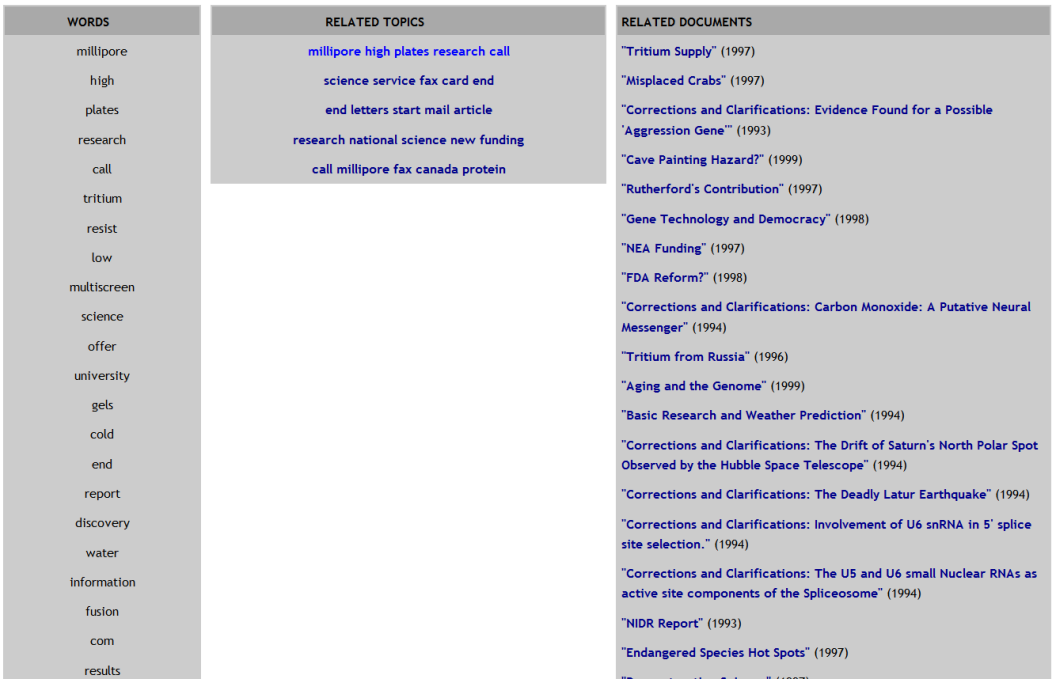
Uporabimo jih lahko tudi za pregled zgodovine ter za odkrivanje sočasnih odkritij in / ali zamujenih priložnosti. Z njimi pridobivamo in razvrščamo informacije. Pomagamo si lahko tudi pri prepoznavanju govora, prav tako pa tudi pri procesiranju naravnega jezika, na primer pri iskanju vzporednic med referencami besed in samimi besedami. V pomoč pridejo celo pri bibliotekarskih podatkih, ko želimo iskati knjige in članke po njihovem opisu.

Lep primer iskanja dokumentov po vsebovanosti posamezne teme je "Discipline Browser"[16]. V iskalno polje vpišemo temo oziroma področje, za katerega si želimo najti dokumente. Iskalnik nam vrne nekaj dokumentov, ki ustrezajo želenemu področju (slika 2.1). Nato lahko, na levi strani iskalnika, določamo uteži posameznim temam, iskalnik pa nam samodejno prilagaja vrnjene dokumente.



Slika 2.1: Semantični brskalnik Discipline Browser

Naslednji primer (slika 2.2) je nekoliko bolj neroden iskalnik člankov glede na temo iz revije "Science"[17]. To je primer modela iskalnika zgrajenega na stotih temah. Teme so opisane s petimi najverjetnejšimi besedami za to temo. S klikom na temo, se nam prikažejo še preostale besede, teme. Prikažejo se teme, ki so podobne izbrani temi in prikažejo se članki, ki pripadajo izbrani temi.



Slika 2.2: Enostavni semantični brskalnik

Pri "BBC Programs"[12] so si zaželeli, da bi lahko TV vsebino avtomatsko razporejali v žanre, glede na njihove kratke opise. Ugotovili so, da se poslušalci glasbe na *last.fm* pogosto strinjajo o čustvenih opisih glasbe. Povezava opisov glasbe iz družbenih omrežij in opisi glasbe, ki jo opisujejo, omogočajo definiranje visoko nivojskih kategorij, ki jih lahko nato uporabimo za opisovanje razpoloženja poslušalca. Iz tega so sklepali, da bi se lahko podobno dogajalo tudi pri gledanju televizijskega programa. Uporabili so algoritem LSA[2] nad pridevniki, ki opisujejo počutje ljudi ob gledanju televizije in meta podatke o posameznih TV vsebinah. Ugotovili so, da bi verjetno z omenjenim algoritmom lahko ugotovili kakšne TV vsebine gledalca zanimajo in mu nato, glede na ugotovljeno, predlagali TV vsebine vredne ogleda.

[2]ang. latent semantic analysis, več o njej v poglavju 3.1.1

Liangcui Shu, Bo Long, Weiyi Meng [13] so uporabili metodo LDA[3] za iskanje in za pripisovanje citatov posameznim avtorjem. Zaradi omejitev računskih virov, so se omejili na srednje velik podatkovni korpus[4], ki so ga dobili kot podmnožico iz digitalnih bibliografij - DBLP [5] bibliografij. Korpus je vseboval 82721 besed in 28678 imen avtorjev, velikost slovarja je bila 7736 in velikost slovarja avtorjev je bila 7470. Za ocenjevanje uspešnosti so uporabili dve meri:

$$natančnost \ = \ \frac{|S_a \cap S_r|}{|S_a|}$$

in

$$vpoklic \ = \ \frac{|S_a \cap S_r|}{|S_r|},$$

kjer je $S_a \ = \ (i,j)|$ *citata i in j sta sestavljena z algoritmom a, in velja* $i \neq j$ in $S_r = (i,j)|$ *citata i in j sta sestavljena v realnem, in* $i \neq j$.

Ali sta citata od istega avtorja je določil klasifikator. Uporabili so dva klasifikatorja: odločitvena drevesa (ang. decision tree) in klasifikator podpornih vektorjev (ang. support vector machines - SVM). Premagati so morali težavo, da si lahko morda več avtorjev deli isto ime oziroma imajo zelo podobno ime. Za učenje klasifikatorjev so uporabili 2200 parov citatov in avtorjev. Za vsak par so izračunali: podobnost imena soavtorja, podobnost naslova, podobnost teme, *venue similarity* in minimalno razlika v imenih soavtorjev. Za računanje podobnosti tem, so uporabili tri različne modele: dualni LDA, LDA in model brez tem, v katerem podobnosti tem niso računali in je zato tudi nadalje niso upoštevali. Ugotovili so, da sta klasifikatorja, ob upoštevanju tem, delovala bolje, kot sicer. Med dualnim LDA in navadnim LDA, je bil dualni LDA boljši.

Thomas K. Landauer, Darrell Laham in Marcia Derr so v članku [11] predlagali algoritem LSA za predobdelavo podatkov za potrebe vizualizacije. Algoritem so testirali na člankih iz PNAS [6], za vizualizacijo podatkov so uporabili program GGobi [7], ki omogoča prikazovanje visokih dimenzij (na primer več kot tridimenzionalen prostor). Program ljudem omogoča iskanje različnih pogledov, ki bi bili morda lahko zanimivi, in bi lahko odkrili, ali potrdili kakšno

[3] ang. latent Dirichlet alocation, več o njem v poglavju 3.1.3

[4] KORPUS je celotna zbirka besedil, ki ima nekaj skupnega, na primer vsa dela istega avtorja, vsa dela o neki temi

[5] DBLP - Digital Bibliografy & Library project: digitalne bibliografije in knjižni projekt

[6] Proceedings of the National Acedemy of Sciences

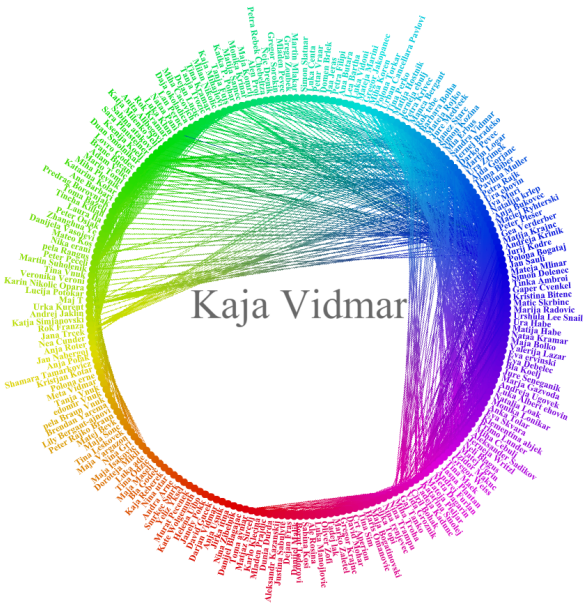
[7] www.ggobi.org

lastnost. Slabost programa se kaže v tem, da je izbor pomembnih dimenzij hevrističen in, da je program z večanjem dimenzij prostora vedno slabši. Ponavadi pa tudi ne sovpada z interesi raziskovalcev. Zato so se odločili, da lahko uporabnik določa začetne dimenzije in nadzira hitrost ter smer rotacije.

Za predstavitev podatkov je na voljo precejšnja izbira možnih vizualizacij od precej dolgočasnih diagramov do bolj zanimivih mrež podatkov.

Vizualno predstavljeni podatki, so nam v veliko pomoč, saj lahko v njih tako hitreje opazimo pomanjkljivosti in razne anomalije. Z vizualiziranimi podatki lahko odkrijemo bančne prevare [18]. Ali pa se le poigramo z družbenimi omrežji in si narišemo krog prijateljev (slika 2.3).

V [19] napovedujejo, da bo vizualizacija podatkov, zaradi zanimivih in ponavadi tudi nepričakovanih rezultatov, postala med ljudmi vse bolj priljubljena.



Slika 2.3: Analiza družabnega omrežja

Poglavje 3

Metodologija

Ideja izdelave semantičnega brskalnika temelji na kognitivističnem pristopu, v katerem je pomen besed sestavljen iz konceptov, ki tvorijo konceptualni prostor. Gärdenfors [5] predlaga predstavitev konceptov v geometrijskem prostoru z vektorji.

Vektorska predstavitev je matematična in zato primerna za simulacije na računalniku. Informacija (v našem primeru pomen besede) je predstavljena z geometrijsko strukturo - točko v večdimenzionalnem konceptualnem prostoru, ki je v resnici vektor. Zaradi vektorske predstavitve je matrična matematika primerna za izračune na konceptualnem nivoju.

3.1 Obdelava podatkov

Čedalje bolj težimo k odkrivanju zakonitosti in povezav v podatkih. Nekatere povezave so bolj očitne, druge manj. Tiste, ki so bolj, morda opazimo že v samem oblaku podatkov, vendar nas ponavadi bolj zanimajo, tiste, ki so skrite. Na pomoč nam priskočijo različne tehnike in algoritmi, ki nam znajo te skrite povezave razkriti. Le-ti so nam v pomoč tudi, ko si zaželimo, da bi računalnik znal sam razvrstiti podatke v pravilno skupino oziroma razred podatkov, glede na podobnost. To pa pomeni, da bi moral algoritem, ki nam podatke razvršča, vedeti, kateri podatki so si med seboj podobni.

Modeliranje tem je klasičen problem v odkrivanju informacij. Informacije lahko odkrijemo s tehnikami kot so: latentna semantična analiza (ang. latent semantic analysis) oziroma latentno semantično indeksiranje (ang. latent semantic indexing), verjetnostna latentna semantična analiza (ang. probabi-

lity semantic analysis) oziroma verjetnostno latentno semantično indeksiranje (ang. probabilistic latent semantic indexing), neodvisna analiza komponent, latentna Dirichletova alokacija (ang. latent Dirichlet allocation) ...

3.1.1 Latentna semantična analiza

Latentna semantična analiza, v nadaljevanju LSA, je algoritem za procesiranje naravnega jezika, ki med dokumenti in besedami išče in analizira povezave. Med njimi ustvarja množice konceptov: dokument - beseda, beseda - beseda, tema - dokument, beseda - tema. Za-to uporablja vektorsko oziroma matrično semantiko.

Leta 1988 so jo patentirali Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum in Lynn Streeter [10]. Kadar se algoritem uporabi za pridobivanje informacij, mu pravimo tudi latentno semantično indeksiranje - LSI.

Matrika pojavitev

LSA uporablja matriko pojavitev beseda - dokument, ki opisuje pojavitve besed v dokumentih. Tipično uteževanje elementov matrike je FB-IFD (frekvenca besed - inverzna frekvenca dokumentov), kar pomeni, da je element matrike odvisen od števila svojih pojavitev v posameznem dokumentu. Zato so lahko besede, ki se malokrat pojavijo precenjene in lahko s tem pokažejo svojo morebitno relativno pomembnost. Matrika take oblike je standardna tudi za ostale semantične modele, a ni vedno izražena v obliki matrike.

Naj bo X matrika (3.1), kjer element (i, j) opisuje pojavitev besede i v dokumentu j (na primer s frekvenco). X zgleda takole:

$$\begin{bmatrix} t_{1,1} & \cdots & t_{1,n} \\ \vdots & \ddots & \vdots \\ t_{m,1} & \cdots & t_{m,n} \end{bmatrix} \tag{3.1}$$

Vrstica v matriki (3.1) je pripadajoč vektor besedi, ki predstavlja relacijo te besede z vsakim od dokumentov: $t_i = [x_{i,1} \cdots x_{i,n}]$. Korelacijo med dvema vektorjema besed preko dokumentov nam določa skalarni produkt $t_i^T t_p$. Element (i, p) je enak elementu (p, i) in vsebuje skalarni produkt $t_i^T t_p$ $(= t_p^T t_i)$. Opazimo, da matrični produkt XX^T vsebuje vse take skalarne produkte. Skalarni produkti $t_i^T t_p$ med dvema vektorjema nam določajo korelacijo med besedami preko dokumentov.

Podobno vektor stolpca v matriki X (3.1) pripada dokumentu, ki predstavlja relacijo dokumenta z vsako od besed: $d_j^T = [x_{i,j} \dots x_{m,j}]$. Korelacija preko vseh besed je določena z $d_i^T d_p$ $(= d_p^T d_i$)), matrika $X^T X$ pa vsebuje vse take skalarne produkte. Podobno matrika $X^T X$ vsebuje vse skalarne produkte med vsemi vektorji dokumentov in nam podaja korelacijo preko vseh besed: $d_i^T d_p$ $(= d_p^T d_i$)) .

Nižanje ranga matrike

Po izgradnji matrike pojavitev, LSA uporabi aproksimacijo z nižanjem ranga matrike. Razlogi za to so različni:

- Za računske vire je lahko matrika pojavitev hitro prostorsko prevelika in časovno prezahtevna. Dobljena matrika z nižjim rangom je aproksimacija originalne matrike (najmanjše in najnujnejše zlo).
- Originalna matrika pojavitev je lahko šumna. Dobljena aproksimacijska matrika je manj šumna in zato boljša od originala.
- Originalna matrika pojavitev vsebuje besede, ki so obstajajo v vsakem dokumentu, medtem ko nas morda zanimajo vse besede v povezavi z vsakim dokumentom.

Z nižanjem ranga matrike pojavitev pričakujemo, da se bodo združile dimenzije povezane s podobnimi pomeni besed. Pri iskanju sopomenk in večpomenk, nam to lahko povzroča težave.

$$\{(drevo), (grm), (avto)\} \rightarrow \{(1.28 * drevo + 0.25 * grm), (avto)\} \qquad (3.2)$$

To namiguje tudi na problem z besedami z več pomeni, saj so komponente teh besed, ki kažejo v pravo smer, dodane h komponentam besed s katerimi si delijo podoben pomen. Nasprotno, komponente, ki kažejo v druge smeri težijo k izginotju, ali v najslabšem primeru so manjše od komponent v smeri, ki ustreza pričakovanemu smislu.

Ob nižanju ranga matrike predpostavimo, da obstaja taka dekompozicija matrike X, da sta U in V ortogonalni matriki, in Σ diagonalna matrika. To imenujemo eno vrednostna dekompozicija - SVD: $X = U\Sigma V^T$. Matrična produkta korelacij besed in dokumentov tako postaneta :

$$\begin{aligned} XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = \\ (U\Sigma V^T)(V^{T^T}\Sigma^T U^T) = \\ U\Sigma V^T V\Sigma^T U^T = \\ U\Sigma\Sigma^T U^T \end{aligned} \tag{3.3}$$

$$\begin{aligned} X^T X = (U\Sigma V^T)^T(U\Sigma V^T) = \\ (V^{T^T}\Sigma^T U^T)(U\Sigma V^T) = \\ V\Sigma^T U^T U\Sigma V^T = \\ V\Sigma^T\Sigma V^T \end{aligned} \tag{3.4}$$

Ker sta $\Sigma\Sigma^T$ in $\Sigma^T\Sigma$ diagonalni matriki, mora U vsebovati lastne vektorje matrike XX^T, V pa mora vsebovati lastne vektorje matrike $X^T X$. Oba matrična produkta imata enake nenegativne lastne vrednosti, podane z nenegativnimi elementi matrike $\Sigma\Sigma^T$ oziroma $\Sigma^T\Sigma$. Kako zgleda SVD dekompozicija je prikazano v enačbi (3.5).

$$\begin{array}{c} \mathrm{X} \\ (d_j) \\ \downarrow \end{array}$$

$$(t_i^T) \rightarrow \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} =$$

$$\begin{array}{ccc} U & \Sigma & V^T \\ & & (\hat{d}_j) \\ & & \downarrow \end{array}$$

$$(\hat{t_i^T}) \rightarrow= \begin{bmatrix} \begin{bmatrix} \\ u_1 \\ \\ \end{bmatrix} & \cdots & \begin{bmatrix} \\ u_l \\ \\ \end{bmatrix} \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix} \begin{bmatrix} [& v_1 &] \\ & \vdots & \\ [& v_l &] \end{bmatrix} \tag{3.5}$$

$\sigma_1 \cdots \sigma_l$ imenujemo singularne vrednosti, $u_1 \cdots u_l$ in $v_1 \cdots v_l$ pa sta levi in desni singularni vektor. Opazimo, da le i-ta vrstica matrike U prispeva k t_i. Označimo to vrstico $\hat{t_i}$. Podobno le j-ti stolpec matrike V^T prispeva k d_j. Označimo ga $\hat{d_j}$. To niso lastni vektorji, ampak so odvisni od vseh lastnih vektorjev.

Izkaže se, da če izberemo k največjih singularnih vrednosti in njihove pripadajoče singularne vektorje iz U in V, dobimo matriko ranga k, ki aproksimira X z najmanjšo napako. Še pomembneje pa je, da lahko sedaj vektorje besed in dokumentov opazujemo kot konceptualni prostor tem dokumentov.

Vektor $\hat{t_i}$ ima k elementov. Vsak izmed njih prikazuje pojavitev besede i v vsaki od k tem. Podobno vektor $\hat{d_j}$ podaja relacijo med dokumentom j in vsako temo. Aproksimacijo lahko zapišemo kot $X_k = U_k \Sigma_k V_k^T$.

Sedaj lahko opazujemo:

- Kako sta si povezana dokumenta j in q v konceptualnem prostoru tem dokumentov s primerjanjem vektorjev d_j in d_q - dobimo gručo dokumentov.

- Kako sta si povezani besedi i in p s primerjanjem vektorjev t_i in t_p - dobimo gručo besed v konceptualnem prostoru tem dokumentov.

- Kako je "mini" dokument v relaciji z dokumenti v konceptualnem prostoru.

  To dosežemo tako, da "mini" dokument najprej transformiramo v konceptualni prostor.

$$\begin{aligned} d_j &= U_k \Sigma_k^T \hat{d_j} \\ \hat{d_j} &= \Sigma_k^{-1} U_k^T d_j \end{aligned} \tag{3.6}$$

  To pomeni, če imamo povpraševalni vektor q, moramo narediti translacijo $\hat{q_j} = \Sigma_k^{-1} U_k^T q_j$ predno ga primerjamo z vektorji dokumentov v našem konceptualnem prostoru. Podobno lahko storimo tudi za besede:

$$\begin{aligned} t_i^T &= \hat{t_i}^T \Sigma_k V_k^T \\ \hat{t_i}^T = t_i^T V_k^{-T} \Sigma_k^{-1} &= t_i^T V_k \Sigma_k^{-1} \\ \hat{t_i}^T &= \Sigma_k^{-1} V_k^T t_i \end{aligned} \tag{3.7}$$

Uporaba konceptualnega prostora

Konceptualni prostor lahko uporabimo za:

- Primerjanje dokumentov v njem (klasifikacija dokumentov, gručenje dokumentov).
- Iskanje podobnih dokumentov med jeziki.
- Iskanje relacij med besedami (sinonimi, večpomenke).
- Primerjanje novega dokumenta z dokumenti v prostoru.
- Iskanje podobnosti med malimi skupinami besed v semantičnem smislu.
- Iskanje sopomenk in večpomenk. Vendar iskanje morda ne najde vseh sopomenk. Pri iskanju večpomenk, pa nam lahko vrne nepomembne dokumente, ki vsebujejo iskano besedo, saj ima večpomenka več pomenov v različnih konceptih (na primer: gori na gori gori).

Omejitve LSA

- Manjšanje dimenzij, lahko pripelje do težje ali celo nemogoče interpretacije rezultatov. Na primer v (3.2) smo še sposobni logično oceniti prvo dimenzijo kot množico rastline, medtem ko pri:

  $$\{(drevo), (kamera), (avto)\} \rightarrow \{(1.28 * kamera + 0.25 * avto), (drevo)\} \tag{3.8}$$

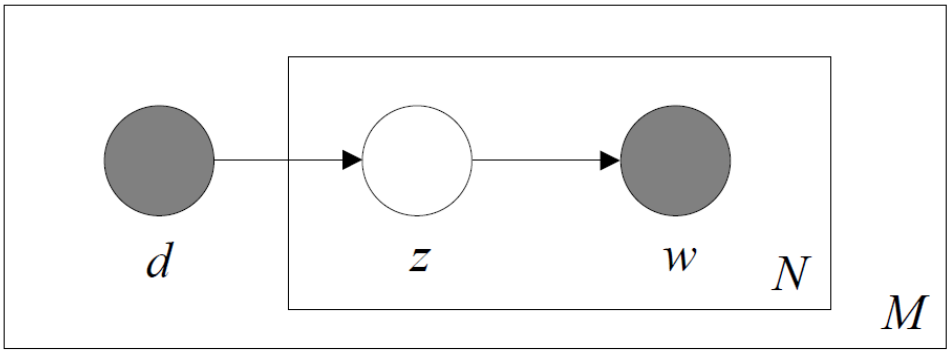
  rezultata ne moremo logično interpretirati.
- LSA se ne zaveda večpomenk, vsaka beseda je predstavljena le z enim pomenom / točko v prostoru.
- BOW (ang. bag of words - prevod vreča besed) je model predstavitve podatkov, kot neurejen seznam besed. Omejitve tukaj so, da model ne določa nobene povezave med posameznimi besedami (torej ni gramatičnih pravil). Edini uporabni informaciji, ki nam jih model ponuja sta obstoj besede in njena frekvenca.
- LSA predvideva, da so besede in dokumenti porazdeljeni po Gaussovi porazdelitvi, medtem ko se dejansko opaža Poissonova porazdelitev. Zato LSA morda ne odraža pravih relacij med besedami in dokumenti ter temami. To popravlja novejša različica - verjetnostni LSA - PLSA.

3.1.2 Verjetnostna latentna semantična analiza

Verjetnostna latentna semantična analiza , v nadaljevanju PLSA, je statistično orodje za analizo sopojavitvenih podatkov. Je v relaciji z ne negativno matrično faktorizacijo. In temelji na mešanici dekompozicije dobljene iz latentnega razreda modela. To rezultira k bolj prvinstvenemu pristopu, ki ima trdne temelje v statistiki.

Če algoritem PLSA uporabimo za pridobivanje informacij, ga podobno kot pri LSA, imenujemo verjetnostno latentno semantično indeksiranje - PLSI. Leta 1999 sta jo predstavila Jan Puzicha and Thomas Hofmann [9].

Naj bo d spremenljivka za dokument, z spremenljivka za temo, potem s $P(z|d)$ označimo porazdelitev tem za dokument d. Naj bo w je beseda pobrana iz porazdelitve besed po temah - $P(w|z)$. Spremenljivki d in w opazujemo, z pa je latentna spremenljivka (slika 3.1).



Slika 3.1: Predstavitev PLSA modela

Če opazujemo sopojavitve oblike par beseda - dokument (w, d), PLSA modelira verjetnost vsake sopojavitve kot mešanico pogojno neodvisnih multinomskih porazdelitev:

$$P(w,d) = \sum_z P(z)P(d|z)P(w|z) \tag{3.9}$$

$$P(w,d) = P(d)\sum_z P(z|d)P(w|z) \tag{3.10}$$

Prva formula (3.9) je simetrična. w in d sta oba dobljena iz latentnega razreda z s pogojnimi verjetnostmi. Medtem ko je druga formula (3.10) asimetrična. Latentni razred z je za vsak dokument d izbran pogojno glede na

porazdelitev $P(z|d)$, beseda je potem dobljena iz dobljenega razreda d glede na porazdelitev $P(w|z)$.

Omejitve PLSA

- PLSA model ima težave s prevelikim prilagajanjem podatkom. Isti model uporabljen na drugačnih podatkih bo zaradi tega morda napačno opisoval relacije.
- Število parametrov hitro raste, saj raste linearno s številom dokumentov.
- Četudi je PLSA generativen model dokumentov v zbirki na katerih je ocenjen, ni generativen model novih dokumentov. To deloma reši Latentna Dirichletova alokacija. Porazdelitev tem v dokumentu je pravilna le za dokumente iz učne množice.
- Ne zgradi verjetnostnega modela na nivoju dokumentov.
- PLSA lahko zlahka razširimo, modelira lahko sopojavitve tudi treh in več spremenljivk. Če za to uporabimo simetrično formulacijo, to naredimo tako, da enostavno dodamo pogojne verjetnosti porazdelitev teh dodatnih spremenljivk.

3.1.3 Latentna Dirichletova alokacija

Latentna Dirichletova alokacija, v nadaljevanju LDA, je generativni model, ki uporablja trinivojsko hierarhijo. Vsaka beseda je zgrajena iz naključnih tem, le te pa so zgrajene iz naključnih porazdelitev. Na primer, če so opazovane besede zbrane v dokumentih, potem določa, da je vsak dokument mešanica nekaj tem, ter da je vsaka beseda atribut neki temi iz dokumenta. Pri čemer je dokument predstavljen z porazdelitvami tem. LDA so leta 2002 predstavili David Blei, Andrew Ng in Michael Jordan [3].

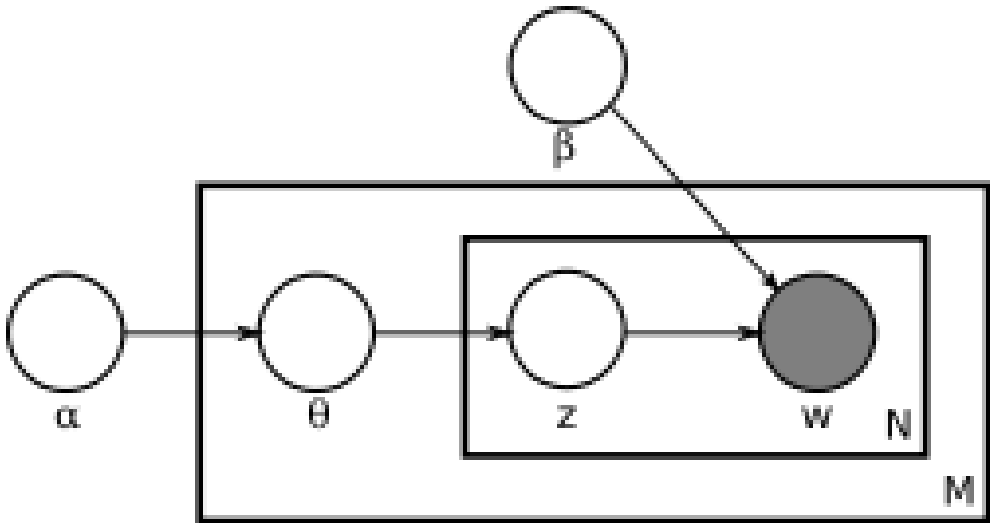
Teme v LDA

Na vsak dokument v LDA gledamo kot mešanico več tem, podobno kot pri PLSA v 3.1.2. Glavna razlika je v tem, da predpostavljamo, da se porazdelitev tem obnaša podobno Dirichletovi porazdelitvi (3.11). Tako dobimo bolj razumljive mešanice tem v dokumentu, saj lahko tudi neznani dokument preslikamo v LDA prostor dokumentov.

$$f(x_1, \cdots, x_K; \alpha_1, \cdots, \alpha_K) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}; \; K \geq 2 \, in \, \alpha_1, \cdots, \alpha_K \geq 0 \tag{3.11}$$

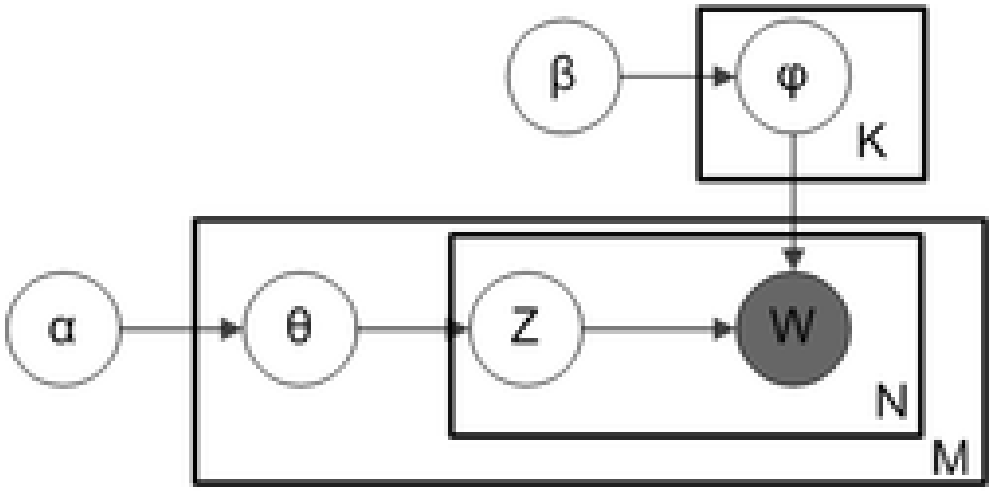
Na primer, imamo LDA model s temama mačke in psi. K temi mačke lahko pripišemo besede kot so *mijav*, *mucek*, *maček*, ... Beseda *maček* bo imela visoko verjetnost, da pripada k temi mačke. Po drugi strani lahko pričakujemo, da bodo imele besede kot so *psiček*, *laježi*, ... veliko verjetnost, da pripadajo k temi psi. Za besede, ki niso pomembne (na primer vezniki, mašila,...), pa pričakujemo, da bodo imele približno enako verjetnost pripadnosti vsem temam.

Model



Slika 3.2: Predstavitev LDA modela

Na sliki 3.2 je predstavljen LDA model. α je parameter uniformne Dirichletove prior porazdelitve tem po dokumentih, β parameter uniformne Dirichletove porazdelitve besed po temah. Θ_i je porazdelitev tem za dokument i, z_{ij} je tema za j-to besedo in i-ti dokument, w_{ij} pa izbrana beseda. Opazujemo le w_{ij}, vse ostale spremenljivke so latentne.



Slika 3.3: Predstavitev mehkega LDA modela

Na sliki 3.3 je prikazan malce drugačen model, ki ponavadi zagotavlja boljše rezultate. K označuje število opazovanih tem v modelu, φ pa je $K * V$ Markova matrika (matrika verjetnosti prehoda v naslednje stanje), v kateri vrstica predstavlja porazdelitev besed po temah, V pa je dimenzija slovarja.

Glede na opisani model LDA, poznamo polno verjetnost modela:

$$P(W, Z, \Theta, \varphi; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\Theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\Theta_j) P(W_{j,t}|\varphi z_{j,t}) \tag{3.12}$$

Želimo oceniti verjetnost teme ob opazovanih besedah glede na dano porazdelitev besed po temah in glede na dano porazdelitev dokumentov po temah: $P(Z|W; \alpha, \beta)$. Z integriranjem po Θ (porazdelitvi tem po dokumentih) in φ (porazdelitvi besed po temah) znamo izraziti verjetnost tem in besed glede na dani porazdelitvi dokumentov in besed po temah: $P(Z, W; \alpha, \beta)$. To verjetnost aproksimiramo z enačbami Gibbsovega vzorčenja (ang. Gibbs sampling) in ker se $P(W|\alpha, \beta)$ ne spremeni za noben Z, jih lahko dobimo direktno iz $P(Z, W; \alpha, \beta)$.

Opis algortima

1. Izberi N po Poissinovi porazdelitvi besed
2. Izberi Θ po Dirichletovi spremenljivki dokumentov
3. Za vsako od besed w_n (1,..., N):
   - Izberi temo z_n $\quad Multinomial(\Theta)$
   - Izberi besedo w_n glede na multinomsko porazdelitev glede na temo z_n : $\quad P(w_n \mid z_n, \beta)$

Namesto Dirichletove porazdelitve, lahko uporabimo tudi katero drugo. Na primer korelacijski model tem, uporabi logistično normalno porazdelitev.

3.1.4 Primerjava modelov

- LSA na manjših množicah podatkov deluje bolje kot PLSA, saj se izogne prevelikemu prilaganju podatkom.
- PLSA model bolje opisuje relacije v podatkih kot LSA.
- LDA model je ekvivalenten PLSA modelu, če pri PLSA modelu zgradimo model tudi na nivoju dokumentov in uporabimo Dirichletovo porazdelitev.
- LSA model za računanje podobnosti uporablja normo L_2, medtem ko PLSA uporablja sosednostno funkcijo multinomskega vzorčenja.
- LDA model se izogne prevelikemu prilagajanju podatkom in podaja bolj razumljivo porazdelitev tem v dokumentih tudi za neznani dokument (dokument izven učne množice).
- PLSA model uporablja predpostavko iz BOW, da vrstni red besed v dokumentu ni važen. LDA pa tudi predpostavko, da vrstni red dokumentov v korpusu ni važen.

Metodi LSA in PLSA so natančneje primerjali v [9]. Da so metodi lahko primerjali, so morali najprej pridobiti podatek o verjetnosti iz navadnega LSA modela.

Če pri PLSA označimo verjetnost dokumenta d_i ob temi z_k $(P(d_i|z_k))_{i,k} = \hat{U}$, verjetnost besede w_i ob temi z_k $(P(w_i|z_k))_{i,k} = \hat{V}$ in verjetnostno porazdelitev tem $(P(z_k))_k = \hat{\Sigma}$, in vse skupaj zapišemo kot: $P = \hat{U}\hat{\Sigma}\hat{V}$, opazimo, da dobimo eno vrednostno dekompozicijo, ki se uporablja pri LSA.

Ugotovili so, da PLSA obeta boljše rezultate kot LSA.

3.2 Vizualizacija

Ljudje si lažje razložimo pojave, pojme, če si jih predstavljamo. Pisatelji za predstavitev svojega pogleda na svet uporabljajo besede, matematiki uporabljajo števila, umetniki slike, risbe in barve ...

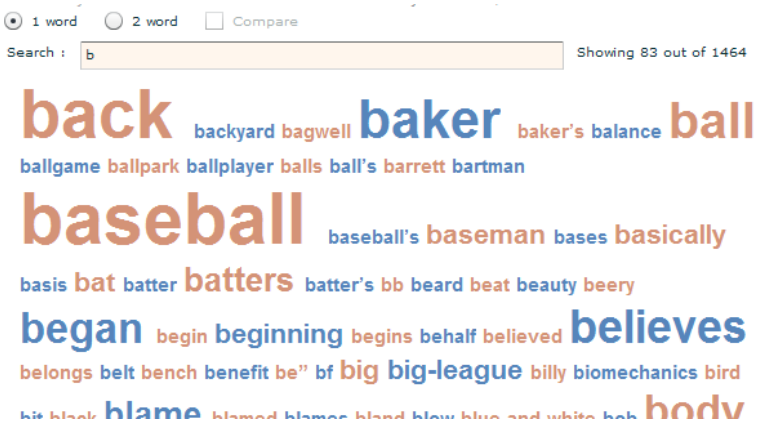
Vizualizacija podatkov nam omogoča, da prikažemo in razkrijemo povezave med podatki in tako lažje najdemo intuitivno razlago zakaj in kako so si nekatere stvari med seboj povezane.

Vizualizacija besedil oziroma povezav med besedami v dokumentih nam lahko razkrije strukturo jezika. Z njo lahko prikažemo značilnosti jezika, besedila, značilnosti pisatelja, ali pa ugotovimo katere besede se v katerih temah najpogosteje pojavljajo. Vsekakor so vizualizirani podatki očem bolj prijazni in razkrijejo marsikatero povezavo in lastnost, ki bi jo sicer lahko spregledali.

Načinov prikaza povezav med podatki je kar nekaj, naj omenim le nekaj najpogostejših.

Oblak značk

Oblak značk (ang. tag cloud) je vizualizacijska metoda, ki besede oziroma opisnike besed uredi po pomembnosti, po neki raziskovalcem pomembni meri (slika 3.4). Ta mera je lahko preprosto število ponovitev te besede v dokumentu, povezave besed med seboj in podobno. Za prikaz razlik uporablja različno velikost besed in/ali odtenke barv.



Slika 3.4: Oblak značk

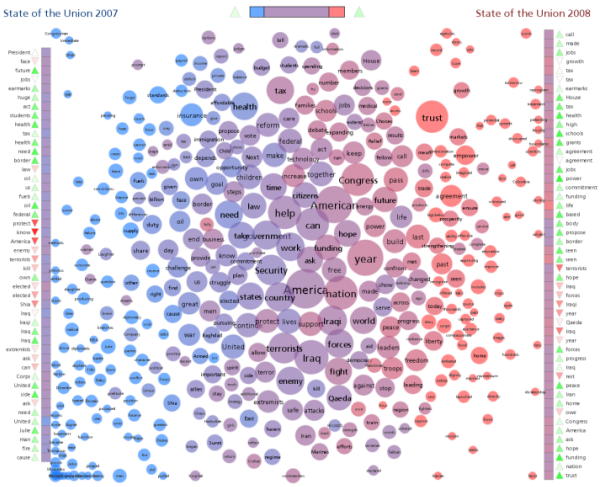
Diagram spektra besed

Z diagramom spektra besed (ang. word spectrum diagram) lahko prikažemo povezavo med besednimi pari (slika 3.5). Z barvo in velikostjo besede opišemo njeno pripadnost eni ali drugi besedi.

Slika 3.5: Spektrični diagram besed american in chinese

Diagram kontrasta za dokumente

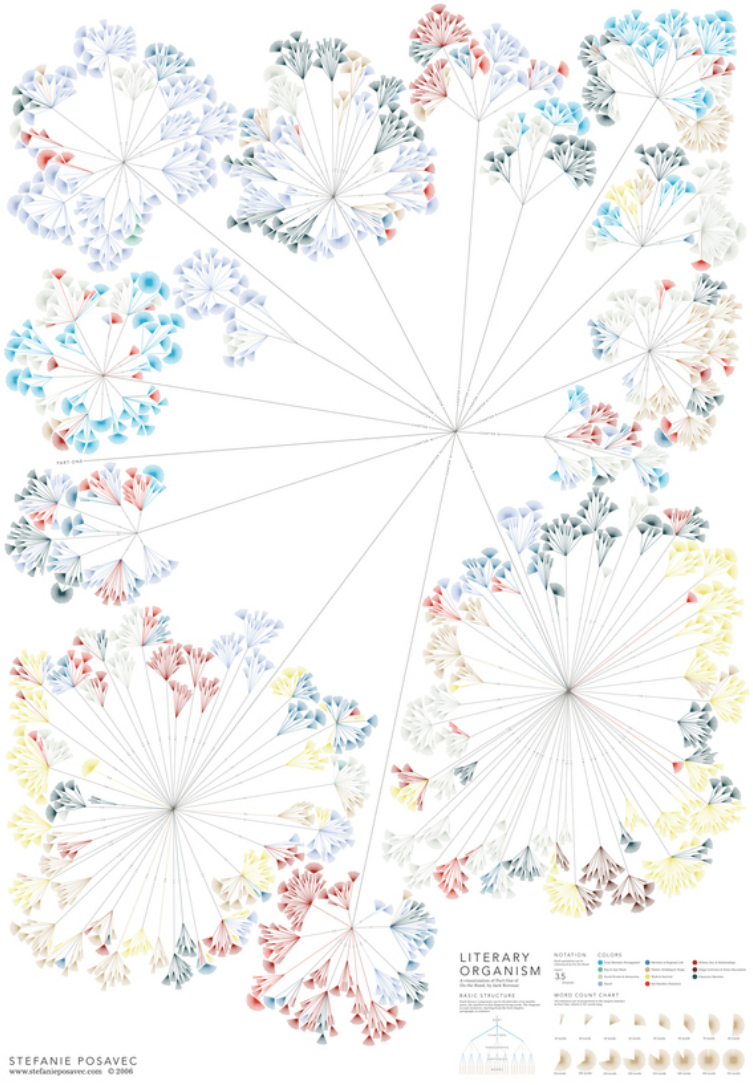
Diagram kontrasta za dokumente (ang. document contrast diagrams) uporablja tehniko mehurčkov (slika 3.6). Velikost mehurčka predstavlja, verjetnost pripadnosti določenemu besedilu. Z barvo pa napoveduje, kateremu besedilu beseda pripada. Iz diagrama mehurčkov lahko razberemo podobnosti in razlike med prikazanima dvema besediloma.



Slika 3.6: Diagram kontrasta za dokumente

Literarna mreža organizmov

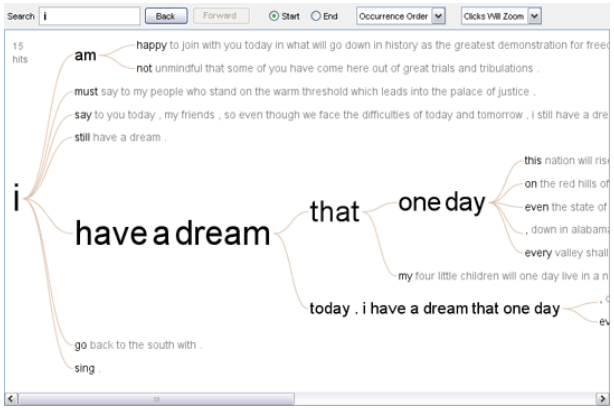
Z umetniškega vidika zanimivejša tehnika vizualizacije je literarna mreža organizmov (ang. Literary Organism Map) (slika 3.7). Predstavlja nekakšen raztegljiv zemljevid, v katerem se vsako poglavje prikaže glede na njegovo temo. Z njo lahko najdemo dele dokumentov, ki se ponavljajo in pa tudi kje in kolikokrat se ponovijo.



Slika 3.7: Literarna mreža organizmov

Drevo besed

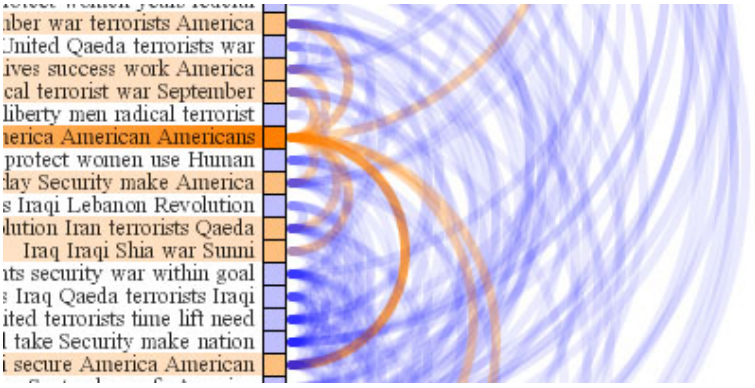
S prikazovanjem relacij med glavnimi besedami, frazami in njihovimi nasledniki v drevesu besed (ang. word tree) lahko že na prvi pogled ugotovimo kontekst k navidez neurejenemu besedilu. Velikost črk prikazuje pomembnost in relativno frekvenco med pari besed. Omogoča lahko iskanje s sledenjem poti od enega do drugega konteksta čez besedilo teksta (slika 3.8).



Slika 3.8: Drevo besed

Diagram puščic za dokument

Če v besedilu najdemo vse dele besedila, ki so medsebojno povezani, in jih povežemo z lokom, dobimo diagram puščic (ang. Document Art Diagrams) (slika 3.9). Omogoča, da se povezave v besedilu hitro pokažejo.



Slika 3.9: Diagram puščic za dokument

3.2.1 Processing

Processing je programski jezik, ki omogoča uporabo programskega okolja v katerem lahko enostavnejše vizualizacije hitro sprogramiramo in to z le malo predhodnega znanja o programiranju. Omogoča tudi integracijo z Javo, kar pride prav naprednejšim uporabnikom za zahtevnejše vizualizacije.

3.2.2 Samoorganizirajoče mreže

Samoorganizirajoče mreže (ang. self organizing maps), v nadaljevanju SOM (slika 3.10), so le eno izmed možnih umetnih omrežij. Predstavil jih je finski profesor T. Kohonen, zato jim včasih pravimo tudi Kohonenove mreže. Mreža iz nenadzorovanega učenja zgradi [20] diskretno predstavitev v nizki dimenziji. Od ostalih umetnih omrežij se loči tako, da uporablja funkcijo sosednosti, da lahko ohrani topološke značilnosti vhodnih učnih primerov. Zato so primerne za vizualizacijo večdimenzionalnih prostorov v več nizko dimenzionalnih pogledih.

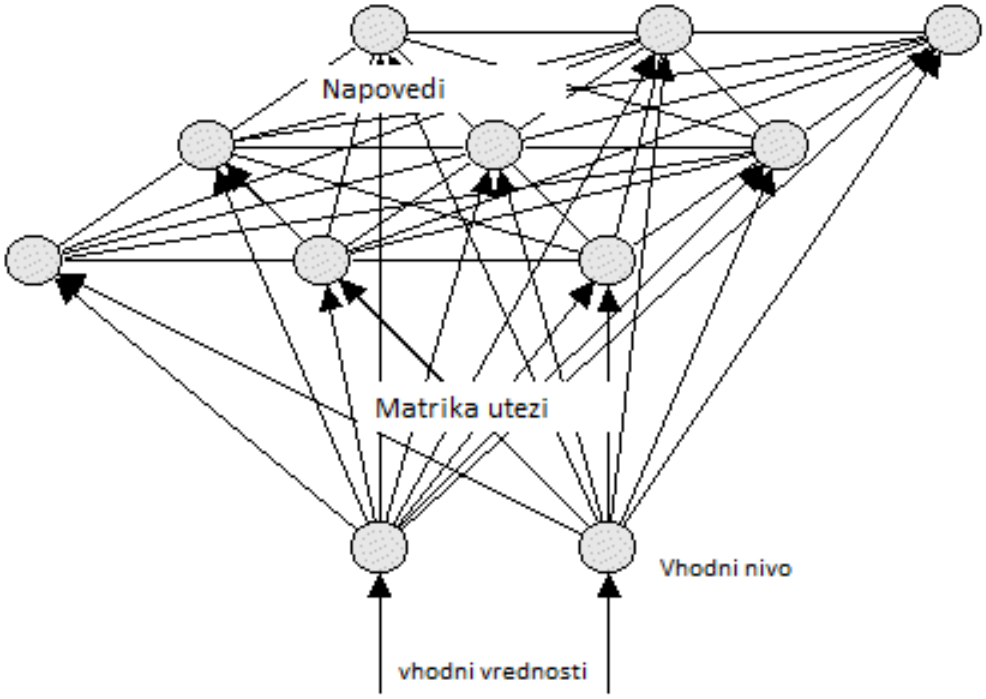
SOM je zgrajen iz vozlišč, ki jim pravimo tudi nevroni. Z njimi so povezani utežni vektorji z isto dimenzijo kot vhodni učni primeri in lokacijo v prostoru mreže. Vozlišča so navadno razporejena v heksagonalni ali pravokotni mreži.

Z vektorsko kvantizacijo SOM zgradi mrežo iz učnih primerov. Vsak nov podatek je nato samodejno klasificiran tako, da je preslikan v vozlišče, ki ima najbližji utežni vektor, vektorju podatka iz vhodnega prostora. S tem je omogočena preslikava iz višje dimenzionalnega vhodnega prostora v nižje dimenzionalen prostor mreže.

Algoritem

Želimo si, da bi mreža delovala podobno kot delujejo naši možgani. Torej, da različni deli mreže reagirajo podobno na določene vhodne podatke.

Da mrežo nečesa naučimo, ji moramo podati veliko vhodnih učnih podatkov, ki so kar se da blizu vektorjem, ki jih bomo preslikavali. Uteži vektorjev vozlišč nastavimo na majhne naključne vrednosti ali pa jih enakomerno vzorčimo iz pod prostora tovorjenega iz dveh največjih lastnih vektorjev. Uporablja se tekmovalno učenje, kar pomeni, da vsa vhodna vozlišča dobijo enak vhod, sprejme pa ga le eden. Izračuna se Evklidska razdalja vhodnega primera do vseh utežnih vektorjev. Glede na vozlišče, ki se najbolje ujema z vhodnim vektorjem (imenujmo ga v^*; ta vektor je tudi sprejel vhodni primer) se nato popravijo uteži utežnih vektorjev. Velikostni razred spremembe se zmanjšuje s časom in razdaljo od najboljšega:



Slika 3.10: Samoorganizirajoča mreža

$$W_v(t+1) = W_v(t) + \Theta(v,t)\alpha(t)(D(t) - W_v(t)) \tag{3.13}$$

Kjer je $W_v(t)$ utežni vektor, ki ga popravljamo, $\alpha(t)$ je monotono padajoč učni koeficient, $D(t)$ vhodni vektor in $\Theta(v,t)$ funkcija sosednosti. $\Theta(v,t)$ je odvisna od razdalje med vozliščem v in v^* in se z časom oža. Na začetku vpliva na vsa vozlišča, nato pa le na nekaj vozlišč, katere uteži konvergirajo k lokalnim ekstremom.

Postopek večkrat ponovimo za vsak vhodni vektor.

Psevdokoda

1. Določi začetne vrednosti uteži utežnih vektorjev.
2. Vzemi vhodni vektor.
3. Dokler $t < \lambda$; (λ je št. iteracij):
4. Za vsako vozlišče v mreži izračunaj:
   - evklidsko razdaljo z vhodnim vektorjem in
   - označi vozlišče z najmanjšo razdaljo.
5. Popravi uteži utežnih vektorjev po enačbi 3.13
6. Povečaj t.

Lastnosti samoorganizirajoče mreže

- Prostor uteži dobro aproksimira vhodni prostor.
- Topološka urejenost: lokacija ustreza legi oziroma lastnosti vhodnega primera.
- V večji gostoti verjetnosti ustreza večje področje nevronov oziroma vozlišč.
- Podobnost z delovanjem možganov.

3.2.3 Voronoijev diagram

Voronoijev diagram je način prikaza za katerega velja, da razdeli prostor na konveksne regije tako, da je vsak element znotraj posamezne regije najbližji centru vsake regije. Postopku, ki zgradi Voronoijev diagram, pravimo Fortuneov algoritem [21].

Ideja algoritma v dvodimenzionalnem prostoru:

Naj bo P množica točk za katero želimo zgraditi Voronoijev diagram. Pometamo premico l preko točk v prostoru.

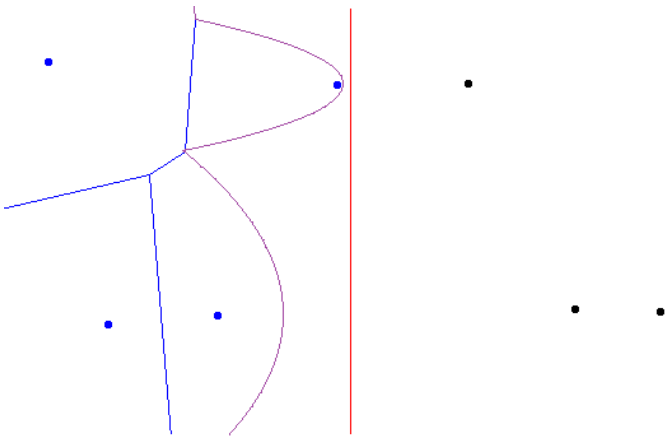
Predpostavimo:

- da se stanje Voronoijevega diagrama na levi strani premice l ne spreminja več;
- za točke $x \in \mathbb{R}^2$ in točko $s \in P$ na levi strani premice l že poznamo regijo oziroma celico kamor pripadajo: $x \in V(s)$; $V(s)$ je celica Voronoijevega diagrama.

Naj za vsako točko p naj velja, da je γ_p parabola.

Slika 3.11: Parabola Voronoijevega diagrama

Naj bo $P_l = P \cap \{x \in \mathbb{R}^2 |\, x\, levo\, od\, l\}$ in naj bo
$R_l = \{x \in \mathbb{R}^2 \mid x\, levo\, od\, \gamma_p\, za\, neko\, p \in P_l\} = \cup_{p \in P_l}\{tocke\, levo\, od\, \gamma_p\}$. Za točke $x \in R_l$ vemo, v katero celico Voronoijevega diagrama spadajo.

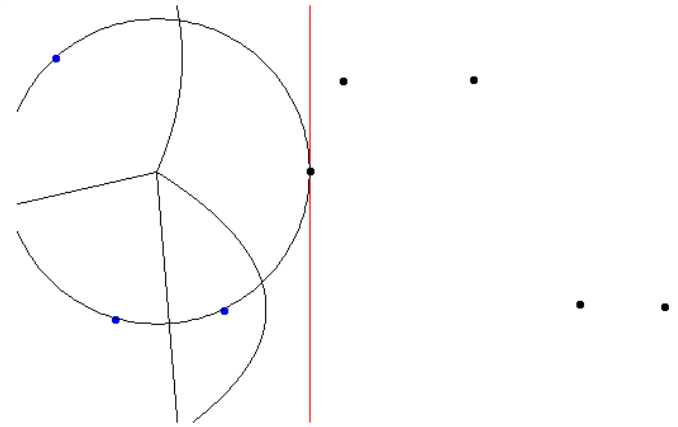


Slika 3.12: Točke na desni strani premice in "beach line"

Mejo med R_l in $\mathbb{R}^2 R_l$ imenujemo "beach-line"[1]. Na sliki 3.12 je označena z vijolično barvo. Ko se beach-line spremeni, pravimo, da smo dobili dogodek. Le-ta je lahko *dogodek vstavljanja*, ob katerem velja, da smo dobili nov kos parabole v "beach-line,"le ko gre l skozi neko točko $p \in P$ ali pa je *dogodek izbrisa*, ki ta se zgodi, ko so si γ_a, γ_b in γ_c zaporedne v "beach-line" in velja, da γ_b ne spada več v "beach-line".

[1] obalna črta

(a) Dogodek vstavljanja



(b) Dogodek brisanja

Slika 3.13: Voronoi dogodek vstavljanja in brisanja

Na začetku vstavimo v prednostno vrsto dogodke vstavljanja. Ko spreminjamo "beach-line,"opazujemo nove zaporedne tri kose "beach-line"in preverjamo ali mogoče definirajo nov dogodek, če ga, potem je le - ta dogodek brisanja.

Poglavje 4

Predobdelava podatkov za vizualizacijo in opis aplikacije

Izdelava konceptualnega iskalnika je zamišljena kot interaktivno orodje, ki omogoča semantično (pojmovno) brskanje in predstavitev vsebine preko topologije konceptualnih prostorov [5].

Konceptualni prostor razdelimo na konveksne regije glede na množico prototipov - besed, ki so najbolj značilne za določeno domeno. Elementi, ki pripadajo isti domeni, so si bolj podobni, kot elementi iz različnih domen, vendar pa ni nujno, da so si elementi v isti domeni enakovredni. Podobnost med elementi je izražena z razdaljo v konceptualnem prostoru. Element je v konceptualnem prostoru predstavljen s točko. Bližje, ko sta si točki, ki predstavljata elementa v prostoru, bolj sta si elementa podobna. Z iskanjem in primerjanjem razdalj med elementi v prostoru, lahko tako raziskujemo podobnosti in relacije med elementi. Regija v konceptualnem prostoru predstavlja relacije (povezave) med elementi.

Zaradi uporabe konveksnih regij, se zdi Voronoijev diagram primeren način vizualizualizacije prostora. Vendar ni primeren za večdimenzionalno (več kot tridimenzionalno) predstavitev, zato je potrebna preslikava večdimenzionalnih podatkov v nižje dimenzije. Vmesni korak je lahko izračun SOM mreže, ki nam zna iz večdimenzij zgraditi trodimenzionalen prostor, v katerem nam predstavi podatke. Nato pa trodimenzionalen prostor preslikamo v dvodimenzionalnega z ortografsko projekcijo in ga uporabimo za izračun Voronoijevega diagrama.

4.1 Nabor dokumentov

Testni korpus vsebuje 1.430.209 besed in okoli 560 dokumentov tematsko različnih vsebin. Slovar besed vsebuje 13588 različnih besed, ob upoštevanju da so bile odstranjene vse angleške " stop " besede (na primer: a, an, the, ...)

4.2 Priprava datotek

Programu za vizualizacijo pomena in podobnosti besed, ne moremo kar podati prvotnega besedila. Besedilo moramo preoblikovati in pripraviti tako, da ga program zna ustrezno interpretirati. Primerno obliko in potrebne datoteke, ki jih program za vizualizacijo sprejeme, dobimo z algoritmi, ki smo jih opisali v prejšnjem poglavju. Vendar, tudi za te algoritme, moramo besedilo primerno pripraviti. Za to moramo najprej narediti podatkovni korpus besedil.

Matlab skripta **analyzeCorpus** sprejme zbirko besedil zapisanih v *.xml* datoteki. Ta datoteka mora imeti sledečo strukturo:

```
<CORPUS>
<DOC>
  <TITLE>  </TITLE>
  <TEXT>   </TEXT>
</DOC>
</CORPUS>
```

Skripta prebere *.xml* datoteko skupaj z datoteko, v kateri so napisane "stop"besede in zgradi podatkovni korpus, sestavljen iz matrike beseda dokument, imen dokumentov, slovarja vseh besed, uteži..., pri čemer izpusti vse "stop"besede.

Nato uporabi enega od algortimov LSA, PLSA ali LDA. Nastaviti moramo število iskanih tem - parameter "NumberOfTopics." Rezultat algoritma vrne matriki "Pw_d"in "Pz_d", to sta porazdelitvi pripadnosti besed k temi in tem k dokumentom.

Na lastne oči iz dobljenih matrik nismo sposobni videti povezav med elementi. Zato si želimo matriki vizualizirati. Kar pomeni, da moramo večdimenzionalen prostor preslikati v nižje dimenzionalen prostor, ki si ga znamo predstavljati. Kot že omenjeno, lahko to storimo s SOM mrežami.

Za izračun SOM mreže moramo skripti povedati, na katerih podatkih naj jo računa - na matriki besed ali na matriki dokumentov, ter ji določiti, kako velika naj bo. Privzeto je velikost nastavljena na 10x10, glede na število podatkov, pa jo lahko povečamo ali zmanjšamo. Za preslikavo se privzeto uporablja analizo glavnih komponent (ang. principal component analysis, PCA), uporabimo pa lahko tudi Sammonovo preslikavo. Pred samim izračunom SOM, vrednosti matrike normaliziramo in izračunamo normalizirana središča tem, ki jih uporabimo v izračunu Voronoijevega diagrama.

Rezultate nato shranimo s skripto **saveData**. Dobimo datoteke:
.somprojection.txt je datotka z tridimenzionalnimi koordinatami vozlišč SOM mreže, v datotekah *.worddictionary.txt* in *.docdictionarz.txt* sta zapisana slovarja besed oziroma dokumentov , v datotekah *.wordprojection.txt*, *.docprojection.txt* je zapisano kako se besede oziroma dokumenti preslikajo na vozlišča SOM mreže, v datoteki *.topicprojection.txt* so zapisana vozlišča, ki predstavljajo središča tem, v datoteki *.somcolor.txt* pa so zapisane barve za vsako od vozlišč na SOM mreži.
Dobljene datoteke lahko sedaj podamo programu za vizualizacijo.

4.3 Opis aplikacije

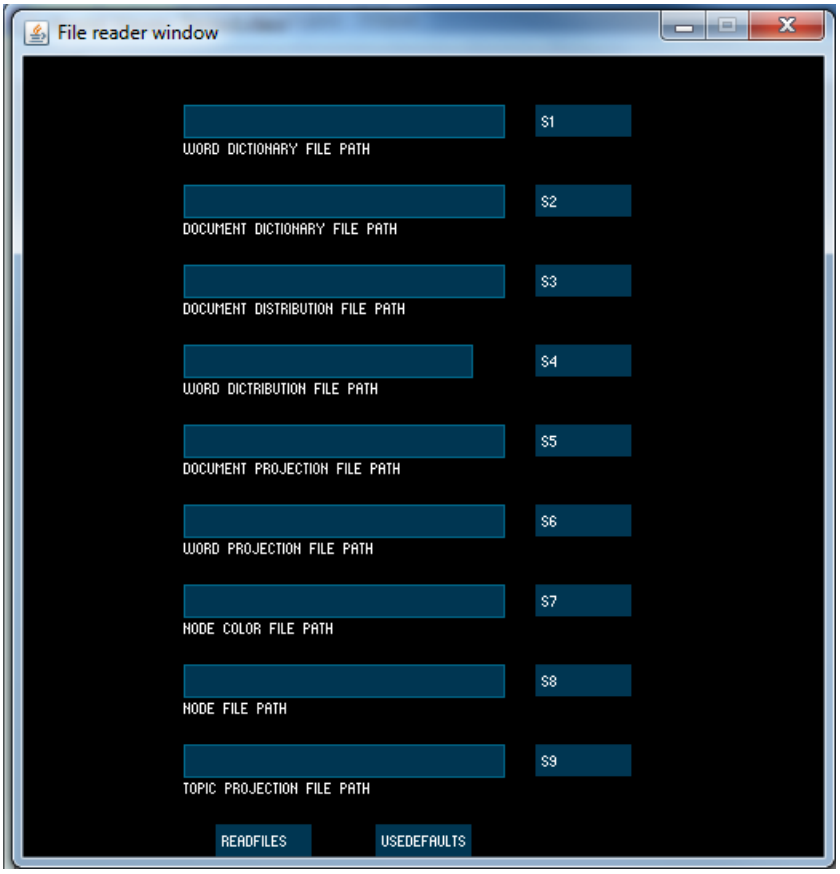
Namen programa je prikazati podatke in olajšati iskanje povezav med njimi. Program sprejme datoteke, ki smo jih ustvarili z Matlabom v prejšnjem poglavju.

4.3.1 Povezava datotek s programom

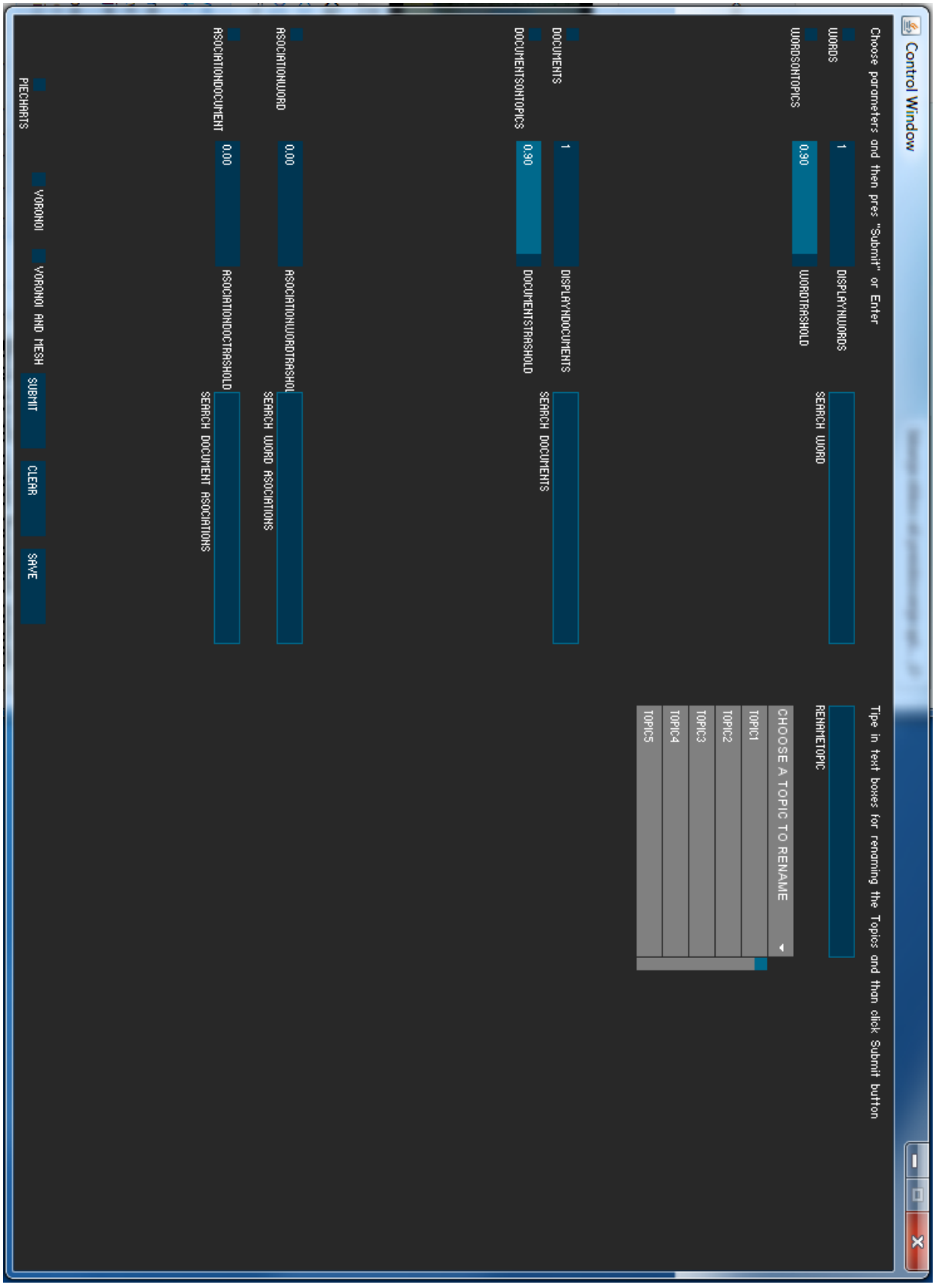
Na začetku se odpre pogovorno okno (slika 4.1), ki zahteva od nas, da vsako od datotek z obdelanimi podatki povežemo s programom. Če želimo, lahko uporabimo tudi privzete poti do datotek.

4.3.2 Upravljanje programa

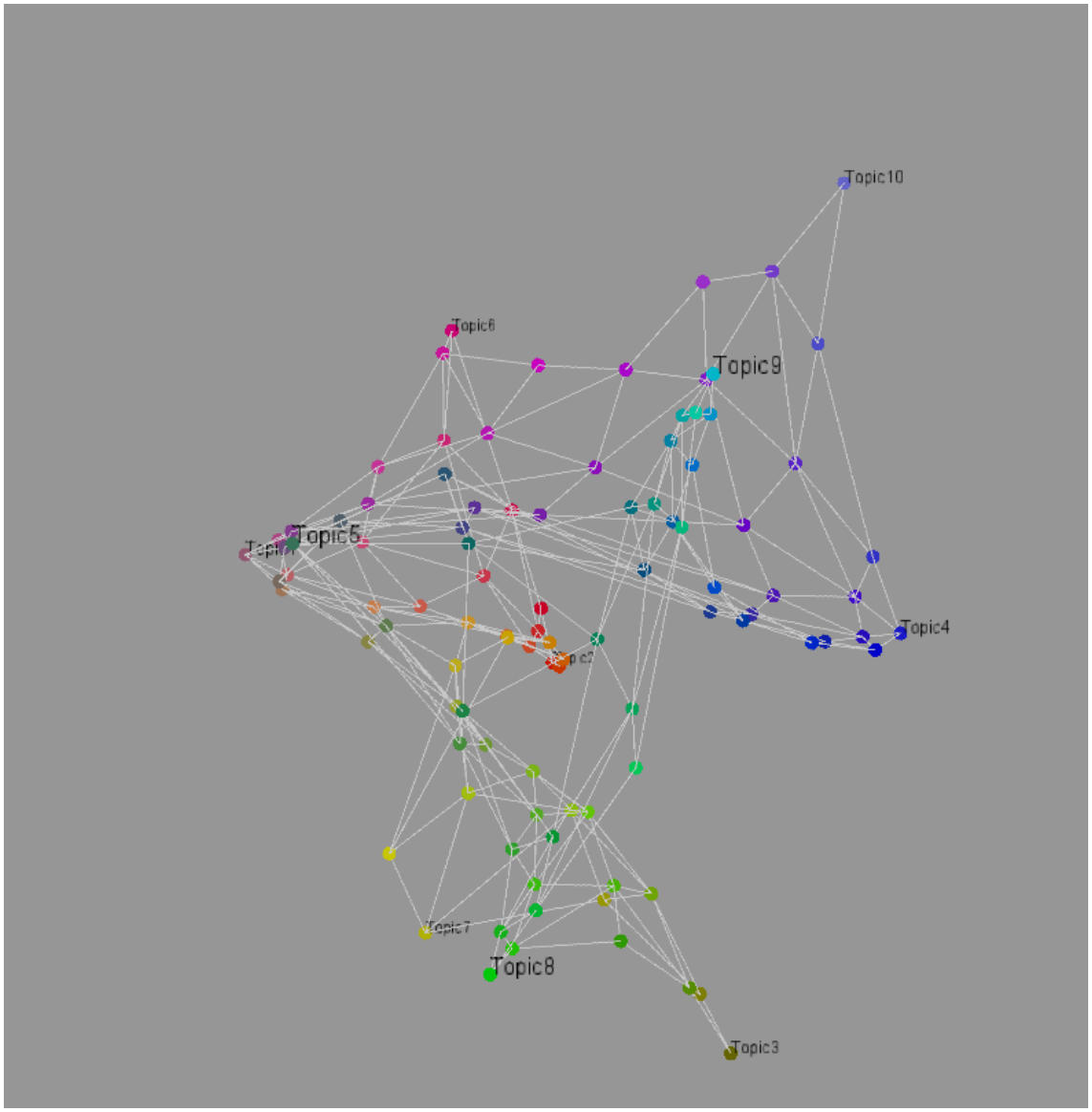
Program upravljamo iz nadzornega okna (slika 4.2) v katerem so vse nadzorne kontrole zbrane skupaj. Spodaj v kontrolnem oknu imamo kontrole, ki spreminjajo in omogočajo prikaz izbranih besed in njihovih lastnosti. Privzeti način prikaza je SOM mreža (slika 4.3(a)). Izberemo si lahko tudi prikaz Voronoijevega diagrama (slika 4.3(b)) ali pa sestavljeni prikaz Voronoijevega diagrama in SOM mreže (slika 4.4).



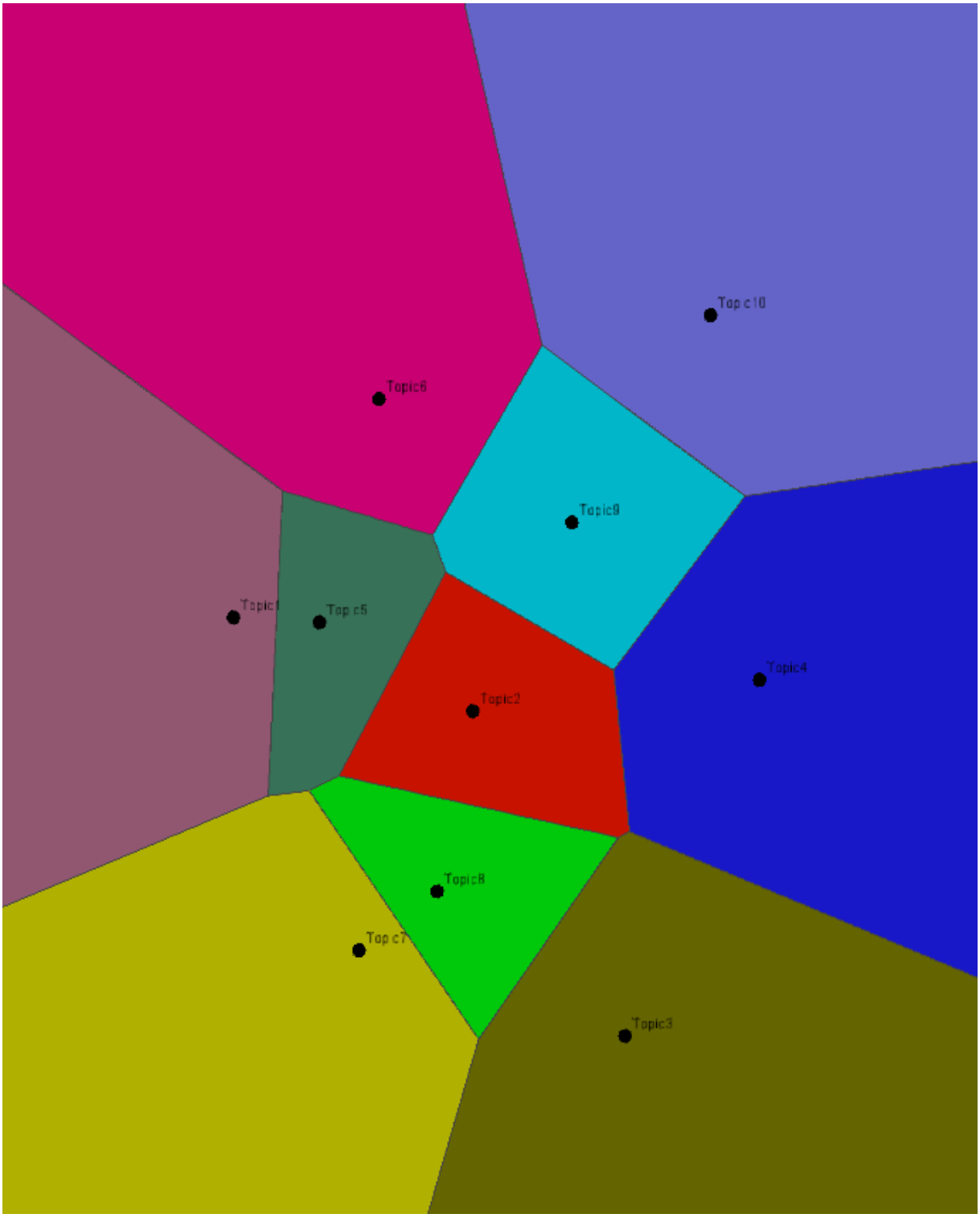
Slika 4.1: Okno za določitev datotek



Slika 4.2: Nadzorno okno

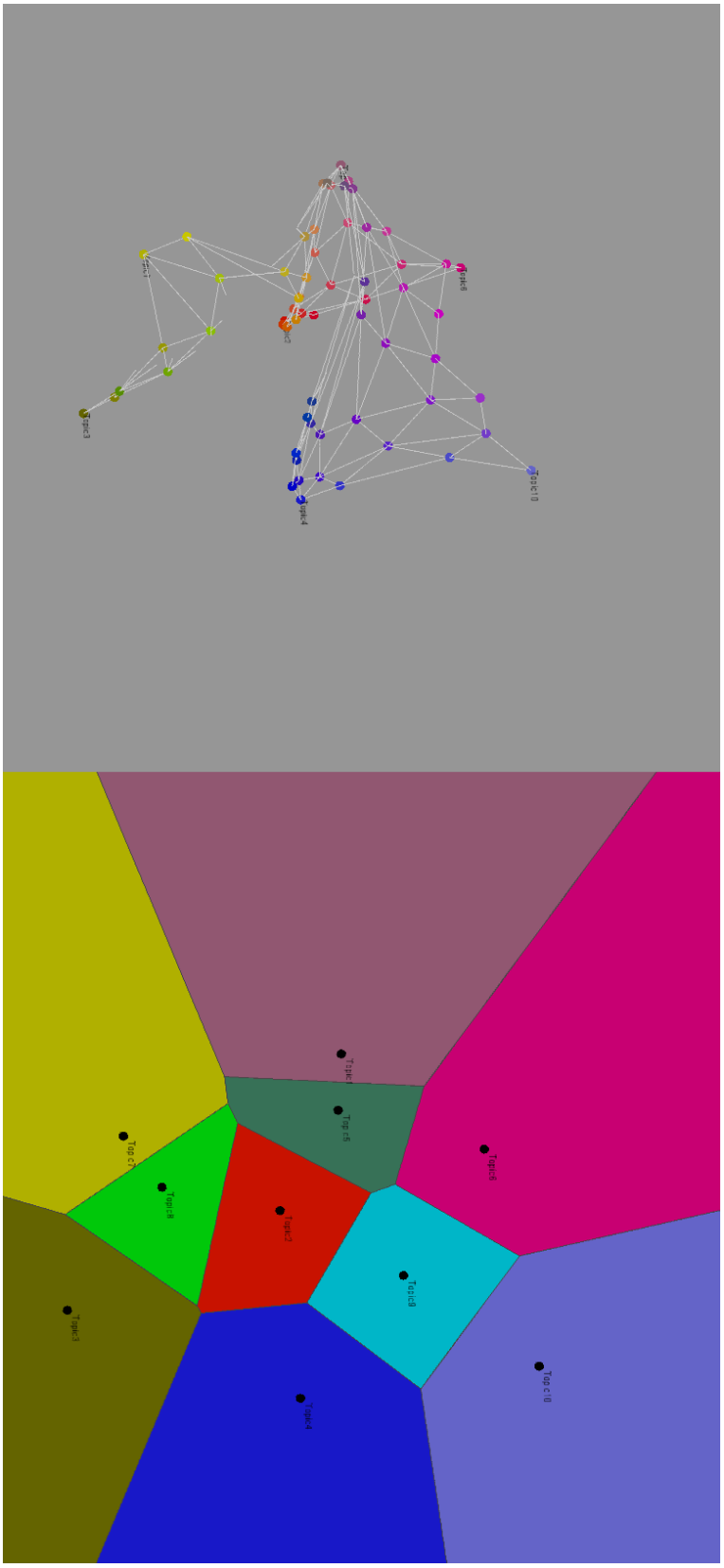


(a) Privzeti prikaz SOM mreže



(b) Privzeti prikaz za Voronoijev diagram

Slika 4.3: Privzeta prikaza za SOM in Voronoijev diagram

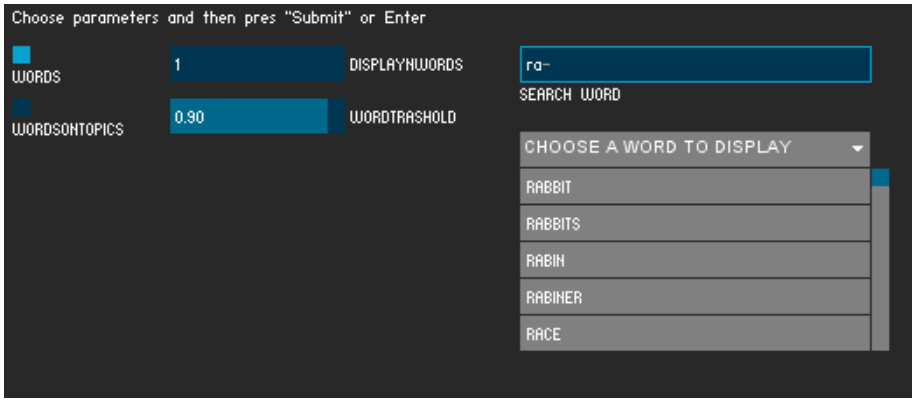


Slika 4.4: Prikaz SOM mreže in Voronoijevega diagrama

Iskanje besed

Program omogoča iskanje posamezne besede v korpusu in nam jo izriše na SOM mreži in ali Voronoijevem diagramu. Skupaj z besedo se prikaže še tortni diagram, ki nam pove, kolikšno verjetnost ima ta beseda na izpisani temi.

Če ne poznamo vseh besed v korpusu, lahko napišemo prvo ali prvi dve črki besede in pritisnemo **- minus**. Prikaže se lista besed, ki se začnejo na vpisane črke (slika 4.5). Posamezne iskane besede so izpisane s črno barvo.



Slika 4.5: Prikaz izbire besed

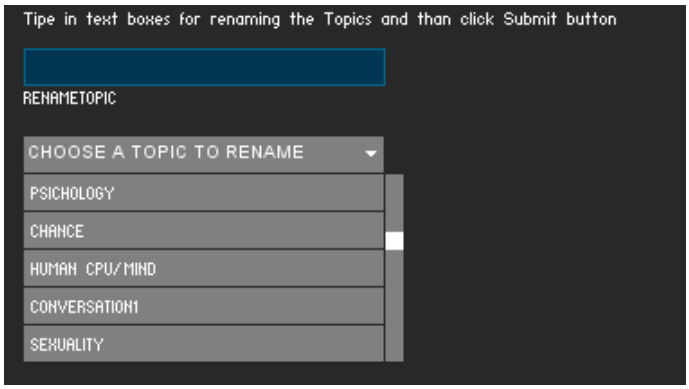
Iskanje najpogostejših besed v temi

Program omogoča tudi iskanje prvih n besed na za posamezno temo. Izpiše se prvih 5 najmočnejših (tistih z najvišjo verjetnostjo). Besede se izpišejo na vozlišču, ki predstavlja središče teme. Barva izpisanih besed je ista kot barva vozlišča. To uporabniku omogoča, da ugotovi tematiko teme.

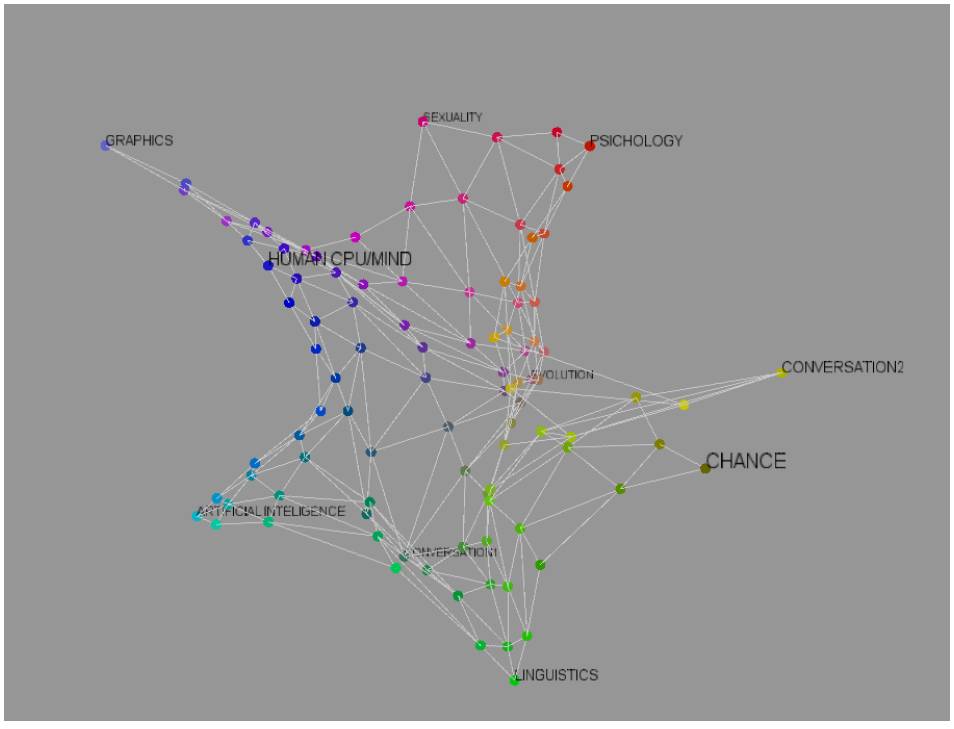
Na primer: na vozlišču *Topic* 4, se izpišejo besede: *memory, cortex, brain, neurons...* (prevod : spomin, korteks, možgani, nevroni ...), iz tega lahko sklepamo, da je to vozlišče povezano s tematiko delovanja možganov. S spreminjanjem praga uravnavamo, kako dobre morajo biti besede, da jih še smatramo za dovolj verjetne za posamezno temo.

Poimenovanje tem

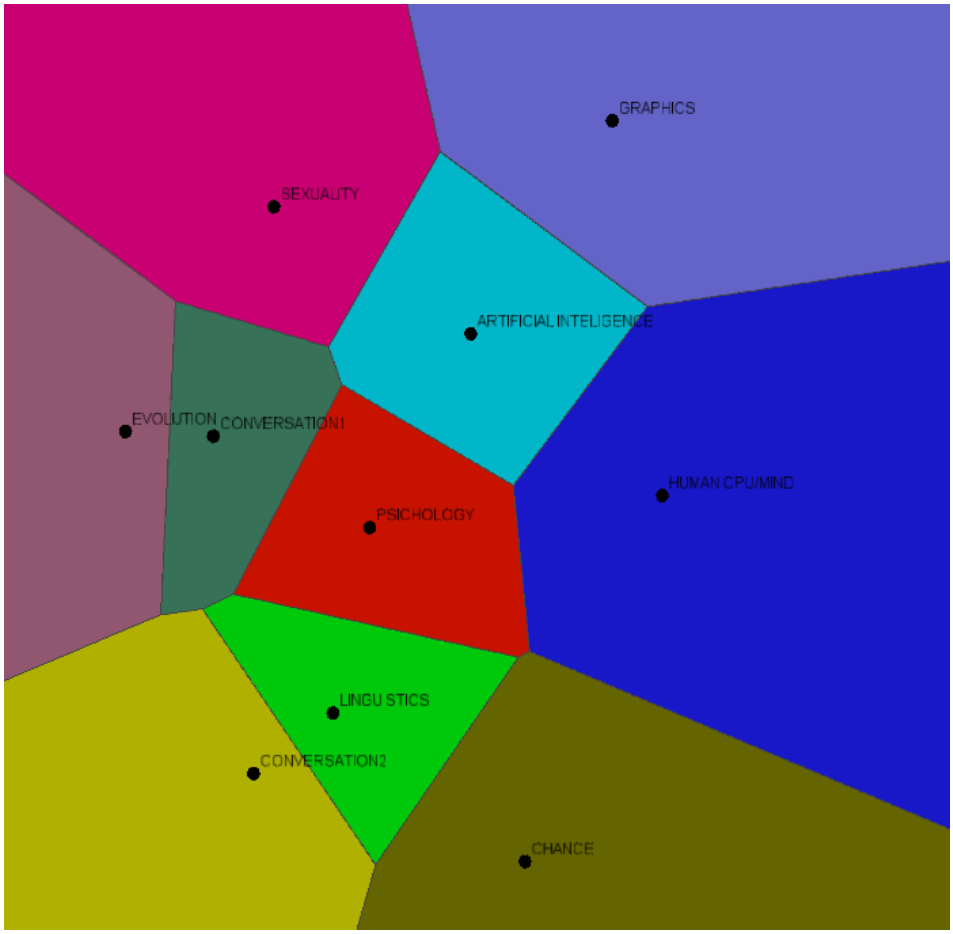
Da uporabniku ni potrebno vedno znova iskati besed za posamezno temo, program omogoča tudi preimenovanje vozlišč. Tako bi lahko vozlišče *Topic* 8 preimenovali v **HUMAN CPU** (prevod: človeška CPE oz. spomin).



Slika 4.6: Kontrola za preimenovanje tem



(a) SOM s preimenovanimi temami



(b) Voronoi s preimenovanimi temami

Slika 4.7: Preimenovanje tem

Iskanje asociacij

Program omogoča tudi iskanje asociacij, to je besed, ki so si po verjetnostni porazdelitvi po temah podobne. Iskana beseda se izpiše kot največja, ostale pa so glede na svojo podobnost iskani besedi nekoliko manjše od nje. S pragom ponovno določamo, koliko mora biti beseda podobna iskani besedi. Podobnost besed je izračunana s kosinusno podobnostjo:

$$\cos\Theta = \frac{a * B}{\|a\| * \|B\|} \tag{4.1}$$

Bolj, ko je $\cos\Theta$ enak 1, bolj podobni sta si besedi.

Niz iskanja asociacij je v celoti izpisan z enako barvo, ki se določa naključno.

Vsa opisana iskanja delujejo tako na besedah, kot na dokumentih. Kar smo iskali in si prikazali s programom si lahko tudi shranimo.

4.3.3 Omogočeni prikazi

Konceptualni prostor je predstavljen na dva načina, ki omogočata njegovo interaktivno raziskovanje - SOM mrežo in Voronoijev diagram. Dovoljuje prikaz vsakega posebej, prikažemo pa ju lahko tudi skupaj.

Prikaz besed in dokumentov na SOM mreži

Prikaz omogoča izpis besed na SOM mreži (slika 4.8). V tem prikazu, se prikazujejo tudi tortni diagrami besed. Mrežo lahko z miško 3D vrtimo in premikamo po prostoru. Vozlišča, ki predstavljajo središče teme, so označena s privzetimi vrednostmi in sicer z besedo "Topic"in zaporedno številko.

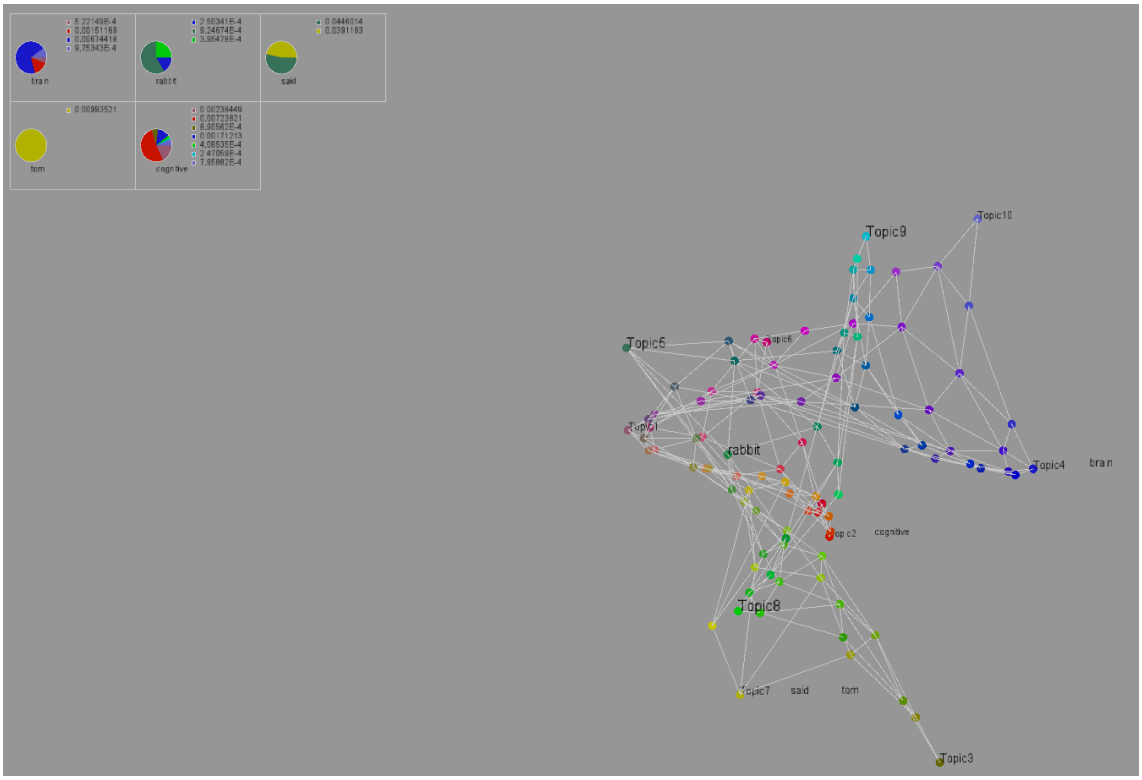
Prikaz besed in dokumentov na Voronoijevem diagramu

Voronoijev diagram (slika 4.9) je zgrajen glede na tista vozlišča SOM mreže, ki predstavljajo središče teme. Klik na posamezno regijo, nam na SOM mreži centrira najbližje vozlišče. Privzeto so preostale točke SOM mreže skrite, s tipko p na tipkovnici, pa jih lahko prikažemo.

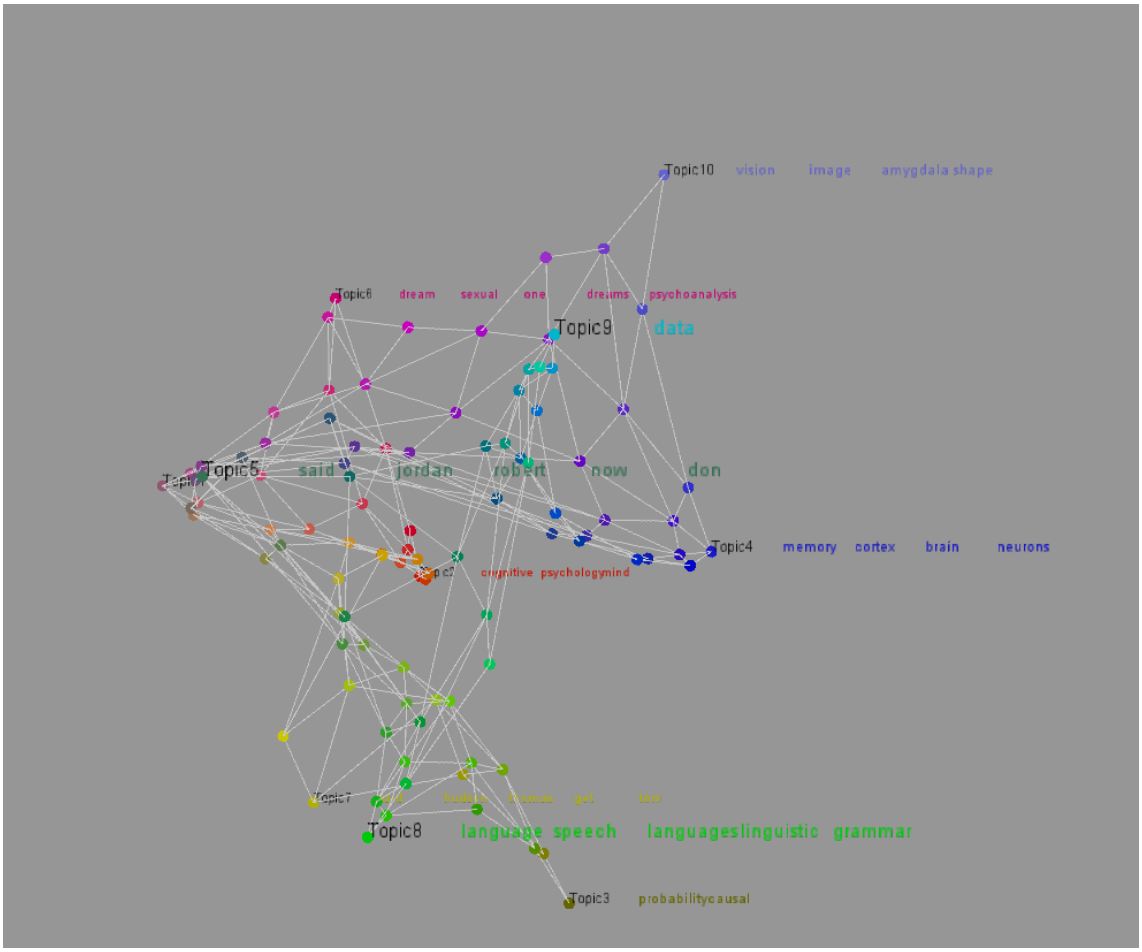
Voronoijev diagram s kontrolami iz tipkovnice lahko 2D vrtimo in ga zmanjšujemo ali povečujemo.

Prikaz besed na SOM mrežami in Voronoijevem diagramu vzporedno

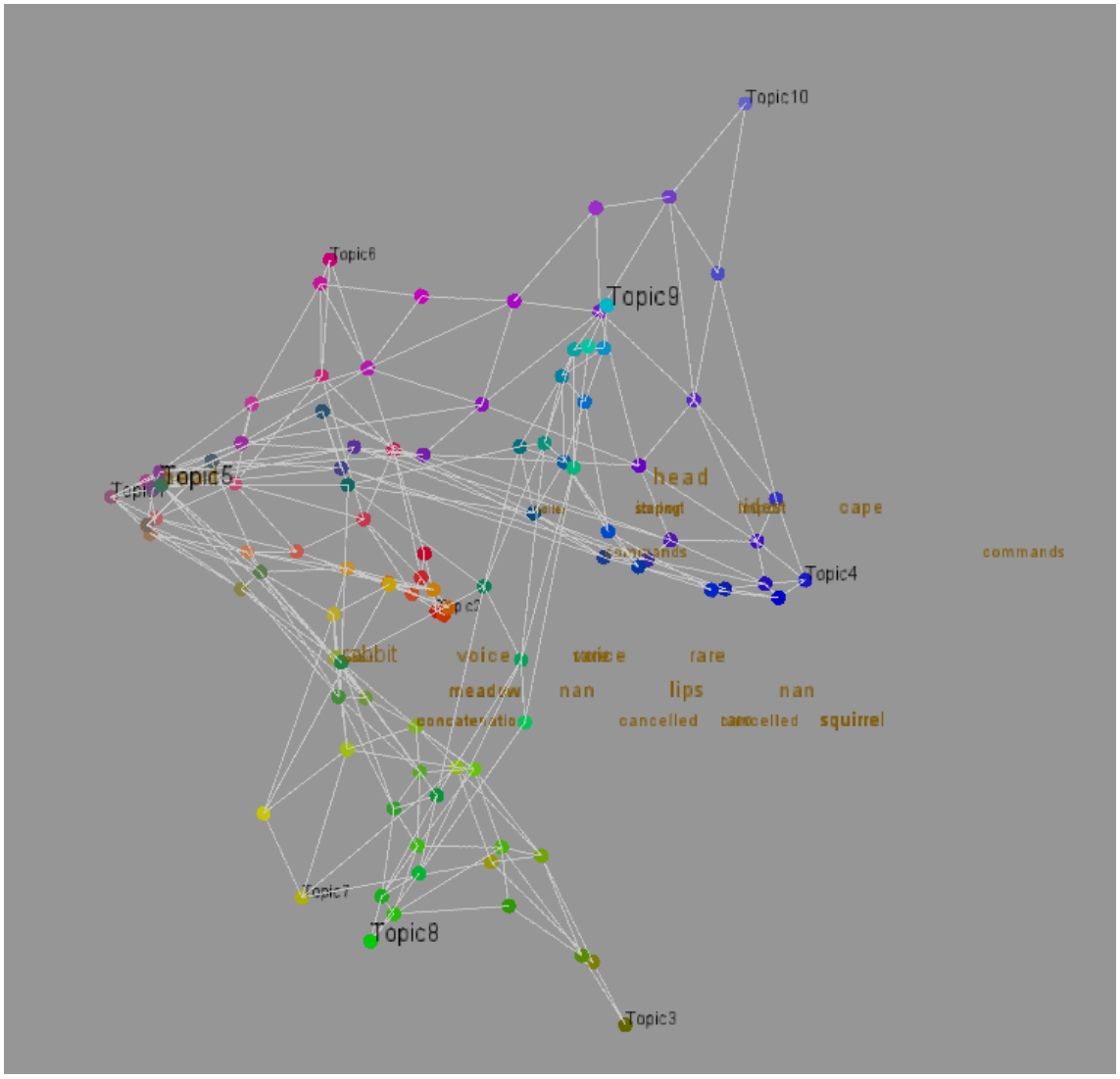
Prikaz ohranja vse lastnosti posameznih prikazov, razen tortnega diagrama, ki zaradi je zaradi pomanjkanja prostora onemogočen. Ko preklopimo v SOM prikaz, se tortni diagrami pokažejo (slika 4.10).



(a) Iskanje besed oz. dokumentov

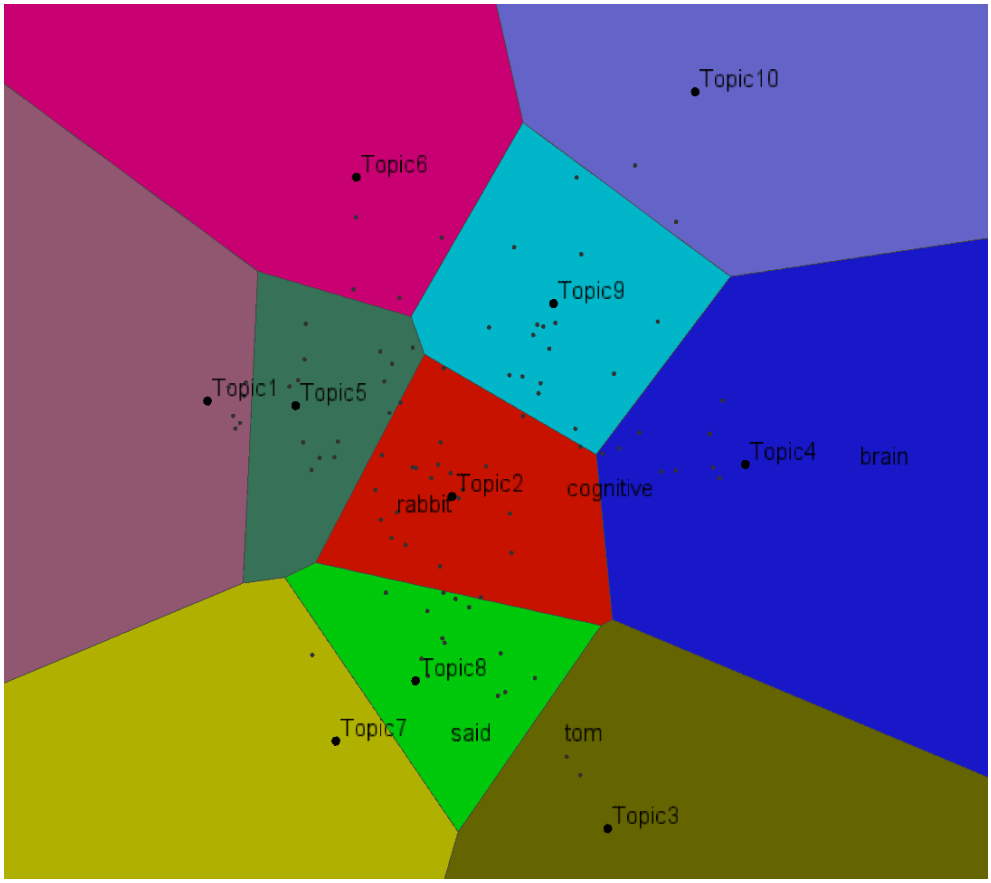


(b) Iskanje n besed na temi

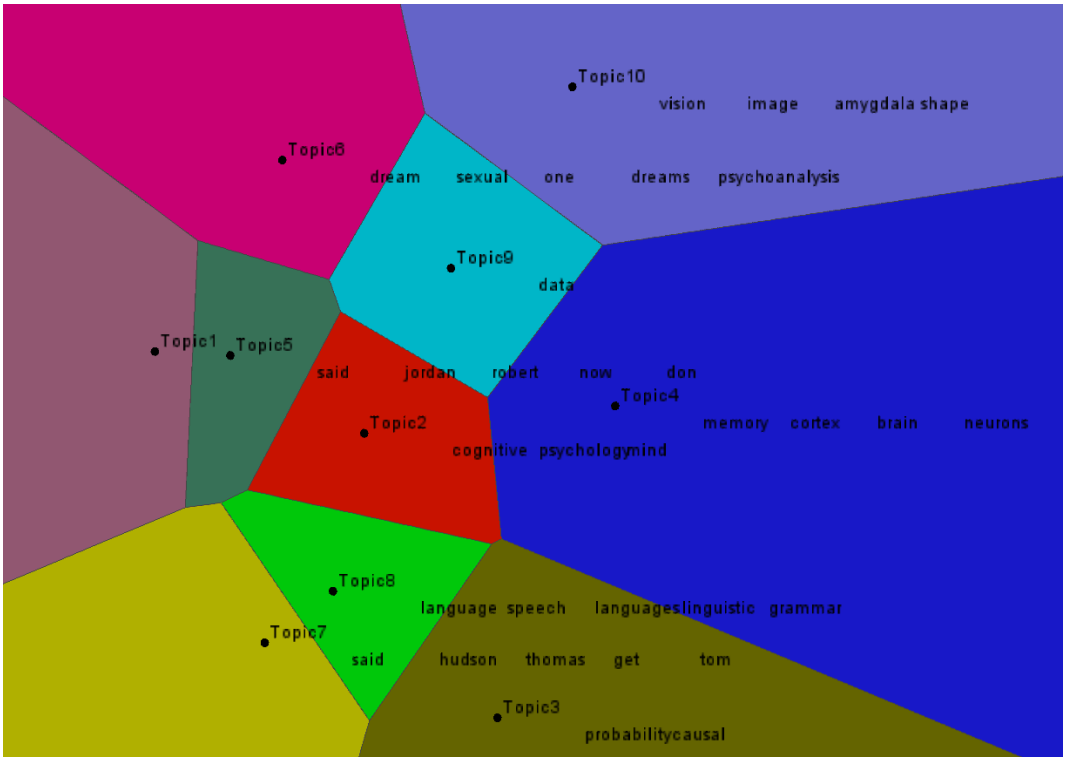


(c) Iskanje asociacij

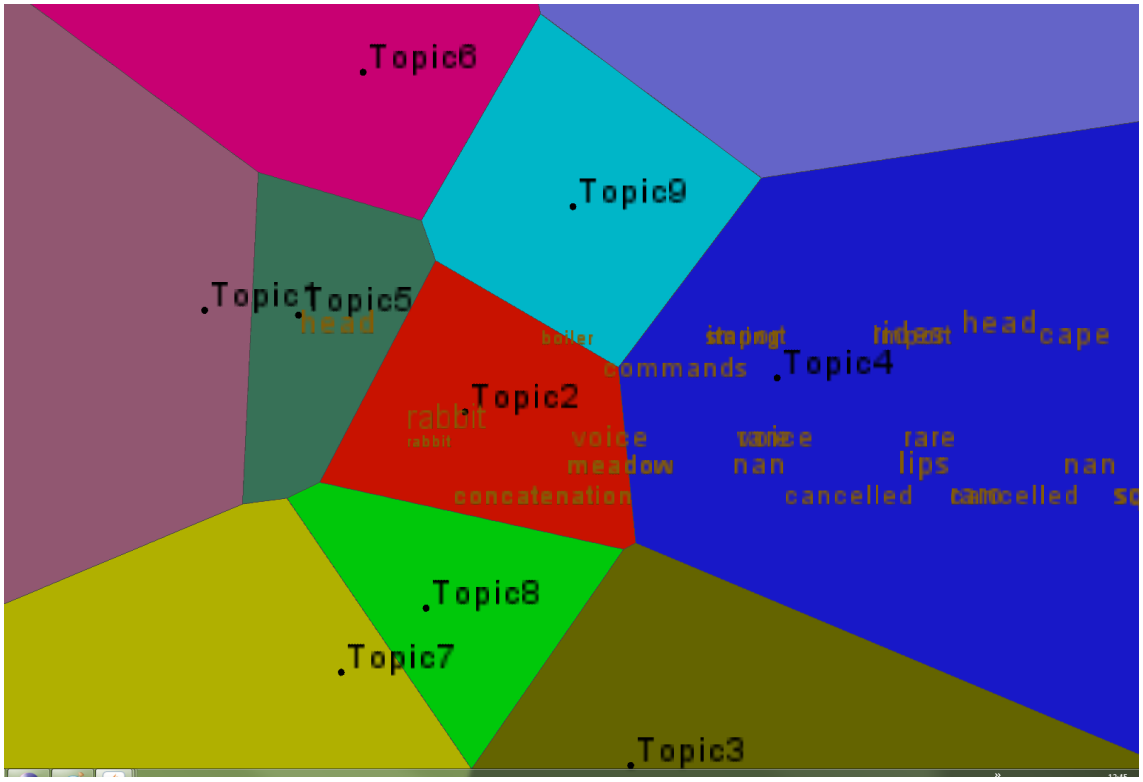
Slika 4.8: Prikazi iskanj na SOM mreži



(a) Iskanje besed oz. dokumentov

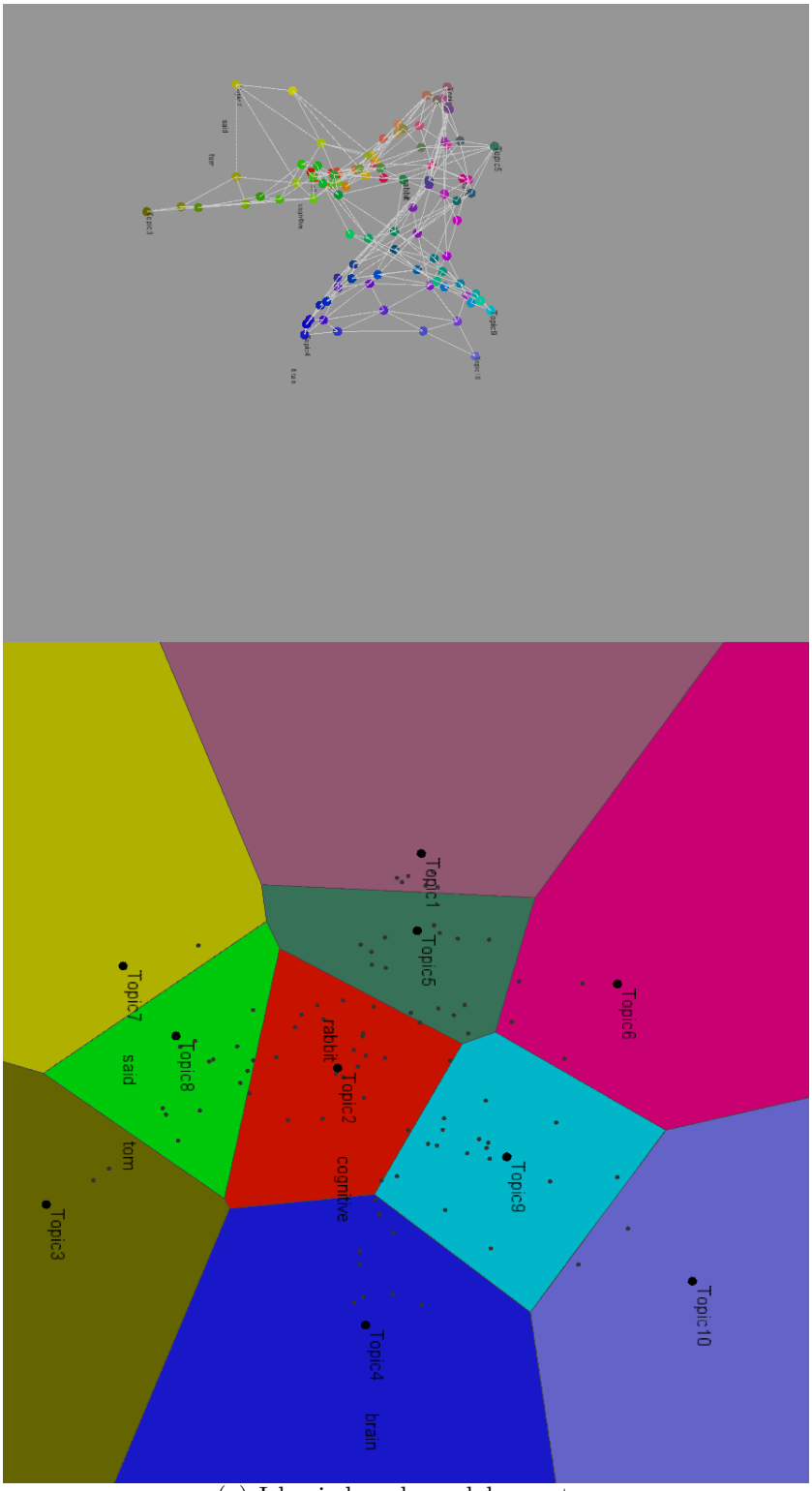


(b) Iskanje n besed na temi

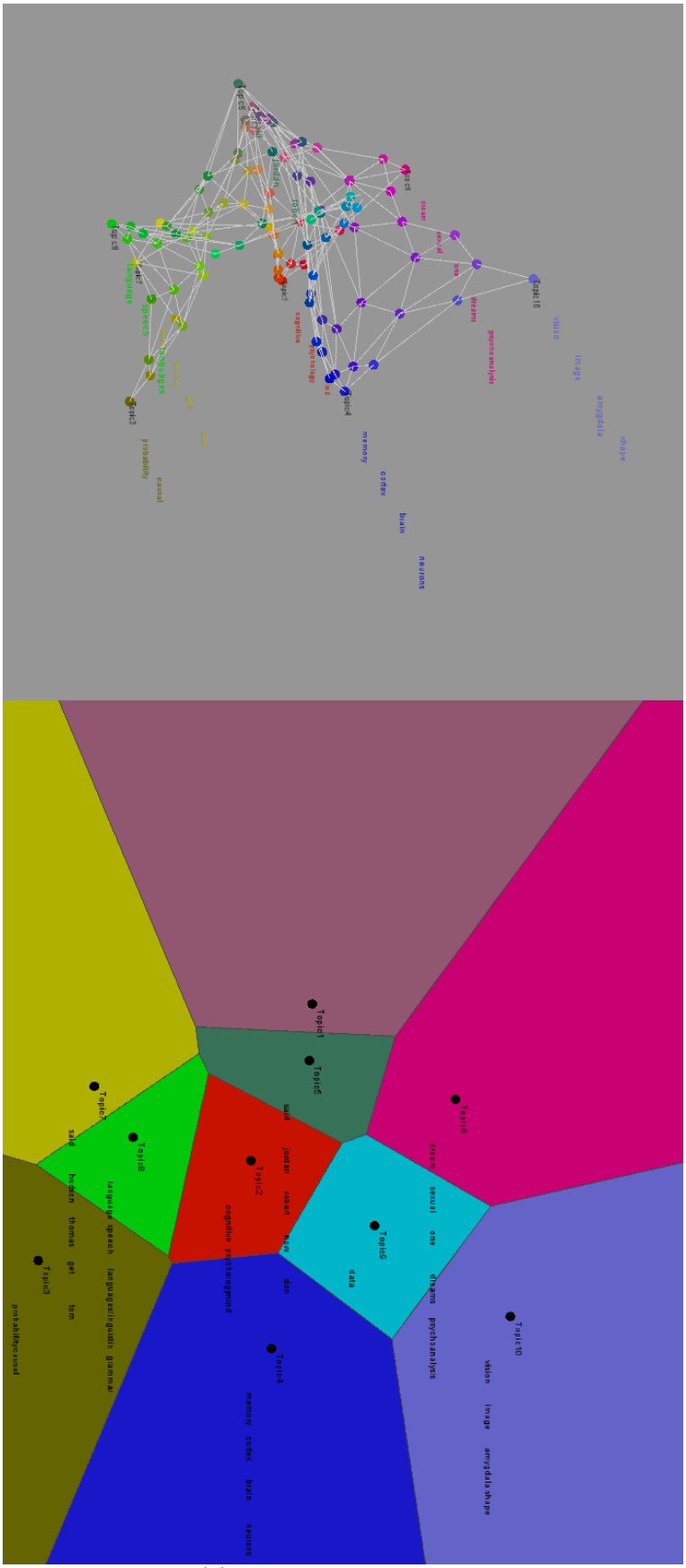


(c) Iskanje asociacij

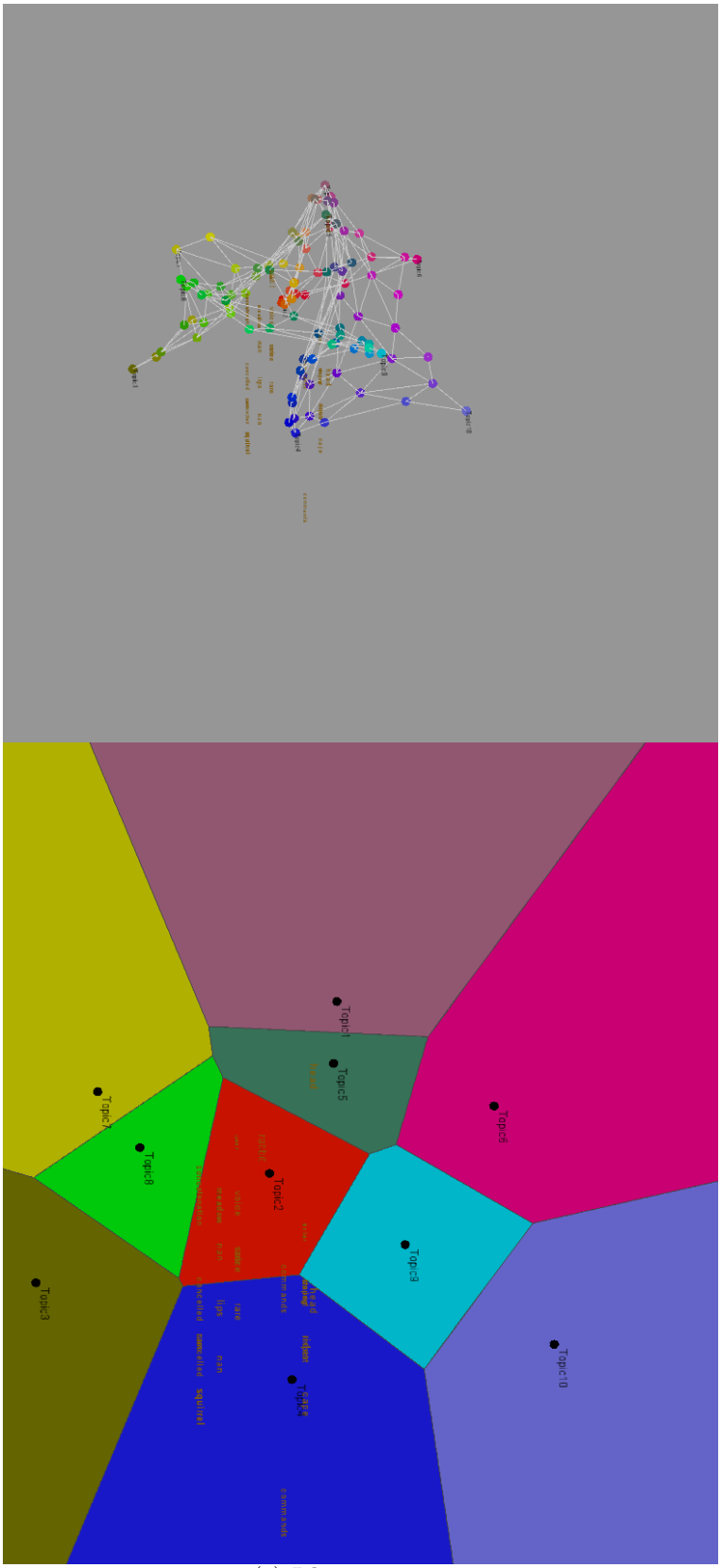
Slika 4.9: Prikazi iskanj na Voronoijevem diagramu



(a) Iskanje besed oz. dokumentov



(b) Iskanje n besed na temi



(c) Iskanje asociacij

Slika 4.10: Prikazi iskanj na Voronoijevem diagramu in SOM mreži

Poglavje 5

Nadaljnje delo

Nadaljnje izboljšave programa so mogoče pri izbiri drugačnega modela za semantično obdelavo podatkov. Takšno izboljšavo omogoča na primer uporaba dinamičnega modela tem (ang. Dynamic Topic Model), v nadaljevanju DTM [1], ki omogoča upoštevanje časovne komponente. Z njim sta David M. Blei in John D. Lafferty analizirala 250 izmed 30000 člankov iz revije *Science* med leti 1881 - 1999.

Korpus je imel skoraj 7,5 milijona besed, slovar besed pa 15966, pri čemer so slovar besed sestavljali koreni besed, izpustili pa so korene, ki so se pojavili manj kot 25-krat. Uporabila sta 20-komponentni DTM in ugotovila, da lahko z modelom ugotovita različne znanstvene teme, ter da lahko z njegovo pomočjo napovesta različne trende uporabe besed v posamezni temi.

Da bi lahko ocenili kako dobro model napoveduje, so ga primerjali še s statičnim modelom tem. Ocenjevali so napoved tem za naslednje leto. Za primerjavo so uporabili dva statična modela, enemu so podali vsa pretekla leta, drugemu pa le prejšnje leto. Izkazalo se je, da je bil DTM boljši od obeh statičnih modelov. DTM bi programu omogočil, da se pri vizualizaciji programa, tako ali drugače upošteva še časovna komponenta dokumentov in/ali tem. Seveda bi bilo potrebno temu primerno prilagoditi tudi vizualizacijo besed. Za vsako temo, bi lahko na primer prikazali njen časovni razvoj, katere besede so se kdaj uporabljale, koliko časa in kdaj so se prenehale uporabljati. To bi programu prineslo kar nekaj interaktivnosti.

Naslednji model, ki tudi ponuja izboljšavo je korelirani model tem (ang. correlated topic model), v nadaljevanju CTM, ki so ga testirali v [2]. Če v LDA algoritmu uporabimo normalno logistično distribucijo, potem dobimo CTM model, ki zna napovedovati tudi elemente povezane z dodatnimi temami, ki so v korelaciji s prvotno temo. Za testiranje modela so uporabili CTM s

100 temami na 16351 člankih prav tako iz revije Science. Članke so vzeli iz let 1990 in 1999. Iz najbolj verjetnih besed posamezne teme in povezav med samimi temami so zgradili graf iz katerega so lahko razbrali, katere teme se medseboj bolj ali manj povezujejo. Model je dostopen na [17]. Na manjši množici člankov - 1452, od leta 1960 dalje, so CTM model primerjali še z LDA modelom. Slovar je vseboval 5612 besed, pri čemer funkcionalne besede niso bile upoštevane in prav tako besede, ki so se pojavile le enkrat v množici podatkov. Ugotovili so, da CTM model bolje napoveduje besede k temam kot LDA. Slabost CTM modela je, da je računsko zahtevnejši od LDA.

Izboljšave so mogoče tudi v sami vizualizaciji. Morda bi bil, poleg SOM in Voronoijevega diagrama, primeren še diagram spektra besed, ki bi pokazal koreliranost dveh besed, dokumentov in / ali tem.

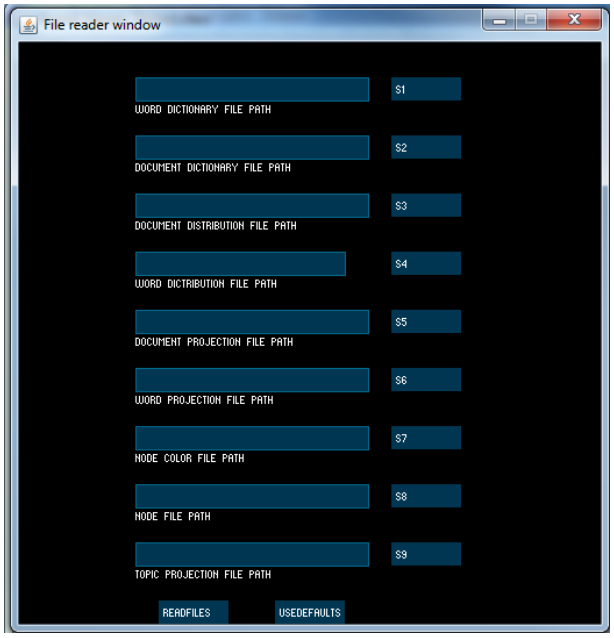
Literarna mapa organizmov bi prikazala dele besedil, ki se ponavljajo, ne samo v enem dokumentu, temveč po vseh dokumentih. Ob upoštevanju še časovne komponente, bi bila zanimiva tudi kakšna animacija spreminjanja pomembnosti besed po temah.

Vsekakor pa bi bilo ob razširjanju programa potrebno razmisliti, ali je programski jezik Processing še dovolj dober. Res da, omogoča lažjo in hitro vizualizacijo, vendar je s količino podatkov vedno počasnejši.

Dodatek A

Navodila za uporabo programa

Za pravilno delovanje programa, je najprej potrebno pravilno povezati datoteke s podatki s programom. Nobena od datotek ne sme manjkati, četudi je prazna. Okno za povezavo datotek s programom zgleda takole:



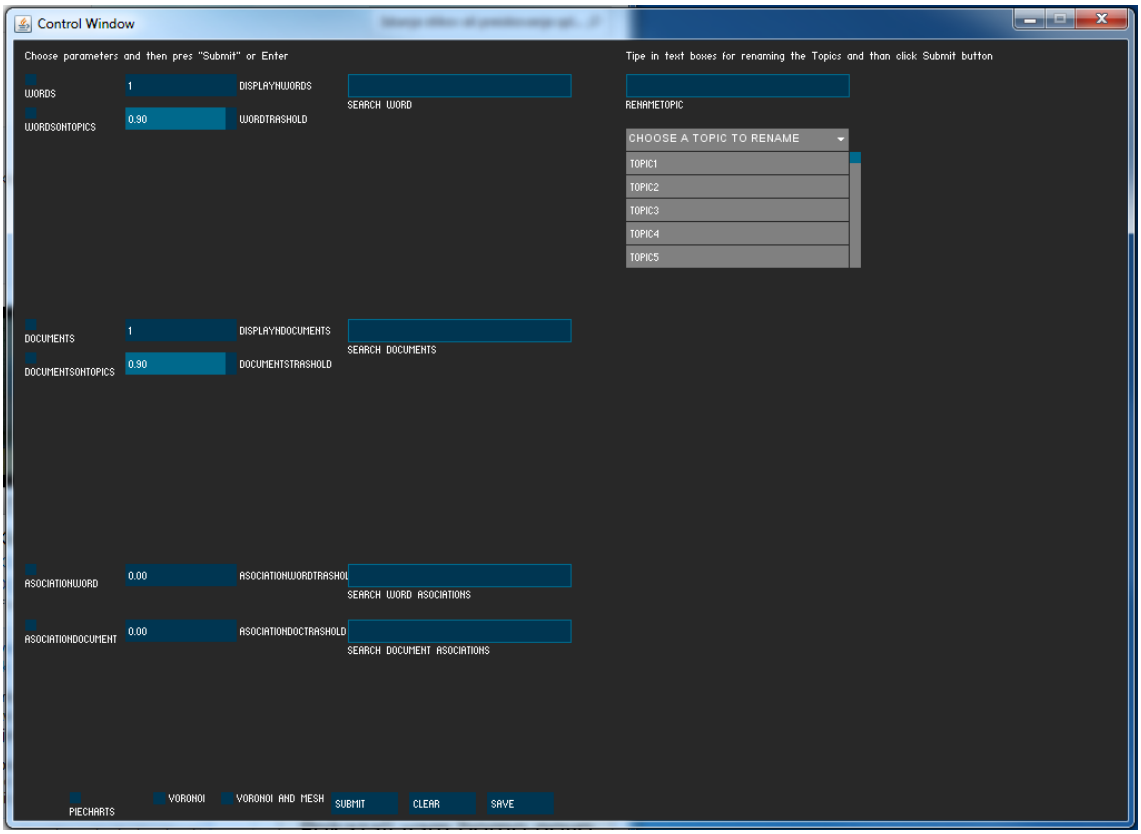
Slika A.1: Okno za določitev datotek

V naslednjem koraku se nam prikažeta dva okna, eno je kontrolno okno, iz katerega nadzorujemo dogajanje v drugem - prikaznem oknu, v katerega se nam izrisujejo iskanja.

A.1 Nadzorno okno

Nadzorno okno nam omogoča iskanje besed, dokumentov in njihovih asociacij. Izpiše nam lahko tudi prvih nekaj besed ali dokumentov na temi. Iz nadzornega okna lahko teme tudi poimenujemo. Na spodnjem delu okna najdemo kontrole, ki jih potrebujemo za nadzor izrisa. Izbiramo lahko ali se nam tortni diagram izrisuje ali ne ter način prikaza: SOM, Voronoi ali oboje. Privzeto so vse kontrole izključene, kar pomeni, da je prikazan SOM.

Če vključimo "pieChart" se ob iskanju besede ali dokumenta na levi strani prikaznega okna nariše tortni diagram z verjetnostnimi pripadnosti temam. Le-ta se zaradi pomanjkanja prostora izrisuje le na SOM prikazu. Naslednja dva gumba medsebojno izključujeta preostala dva možna prikaza. Gumb "Submit" izriše iskanje, gumb "Clear" počisti vsa dosedanja iskanja, gumb "Save" shrani prikazano (razeb izpisa besed ali dokumentov na temah), da se lahko ob naslednjem zagonu programa, progam vrne v zadnji shranjen prikaz podatkov.



Slika A.2: Okno za določitev datotek

Iskanje besed, dokumentov in njihovih asociacij

Za iskanje besed ali dokumentov mora biti vključen gumb "words" oziroma "documents." Drsnik, ki se nahaja zraven, nam določa na koliko vozliščih se bo beseda oz. dokument izpisal. Le - to je odvisno od vhodnih podatkov. Besedo oz. dokument, ki ga iščemo vpišemo v polje zraven drsnika in pritisnemo gumb "Submit" ali "Enter". Podobno mora biti pri iskanju asociacij vključen gumb "asociationWord" za besede oziroma "asociationDocuments" za dokumente. Drsnik zraven določa prag, kako podobne si morajo besede biti. Večji kot je prag, bolj podobne so si besede. Besedo kateri želimo poiskati asociacije pa vpišemo v polje zraven drsnika.

Kaj, če ne poznamo besed, ki so v slovarju?

Če ne poznamo slovarja besed, lahko v polje za iskanje besede oz. dokumenta vpišemo prvo črko in pritisnemo **- minus**. Izpiše se nam seznam besed, ki se začnejo na to črko. To deluje tudi, če vpišemo prvi dve črki. Besede se izpišejo v abecednem vrstnem redu. Nato v seznamu izberemo želeno besedo. Opazimo, da se nam je tako zapisala v polje. Sedaj pritisnemo gumb "Submit" ali pritisnemo "Enter" za izris besede. Slovar besed deluje tudi pri iskanju asociacij.

Izpis besed oziroma dokumenta na temo

Za izpis prvih nekaj besed oziroma dokumentov na temo mora biti vključen gumb "wordsontopics" oziroma "documentsontopics". Drsnik zraven določa prag, kako dobro pripadnost morajo imeti besede, da se izpišejo na temi. Večji kot je prag, večjo verjetnost pripadnosti temi mora imeti beseda. Ponovno, za prikaz moramo pritisniti gumb "Submit" ali tipko "Enter".

Preimenovanje tem

Na desni strani nadzornega okna imamo polje in seznam tem. Če želimo temo preimenovati, jo najprej kliknemo, da se nam izpiše v polju nad seznamom, ter jo nato preimenujemo. Za potrditev preimenovanja moramo ponovno pritisniti gumb "Submit" ali tipko "Enter".

A.2 Prikazno okno

SOM

Privzeti prikaz je prikaz SOM mreže. Mrežo lahko v tridimenzionalnem prostoru vrtimo in premikamo s pomočjo miške. S koleščkom mrežo približujemo in oddaljujemo. Leva tipka nam omogoča vrtenje okoli središča. Leva in desna tipka skupaj, pa premikanje mreže po prostoru. Dvo - klik leve tipke vrne mrežo v začetni položaj.

Voronoijev diagram

Ko vključimo prikaz Voronoijevega diagrama ga lahko premikamo po ravnini s pomočjo tipkovnice. S smernimi tipkami nadzorujemo približevanje oziroma oddaljevanje (tipki **gor**, **dol**) in vrtenje (tipki **levo**, **desno**) diagrama. Prikaz omogoča še naslednje možnosti:

- vklop/izklop Voronoijevega diagrama s tipko 1;
- vklop/izklop Delaunijevega diagrama s tipko 2;
- vklop/izklop prikaza konveksne ovojnice s tipko 3;
- prikaz preostalih vozlišč mreže s tipko p.

Klik v regijo Voronoijevega diagrama, nam bo pogled na mreži spremenil na najbližje vozlišče glede na klik v Voronoijevi regiji.

Slike

Literatura

[1] Blei D. M., Lafferty, J. D.,*Dynamic topic models. In Proceedings of the 23rd international conference on Machine learning*, str. 120, 2006

[2] Blei D. M., Lafferty, J. D.,*Correlated Topic Models*

[3] Blei D. M., Ng, A. Y., Jordan, M. I., "Latent dirichlet allocation," *The Journal of Machine Learning Research*, št.3, str. 993-1022,2003

[4] Eglen S.J., Lofgreen D. D., Raven M.A., Reese B.E., " Analysis od spacial relationships in three dimensions: tools for the study of nerve cell patterning," *BMC Neuroscience*, 2008 članek dostopen na: *www.biomedcentral.com/1471-2202/9/68*

[5] Gärdenfors P., *Conceptual Spaces: The Geometry of Thought*, The MIT Press, 2004

[6] Gärdenfors P., *How to make semantic web more semantic: The Geometry of Thought,Formal ontology : information systems*, str.19-36,2004

[7] Gärdenfors P., Zenker F., " Using Conceptual Spaces to Model the Dynamisc of Emirical Theories, " *Science in Flux: Philosophy pf Science Meets Belief Revision Theory*, str.137 - 158 , izide 2010, članek dostopen na: *http://www.springer.com/philosophy/epistemology+and+philosophy+of+science /book/978−90−481−9608−1*

[8] Griffiths, T. L., Steyvers, M., Tenenbaum, J. B. "Topics in semantic representation," *Psychological Review*,114(2),str. 211-244,2007

[9] Hoffman T., "Probabilistic Latent Semantic Analysis," *Uncertainty in Artificial Intelligence*, 1999

[10] Landauer, T. K., Dumais, S. T.," A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological review*, 104(2), str. 211-240, 1997

[11] Landauer T.K.,Laham D., Derr M., "From paragraph to graph : Latent semantic analysis for information visualization," Dostopno na: *www.pnas.orh/cgi/doi/10.1073/pnas.0400341101*

[12] Petersen M.K., Butkus A., " Modeling Moods in BBC Programs Based on Emotional Context," *EuroITV, LNCS 5066*, str. 112-116, 2008

[13] Shu L., Long B., Meng W., " A Latent Topic Model for Complete Entity Resolution"

[14] Steyvers, M., Griffiths, T., "Probabilistic topic models," *Handbook of Latent Semantic Analysis*,str. 424-440, 2007.

[15] G . Strle, *Semantics within: The Representation of Meaning Through Conceptual Spaces*, *Doktorska disertacija v pripravi*

[16] *http://dbrowser.jstor.org*

[17] *http://www.cs.cmu.edu/ lemur/science/1.html*

[18] *http://visualizeit.wordpress.com/*

[19] (2009) Hearst M.A., Information vizualisation fot text analysis, dostopno na: *http://searchuserinterfaces.com/book/sui_ch11_text_analysis_visualization.html*

[20] nevronske mreže, dostopno na: *http://en.wikipedia.org/wiki/Artificial_neural_network*

[21] *http://www.diku.dk/hjemmesider/studerende/duff/Fortune/*