GRADNJA HIERARHIČNE STRUKTURE KONCEPTOV IZ NESTRUKTURIRANIH TEKSTOVNIH DOKUMENTOV

Robert T. Leskovar, József Györkös, Ivan Rozman
Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko, Smetanova ulica 17, Maribor
RLeskovar@uni-mb.si

Povzetek

Metode za iskanje po obsežnih bazah dokumentov in za klasificiranje dokumentov pogosto uporabljajo za učinkovitejše delovanje vnaprej pripravljene opise področnih kategorij. V prispevku je predstavljena izvirna metoda za samodejno gradnjo hierarhične strukture vsebinskih konceptov, ki jih določi z analizo množice dokumentov. Struktura omogoča učinkovito brskanje in iskanje po vsebinskih konceptih ter zelo poenostavi oblikovanje opisa področnih kategorij za poljubno množico dokumentov. Metoda je zasnovana neodvisno od jezika, v katerem je zapisana vsebina dokumentov. Predstavljeni sta tudi razširitvi za semantično povezovanje konceptov in za pospešitev delovanja metode.

Abstract

CONSTRUCTION OF A HIERARCHICAL CONCEPT STRUCTURE FROM UNSTRUCTURED TEXT DOCUMENTS

Document retrieval methods, implemented for large document collections and document classification methods often use predefined descriptions of subject categories to improve their efficiency. In the article the original method for automated construction of a hierarchical concept structure is presented. The structure is built through an analysis of the documents in a collection. Such a structure makes efficient concept browsing and searching feasible and simplifies defining of subject category descriptions for arbitrary document collections. The method is independent of language used in documents. Two improvements are introduced as well, aiming at connecting the concepts semantically and improving a processing capacity of the method.

■ ■ ■

1. Uvod

1.1 Pridobivanje informacij

Količina informacij, zapisanih v nestrukturirani ali delno strukturirani tekstovni elektronski obliki (nadalje: informacij), s katerimi se kot informacijska družba srečujemo v okoljih delovnih organizacij in v spletu, je vse manj obvladljiva in zahteva nove metode iskanja in predstavitve želenih informacij. Znanje, zapisano s temi informacijami, ima visoko vrednost in njegovo učinkovito upravljanje je ključnega pomena za kakovost delovnih procesov ter posledično konkurenčnost organizacij, tako gospodarskih kot negospodarskih.

Metode pridobivanja informacij (ang. 'information retrieval') so se v zadnjih dvajsetih letih razvile od preprostega iskanja dokumentov do zahtevnejšega iskanja, klasificiranja, filtriranja in grupiranja dokumentov. Področje pridobivanja informacij ne zajema le sintaktičnega vidika dokumentov, ki je predmet preprostega iskanja, ampak tudi na semantični in pragmatični vidik dokumentov. *Preprosto iskanje* (v literaturi je običajno omenjeno kot 'prosto iskanje po tekstu') na podlagi uporabnikovega povpraševanja vrne seznam dokumentov, ki vsebujejo natančno takšne izraze, kot so bili zapisani v povpraševanju. Pomanjkljivosti, ki jih vsebuje ta postopek, so neprepoznavanje večpomenskih besed, sinonimov, izvorov besed, sklanjatev in spregatev. Učinkovitost iskanja je temu primerno nizka.

Osnovne veje področja pridobivanja informacij so danes:

- *Iskanje dokumentov* (najpogosteje ga imenujemo kar 'pridobivanje informacij') temelji predvsem na enem od štirih modelov ter njihovih variantah – verjetnostnem, Boolovem modelu na podlagi mehke logike in vektorskem. Kot rezultat procesa dobimo iz baze dokumentov seznam dokumentov, ki so urejeni po stopnji ustreznosti glede na povpraševanje. Več o posameznih modelih je na voljo v [Yates, 99].
- *Klasificiranje dokumentov* (v literaturi tudi kategoriziranje, ang. 'classification' oz. 'categorization') izvaja razporejanje dokumentov, ki se nahajajo v skupni bazi, v vnaprej določeno množico področnih kategorij. Razporejanje temelji na vsebini dokumentov in ne na metainformacijah, ki jih lahko dokumenti vsebujejo. Zato je razporejanje nedeterminističen proces.

- *Filtriranje dokumentov* je oblika klasificiranja, ki razporeja dinamični tok dokumentov v vnaprej določene kategorije, najpogosteje glede na profil uporabnika, v katerem so v obliki pravil zapisani njegovi izbori. Primeri so sistemi za filtriranje novic, reklamnih sporočil ali Usenet novic.
- *Grupiranje dokumentov* (v literaturi tudi grozdenje ali združevanje, ang. 'clustering') združuje dokumente v grupe na podlagi določenih skupnih značilnosti.

Rezultati naše raziskave predstavljajo doprinos iskanju in klasificiranju dokumentov. V nadaljevanju bomo dodatno utemeljili potrebo po tovrstnih raziskavah.

1.2 Ovire za učinkovito pridobivanje želenih informacij in motivacija za raziskavo

Pri interakciji med človekom in informacijskim sistemom prihaja do več ovir, ki otežujejo učinkovito pridobivanje želenih informacij. Ena od ovir je nezmožnost uporabnikov, da bi oblikovali *ustrezno* povpraševanje. Pri iskanju v spletu je posledica tega ogromna množica vrnjenih dokumentov, med katerimi uporabniki ne najdejo želenih. Študija rabe spletnega iskalnika Excite je pokazala, da so povpraševanja v povprečju krajša od treh besed [Jansen, 98]. Osnovna razloga za to sta, da obstajajo med različnimi avtorji dokumentov razlike v rabi jezika, ki so pogojene med drugim z njihovo izobrazbo, kulturo, razgledanostjo in namenom pisanja, in da uporabniki, ki niso strokovnjaki računalniške stroke, ne poznajo principov, po katerih delujejo metode za iskanje informacij. K reševanju tega problema so se usmerile tehnike za samodejno razširjanje povpraševanj z ustreznejšimi izrazi in za usmerjanje povpraševanj na podlagi vnaprej zgrajenih seznamov izrazov, ki jih uporablja določeno vsebinsko področje (imenujemo jih vertikalni tezavri), vendar povzroči večje število izrazov pogosto tudi večji obseg pridobljenih dokumentov, tako ustreznih kot neustreznih.

Druga ovira za uspešno iskanje informacij se pojavi, če naša informacijska potreba ni jasno definirana. V takšnih primerih se raje lotimo *brskanja* po katalogih informacij (če so na voljo), kjer so dokumenti razvrščeni v področne kategorije. To nam omogoča, da pridemo prek kategorij do želenih informacij, ali da odkrijemo informacije, ki bi nas utegnile zanimati.

Za klasificiranje dokumentov v kategorije skrbi klasifikator – sistem, ki skuša s pomočjo različnih metod ugotoviti, v katero kategorijo (ali več kategorij) je potrebno dokument uvrstiti glede na njegovo vsebino. Najpogosteje so v rabi metode strojnega učenja, s katerimi klasifikator z 'opazovanjem' značilnosti množice dokumentov, ki jih je poprej klasificiral v vnaprej določeno množico kategorij strokovnjak, določajo uvrstitev v te kategorije tudi za ostale dokumente. Na mnogih področjih so se zato oblikovale taksonomije kategorij. V spletu so to splošnonamenski katalogi, kot sta Yahoo!-jev in Infoseek-ov, ki so osnova večini iskalnikov, ameriška Kongresna knjižnica ima sistem oznak Library of Congress Subject Headings (LCSH), v mnogih knjižnicah se uporablja Deweyeva decimalna klasifikacija (DDC), v medicinski domeni uporabljajo množico medicinskih oznak Medical Subject Headings (MeSH) itn..

Najnatančneje lahko klasificiramo dokumente, če so v njih navedeni metapodatki - klasifikacijske oznake ali ključne besede. Kadar niso, jih lahko doda urednik ali katalogizator, ki pozna oziroma preuči vsebino, vendar je to drag in dolgotrajen proces. Na človeško izbiro oznak in ključnih besed vpliva tudi več dejavnikov - predvsem izkušnje, razgledanost, izobrazba in konsistentnost pri uporabi oznak. Raziskave konsistentnosti pri izbiri oznak za opis vsebine določenega dokumenta so pokazale, da se oznake, ki jih izbereta dva različna profesionalna katalogizatorja, razlikujejo v povprečju za 20 odstotkov [Harman, 95].

Našo raziskavo smo usmerili v razvoj metode, ki omogoča, da lahko dobi uporabnik – iskalec informacij – hiter, strukturiran pregled vsebine množice dokumentov, ko ni na voljo vnaprej pripravljena taksonomija kategorij, v katere bi jih lahko klasificirali, in ko dokumenti niso opremljeni z metapodatki. V ta namen pretvori metoda vsebino dokumentov v hierarhično strukturo, v kateri se nahajajo med seboj povezani vsebinski koncepti, kar omogoča učinkovito brskanje in iskanje želenih informacij in predstavlja osnovo za gradnjo taksonomije kategorij. Metodo smo zasnovali neodvisno od jezika, v katerem so dokumenti. Z razširitvami, prilagojenimi določenemu jeziku, lahko še dodatno povečamo njeno učinkovitost.

1.3 Definicije

Najprej bomo natančneje definirali osnovne termine, ki jih uporabljamo v prispevku. Termina 'dokument' in 'izraz' sta privzeta termina iz področja pridobivanja informacij.

Dokument pomeni določeno tekstovno enoto, npr. naslov, povzetek, odstavek, prispevek, poglavje. Množica dokumentov je množica tekstovnih enot.

Izraz je ključna beseda (lahko besedna zveza), ki sodeluje pri opisu vsebine dokumentov. Za ključne besede izberemo predvsem tiste, ki nosijo večji pomen (predvsem samostalniške besedne oblike).

Koncept imenujemo množico izrazov, ki soodvisno nastopajo pri opisu določenih vsebinskih delov. Samodejno prepoznavanje konceptov temelji na dejstvu, da pri opisovanju podobnih stvari uporabljamo podobne besede. Koncepte lahko natančneje določimo ob primerjavi več vsebinsko sorodnih dokumentov.

2. Sorodna dela

Forsyth in Rada sta v svojem delu določala splošnost in specifičnost izrazov na podlagi števila dokumentov, v katerih se pojavljajo [Forsyth, 86]. Uporabila sta predpostavko, da je izraz bolj splošen, če se pojavi v več dokumentih. Na podlagi tega sta zgradila manjši večnivojski graf, ki je predstavljal relacije med izrazi.

Woods je v svojem delu uporabil analizo angleških fraz v povezavi z veliko bazo znanja, na podlagi česar je organiziral izraze v hierarhije konceptov [Woods, 97]. Uspeh tehnike je temeljil na bogati bazi morfološkega znanja, s katerim je identificiral komponente fraz. Hierarhijo konceptov je uporabil za samodejno razširjanje povpraševanj pri iskanju dokumentov.

Sanderson in Croft sta se lotila gradnje hierarhije konceptov iz dokumentov, ki so bili pridobljeni na podlagi povpraševanja, s primerjanjem medsebojne vsebovanosti izrazov [Sanderson, 99]. Iz množice dokumentov sta najprej s pomočjo določenega povpraševanja pridobila tiste dokumente, ki so vsebinsko vezani na povpraševanje. Množico dokumentov so zato že sestavljali dokumenti, ki uporabljajo izraze na podoben način. Množico izrazov, ki bodo vsebovani v hierarhiji, sta pridobila iz izrazov razširjenega povpraševanja in izrazov, ki se pojavljajo v več kot 10% v pridobljenih dokumentih glede na celotno bazo. Izraze sta nato primerjala glede na relacijo 'vsebovanje' (ang. 'subsumption'): izraz A vsebuje izraz B, če nastopa v več kot 80% dokumentov, v katerih nastopa izraz B. Glede na medsebojno vsebovanje sta izraze uredila v hierarhijo, pri čemer sta izločila izraze z nizko frekvenco pojavljanja. Izrazi v njuni hierarhiji lahko imajo več staršev.

Yang in Lee pristopata h gradnji hierarhije z metodo nenadzorovanega učenja nevronskih mrež, ki se imenuje *samoorganizirajoče mape* [Yang, 00]. Avtorja sta z učnimi primerki dokumentov vzpostavila dve mapi, mapo grup dokumentov in mapo grup izrazov. Dominantne nevrone v mapi grup dokumentov (ki predstavljajo določeno grupo dokumentov) sta vzela kot centroide nadgrup, ki predstavljajo bolj splošno kategorijo. Izraze, ki so povezani z istoležečim nevronom v mapi grup izrazov, sta nato uporabila kot izraze, ki opisujejo kategorijo. Hierarhijo kategorij sta zatem gradila z iskanjem korelacij med nevroni v obeh mapah. Rezultati so žal predstavljeni s kitajskimi pismenkami (avtorji so testirali pristop na množici dokumentov v kitajščini), zato jih ne moremo ovrednotiti. Avtorja omenjata nujnost zmanjšanja dimenzij vhodnih podatkov, vendar tega postopka nista opisala. Iz prispevka tudi niso razvidni načini za določanje začetne velikosti samoorganizirajoče mreže, topologije mreže (kar lahko naredimo le empirično) in faktorja učenja, ki so pomembni parametri za uspešno delovanje samoorganizacije.

Pričujoče delo se od omenjenih razlikuje v bolj univerzalnem načinu za določanje splošnosti izrazov, ki upošteva za vsak izraz število in moč njegovih korelacij, in v načinu gradnje hierarhične strukture konceptov, ki omogoča natančnejše določanje konceptov. Za gradnjo strukture ne potrebujemo pripravljenih učnih primerkov. Struktura konceptov, ki jo zgradi v nadaljevanju predstavljena metoda, je sestavljena iz *več hierarhij*, znotraj katerih so koncepti jasno izraženi glede na dokumente, v katerih se pojavljajo, med hierarhijami pa so šibko povezani, kar omogoča hiter pregled različnih delov vsebine celotne množice dokumentov. Vsebina množice dokumentov je lahko raznovrstna.

3. Metoda za gradnjo hierahične strukture konceptov

Med opisom metode bomo predstavljali njeno delovanje na majhnem, preglednem primeru – tekstu desetih naslovov prispevkov, od katerih jih šest opisuje pridobivanje informacij in štirje aplikacije teorije grafov.

Delovanje metode smo testirali na bazah dokumentov velikosti od 10 do 300 dokumentov, ki smo jih tvorili iz standardnih baz dokumentov TIME (novice iz vsega sveta, objavljene v reviji TIME leta 1963), CRAN (povzetki člankov s področja aerodinamike) in na bazi 110 dokumentov, ki govorijo o terorističnih napadih in njihovem ozadju.

Za predstavitev konceptov v zgoščeni obliki smo izbrali *hierarhično drevesno strukturo* – neusmerjen, necikličen graf, v katerem lahko ima vsak otrok le enega starša. Struktura konceptov je sestavljena iz več dreves (hierarhij), ki vsebujejo koncepte. Koncepti znotraj hierarhij so močno, med hierarhijami pa šibko povezani. Koreni hierarhij so izrazi, ki so v konceptih hierarhij najmočneje zastopani.

Kot model za predstavitev vsebovanosti izrazov v dokumentih smo izbrali *model vektorskega prostora* [Salton, 71], ki je v praksi najbolj uporabljan model, saj omogoča učinkovito obdelavo s statističnimi algoritmi in algoritmi strojnega učenja, ki izvajajo združevanje, klasificiranje, analizo glavnih komponent in druge vrste obdelav.

V tem modelu so izrazi predstavljeni z vrsticami in dokumenti s stolpci matrike. Elementi matrike so uteži, ki jih priredimo posameznemu izrazu za določen dokument.

Metoda za samodejno gradnjo hierarhične strukture konceptov zajema štiri osnovne korake, ki jih bomo podrobneje predstavili v nadaljevanju:

1. Gradnja matrike izrazov ter dokumentov.
2. Računanje matrike korelativnosti izrazov.
3. Računanje globalnih korelacijskih stopenj in iskanje korenov hierarhij.
4. Gradnja hierarhične strukture.

1. Gradnja matrike izrazov ter dokumentov

Ta korak je običajen začetni korak v pridobivanju informacij in zajema večino naslednjih aktivnosti:

(a) izbor izrazov iz baze dokumentov,
(b) krnjenje izrazov,
(c) uteževanje in normaliziranje izbranih izrazov ter
(d) združevanje izrazov, predstavljenih z enakimi vektorji.

V prvi aktivnosti iz množice vseh izrazov, ki nastopajo v bazi dokumentov, *izločimo* blokirane izraze. V splošnem lahko izločimo pomensko šibke besede, ki se zelo pogosto pojavljajo v dokumentih in jih določimo s seznamom blokiranih besed (npr. vezniki in števniki) in pomensko šibke besede, ki jih določimo s pomočjo slovarja jezika (obdržimo npr. samo samostalniške besedne oblike). V sklopu te aktivnosti izločimo samodejno tudi izraze z zelo nizko frekvenco pojavljanja v bazi dokumentov, saj njihov delež pri preglednem opisu vsebine dokumentov ni pomemben (imajo nizko funkcionalno vrednost), obenem pa s tem precej zmanjšamo dimenzije matrike izrazov in dokumentov. Frekvenčni prag je odvisen tudi od tega, koliko dokumentov je v bazi in koliko specifičnih izrazov želimo vstaviti v hierarhično strukturo. Obdržanim izrazom nato priredimo frekvence, s katerimi nastopajo v določenem dokumentu.

Po predhodni aktivnosti lahko izraze tudi *krnimo*, torej izluščimo korene izrazov z odstranjevanjem predpon in predvsem pripon. Študija uporabe krnjenja v angleščini (kjer se večinoma uporablja Porterjev algoritem [Porter, 80; Porter, 01]) ni pokazala, da bi leto pri pridobivanju informacij prispevalo k natančnosti [Frakes, 92]. Nedvomno ima krnjenje večji pomen za morfološko kompleksne jezike, kot je slovenščina. Temu primerno so kompleksni tudi algoritmi, ki se lotevajo krnjenja v teh jezikih [Popovič, 91; Dimec, 99]. V naši raziskavi krnjenja nismo uporabili, menimo pa, da velja pri aplikaciji metode za določen jezik razmisliti o njegovi uporabi.

Uteževanje izrazov priredi elementu a_{ij} v matriki izrazov in dokumentov A določeno vrednost. Vrednost je odvisna od pomena izraza i za dokument j (temu faktorju pravimo *lokalna utež*) in pomena izraza i za celotno bazo dokumentov (faktor imenujemo *globalna utež*). Obstaja več načinov za računanje lokalnih in globalnih uteži, ki so bili razviti empirično

Dokumenti (naslovi prispevkov)

D1: CAFE: A Conceptual Model for Managing Information in Electronic Mail
D2: Agent-based approach for information gathering
D3: Dunhuang Frescoes retrieval based on similarity calculation
D4: Information Retrieval on the Web
D5: The Retrieval of Document Images: A Brief Survey
D6: Survey: Introduction to Content-Based Image Retrieval
D7: Graph Visualization and Navigation in Information Visualization: A Survey
D8: Floorplan generation and area optimization using AND-OR graph search
D9: Visual programming with graph rewriting systems
D10: A graph grammar approach to graphical parsing

		D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1.	cafe,conceptual,model,managing,electronic,mail	1	0	0	0	0	0	0	0	0	0
2.	information	0,5	0,5	0	0,5	0	0	0,5	0	0	0
3.	agent-based,gathering	0	1	0	0	0	0	0	0	0	0
4.	approach	0	0,7	0	0	0	0	0	0	0	0,7
5.	dunhuang,frescoes,based,similarity,calculation	0	0	1	0	0	0	0	0	0	0
6.	retrieval	0	0	0,5	0,5	0,5	0,5	0	0	0	0
7.	web	0	0	0	1	0	0	0	0	0	0
8.	document,images	0	0	0	0	1	0	0	0	0	0
9.	survey	0	0	0	0	0,6	0,6	0,6	0	0	0
10.	introduction,content-based,image	0	0	0	0	0	1	0	0	0	0
11.	graph	0	0	0	0	0	0	0,5	0,5	0,5	0,5
12.	visualization,navigation	0	0	0	0	0	0	1	0	0	0
13.	floorplan,generation,area,optimization,and-or,search	0	0	0	0	0	0	0	1	0	0
14.	visual,programming,rewriting,systems	0	0	0	0	0	0	0	0	1	0
15.	grammar,graphical,parsing	0	0	0	0	0	0	0	0	0	1

Tabela 1: **Demonstracijski primer - matrika izrazov in dokumentov po prvem koraku**

in na podlagi teorije informacij [Kowalski, 97]. V praksi izkazujeta večjo učinkovitost uporaba logaritemske uteži za lokalno utež in entropijske uteži za globalno utež [Dumais, 91]. V našem primeru globalne uteži nismo uporabili, saj se vektorji izrazov v drugem koraku pri računanju matrike korelacij normalizirajo in se globalna utež, ki sorazmerno uteži komponente vektorjev izrazov, s tem izniči. Globalna utež, ki vključuje inverzno frekvenco izrazov za določen dokument [Yates, 99] in zato ne uteži vseh komponent vektorja izraza sorazmerno, pa je dajala na testih slabše rezultate, kot če globalne uteži nismo uporabili.

Kot lokalno utež smo uporabili logaritemsko, torej za vse elemente v matriki A je $a_{ij}=log(tf_{ij} + 1)$, pri čemer je tf_{ij} frekvenca pojavljanja izraza i v dokumentu j. S tem smo zmanjšali učinek večjih razlik pojavljanja izrazov v frekvencah, ker je za tvorjenje konceptov bolj pomembno, da neki izraz sodeluje v konceptu, kot to, kolikokrat sodeluje.

Vektorje izrazov na koncu *normaliziramo* in *združimo* izraze, ki so predstavljeni z enakimi vektorji (imenujemo jih bratje, saj nastopajo v istih dokumentih z istimi frekvencami in za tovrstno samodejno obdelavo med njimi ni razlik).

Rezultat prvega koraka na demonstracijskem primeru prikazuje tabela 1. Iz naslovov nad tabelo smo s pregledovalnikom izločili blokirane izraze s pomočjo seznama angleških blokiranih izrazov SMART [SMART], ki je nastal v okviru dolgoletnega projekta na univerzi Cornwell. Dimenzije začetne matrike A so bile 39 (izrazov) x 10 (dokumentov). Elemente matrike smo utežili z lokalno logaritemsko utežjo, vektorje izrazov normalizirali in združili brate. Tabela 1 prikazuje končno matriko A, ki je osnova za nadaljnje korake metode.

2. Računanje matrike korelativnosti izrazov

V drugem koraku metode izračunamo korelacije (podobnost, soodvisnost) poljubnih dveh izrazov. Vse korelacije predstavimo v matriki korelacij C. Korelacije dobimo z računanjem kosinusa kota w_{ij} med vektorji vseh izrazov i in j, $i<>j$. Kosinus kota nam pove, kako blizu so vektorji izrazov oziroma koliko sodelujejo pri opisu vsebine dokumentov. Ker so vektorji normalizirani, je računanje kosinusa enako množenju vektorjev:

$$c_{ij} = \cos\varpi_{ij} = (\boldsymbol{e}_i^T \boldsymbol{A})(\boldsymbol{A}^T \boldsymbol{e}_j), \qquad (1)$$

za $i,j=1,\ldots,m$, $\boldsymbol{e}_j$ je enotski vektor dimenzije n; m predstavlja št. izrazov, n predstavlja št. dokumentov.

Če je vrednost produkta blizu 1, pomeni, da sta vektorja izrazov skoraj vzporedna oziroma da sta izraza na podoben način zastopana v vsebini množice dokumentov.

Matrika C je simetrična, na diagonalo postavimo ničle. Algoritem za računanje matrike C je v splošnem reda $O(m^2n/2)$, vendar ga lahko v našem primeru pospešimo, ker so vektorji izrazov v matrikah izrazov in dokumentov A redki. Pri običajnih dimenzijah matrike A, to je nad nekaj tisoč izrazov in nad nekaj sto ali tisoč dokumentov, je v povprečju 99 odstotkov elementov enakih 0 [Berry, 92]. Redkost ne nastopa v obliki pravilnih vzorcev, lahko pa jo izkoristimo za pospešitev algoritmov za računanje z matrikami [Berry, 99].

Prvi korak metode za gradnjo hierarhične strukture konceptov je običajen korak pri metodah pridobivanja informacij. Drugi je običajen v primerih, ko želimo grupirati izraze po podobnosti. Naslednji koraki so izvirni in specifični za metodo, ki je predmet tega prispevka.

3. Računanje globalnih korelacijskih stopenj in iskanju korenov hierarhij

Elementi c_{ij} matrike korelacij C izražajo podobnost parov izrazov i in j. Da bi lahko ugotovili, kako so posamezni izrazi korelativni z vsemi izrazi v množici dokumentov, smo uvedli tudi globalno korelacijsko stopnjo Gks_i za vsak izraz i:

$$Gks_i = \frac{1}{m}\left(\sum_{j=1}^{m} c_{ij} + k_i\right), \qquad (2)$$

m predstavlja število izrazov; k_i predstavlja število izrazov, s katerimi je korelativen izraz i.

Globalna korelacijska stopnja, izračunana po enačbi 2, je dajala najustreznejše empirične rezultate, kar je potrdilo intuitivno pričakovanje, da je najustrezneje upoštevati enakomeren vpliv razmerja med številom korelacij izraza i in št. vseh izrazov ter povprečne višine korelacij za izraz i.

Izvedli smo tudi teste, pri katerih smo prišteli h Gks_i globalno entropijsko utež izraza i, ki smo jo omenili v prvem koraku, saj se v prejšnjih korakih metode ni mogla postati opazna. Vendar se je pokazalo, da njen vpliv ni dovolj velik (ali dovolj različen od že upoštevanih sumandov), da bi spremenil izbor korenov hierarhij ali izboljšal njihovo gradnjo.

Po izračunu vseh Gks, ki jih uredimo po velikosti od najvišje do najnižje, lahko določimo korene hierarhij po naslednjem postopku:

- prvi koren postane izraz z najvišjo Gks,
- vsak naslednji koren postane izraz z nižjo Gks, ki je z dosedaj izbranimi koreni korelativen nižje od korenskega praga Kp (ali z drugimi besedami: kosinus kota med kandidatom za koren in dosedanjimi koreni mora biti nižji od Kp).

Za določitev korenskega praga Kp so se pokazale kot najbolj uporabne vrednosti med 0,2 in 0,5, za večino

primerov lahko vzamemo vrednost 0,3. Višja vrednost vodi k več korenom, ki so večinoma močneje zastopani v konceptih hierarhij, nižja vrednost pa k manj korenom, od katerih so nekateri šibkeje zastopani (zaradi nizke *Gks*). Če želimo minimizirati število hierarhij, lahko določimo prag samodejno, z računanjem korenov na določenem intervalu in izborom praga, pri katerem je njihovo število minimalno.

Na opisani način izberemo za korene izraze, ki nastopajo večinoma v različnih konceptih in so najbolj zastopani v ustrezni hierarhiji.

Korene lahko določi tudi *uporabnik*, npr. na podlagi seznama izrazov, ki so urejeni po višini *Gks*. Ko jih izbere, se lahko določijo po zgornjem postopku samodejno še drugi koreni, da bodo v hierarhični strukturi zajeti vsi koncepti v dokumentih.

Rezultate drugega in tretjega koraka metode, uporabljene na demonstracijskem primeru, prikazuje tabela 2. V prvem stolpcu so globalne korelacijske stopnje (*Gks*) izrazov 1 do 15 (indeksi izrazov iz tabele 1), v drugem stolpcu so izrazi, ki so bili izbrani kot koreni pri pragu 0,3.

	Gks				Koreni
1.	0,1	9.	0,592	2.	information (0,743)
2.	0,743	10.	0,205	11.	graph (0,659)
3.	0,214	11.	0,659	9.	survey (0,592)
4.	0,408	12.	0,305	5.	dunhuang, frescoes, based, similarity, calculation (0,1)
5.	0,1	13.	0,1		
6.	0,588	14.	0,1		
7.	0,2	15.	0,214		
8.	0,205				

Tabela 2:
Demonstracijski primer – globalne korelacijske stopnje in izbrani koreni

Štirje koreni določajo štiri ločene hierarhije, znotraj katerih bodo umeščeni koncepti iz dokumentov. Četrti koren z nizko Gks_i je bil izbran kot koren zato, ker je z vsemi prejšnjimi korelativen nižje od praga *Kp* in bi lahko manjkal v celotni hierarhični strukturi konceptov, v katerem se nahaja, če bi ga izpustili.

4. Gradnja hierarhične strukture

Hierarhična struktura konceptov, ki jo zgradi metoda iz vsebine baze dokumentov, je sestavljena iz več drevesnih hierarhij, katerih korene smo določili v prejšnjem koraku. Gradnja vsake hierarhije poteka ločeno in po nivojih.

Preden začnemo z gradnjo, pripravimo za vsak izraz *j* korelacijske sezname Ks_j, to je sezname izrazov, ki so z njim korelativni. Sezname uredimo v padajočem redu po njihovi globalni korelacijski stopnji. S tem pospešimo gradnjo konceptov v hierarhiji, sicer bi ob iskanju otrok določenega izraza (prvi korak algoritma, ki je opisan v tem razdelku) iskali vsakič kandidata z maksimalno *Gks*.

Začnemo z uvrstitvijo korenskega izraza v hierarhijo. Ta je skupen vsem konceptom znotraj te hierarhije. Gradnja konceptov se nato nadaljuje po nivojih do listov – izrazov, ki nimajo otrok, kar pomeni, da ni več v okviru določenega koncepta korelativen z njimi noben drug izraz. Vsak otrok v hierarhiji ima le enega starša. Otroci določenega starša se uvrščajo v hierarhijo po višini *Gks*, najprej tisti z višjo stopnjo.

Koncepti so tako predstavljeni v drevesni hierarhiji kot *vse neciklične poti* od korena do lista.

Sledi postopek gradnje konceptov po nivojih:

Najprej vstavimo na prvi nivo vsake hierarhije indeks izraza, ki je njen koren, in mu dodamo množico indeksov dokumentov, v katerih nastopa (to so indeksi neničelnih elementov v njegovi vrstici matrike *A*). Množico indeksov dokumentov, ki so prirejeni indeksu izraza *j* v okviru določenega koncepta, imenujemo v nadaljevanju 'dokumenti izraza *j*'. Dokumenti se prenašajo navzdol do listov, s čimer zagotavljamo, da so izrazi (otroci) na nižjih nivojih vsebovani v istih konceptih kot njihovi starši.

Koncepte nato gradimo po nivojih. Izrazu *j*, ki se že nahaja v hierarhiji na nivoju *l*, dodamo otroke, ki nastopajo v okviru istega koncepta ali več konceptov, iterativno po naslednjih fazah:

I. Preverimo, ali obstaja v korelacijskem seznamu Ks_j kandidat za njegovega otroka. Če ne obstaja, preverimo, ali je izraz *j* list, kar pomeni, da v okviru tega koncepta noben izraz iz Ks_j ni mogel postati otrok. Če ni list, odstranimo iz njegovih dokumentov tiste, v katerih so vsebovani tudi njegovi otroci, da niso vsem izrazom v konceptu prirejeni dokumenti, ampak le listom. V obeh primerih označimo, da je izraz j v okviru tega koncepta zadnji - komponente vrstice *j*, ki ustrezajo njegovim dokumentom, v matriki *A* postavimo na 0 in končamo z gradnjo koncepta. Z ažuriranjem matrike *A* preprečimo, da bi se koncepti ali deli konceptov v hierarhiji po nepotrebnem ponavljali.

II. Vzamemo naslednjega kandidata za otroka iz seznama Ks_j. Pri njem preverimo:
   a. ali nastopa v okviru koncepta, ki ga gradimo – torej ali se pojavlja v dokumentih izraza *j*. Če se ne, začnemo ponovno s fazo I.
   b. ali se nahaja med predniki tega izraza. Če se nahaja, začnemo ponovno s fazo I.
   c. ali nastopa v enakem konceptu, kot kateri od že dodanih otrok svojega starša – torej ali je korelativen s katerim od njih. Če je, je smiselno, da ga

uvrstimo pod njega, zato ga ne dodamo pod izraz j. S tem zagotovimo celovitost konceptov. Če je iz korelacijskih seznamov takšnega otroka in izraza, ki ga dodajamo, bila izbrisana njuna relacija v katerem od prejšnjih konceptov, jo povrnemo, da ga bomo lahko kasneje res uvrstili pod njega.

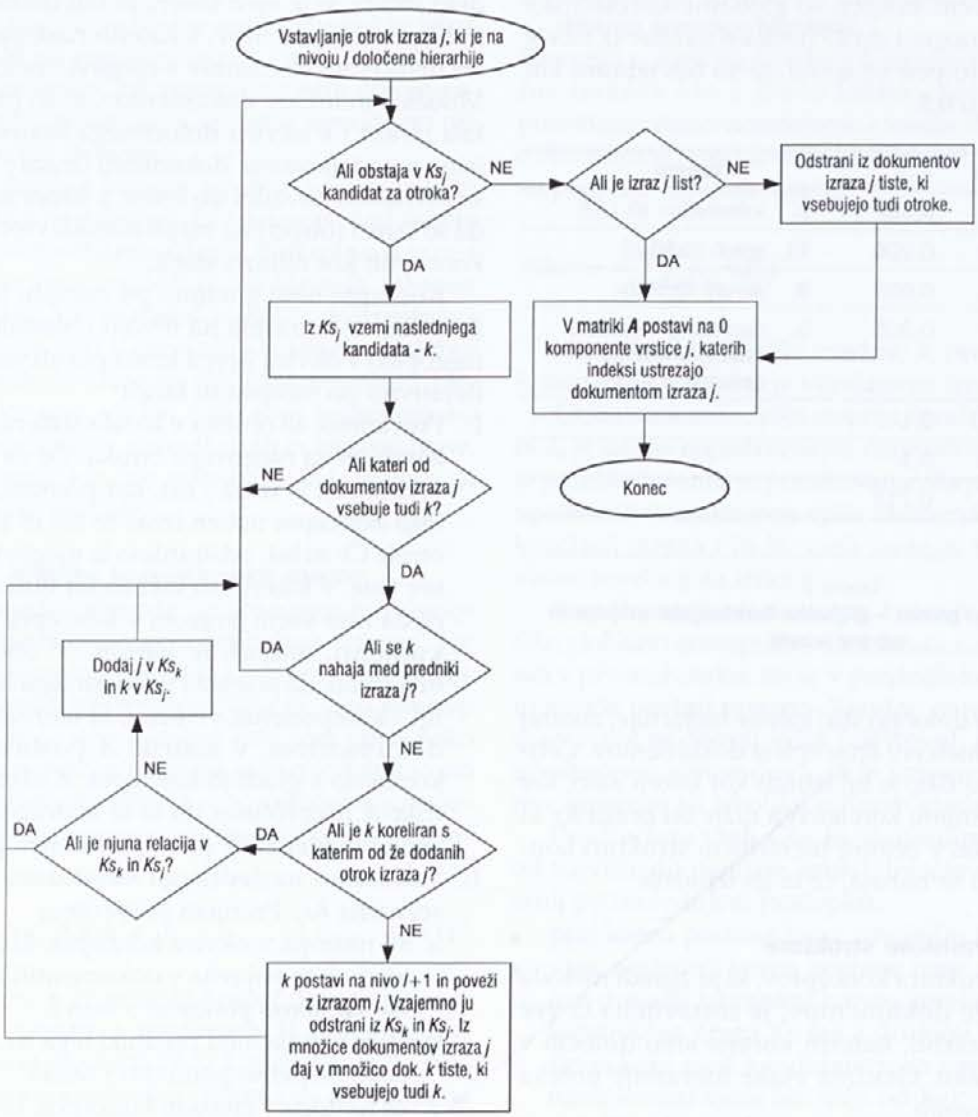
III. Izraz, ki je postal otrok, postavimo na nivo $l+1$ in ga povežemo z izrazom j. Dodamo mu podmnožico dokumentov izraza j, v katerih nastopa v okviru tega koncepta, in zbrišemo otroka iz korelacijskega seznama izraza j ter izraz j iz korelacijskega seznama otroka, da se ne ponovi njuna relacija (lahko jo povrnemo v fazi II).

Slika 1 prikazuje diagram opisanega algoritma za gradnjo konceptov. Na sliki 2 je predstavljen rezultat tega koraka metode, uporabljene na demonstracijskem primeru. Hierarhije so zgrajene in oštevilčene pri korenskih izrazih. Pri izrazih na nižjem nivoju so navedene v okroglih oklepajih njihove globalne korelacijske stopnje. Pri listih se nahajajo v koničastih oklepajih seznami indeksov dokumentov. Koncept, ki ga tvorijo izrazi od lista do korena, nastopa v vseh teh dokumentih.

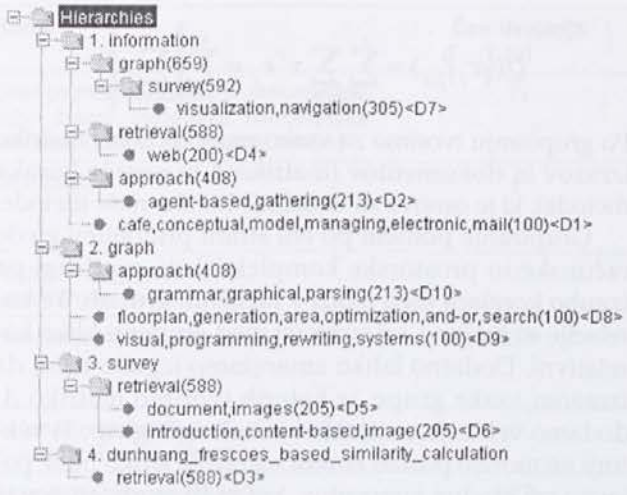
4. Razširitve metode v smeri večje učinkovitosti, neodvisne od jezika

4.1 Uporaba analize glavnih komponent

Pri pridobivanju informacij na osnovi vektorskega prostora se pogosto uporablja metoda za analizo glavnih



Slika 1: Algoritem gradnje konceptov – vstavljanje otrok izraza j, ki se nahaja na nivoju l določene hierarhije



Slika 2: **Demonstracijski primer – hierarhična struktura konceptov**

komponent, ki se imenuje razcep na singularne vrednosti (RSV) [Furnas, 88; Berry, 99]. RSV se uporablja sicer tudi pri procesiranju slik, obdelovanju signalov, stiskanju podatkov in v kriptografiji.

S to metodo lahko odstranimo iz začetne matrike izrazov in dokumentov "šum", to je vpliv, ki ga imajo dokumenti in izrazi, ki z ostalo večino vsebinsko niso povezani, in vpliv, ki nastane zaradi različne rabe besed. Uporaba RSV se je pokazala kot učinkovita pri prepoznavanju sopomenskih in večpomenskih besed. Ena od raziskav je pokazala, da je pridobivanje informacij na osnovi te metode, po učenju na študentski enciklopediji, vračalo enake odgovore na standardnem testu angleških sinonimov, kot so jih dajali uspešni tuji kandidati za študij v Ameriki [Landauer, 96]. Z RSV razcepimo osnovno matriko A (m x n) na

$$A = U\Sigma V^T,$$

kjer je U ortogonalna matrika dimenzij m x m, katere stolpci so levi singularni vektorji A, V je ortogonalna matrika dimenzij n x n, katere stolpci so desni singularni vektorji A in Σ je matrika dimenzij m x n, katere diagonalni elementi so singularne vrednosti A, urejene od najvišje do najnižje. Po razcepu lahko odrežemo k najnižjih singularnih vrednosti, s čimer aproksimiramo matriko A z matriko A_k (rečemo tudi, da zmanjšamo rang matrike na k). A_k, izračunana z RSV, velja za optimalno aproksimacijo ranga k matrike A [Eckart, 36; Stewart, 00].

Zaradi lastnosti metode RSV smo želeli ovrednotiti njen vpliv tudi v okviru naše metode, torej ugotoviti, kako bi RSV vplival na strukturo in povezavo konceptov. Razcep uporabimo v našem kontekstu na matriki A takoj po prvem koraku metode za gradnjo hierarhične strukture. Enačbo 1, po kateri smo v drugem koraku računali korelacije med vektorji izrazov, lahko sedaj preoblikujemo, saj ni potrebno po razcepu izračunati aproksimirane matrike A_k, da dobimo vektorje izrazov, ampak je dovolj, da izračunamo $\Sigma_k U_k^T$ (dimenzij k x m). Če definiramo $w_j = \Sigma_k U_k^T e_j$ (j=1,...,n), lahko enačbo 2 zapišemo kot:

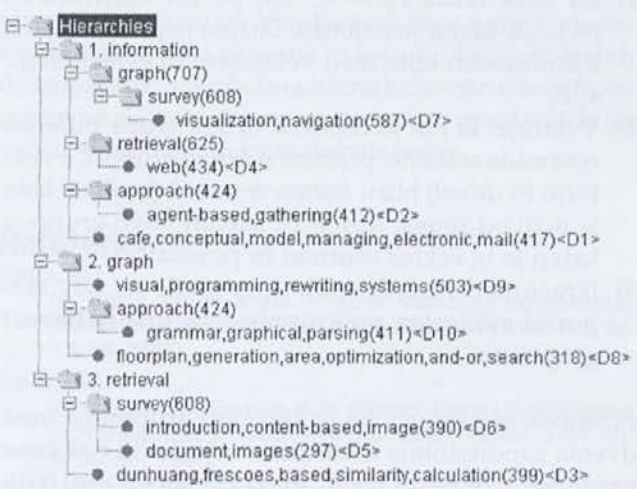
$$c_{ij} = \cos\varpi_{ij} = \frac{w_i^T w_j}{\|w_i\| \|w_j\|},$$

za i,j=1,...,m. Če se pojavijo po razcepu matrike A zaradi aproksimacije novi bratje (izrazi, katerih korelacija je 1), jih združimo, tako kot smo to naredili že v prvem koraku.

Razvite so bile hitre iterativne metode za izračun RSV, ki so prilagojene tudi redkim matrikam, kot sta Arnoldi-jeva [Lehoucq, 96] in Lanczos-eva [Parlett, 80], s katerimi se osnovna časovna in prostorska zahtevnost razcepa ($O(mn^2)$ za polno matriko) precej zmanjša (manj kot $O(mn)$).

Testi so pokazali, da z večjo aproksimacijo matrike A izgubljamo poleg šuma tudi natančnost korelacij, zaradi česar se vzpostavlja med izrazi vse več bratov in postajajo koncepti vse bolj splošćeni. Prednost uporabe RSV v gradnji hierarhične strukture konceptov je, da se združujejo koncepti, ki vsebujejo veliko skupnih izrazov, s čimer dosežemo manjšo občutljivost za variacije v rabi besed in za sinonime, kar je tudi sicer splošna značilnost RSV, znana iz pridobivanja informacij.

Težava pri uporabi te metode je, da se ne da vnaprej določiti optimalnega ranga aproksimirane matrike, ampak se to počne empirično. V pomoč so nam lahko raziskave, ki so v praksi [Berry, 95] in s statistično analizo [Ding, 99] pokazale, da je optimalni rang, v katerem se ohrani večina pomembne vsebinske strukture dokumentov, okrog 100 do največ 300.



Slika 3: **Demonstracijski primer – hierarhična struktura konceptov po razcepu na sing. vred., ohranili smo 6 sing.vred.**

Tak rang je optimalen pri bazah dokumentov, ki vsebujejo vsaj nekaj sto dokumentov.

Slika 3 prikazuje hierarhično strukturo, zgrajeno z aproksimacijo matrike A (iz demonstracijskega primera) z matriko ranga 6. Vidimo lahko, da se je tretji koren spremenil glede na hierarhijo na sliki 2, s čimer se je koncept, zajet v četrti hierarhiji na sliki 2, pravilno povezal s koncepti v tretji hierarhiji. Pri rangu 6 torej še nismo imeli nobene izgube v vsebini konceptov, pri nižjih pa je že prihajalo do izgub.

4.2 Pospešitev metode z grupiranjem izrazov

Računsko in prostorsko kompleksnost drugega in četrtega koraka gradnje hierarhične strukture konceptov lahko učinkovito zmanjšamo z uvedbo grupiranja izrazov. V ta namen smo prilagodili algoritem za sferično grupiranje v k grup [Dhillon, 01], ki je različica evklidskega algoritma za grupiranje v k grup - $W_1,\ldots, W_k$. Algoritem smo razširili z možnostjo dinamičnega ustvarjanja novih grup, kajti ne da se vnaprej oceniti, koliko grup bodo tvorili izrazi, ki opisujejo podobno vsebino, iz poljubne baze dokumentov.

Grupiranje izrazov izvedemo po koraku gradnje matrike izrazov in dokumentov. Najprej razporedimo m normaliziranih vektorjev izrazov $(x_1, \ldots, x_m)$ matrike A naključno v k grup. Nato izračunamo vsem grupam normalizirane centroide:

$$c_j = \frac{\sum_{x\in\Omega_j} x}{\left\| \sum_{x\in\Omega_j} x \right\|},$$

za $j=1,\ldots,k$. (3)

Iterativni postopek za grupiranje je naslednji:

1) Za vsak izraz x_i, $i=1,\ldots,m$, poišče centroid c_j, $j=1,\ldots,k$, ki mu je najbližji. Bližina je predstavljena s kosinusom kota med vektorjem in centroidom: $x_i^T c_j$.
2) Vektorje, ki jim je centroid druge grupe bliže od centroida matične, prestavi v drugo grupo. Če vektorju ni dovolj blizu noben centroid (kosinus kota je nižji od praga, npr. 0,1), ustvari novo grupo, v kateri je ta vektor centroid in poveča k.
3) Izračuna centroide vseh grup glede na nov razpored vektorjev po grupah. Izračuna kakovost grupiranja.

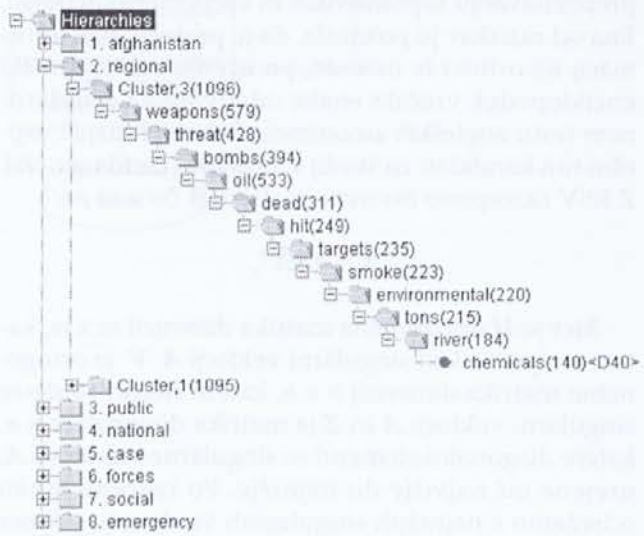
Postopek se zaključi, ko je kakovost grupiranja med dvema zaporednima iteracijama manjša od določene vrednosti e (v naših testih smo uporabljali e=0.001). Kakovost grupiranja izračunamo kot vsoto kakovosti ali koherenc posameznih grup in ima limito, h kateri hitro konvergira [Dhillon, 01]:

$$Q(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^{k} \sum_{x\in\pi_j} x^T c_j = \sum_{j=1}^{k} \left\| \sum_{x\in\pi_j} x \right\|.$$

Po grupiranju tvorimo za vsako grupo posebej matriko izrazov in dokumentov (matriko A iz prvega koraka metode), ki je osnova za nadaljnje korake naše metode.

Grupiranje pomeni po eni strani pridobitev glede računske in prostorske kompleksnosti, po drugi pa izgubo korelacij med izrazi v različnih grupah. Te korelacije so majhne, saj so izrazi med grupami šibko korelativni. Dodatno lahko zmanjšamo izgubo tako, da izrazom vsake grupe, iz katerih tvorimo matriko A, dodamo vektorje centroide vseh drugih grup. Ti vektorji ne morejo postati koreni hierarhij, lahko pa se pojavijo pri gradnji konceptov, kot indikatorji izrazov iz druge grupe, ki nastopajo v tem konceptu.

Primer hierarhične strukture, zgrajene po grupiranju izrazov, predstavljamo na sliki 4. Prikazana je struktura, zgrajena iz druge grupe. Centroidi drugih grup, ki so vključeni v koncepte, so predstavljeni z izrazom 'Cluster', številko grupe in globalno korelacijsko stopnjo. Baza dokumentov za ta primer je enaka, kot v naslednjem poglavju pri prikazu časovne zahtevnosti. Začeli smo z grupiranjem v dve grupi, tretja je bila ustvarjena med grupiranjem. Grupiranje je bilo zaključeno po 19 iteracijah.



Slika 4: Hierarhična struktura po grupiranju

5. Časovna zahtevnost metode

Časovno zahtevna sta koraka metode za računanje matrike korelacij ($O(m^2n)$) in za gradnjo hierarhične strukture, katerega zahtevnost je zelo odvisna od korelacij med izrazi. Zahtevna je tudi razširitev z uvedbo razcepa s singularnimi vrednostmi ($O(mn^2)$), ki pa se z iterativnimi postopki zmanjša na $O(mn)$.

Korak	Čas izvajanja (s)	%	Čas izvajanja z razcepom (s)	%	Čas izvajanja z grupiranjem (s)	%
Gradnja matrike izrazov in dokumentov (izbor izrazov, uteževanje, normalizacija vektorjev izrazov, združevanje bratov)	77	13	77	13	77	27
Grupiranje	/		/		10 (3 grupe, 19 iteracij)	3,5
Razcep s sing. vrednostmi (aproksimac. na rang 40)	/		15	2,5	/	
Računanje matrike korelacij	118	21	137	23	55 (skupaj, za vse grupe)	19
Računanje globalnih korelacijskih stopenj in iskanje korenov hierarhij	3	0,5	3	0,5	3	1
Gradnja hierarhične strukture	373 (23 hierarhij)	65,5	354 (15 hierarhij)	61	141 (skupaj, za vse grupe, skupno 32 hierarhij)	49,5
Skupno	**571**	**100**	**586**	**100**	**286**	**100**

Tabela 3: **Razmerja časovnih zahtevnosti korakov metode**

Metodo smo implementirali v okolju za matematično modeliranje Scilab 2.6, ki ga je razvil INRIA – Francoski nacionalni inštitut za raziskave na področju računalništva in avtomatike (http://www-rocq.inria.fr/scilab/). V tem okolju se programski ukazi interpretirajo. Zelo počasno je v tem okolju tudi obdelovanje netipiziranih seznamov, ki jih uporabljamo pri gradnji strukture in obdelovanje redkih matrik. Zaradi teh razlogov so časovne vrednosti, ki jih navajamo v tabeli 3, namenjene zgolj približni oceni razmerij časovne zahtevnosti med posameznimi koraki metode. Hitrost izvajanja metode, implementirane s prevajanim programskim jezikom, npr. C++, v katerem bi lahko uporabili tudi bolj učinkovite podatkovne strukture, bi bila precej večja.

Baza dokumentov za prikaz časovne zahtevnosti je vsebovala 110 odstavkov desetih daljših člankov o globljih vzrokih terorizma in o vojni v Afganistanu. Od 3078 izrazov, ki so ostali po izločanju blokiranih besed, smo ohranili izraze, ki so se pojavljali v celotni bazi dokumentov s frekvenco, višjo od 4. Teh izrazov je bilo 305.

6. Zaključek

Metoda, ki smo jo razvili, gradi hierarhično strukturo konceptov iz vsebine baze dokumentov neodvisno od jezika, ki je uporabljen v dokumentih. Učinkovitost kompleksnih metod za pridobivanje informacij je v praksi odvisna od integracije zmogljivih statističnih pristopov z izčrpnimi jezikovnimi bazami – predvsem slovarji in tezavri. S slovarjem določenega jezika in/ali tezavrom določenega vsebinskega področja bi lahko naredili natančen izbor izrazov, ki jih želimo vključiti v hierarhično strukturo, in jih ustrezno utežili.

Vendar v praksi slovarji in tezavri večinoma niso na voljo in smo v teh primerih odvisni predvsem od statističnih značilnosti teksta. Na njihovi podlagi in v povezavi z uporabniškim izborom korenov hierarhij ter razcepom s singularnimi vrednostmi zgradi opisana metoda zelo uporabno hierarhično strukturo, ki omogoča učinkovit pregled konceptov in iskanje po njih ali njihovih delih – tako v hierarhijah kot v samih dokumentih, saj vsebujejo hierarhije informacije o dokumentih, v katerih se nahajajo posamezni koncepti.

Časovno zahtevnost lahko precej zmanjšamo z grupiranjem izrazov. Vendar tudi brez grupiranja časovna zahtevnost ni resna ovira pri običajnih bazah dokumentov, saj gradnja hierarhične strukture konceptov ni pogosta dejavnost, ampak jo izvedemo le ob večjih spremembah baze dokumentov.

Reference:

[Berry, 92] M. Berry, Large scale singular value computations. *International Journal of Supercomputer Applications*, 6:1, str. 13-49, 1992.

[Berry, 95] M. W. Berry, S.T. Dumais, G.W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review*, 37:4, str. 573-595, 1995.

[Berry, 99] M. W. Berry, Z. Drmač, E. R. Jessup, Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*, 41:2, str. 335-362, 1999.

[Dhillon, 01]
I. S. Dhillon, D. S. Modha, Concept Decompositions for Large Sparse Text Data Using Clustering. *Machine Learning*, 42:1/2, str.143-175, 2001.

[Dimec, 99]
J. Dimec, L. Todorovski, D. Hristovski, S. Džeroski, The personalized search engine for Slovenian and English medical documents. V *Managing multimedia collections*, Bled, str.56-63, 1999.

[Ding, 99]
C. H. Q. Ding, A Similarity-Based Probability Model for Latent Semantic Indexing. V *Proceedings of 22nd International Conference on Research and Development in Information Retrieval*, Berkeley, str. 58-65, 1999.

[Dumais, 91]
S. T. Dumais, Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments & Computers*, 23:2, str. 229-236, 1991.

[Eckart, 36]
C. Eckart, G. Young, The approximation of one matrix by another of lower rank. *Psychometrika*, 1, str. 211-218, 1936.

[Forsyth, 86]
R. Forsyth, R. Rada, Adding an edge in Machine Learning. V *Machine Learning: Applications in Expert Systems and Information retrieval*, str. 198-212, 1986.

[Frakes, 92]
W. B. Frakes, R. Baeza-Yates, Information Retrieval: Data Structures & Algorithms, Prentice Hall, ZDA, 1992.

[Furnas, 88]
G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, K. E. Lochbaum, Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure. V *Proceedings of ACM SIGIR*, str. 465-480, 1988.

[Harman, 95]
D. Harman. Overview of the third text REtrieval conference (TREC-3). V *Overview of the Third Text REtrieval Conference, National Institute of Standards and Technology Special Publication*, NIST, str. 1-21, 1995.

[Jansen, 98]
B. Jansen, A. Spink, J. Bateman, Searchers, the Subjects they Search, and Sufficiency: A Study of a Large Sample of EXCITE Searches. V *Proceedings of World Conference on the WWW, Internet and Intranet (WebNet-98)*, AACE Press, 1998.

[Kowalski, 97]
G. Kowalski, Information Retrieval Systems: Theory and Implementation, Kluwer Academic Publishers, 1997.

[Landauer, 96]
T. K. Landauer, S. T. Dumais, How come you know so much? From practical problem to theory. V *Basic and applied memory: Memory in context*, NJ: Erlbaum, str. 105-126, 1996.

[Lehoucq, 96]
R. Lehoucq, D. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, *SIAM Journal on Matrix Analysis and Applications*, 17:4, str. 789-821, 1996.

[Parlett, 80]
B. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, NJ, 1980.

[Popovič, 91]
M. Popovič, Implementation of a Slovene language free-text retrieval system. Doktorska disertacija, University of Sheffield, Department of Information Studies, Združeno kraljestvo, 1991.

[Porter, 80]
M. F. Porter, An algorithm for suffix stripping. *Program*, 14:3, str. 130-137, 1980.

[Porter, 01]
M. F. Porter, Snowball: A language for stemming algorithms, http://snowball.sourceforge.net/texts/introduction.html, 2001.

[Salton, 71]
G. Salton, The SMART Retrieval System, Prentice Hall, 1971.

[Sanderson, 99]
M. Sanderson, B. Croft, Deriving concept hierarchies from text. V *Proceedings of the 22nd Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, Kalifornija, str. 206-213, 1999.

[SMART]
SMART English stopword list, SMART project, Cornell University, ftp://ftp.cs.cornell.edu/pub/smart/.

[Stewart, 00]
G. W. Stewart, The Decompositional Approach to Matrix Computation. *IEEE Computing in Science & Engineering*, 2:1, str. 50-59, 2000.

[Woods, 97]
W. A. Woods, Conceptual Indexing: A Better Way to Organize knowledge. *Sun Labs Technical Report: TR-97-61*, Technical Reports, Palo Alto, 1997.

[Yang, 00]
H.C. Yang, C. H. Lee, Automatic Category Generation for Text Documents by Self-Organizing Maps. V *Proceedings of the IEEE-INNS-ENNS Int. Joint Conference on Neural Networks (IJCNN'00)*, Italija, str. 3581-3586, 2000.

[Yates, 99]
R.B. Yates, B.R. Neto, Modern Information Retrieval, Addison-Wesley Pub. Co., New York, 1999.

◆

Robert Leskovar je asistent – mladi raziskovalec na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Diplomiral je leta 1998 in končuje doktorski študij računalništva in informatike. Področja njegovega raziskovalnega dela zajemajo pridobivanje informacij s poudarkom na prepoznavanju konceptov v nestrukturiranem tekstu, tehnologije računalniško posredovanega komuniciranja in zagotavljanje kakovosti pri razvoju informacijskih sistemov ter drugih delovnih procesov.

◆

Dr. József Györkös je izredni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Področja njegovega dela zajemajo računalniško posredovano komuniciranje, teorijo upravljanja zahtev in zagotavljanje kakovosti pri razvoju informacijskih sistemov ter širše. Znanje in izkušnje je s sodelavci apliciral v nekaterih večjih družbah in na ravni informatike v vladnih institucijah. Objavil je več strokovnih in znanstvenih publikacij v mednarodnih revijah in knjigah. Trenutno opravlja delo dr•avnega sekretarja na Ministrstvu za informacijsko družbo.

◆

Dr. Ivan Rozman je redni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Je dekan fakultete in vodja Inštituta za informatiko. Področja njegovega pedagoškega, znanstvenega in raziskovalnega dela zajemajo programsko inženirstvo, objektno tehnologijo, vodenje projektov in kakovost v programskem inženirstvu. Je vodja več raziskovalnih projektov in avtor številnih znanstvenih ter strokovnih publikacij.