SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1552

REDUKCIJA DIMENZIONALNOSTI TEMELJENA NA SVD-u I ALGORITMI GRUPIRANJA PODATAKA

Boris Bereček

Zagreb, rujan 2005.

Zahvaljujem prof. dr. sc. Bojani Dalbelo-Bašić na stručnom vodstvu, pomoći i savjetima pri izradi diplomskog rada.

Sadržaj

1. Uvod

Od davnih vremena ljudsko se znanje čuvalo u obliku pisane riječi u knjigama. Ni danas nije ništa drugačije, jedino su računala dodala još jedan medij za pohranu - Internet. Veliko širenje Interneta kao glavnog medija za razmjenu informacija i njegova velika dostupnost potiču ljude da sve više stvaraju i razmjenjuju informacije i znanje. A većina je toga u obliku teksta…

Motivacija

Zamislite samo velike količine teksta koje se svakodnevno generiraju u obliku novinskih članaka, knjiga, internetskih stranica, elektronske pošte, raznih dopisa, istraživačkih studija itd. Ako nas zanima neka od tih tema, jednostavno nemamo dovoljno vremena da pročitamo sve te dokumente, a kamo li da izvučemo važne informacije koje se u njima nalaze. Dakle postoji očita potreba za automatiziranim računarskim sustavom koji će najprije vršiti odabir svih, a zatim samo najvažnijih dokumenata iz određene grupe koja vas zanima.

Ako ste posljednjih godina pročitali uvod nekoliko radova koji obrađuju ovakvu temu, mogli ste primijetiti da svi oni počinju na isti način – baš kao što počinje i ovaj rad. Svi autori kao glavnu motivaciju spominju Internet i sve veći broj dostupnih tekstualnih dokumenata. Nije to zbog nemaštovitosti autora, već zato jer je to uistinu najveći i najvažniji razlog zašto se na ovom području toliko intenzivno radi. Istraživanja pokazuju da se čak 80% informacija i znanja nalazi pohranjeno u obliku teksta! Dakle, kako to iskoristiti?

Rješenje

Jedno rješenje tog problema daje postupak dubinske analize teksta (engl. t*ext mining*). Dubinska analiza teksta odnosi se na traženje zanimljivih i netrivijalnih informacija i znanja u nestrukturiranom tekstu, zatim **grupiranje** i klasifikaciju teksta. To je mlada znanstvena disciplina koja ima svoje korijene u disciplinama kao što su dohvat informacija (engl. *information retrieval*), dubinska analiza podataka (engl. *data mining*), strojno učenje (engl. *machine learning*), statistika (engl. *statistics*) i računalna lingvistika (engl. *computational linguistics*).

Do sada već postoji niz alata koji rade dubinsku analizu teksta. No svi se oni susreću s istim problemima. Naime, prirodni jezik, kakav se obično nalazi u dokumentima, nije namijenjen za analitičku obradu. Takav je jezik nestrukturiran i da bi se mogao obrađivati na računalu mora proći niz postupaka koji se nazivaju pretprocesiranje. Taj postupak stvara numeričku reprezentaciju dokumenata u obliku matrice. Zbog velikog broja različitih riječi koje se susreću u dokumentima, potrebno je obaviti još jedan korak – ukloniti nepotrebne riječi. Jedan od algoritama kojim je to moguće obaviti naziva se rastav matrice pomoću singularnih vrijednosti ili engl. *Singular Value Decomposition* (***SVD***). Nakon tog koraka primjenjuju se standardni algoritmi za grupiranje podataka. U ovom će radu biti prikazano nekoliko takvih algoritma.

Na tržištu postoji niz gotovih komercijalnih alata namijenjenih dubinskoj analizi teksta. Jedan je od njih SAS *Text Miner* čije je korištenje detaljno objašnjeno u nastavku. Taj je alat također korišten i u svim eksperimentima opisanima u ovom radu.

Zadatak je ovog rada objasniti sve korake dubinske analize teksta s gotovim alatom te usporediti rezultate grupiranja dokumenata na hrvatskome i engleskom jeziku.

2. Dubinska analiza teksta

Svrha dubinske analize teksta jest pomoći ljudima u razumijevanju sadržaja grupe dokumenata bez potrebe da se pročita cijeli tekst svih dokumenata. Primjena dubinske analize teksta može se svesti na dva njezina najvažnija dijela:

- opisnu analizu ili grupiranje (engl. *clustering)* i
- prediktivnu analizu ili klasifikaciju (engl. *classification*).

Ovaj će se rad baviti isključivo postupcima dubinske analize teksta grupiranjem.

Oba navedena postupka dubinske analize podataka dijele neke zajedničke zahtjeve. Naime, tekstualne dokumente koje ljudi mogu jednostavno razumjeti najprije je potrebno pretvoriti u oblik koji omogućuje dubinsku analizu računalom. Dokumenti pisani prirodnim jezikom trebaju biti pretprocesirani prije nego se mogu otkriti uzorci i veze među pojedinim dokumentima. Iako ljudi razumiju poglavlja, odlomke i rečenice, računala zahtijevaju strukturirane podatke. Kao posljedica toga, nestrukturirani dokumenti trebaju biti prebačeni u strukturirani oblik prije nego što se na njima može napraviti dubinska analiza.

2.1. Pregled postupka dubinske analize teksta

U prirodnom jeziku postoje neke riječi koje ne govore ništa o sadržaju dokumenta u kojem se nalaze. Najbolji primjer za te riječi su npr. veznici, prilozi, prijedlozi (i, ili, ali, uvijek, sada, danas, na, pod itd.). Takve riječi moramo izbaciti iz dokumenata jer nam ne pomažu u utvrđivanju njihova sadržaja, nego samo uzrokuju šum u podacima.

Još jedan od koraka koji je potrebno obaviti jest smanjivanje broja pojavnica pojedine riječi. Naime, riječi se mogu pojavljivati u jednini, množini, raznim padežima, rodovima i sl. Sve one imaju isto ili slično značenje, ali se različito pišu, a to računalima predstavlja problem. Takve riječi moramo svesti na njihov korijen ili osnovni oblik. Treba također pripaziti i na sinonime i grupe riječi. I njih isto treba obraditi na odgovarajući način.

Nakon ovih postupaka potrebno je od tako pretprocesiranog teksta napraviti njegovu numeričku reprezentaciju pogodnu za daljnju obradu na računalu. U tom koraku stvara se matrica riječ-dokument. Ovako dobivena matrica riječ-dokument može imati goleme dimenzije. Nad takvom matricom algoritmi za grupiranje ne bi mogli efikasno raditi. Zato je potreban još jedan važan korak, a to je redukcija dimenzionalnosti.

Konačan korak čini postupak grupiranja dokumenata koji radi nad reduciranom matricom riječ-dokument. Za postupak grupiranja postoji velik broj algoritama.

U nastavku slijedi detaljniji opis svakog od navedenih koraka s pripadajućim algoritmima.

2.2. Pretprocesiranje

Pretprocesiranje teksta nužno je za njegovu što efikasniju analizu. Za početak, uklanjanje stop-riječi gotovo uvijek pomaže. Zatim, kod morfološki bogatih jezika, kao što je hrvatski, posebno je bitan postupak smanjivanja broja pojavnica riječi što je moguće na dva načina. Osim što hrvatski jezik ima raznolik rječnik, te se riječi mogu nalaziti u jako mnogo oblika, čineći pritom ukupan broj pojavnica riječi većim i matricu riječ-dokument još većom i rjeđom. Još jedan korak koji također pripada u pretprocesiranje naziva se pročišćavanje (engl. *pruning*). Taj postupak u nekim slučajevima može popraviti rezultate, ali u manjoj mjeri nego prva dva koraka.

2.2.1. Stop-riječi

U uvodu ovog poglavlja bilo je rečeno da su stop-riječi one riječi koje ne nose neko relevantno značenje za temu koju promatramo. Postoje određene vrste riječi za koje je to u potpunosti istinito (npr. za veznike).

Za neke druge slučajeve i druge tipove riječi nije tako. U tim slučajevima odabir stop-riječi ovisi o kontekstu u kojem želimo promatrati dokumente. Ako npr. želimo grupirati dokumente koji govore o današnjim i prošlim događajima tada u listu stop-riječi nećemo uključiti priloge (npr. *danas*, *jučer*, *sada* itd.). Da smo iste dokumente htjeli grupirati po značenju tada bismo najvjerojatnije navedene priloge uključili kao stop-riječi jer nam u tom kontekstu nisu zanimljive.

Imenice i glagoli rijetko se stavljaju kao stop-riječi no i to je moguće. To opet ovisi o kontekstu u kojem promatramo dokumente. Ako u dokumentima postoji nekoliko riječi koje jako utječu na rezultate analize teksta, a nama nisu zanimljive, tada ih je moguće izbaciti iz analize tako da ih stavimo u listu stop-riječi.

Općenita lista engleskih stop-riječi sastoji se od oko 600 riječi, a u SAS-u znatno manje – samo 330. Za usporedbu, općenita lista hrvatskih stop-riječi sastoji se od 2000 riječi. To je samo jedan od pokazatelja morfološke raznolikosti i bogatstva hrvatskog jezika – ovom slučaju bogatstva koje nam donosi određene probleme kod analize teksta.

2.2.2. Smanjivanje broja pojavnica

Ovaj je korak posebno važan kod morfološki bogatih jezika kao što je hrvatski i potrebno ga je obaviti da bi se olakšao rad sljedećih koraka u postupku analize teksta: npr. riječi *kućama*, *kuće*, *kući*, *kućice* imaju za osnovu riječ *kuća*. No, zbog moguće višeznačnosti riječi ovaj korak nije nimalo lagan. Riječ *mora*, na primjer, može doći iz konteksta *sinja mora* ili

noćna mora. U jednom slučaju radi se o riječi *more*, a u drugom *mora*. Kod morfološki jednostavnijih jezika taj je postupak nešto lakše napraviti.

Osim što na taj način smanjujemo dimenzionalnost ulaznih podataka, povećavamo i kvalitetu teksta koji želimo kategorizirati jer uklanjamo rasipanje značenja istog pojma na više leksički različitih oblika (Šnajder,2005).

Postoje dva načina kako se može ostvariti ovaj korak. Jedan od njih se naziva svođenje riječi na osnovni oblik (lemu) tj. lematizacija, a drugi je svođenje riječi na korijen (engl. *stemming*).

2.2.2.1. Lematizacija

Lematizacija označava svođenje riječi na njezin osnovni oblik – onakav kakav je napisan u rječniku (za imenice je to nominativ jednine muškog roda, za glagole infinitiv itd.).

Za hrvatski jezik napravljen je Automatski Morfološki Normalizator (AMN). To je postupak kojim se pojavnice u tekstu svode na svoje morfološke norme. Idealno, ali praktično nemoguće bi bilo imati lematiziranu bazu. U tom bi slučaju sve riječi bile svedene na osnovni oblik. Postupkom normalizacije pokušava se što više približiti lematiziranoj bazi. Više o tome u (Šnajder,2005).

2.2.2.2. Svođenje na korijen riječi

Najčešće korišteni algoritam za svođenje na korijen riječi jest Porterov algoritam. No, njegova je primjena pogodna samo za engleski jezik i slične morfološki jednostavnije jezike. Ovaj algoritam uklanja nastavke riječi svodeći ih na njihov korijen. Treba napomenuti da korijen riječi ne mora biti stvarna riječ. Tako bi engleske riječi *baking*, *baked* bile bi svedene na korijen *bak* za koji vidimo da nije stvarna riječ. Također, ne mora biti nužno da riječ s istim korijenom predstavlja riječ sa istim značenjem.

Možemo uočiti, dakle, da je lematizacija precizniji način za smanjenje broja pojavnica, ali je i računalno mnogo zahtjevniji i teži za izvođenje.

2.2.3. Sinonimi

Prirodni jezik može sadržavati sinonime – riječi koje se različito pišu, a imaju isto značenje (npr. *put* i *cesta*), Izjednačavanje sinonima također je korak koji bi trebali poduzeti da bi se na kraju dobili što bolji rezultati analize teksta. Za sinonime, kao i za stop-riječi, mora postojati unaprijed zadana lista. Takva se lista sinonima može prilagođavati ovisno o kontekstu dokumenata koji se analiziraju.

2.2.4. Grupe riječi

Grupom riječi nazivamo nekoliko riječi koje opisuju određeni pojam što u analizi teksta može predstavljati problem. Taj problem možemo riješiti

preko liste sinonima, ali samo ako znamo sve grupe riječi (što nije uvijek slučaj). Npr. grupu riječi „*Hrvatsko Narodno Kazalište"* trebalo bi grupirati kao jedan pojam i izjednačiti s riječi *kazalište*.

Drugi pristup problemu može biti statistički. Kroz sve dokumente možemo pratiti pojavljivanje dviju, triju ili više riječi zajedno. Ako se taj skup riječi pojavi više puta od nekog unaprijed određenog praga, tada taj skup možemo nazvati grupom riječi i zamijeniti ih s jednom umjetnom pojavnicom.

2.2.5. Pročišćavanje riječi

Posljednji je korak kod pretprocesiranja pročišćavanje riječi (engl. *pruning*). To je postupak u kojem se uklanjaju riječi koje se pojavljuju rjeđe od unaprijed određenog postotka. Taj postotak obično iznosi oko 1% što znači da će se riječi koje se pojavljuju u manje od 1% dokumenata jednostavno izbaciti iz daljnje analize. Takve su riječi obično nastale zbog pogreške u pisanju dokumenata pa ih je korisno izbaciti jer uzrokuju šum. Isto vrijedi i sa riječima koje se često pojavljuju (npr. u više od 20% dokumenata).

Ovaj korak nije toliko važan ako se pri stvaranju matrice riječ-dokument koriste težinske funkcije. U tom slučaju težinske će funkcije obaviti upravo taj posao – diskriminirat će riječi koje se izrazito rijetko ili prečesto koriste.

2.3. Numeričko predstavljanje dokumenata

Postupcima koji ulaze u fazu pretprocesiranja dobili smo obrađeni tekst koji se sada može pretvoriti u numerički oblik potreban kao ulaz za algoritme grupiranja. Sada se stvara standardna matrica riječ-dokument.

2.3.1. Matrica riječ-dokument

Svaki koeficijent a_{ij} u toj matrici čini broj koliko se puta određena riječ w_i našla u određenom dokumentu d_j. Matrica riječ-dokument služi kao osnova za analizu skupa dokumenata jer se na taj način svaki dokument predstavlja kao vektor.

	$\mathbf{d_1}$	$\mathbf{d_2}$	…	$\mathbf{d_n}$
$\mathbf{w_1}$	a_{11}	a_{12}	…	a_{1n}
$\mathbf{w_2}$	a_{21}	a_{22}	…	a_{21}
$\mathbf{w_3}$	a_{31}	a_{32}	…	a_{31}
…	…	…	…	…
$\mathbf{w_m}$	a_{m1}	a_{m2}	…	a_{mn}

Slika 1. Matrica riječ-dokument.

2.3.2. Težinske funkcije

U većini slučajeva performanse sustava za analizu teksta se mogu povećati korištenjem raznih težinskih funkcija. Te funkcije svoj rad temelje na podatku koliko se pojedina riječ pojavljuje u dokumentu i u skupu dokumenata kao cjelini. Na taj se način favoriziraju riječi koje se pojavljuju umjereno u odnosu na one riječi koje se nalaze u jako puno ili jako malo dokumenata. Prečeste ili izrazito rijetke riječi ne koriste pri analizi teksta, već samo unose šum u podatke i troše vrijeme zbog povećane potrebe za izračunavanjem. Težinske funkcije takvim riječima postavljaju jako male koeficijente u matricu riječ-dokument pa je njihov doprinos u kasnijim koracima postupka zanemariv.

Ukupnu težinu pojedine riječi $\hat{a}_{ij}$ određuju dva faktora: težinska funkcija frekvencije L_{ij} i težina same riječi G_{ij}, tj.

$$\hat{a}_{ij} = L_{ij} G_{ij} . \tag{1}$$

Težinske funkcije frekvencije mogu biti:

- binarne – Frekvencija može biti 0 ili 1. Binarna težinska funkcija se najčešće koristi za dokumente koji sadrže malen rječnik, kao, recimo, skup računalnih programa koji su svi napisani u istim programskim jezikom.

$$L_{ij} = \begin{cases} 1, \text{ako rijec postoji u dokumentu} \\ 0, \text{inace} \end{cases} \tag{2}$$

- logaritamske – Uzima se log_2 svake od frekvencija zbrojen sa 1. Logaritamske težinske funkcije umanjuju utjecaj neke riječi koja se prečesto spominje u dokumentu. Ova se težinska funkcija najčešće i koristi u praksi, jer očito je da riječ koja se spominje deset puta značajnija od riječi koja se spominje jedan ili dva puta.

$$L_{ij} = \log_2 (a_{ij} + 1) . \tag{3}$$

Na težine pojedinih riječi utječe broj pojavljivanja riječi u cijelom skupu dokumenata. Težinske funkcije riječi mogu biti:

- entropija – dodjeljuje najveće težine riječima koje se rijetko pojavljuju u skupu dokumenata.

$$G_i = 1 + \sum_j \frac{p_{ij} \log_2 (p_{ij})}{\log_2 (n)} . \tag{4}$$

- *idf* (engl. *Inverse Document Frequency*) – koristi recipročnu vrijednost broja dokumenata u kojima se određena riječ pojavljuje. Daje sličan rezultat kao i entropija tj. daje veće težine riječima koje se pojavljuju samo nekoliko puta u skupu dokumenata.

$$G_i = \frac{g_i}{d_i}. \tag{5}$$

- *gf*idf* (engl. *Global Frequency times Inverse Document Frequency*) – povećava *idf* vrijednost riječi tako da je množi sa globalnom frekvencijom te riječi. Ova metoda daje veće težine riječima koje se češće pojavljuju nego što im daju metode *idf* i Entropija.

$$G_i = \log_2\left(\frac{n}{d_i}\right) + 1. \tag{6}$$

U jednadžbama f_{ij} označava frekvenciju riječi i u dokumentu j, d_i je broj dokumenata u kojima se riječ i pojavljuje, g_i je broj koliko se puta riječ *i* pojavljuje u cijelom skupu dokumenata, n je broj dokumenata u skupu, a

$$p_{ij} = \frac{f_{ij}}{g_i}. \tag{7}$$

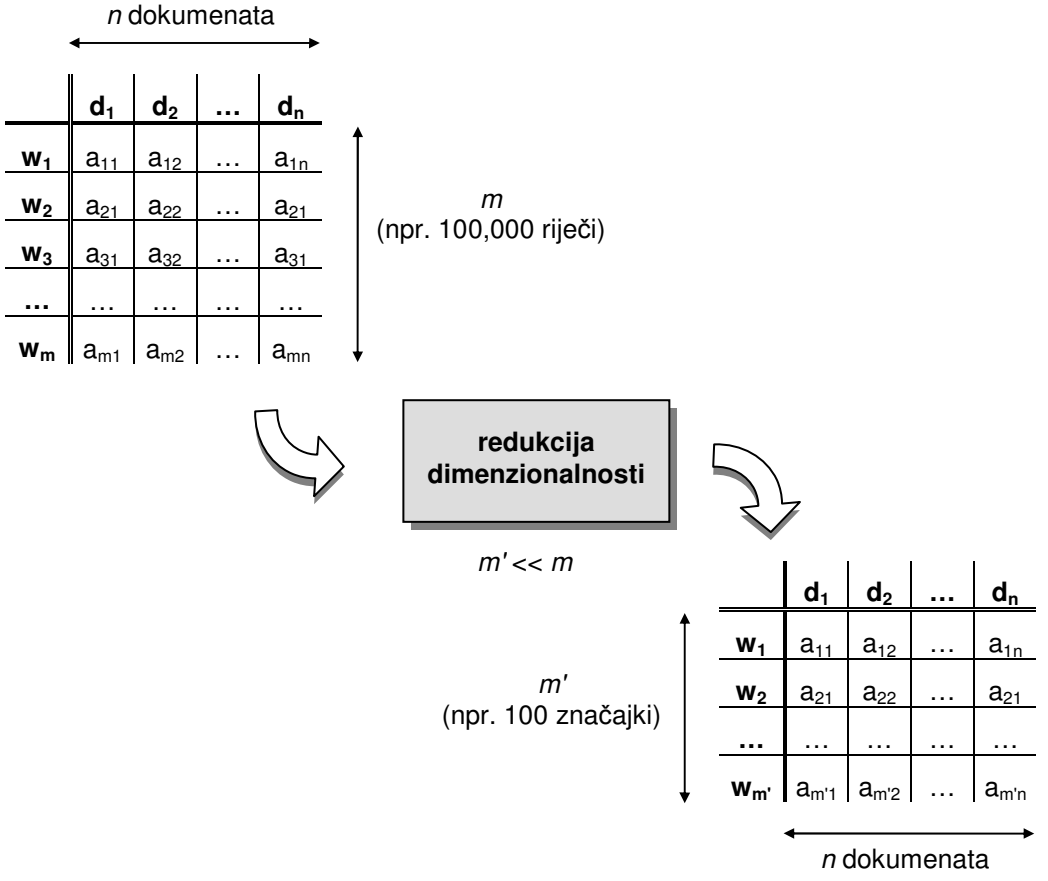
Za potrebe klasifikacije teksta rabe se i težinske funkcije Chi-squared, zajednička informacija (engl. *Mutual Information*) i informacijska dobit (engl. *Information Gain*).

Težinskim funkcijama nismo smanjili dimenzionalnost matrice riječ-dokument već smo samo pojačali ili smanjili utjecaj nekih riječi. Time smo olakšali posao postupcima za redukciju dimenzionalnosti.

3. Redukcija dimenzionalnosti

Uz sve postupke pretprocesiranja i težinskih funkcija matrica riječ-dokument još je uvijek velikih dimenzija. Svaki dokument može biti predstavljen s vektorom od 10 000 ili čak do 100 000 dimenzija. Uz to još treba spomenuti da je ta matrica jako rijetka (engl. *sparse*) – većina elemenata u njoj ima vrijednost nula. Na većim matricama broj nul-elemenata može doseći čak 99%. Rad algoritama za grupiranje bio bi gotovo nemoguć u takvim uvjetima. Oni u pravilu najefikasnije rade sa stotinu ili najviše nekoliko stotina dimenzija.

Postoje dvije metode za redukciju dimenzionalnosti: uzimanje riječi sa najvećim težinama i rastav matrice pomoću singularnih vrijednosti (*SVD*). Obje metode su podržane u SAS *Text Miner* programu, a njihovo će objašnjenje slijediti nešto niže.



Slika 2. Redukcija dimenzionalnosti.

3.1. Riječi s najvećim težinama

Metoda uzimanja riječi s najvećim težinama (engl. *roll-up terms*) heuristička je metoda. Ona sortira matricu riječ-dokument tako da riječi s najvećim težinama stavlja na vrh. Zatim uzima prvih n riječi, a ostale odbacuje. Broj n može se uzeti po potrebi, ali je također moguće pregledom težina riječi doći do optimalnog broja n. Npr. kada težine riječi u sortiranoj riječ-dokument matrici padnu ispod nekog unaprijed određenog praga, tada riječi s manjim pragom možemo odbaciti.

U novoj matrici svaki će dokument biti predstavljen s vektorom čije će vrijednosti i dalje biti težinski koeficijenti riječi. Taj će se vektor sastojati od *n* dimenzija.

Koliko je ova metoda jednostavna, toliko ima i ozbiljnih nedostataka te se ne koristi zasebno za veće skupove dokumenata. Najkorisnija je kada su dokumenti kratki tako da se među njima preklapa jako malo riječi.

3.2. SVD

Rastav matrice pomoću singularnih vrijednosti (*SVD*) važna je tehnika za redukciju dimenzionalnosti, a rabi se u području znanosti koje se bavi dohvatom informacija (engl. *information retrieval*). Njezina dobra svojstva tek su se nedavno počela koristiti u području dubinske analize podataka. Rastuće mogućnosti računalnih sustava omogućile su sve veće iskorištavanje ove računski zahtjevne metode.

3.2.1. Definicija

Pretpostavimo da je, bez gubitka općenitosti, A rijetka matrica dimenzija m puta n koja ima svoj rang r. Rastav matrice A pomoću singularnih vrijednosti definira se kao

$$A = U\Sigma V^T \tag{8}$$

gdje su matrice U i V ortogonalne tj. vrijedi da je $U^T U = V^T V = I_n$ i da je Σ dijagonalna matrica tj. $\Sigma = diag(\sigma_1,...,\sigma_n)$, $\sigma_i > 0$ za $1 \le i \le r$, $\sigma_i = 0$ za $i \ge r+1$. Prvih r stupaca ortogonalnih matrica U i V definira ortonormalne svojstvene vektore vezane sa r svojstvenih vrijednosti različitih od nule matrica AA^T i A^TA. Singularne vrijednosti matrice A definirane su kao dijagonalni elementi matrice Σ, a to su zapravo nenegativni kvadratni korijeni n svojstvenih vrijednosti matrice AA^T. Skup $\{ u_i, \sigma_i, v_i \}$ naziva se *i*-ta singularna trojka.

Pokažimo SVD algoritam na primjeru za koji će nam poslužiti matrica A,

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \in R^{3x2}.$$

Najprije računamo svojstvene vrijednosti matrice A tako da izračunamo

$$\det(A^T A - \lambda I) = 0$$

tj.

$$\det\left(\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0,$$

$$\det\left(\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0,$$

$$\det\left(\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} \right) = 0.$$

Svojstvene vrijednosti su λ_1 = 3, λ_2 = 1. Matrica ima dvije svojstvene vrijednosti različite od nule pa je prema tome rang matrice r = 2. Ortonormalni svojstveni vektori matrice $A^T A$ su

$$\mathbf{v}_1 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}$$

i oni čine matricu

$$V = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix} \in R^{2x2}.$$

Sada tražimo singularnu vrijednost matrice $\Sigma \in R^{2x2}$:

$$\Sigma = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{bmatrix}$$

čiju dijagonalu čine kvadratni korijeni svojstvenih vrijednosti matrice $A^T A$ u padajućem nizu, a ostatak matrice čine nule. Sada tražimo prva dva vektora matrice $U \in R^{3x2}$

$$\mathbf{u}_1 = \sigma_1^{-1} A\mathbf{v}_1 = \frac{\sqrt{3}}{3}\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\sqrt{2}/2\\ \sqrt{2}/2\end{bmatrix} = \begin{bmatrix}\sqrt{6}/3\\ \sqrt{6}/6\\ \sqrt{6}/6\end{bmatrix}$$

i

$$\mathbf{u}_2 = \sigma_1^{-1} A\mathbf{v}_2 = \begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\sqrt{2}/2\\ -\sqrt{2}/2\end{bmatrix} = \begin{bmatrix}0\\ -\sqrt{2}/2\\ \sqrt{2}/2\end{bmatrix}.$$

Vektor $\mathbf{u}_3$ izračunamo Gram-Schmitd postupkom

$$\overset{*}{\mathbf{u}}_3 = \mathbf{e}_1 - (\mathbf{u}_1^T\mathbf{e}_1)\mathbf{u}_1 - (\mathbf{u}_1^T\mathbf{e}_2)\mathbf{u}_2 = \begin{bmatrix}1/3\\ -1/3\\ -1/3\end{bmatrix}$$

koji nakon normiranja postaje

$$\mathbf{u}_3 = \begin{bmatrix}\sqrt{3}/3\\ -\sqrt{3}/3\\ -\sqrt{3}/3\end{bmatrix}.$$

Sada je

$$U = \begin{bmatrix}\sqrt{6}/3 & 0 & \sqrt{3}/3\\ \sqrt{6}/6 & -\sqrt{2}/2 & -\sqrt{3}/3\\ \sqrt{6}/6 & \sqrt{2}/2 & -\sqrt{3}/3\end{bmatrix}.$$

Konačno, rastav matrice A pomoću singularnih vrijednosti je

$$A = \begin{bmatrix}\sqrt{6}/3 & 0 & \sqrt{3}/3\\ \sqrt{6}/6 & -\sqrt{2}/2 & -\sqrt{3}/3\\ \sqrt{6}/6 & \sqrt{2}/2 & -\sqrt{3}/3\end{bmatrix}\begin{bmatrix}\sqrt{3} & 0\\ 0 & \sqrt{1}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\sqrt{2}/2 & \sqrt{2}/2\\ \sqrt{2}/2 & -\sqrt{2}/2\end{bmatrix}.$$

Da bismo pokazali na koji način SVD algoritam može otkriti važne informacije o strukturi matrice, napisat ćemo dva teorema.

Teorem 1. *Neka je SVD matrice A zadana jednadžbom (8), neka je $r=rang(A)$ i neka je*

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq \sigma_{r+1} = ... = \sigma_n = 0 .$$

Tada vrijedi svojstvo dekompozicije:

$$A = \sum_{i=1}^{r} u_i \sigma_i v_i^T . \tag{9}$$

Svojstvo dekompozicije, koje služi kao osnova za redukciju i kompresiju podataka u mnogo aplikacija, opisuje matricu kao sumu r matrica ranga jedan, i to po padajućoj važnosti, kao što pokazuju singularne vrijednosti. Sada možemo napisati teorem o dekompoziciji odbacivanjem:

Teorem 2. [Eckart i Young] *Neka je SVD matrice A zadana jednadžbom (8) i neka je $r=rang(A) \leq p=min(m,n)$. Tada je*

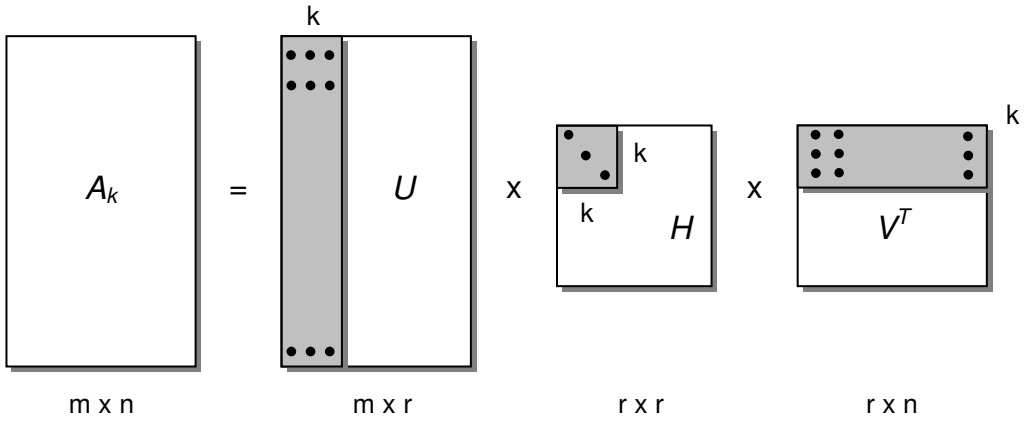
$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T , \quad \text{uz k} < \text{r} \tag{10}$$

i vrijedi

$$\min_{r(B)=k} \left\| A - B \right\|_F^2 = \left\| A - A_k \right\|_F^2 = \sigma_{k+1}^2 + ... + \sigma_p^2 . \tag{11}$$

Ovaj važan rezultat, koji pokazuje da je A_k najbolja aproksimacija ranga k matrice A (u smislu najmanje kvadratne pogreške), služi kao osnova za redukciju dimenzionalnosti.

Nakon što se izračuna SVD matrice, svaki stupac u riječ-dokument matrici (tj. dokument kojeg taj stupac predstavlja) može se projicirati na prvih k stupaca matrice U. Matematički, ta projekcija stvara k-dimenzionalni potprostor koji najbolje opisuje skup dokumenata. Projekcija po stupcima (ili projekcija dokumenata) matrice riječ-dokument metoda je koja predstavlja svaki dokument s k različitih koncepata. Drugim riječima, skup dokumenata se mapira u k-dimenzionalni prostor u kojem je jedna dimenzija rezervirana za svaki koncept. Slično tome, svaki red (ili riječ) riječ-dokument matrice može se projicirati na prvih k stupaca matrice V.



Slika 3. Redukcija dimenzionalnosti matrice riječ-dokument SVD algoritmom.

3.2.2. Primjena

Problemi s velikim i rijetko popunjenim matricama obično se pojavljuju pri modeliranju problema iz stvarnog života. Korištenje SVD algoritma može pomoći svima koji se bave područjima analize teksta i dohvata informacija. Osim kod klasifikacije i grupiranja dokumenata, ovakav se pristup rabi i kod postupka indeksiranja.

Spomenuti pristup obradi tekstualnih dokumenata uklanja neke osnovne probleme s kojima se susreću sustavi za dohvat i obradu tekstualnih podataka ako rade na principu uspoređivanja riječi u upitima korisnika s riječima u dokumentima. To su najčešće sustavi za indeksiranje. Problem je što korisnici zadaju upite koji se temelje na bazi koncepta ili značenja dokumenta. Postoji mnogo načina da se izrazi zadani koncept (npr. preko sinonima) tako da se zadane riječi u upitu neće moći pronaći u relevantnom dokumentu. Također, mnogo riječi u prirodnom jeziku ima višestruka značenja (polisemija) tako da će riječi u upitu odgovarati riječima u dokumentima koji nisu konceptualno vezani za upit. Korištenje SVD-a eliminira takve probleme jer sada se za jednu vektorsku dimenziju u reduciranoj matrici riječ-dokument veže više riječi koje opisuju jedan koncept.

3.2.3. Algoritmi

Postoji nekoliko standardnih algoritama koji se koriste za izračunavanje SVD-a gusto popunjene matrice kao npr. metoda Golub-Kahan-Reinsch ili Jakobijeve metode. Takve metode nisu optimalne za velike i rijetko popunjene matrice. Ove metode primjenjuju ortogonalne transformacije direktno na rijetko popunjenu matricu i na taj način troše jako puno memorijskog prostora. Drugi je nedostatak ovih metoda za izračunavanje SVD-a taj što će one izračunati sve singularne trojke matrice,

a to je nepotrebno jer obično se traži samo nekoliko najznačajnijih singularnih trojki. Na taj se način bez potrebe troše vrijeme i resursi računala.

Iterativne metode koje se koriste za izračunavanje SVD-a, a koje nemaju prije navedene nedostatke, jesu:

- Subspace Iteration
- Trace Minimization Method
- Single-Vector Lanczos Method
- Block Lanczos Method

U praksi se najčešće rabe posljednje dvije metode.

3.3. Kombinacija metoda

Korisna stvar u metodi uzimanja riječi s najvećim težinama jest što se može koristiti zajedno sa SVD metodom. Naime, SVD je i računski i memorijski zahtjevna metoda. Ako je memorijski prostor na računalu ograničen, moguće je najprije iz matrice riječ-dokument, koja može imati npr. 20 000 riječi, odabrati 2 000 riječi s najvećim težinama, a zatim takvu matricu proslijediti SVD algoritmu koji će je smanjiti na npr. 50 dimenzija. Na taj se način može uštedjeti na brzini izvođenja koraka redukcije dimenzionalnosti bez loših utjecaja na točnost dobivenih rezultata.

4. Grupiranje dokumenata

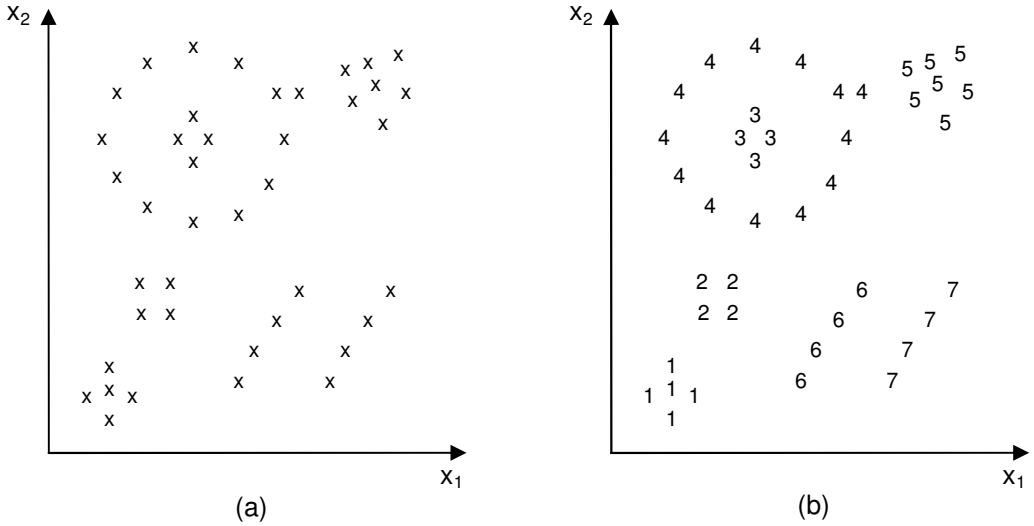
U ovom će poglavlju biti riječ o postupku koji čini posljednju fazu dubinske analize teksta tj. grupiranje (engl. *clustering*). To je nenadgledana (engl. *unsupervised*) procedura koja koristi dokumente (ili općenito uzorke) kojima nije unaprijed određena kategorija. Vidjet ćemo što se može zaključiti iz skupa dokumenata za koje ne znamo kojim kategorijama pripadaju.

4.1. Motivacija

Mogli bismo se i zapitati zašto bismo uopće trebali rješavati takav težak problem i da li je uopće moguće išta naučiti iz nekategoriziranih dokumenata. Postoji nekoliko razloga za interes za takve nenadgledane procedure. Prvo, skupljanje i ručno kategoriziranje velikog skupa dokumenata može biti izuzetno skupo – npr. snimanje govora je jednostavno, no pretvaranje govora u tekst i zatim označavanje značenja svake riječi, skupa riječi i cijelog razgovora može biti vrlo težak i vremenski zahtjevan posao. Ako bismo mogli napraviti klasifikator na malom skupu označenih riječi i dokumenata i zatim ga pustiti da radi sam bez nadzora na velikome, neoznačenom skupu tada bismo uštedjeli mnogo vremena i muke. Drugo, mogli bismo krenuti i obrnutim putem: trenirati nenadzirani algoritam s velikim količinama nekategoriziranih podataka (što je računalno manje zahtjevna operacija) i zatim koristiti nadgledani algoritam za imenovanje pronađenih grupa dokumenata. Ovaj pristup može biti primjenjiv za veće analize podataka gdje se sadržaj promatranih skupova podataka ne zna unaprijed. Treće, u mnogo se slučajeva karakteristike podataka mogu polagano mijenjati s vremenom. Primjer za to mogu biti članci iz kategorije „Šport“ u dnevnim novinama – promjenom godišnjih doba mijenja se i vrsta športova koji se u to vrijeme odvijaju. Ako se takve promjene mogu pratiti klasifikatorom koji radi u nenadziranom načinu rada, tada se može mnogo dobiti na performansama sustava. Četvrto, nenadzirane metode možemo koristiti za pronalazak značajki koje će zatim biti korisne za kategorizaciju. Postoje nenadzirane metode koje obavljaju pametno pretprocesiranje ili pametni odabir značajki. Postoje nenadzirane metode koje se koriste za redukciju dimenzionalnosti. I posljednje, peto, u ranim fazama istraživanja može biti jako korisno napraviti preliminarno istraživanje podataka nad kojima će se kasnije vršiti daljnje preciznije analize i na taj način dobiti uvid u stvarnu prirodu i strukturu podataka. Otkrivanje različitih skupova i podskupova podataka – podataka koji su sličniji jedni drugima od nekih drugih podataka – ili velikih odstupanja od očekivane strukture podataka mogu u velikoj mjeri utjecati na daljnji pristup pri izradi drugog mogućeg završnog koraka dubinske analize teksta tj. klasifikatora.

4.2. Definicija grupiranja

Analiza grupa organizacija je skupa uzoraka (koji su obično predstavljeni kao vektor u višedimenzionalnom prostoru) u grupe temeljene na sličnosti. Intuitivno, uzorci unutar valjane grupe sličniji su jedni drugima nego što su to uzorci koji pripadaju drugoj grupi (slika 4). Ulazni je skup točaka prikazan na slici (a), a željene grupe na slici (b). Točke koje pripadaju istim grupama imaju postavljene iste nazive. Postoji velik skup tehnika za prikaz podataka, mjere udaljenosti (sličnosti) između njih i njihovo grupiranje.



Slika 4. Grupiranje podataka.

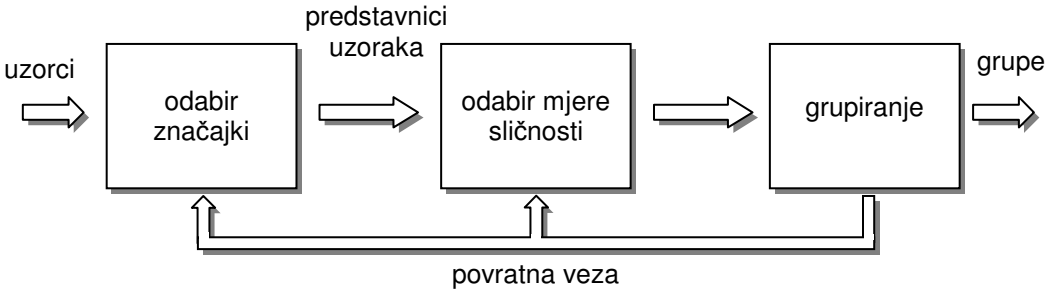
Važno je razumjeti razliku između dva posljednja koraka postupka dubinske analize podataka: grupiranja (nenadgledani postupak) i klasifikacije (nadgledani postupak ili postupak s učiteljem). Pri nadgledanoj klasifikaciji raspolažemo sa skupom klasificiranih uzoraka (dokumenata) – problem je klasificirati nove, još neklasificirane uzorke (dokumente). Klasificirani se uzorci rabe za učenje opisa grupe koja služi za opis novih uzoraka. U slučaju grupiranja problem je grupirati zadani skup neklasificiranih uzoraka u smislene grupe. Takve smislene grupe se također imenuju, ali ne u unaprijed zadana imena, već isključivo u imena dobivena iz samih podataka.

4.2.1. Koraci postupka

Općeniti postupak grupiranja sastoji se od sljedećih koraka:

1. odabir značajki uzoraka (u slučaju teksta ovaj korak uključuje pretprocesiranje i redukciju dimenzionalnosti)
2. definicija mjere sličnosti koja odgovara podacima nad kojima će se vršiti grupiranje
3. grupiranje

4. zaključivanje na temelju grupa (opcionalno)
5. evaluacija rezultata grupiranja (opcionalno)



Slika 5. Komponente postupka grupiranja.

Slika 5 prikazuje tipični redoslijed prva tri koraka uključujući povratnu vezu tako da postupak grupiranja može utjecati na odabir značajki i mjere sličnosti ako se pokaže da rezultati nisu zadovoljavajući.

Sličnost (udaljenost) uzoraka obično se mjeri *funkcijom sličnosti* (udaljenosti) na paru uzoraka. Postoje različite funkcije sličnosti za različite potrebe. Jednostavne funkcije udaljenosti (poput, najpoznatije, Euklidske udaljenosti) mogu se rabiti kao mjera različitosti među uzorcima. Ostale mjere su objašnjene u nastavku.

Korak *grupiranja* može se izvesti na više načina. Izlazne grupe mogu biti čvrste (engl. *hard partitioning*) ili neizrazite (engl. *fuzzy partitioning*). Kod čvrstih grupa svaki uzorak pripada samo jednoj izlaznoj grupi. Pri neizrazitim grupama svaki uzorak ima različiti stupanj pripadnosti svakoj od izlaznih grupa. Algoritmi za hijerarhijsko grupiranje stvaraju ugniježđenu strukturu grupa s kriterijem spajanja ili dijeljenja grupa ovisno o sličnosti. Algoritmi koji stvaraju particije početne grupe obično rade na principu (najčešće lokalne) optimizacije kriterija grupiranja. Dodatne tehnike grupiranja uključuju vjerojatnosne metode i metode grupiranja zasnovane na grafovima. One ovdje neće biti dodatno objašnjene.

Zaključivanje na temelju generiranih grupa zapravo je stvaranje jednostavnoga, malog i preciznog opisa pojedinih grupa i cijelog skupa podataka. Jednostavnost se može gledati sa strane automatske analize (tako da računalo može obaviti daljnje procesiranje rezultata što jednostavnije) ili sa strane čovjeka (tako da se rezultati predstave u obliku koji je intuitivan i čovjeku lako razumljiv).

Kako se radi *evaluacija rezultata grupiranja* kod grupiranja? Što karakterizira „dobar“ rezultat grupiranja, a što „loš“ ako znamo da nemamo nikakvih početnih kategorija za usporedbu? Svi algoritmi za grupiranje generiraju grupe podataka u koje smještaju uzorke koje su dobili kao ulazne parametre. Oni će generirati grupe bez obzira sadrže li podaci ikakve grupe ili ne. Ako podaci sadrže strukturu u obliku grupa, tada neki algoritmi mogu

izdvojiti „bolje“ grupe od nekih drugih algoritama. Evaluacija rezultata grupiranja ima prema tome nekoliko zadaća. Jedna je od njih evaluacija domene podataka prije samog postupka grupiranja – podaci koji ne sadrže grupe uopće ne trebaju biti procesirani algoritmom za grupiranje. Evaluacija grupa gleda ispravnost samih rezultata algoritma za grupiranje. Ova analiza često rabi neki od specifičnih kriterija optimalnosti koji se obično određuju – subjektivno. No, postoje i neki objektivni kriteriji po kojima se računa kvaliteta algoritma za grupiranje i ispravnost strukture grupa. Za strukturu grupe kaže se da je ispravna ako nije nastala slučajno ili kao artefakt algoritma za grupiranje.

Ne postoji tehnika grupiranja koja je univerzalno primjenjiva za otkrivanje raznih struktura u svim vrstama višedimenzionalnih podataka. Uzmimo za primjer dvodimenzionalni skup podataka na slici 4 (a). Neće sve tehnike grupiranja otkriti sve grupe s jednakom točnošću (kao na slici (b)) zato jer algoritmi za grupiranje često sadrže implicitne pretpostavke o obliku i hijerarhiji grupa koje se baziraju na korištenim mjerama sličnosti i kriterijima grupiranja.

Ljudi mogu jako dobro obavljati grupiranje podataka u jednoj ili dvije dimenzije, ali većina realnih problema uključuje grupiranje u mnogo većim dimenzijama. Ljudima je teško intuitivno predočiti podatke koji se nalaze u obliku višedimenzionalnih vektora najviše zbog toga jer takve podatke ne možemo jednostavno grafički prikazati. Također, u takvim prostorima podaci teško prate „idealne“ strukture kao što je pokazano na slici 4 (kružnice, pravokutnici, linije, elipse…). To objašnjava velik broj postojećih algoritama za grupiranje. No, svaki je od njih pogodan samo za otkrivanje određene strukture u podacima. Na istraživaču je da odabere koju.

4.3. Mjere sličnosti

Problem grupiranja definirali smo kao pronalaženje prirodnih grupa unutar skupa podataka. No, što smo zapravo mislili pod izrazom „prirodne grupe“? I u kojem su smislu uzorci u jednoj grupi sličniji od uzoraka u drugoj grupi?

Najočitija mjera sličnosti (ili različitosti) između dva uzorka njihova je međusobna udaljenost. Jedan od načina na koji bi se moglo izvesti grupiranje jest definiranje prikladne mjere i izračunavanje matrice sličnosti između svih parova uzoraka. Ako je udaljenost dobra mjera različitosti, tada bi udaljenost između uzoraka iste grupe bila značajno manja od uzoraka koji pripadaju nekoj drugoj grupi.

U prethodno navedenom sustavu za grupiranje spomenuli smo da dva uzorka pripadaju istoj grupi ako je udaljenost među njima manja od nekog praga d_0. Uočavamo da je odabir d_0 jako bitan. Ako je d_0 velik, tada će svi uzorci biti dodijeljeni jednoj grupi. Ako, pak, je d_0 jako malen tada će svaki uzorak činiti svoju grupu. Da bismo dobili „prirodne“ grupe, trebamo odrediti

d_0 takav da bude veći od tipične udaljenosti uzoraka unutar grupe, a manji od tipične udaljenosti uzoraka između grupa. To nikako nije lagan zadatak.

U nastavku će biti prikazane najčešće rabljene mjere sličnosti (različitosti).

4.3.1. Euklidska udaljenost

Najpopularnija i najviše korištena mjera udaljenosti jest Euklidska mjera

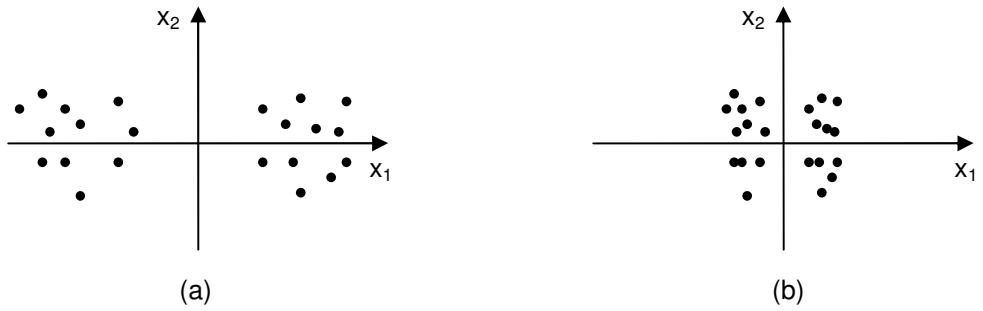
$$d_2(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \tag{12}$$

koja je poseban slučaj (p = 2) mjere Minkowskog

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^p \right)^{\frac{1}{p}} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^p . \tag{13}$$

Postavljanje parametra p na vrijednost 1 daje Manhattan mjeru (engl. *city block metric*) – sumu svih apsolutnih udaljenosti duž svake od d koordinata.

Problem pri ovakvim mjerama jest tendencija da značajka posjeduje najveće vrijednosti obično ima dominantan utjecaj nad ostalim značajkama. Rješenje je ovog problema normalizacija kontinuiranih značajki, iako s tim postupkom treba biti oprezan. Ako, na primjer, podaci pripadaju dvjema dobro odvojenim grupama (slika 6 (a)), tada normalizacija skaliranjem može smanjiti njihovu odvojenost i u tom slučaju postupak normalizacije nije poželjan (slika 6 (Slika 6b)).



Slika 6. Utjecaj normalizacije na odvojenost grupa podataka.

4.3.2. Kosinusna sličnost

Općenito gledajući, možemo i odbaciti udaljenost kao mjeru te uvesti mjeru sličnosti koja uspoređuje dva vektora $\mathbf{x}_i$ i $\mathbf{x}_j$. Ta mjera može biti bilo koja simetrična pozitivna funkcija čija je vrijednost velika kada su $\mathbf{x}_i$ i $\mathbf{x}_j$ u neku ruku „slični“. Primjer takve mjere može biti kut između dva vektora, a njihov produkt

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \tag{14}$$

može biti pripadajuća funkcija sličnosti. Ova mjera je zapravo kosinus kuta između $\mathbf{x}_i$ i $\mathbf{x}_j$ što je ta vrijednost veća (bliža jedan), to su vektori sličniji.

4.4. Kriterijske funkcije

Nakon definiranja prve važne stvari pri grupiranju tj. odabiru mjere, definirat ćemo kriterijsku funkciju koju će algoritam grupiranja optimizirati. Pretpostavimo da skup $D = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ ima n uzoraka koje želimo grupirati u c grupa $D_1, ..., D_c$. Jedan od načina da riješimo ovaj problem jest da definiramo kriterijsku funkciju koja mjeri kvalitetu grupiranja podataka. Tada se problem grupiranja svodi na traženje ekstrema kriterijske funkcije. U nastavku će biti objašnjena najpoznatija takva kriterijska funkcija.

4.4.1. Kriterij sume najmanje kvadratne pogreške

Najjednostavnija i najšire korištena kriterijska funkcija u postupku grupiranja jest suma najmanje kvadratne pogreške. Neka je n_i broj uzoraka u D_i i neka je $\mathbf{m_i}$ srednja vrijednost uzoraka

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}. \tag{15}$$

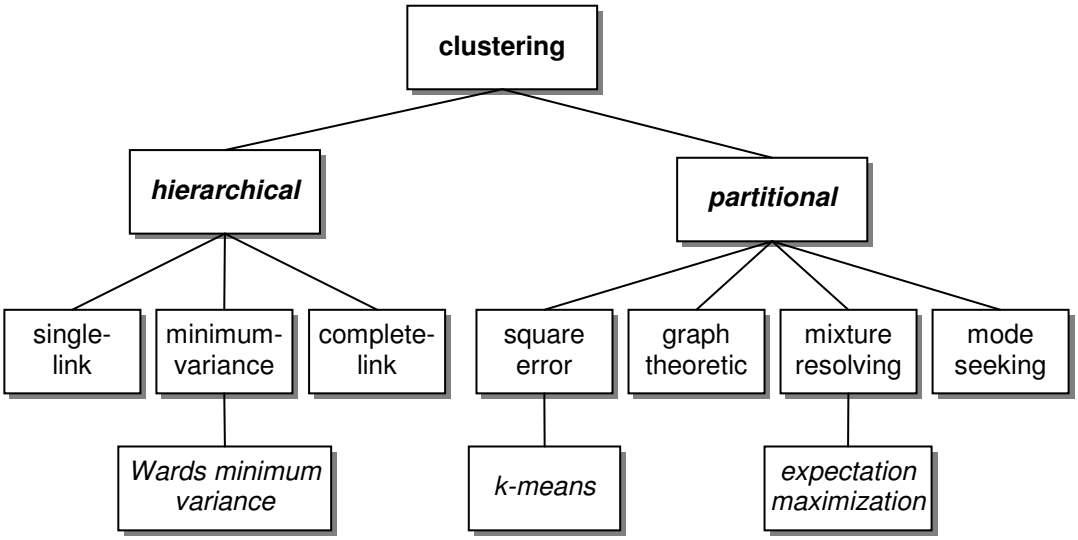
Tada je suma najmanje kvadratne pogreške jednaka

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \|\mathbf{x} - \mathbf{m}_i\|^2 . \tag{16}$$

Vrijednost J_e ovisi o tome kako su uzorci grupirani u grupe i broju grupa. Optimalna je podjela ona koja minimizira J_e. Grupiranje ovog tipa često se zove grupiranje na temelju najmanje promjene (engl. *minimum variance partition*).

4.5. Algoritmi za grupiranje

Različiti pristupi grupiranju podataka mogu se prikazati pomoću hijerarhije prikazane na slici 7. Na vrhu hijerarhije napravljena je podjela između hijerarhijskog i particijskog pristupa. Hijerarhijske metode generiraju ugniježđeni skup grupa, a particijske metode generiraju samo jedan skup grupa kao što prikazuje slika 8.



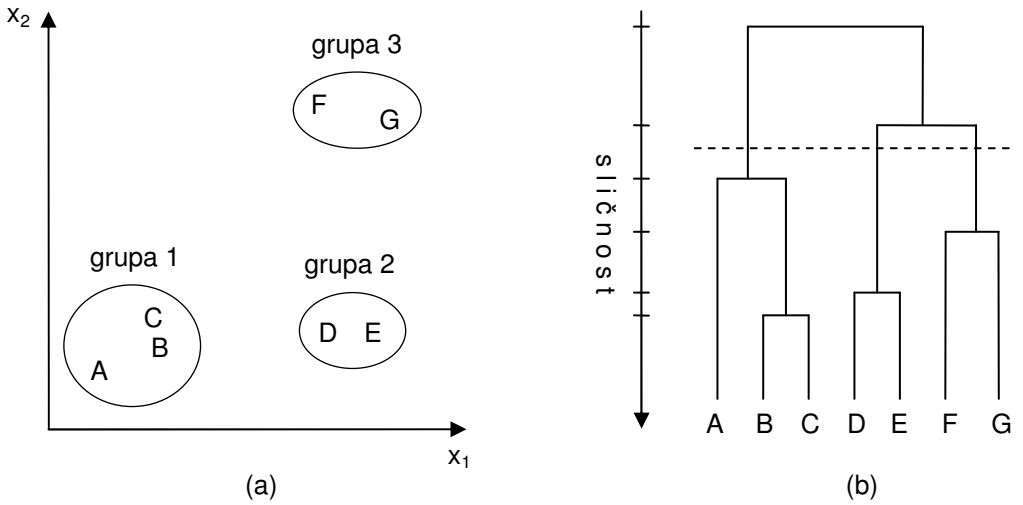
Slika 7. Podjela algoritama za grupiranje.

Postoji još jedna važna podjela algoritama za grupiranje: na tzv. *agglomerative* i *divisive*. Ta je podjela vezana uz samu strukturu i način rada algoritma. Algoritmi s *agglomerative* pristupom počinju tako da svaki uzorak stave u zasebnu grupu, a zatim postupno spajaju grupe sve dok se ne zadovolji uvjet zaustavljanja. Algoritmi s *divisive* pristupom, pak, počinju tako da sve uzorke grupiraju u jednu grupu, a zatim tu grupu dijele sve dok se ne zadovolji uvjet zaustavljanja. Ova podjela jednako vrijedi i za hijerarhijske i za particijske algoritme.

4.5.1. Hijerarhijsko grupiranje

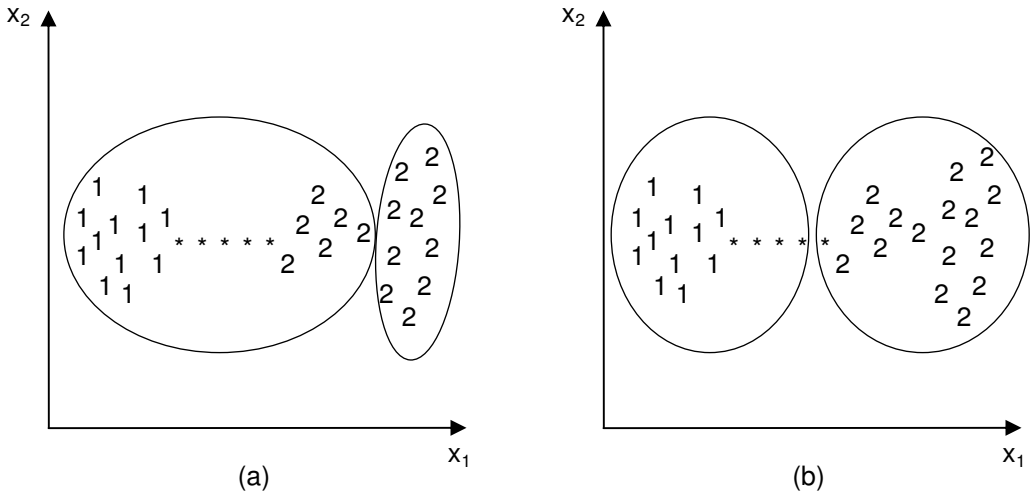
Rezultat grupiranja hijerarhijskog algoritma prikazan je na dvodimenzionalnom skupu podataka na slici 8 (a). Ona prikazuje sedam uzoraka imenovanih A, B, C, D, E, F i G raspoređenih u tri grupe. Hijerarhijski algoritam grupiranja kao rezultat daje *dendogram* koji predstavlja ugniježđenu strukturu grupa i nivoe sličnosti na kojima se mijenja grupiranje. Dendogram koji odgovara tim točkama prikazan je na slici 8 (b). Dendogram se može prelomiti na različitim nivoima i tako se može dobiti drugačije grupiranje podataka.

Većina hijerarhijskih algoritama grupiranja jesu varijante *single-link*, *complete-link* i *minimum-variance* metoda. Od tih su metoda prve dvije najpopularnije. Treću metodu u svome radu koristi SAS *Text Miner*.



Slika 8. Particijsko (a) i hijerarhijsko (b) grupiranje

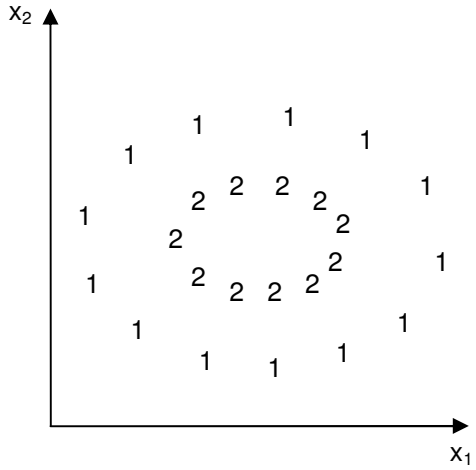
Single-link i *complete-link* metode razlikuju se u načinu na koji određuju razliku u sličnosti između dvije grupe. U *single-link* metodi udaljenost između dvije grupe *minimum* je udaljenosti između svih parova uzoraka koji se nalaze u tim grupama (jedan uzorak iz prve grupe i jedan uzorak iz druge grupe). U *complete-link* metodi udaljenost je između grupa *maksimum* svih udaljenosti između parova uzoraka iz obiju grupa.



Slika 9. Usporedba *single-link* (a) i *complete-link* (b) metoda grupiranja.

U svakom slučaju, dvije se grupe spajaju u jednu veću po kriteriju minimalne udaljenosti. *Complete-link* metoda stvara čvršće vezane i

kompaktnije grupe. Nasuprot tome, s*ingle-link* metoda pati od efekta ulančavanja tj. ima tendenciju stvaranja rastegnutih i produljenih grupa. Na slikama 9 (a) i (b)postoje dvije grupe odvojene „mostom“ kojeg čine uzorci koji predstavljaju šum. *Single-link* algoritam stvara grupe prikazane na slici (a), a *complete-link* stvara grupe prikazane na slici (b). Grupe dobivene *complete-link* algoritmom kompaktnije su nego one dobivene sa s*ingle-link* algoritmom. Grupa označena brojem 1 korištenjem s*ingle-link* algoritma produžena je zbog uzoraka označenih znakom „*“ koji predstavljaju šum. No, upravo zbog toga s*ingle-link* algoritam može odrediti koncentrične grupe prikazane na slici 10, što se s *complete-link* algoritmom ne može. Ukupno gledajući, *complete-link* algoritam daje bolje rezultate i gradi korisnije i logičnije hijerarhije od s*ingle-link* algoritma.



Slika 10. Dvije koncentrične grupe.

U nastavku će detaljnije biti objašnjen jedan algoritam za hijerarhijsko grupiranje koji je implementiran u SAS *Text Miner*-u.

4.5.1.1. Ward's minimum-variance method

Ova metoda radi na *agglomerative* principu. Spajaju se oni parovi grupa iz prethodne generacije čije spajanje minimizira povećanje ukupne kvadratne pogreške unutar grupe. Za mjeru se sličnosti uzima Euklidska mjera ili modificirana Euklidska mjera

$$d_w(\mathbf{x}_i, \mathbf{x}_j) = \frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}{\frac{1}{N_i} + \frac{1}{N_j}}. \qquad (17)$$

Wardova metoda teži spajanju grupa s manjim brojem uzoraka i ima pristranost prema stvaranju grupa s približno istim brojem uzoraka. Stvara simetričnu i preciznu hijerarhiju grupa i dobra je za otkrivanje same strukture grupa. Nedostaci su joj to što je osjetljiva na šum tj. na članove grupe koji su jako udaljeni od ostataka grupe (engl. *outliers*). Također, daje loše rezultate

pri grupama koje imaju produženi oblik (engl. *elongated*) što ga čini sličnim *complete-link* postupku. Vremenska složenost mu je $O(n^2)$, a prostorna $O(n)$.

4.5.2. Particijsko grupiranje

Za razliku od hijerarhijskih metoda, algoritmi za particijsko grupiranje kao rezultat daju jednu podjelu uzoraka u grupe. Particijske metode imaju prednosti pri velikim skupovima podataka jer bi korištenje hijerarhijskih metoda i konstrukcija dendograma bilo računski neizvedivo. Najveći problem particijskoga grupiranja odabir je broja željenih grupa. Particijske metode obično stvaraju grupe optimizirarajući kriterijsku funkciju definiranu lokalno (na podskupu uzoraka) ili globalno (na svim uzorcima). Kombinatorno pretraživanje skupa svih mogućih podjela za optimalnu vrijednost kriterijske funkcije računski je nemoguće izvesti. Zbog toga se u praksi algoritmi pokreću više puta s različitim početnim stanjima, a najbolja dobivena konfiguracija grupa koristi se kao rezultat grupiranja.

4.5.2.1. K–means algoritam

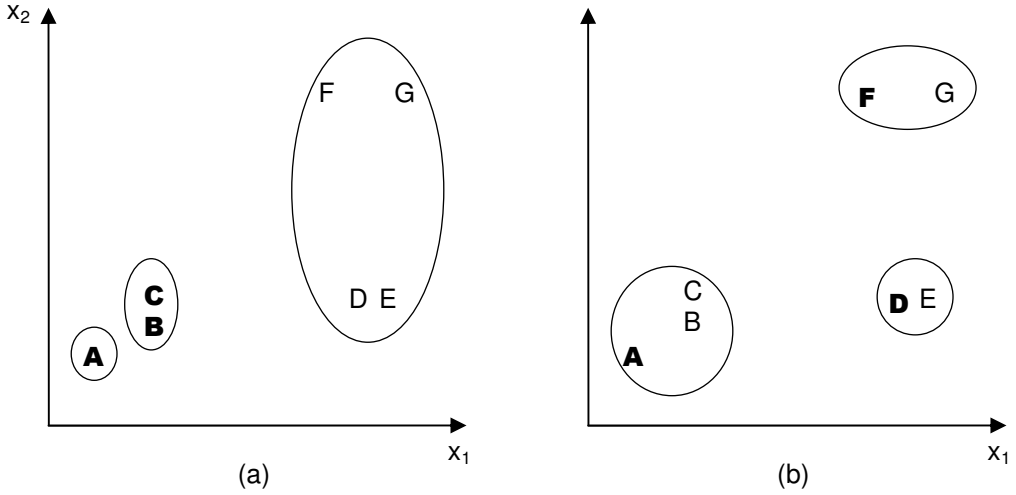
K–means najjednostavniji je i najčešće korišteni algoritam koji rabi kriterij najmanje kvadratne pogreške (objašnjen u poglavlju 4.4.1). Počinje sa slučajnom inicijalnom podjelom uzoraka u grupe i zatim smješta uzorke u druge grupe ovisno o uzorku i središtu grupe sve dok se ne zadovolji kriterij konvergencije (tj. sve dok se niti jedan uzorak ne dodijeli nekoj drugoj grupi ili se kvadratna pogreška značajno ne smanji nakon određenog broja iteracija).

Ovaj je algoritam popularan zato jer ga je lagano implementirati i zato jer mu je vremenska ovisnost $O(n)$, gdje je n broj uzoraka. Glavni je problem u ovom algoritmu osjetljivost na odabir početne podjele tako da algoritam može konvergirati u lokalni minimum kriterijske funkcije ako početna podjela nije odgovarajuća. Na slici 11 (a) je prikazan rezultat grupiranja ako su za početna središta grupa izabrani uzorci A, B i C, a na slici 11 (b) je prikazan ispravan rezultat grupiranja koji se dobije ako se za središta grupa uzmu uzorci A, D i F.

Algoritam je sljedeći:

1. Izaberi k središta grupa tako da se izabere k slučajno odabranih uzoraka ili k slučajno odabranih točaka unutar prostora uzoraka.
2. Svakoj grupi dodijeli najbliže uzorke.
3. Ponovno izračunaj središta svih grupa koristeći uzorke koji se trenutno nalaze u grupama.
4. Ako nije zadovoljen kriterij konvergencije idi na korak 2. Tipični uvjeti konvergencije su: nema ponovne dodjele uzoraka u druge grupe (ili je broj ponovno dodijeljenih uzoraka minimalan) ili se kvadratna pogreška minimalno smanjila.

Postoji i varijacija algoritma koja dopušta podjelu i spajanje rezultirajućih grupa. Poznati takav algoritam je ISO-DATA.



Slika 11. Ovisnost rezultata grupiranja *k-means* algoritmom o izboru početnih središta grupa.

4.5.2.2. Expectation Maximization (EM) algoritam

Ovaj algoritam spada u grupu *mixture-resolving* algoritama. Pretpostavka u ovoj metodi jest da su svi uzorci koji se grupiraju raspoređeni po jednoj od nekoliko razdioba, a cilj je pronaći parametre svake razdiobe i po mogućnosti njihov broj. U većini se slučajeva uzima da su razdiobe Gaussove. EM algoritam koristi se za pronalaženje njihovih parametara.

Funkcija gustoće vjerojatnosti EM algoritma u točki x evaluira se sa

$$p(x) = \sum_{h=1}^{k} w_h f_h(x \mid \mu_h, \Sigma_h) \tag{18}$$

gdje je k broj grupa, f_h funkcija gustoće grupe h, a w_h je postotak podataka koji pripadaju primarnoj grupi h. Svaka se grupa modelira s d-dimenzionalnom Gaussovom razdiobom

$$f_h(x \mid \mu_h, \Sigma_h) = \frac{e^{-\frac{1}{2}(x-\mu_h)^T (\Sigma_h)^{-1} (x-\mu_h)}}{\sqrt{(2\pi)^d |\Sigma_h|}} \tag{19}$$

gdje je d broj varijabli vektora, μ_h je srednji vektor, a Σ_h je matrica devijacije za svaku grupu h. Za svaki uzorak x u skupu uzoraka h u svakoj iteraciji j parametri EM algoritma predviđaju se po sljedećim koracima:

1. (Expectation korak.) Izračunaj vjerojatnost pripadnosti uzorka x svakoj grupi h, $h=1,\ldots,k$ formulom

$$w_h^j(x) = \frac{w_h^j f_h(x \mid \mu_h^j, \Sigma_h^j)}{\sum_{i=1}^{k} w_i^j f_i(x \mid \mu_i^j, \Sigma_i^j)}. \quad (20)$$

2. (Maximization korak.) Promijeni parametre modela za svaku grupu $h=1,\ldots,k$ formulom

$$w_h^{j+1} = \sum_x w_h^j(x), \quad (21)$$

$$\mu_h^{j+1} = \frac{\sum_x w_h^j(x) x}{\sum_x w_h^j(x)} \quad \Sigma_h^{j+1} = \frac{\sum_x w_h^j(x)(x-\mu_h^{j+1})(x-\mu_h^{j+1})^T}{\sum_x w_h^j(x)}. \quad (22)$$

3. Ponavljaj korake 1. i 2. sve dok

$$\left| L(\Phi^j) - L(\Phi^{j+1}) \right| \le \varepsilon, \varepsilon > 0, \quad (23)$$

$$L(\Phi) = \sum_x \log\left[\sum_{h=1}^{k} w_h f_h(x \mid \mu_h, \Sigma_h) \right]. \quad (24)$$

EM algoritam također je partitivna metoda, ali za razliku od K-means algoritma on dopušta bilo kakav oblik i veličinu grupa.

4.6. Evaluacija rezultata grupiranja

Evaluacija rezultata grupiranja nije lagana zadaća. Ako već unaprijed znamo kategorije, a želimo vidjeti kvalitetu grupiranja nekog algoritma, tada možemo naići na niz problema. Prvo, zadane kategorije mogu biti subjektivno određene i samim time umjetne i „neprirodne“ za algoritam koji testiramo. Drugo, nakon što dobijemo rezultate grupiranja, moramo ručno pregledati sve grupe i zatim svaku grupu povezati s određenom kategorijom. To je također subjektivni postupak.

Povezivanjem grupa i kategorija zapravo dobivamo matricu koja pokazuje odnos unaprijed zadanih kategorija i dobivenih grupa ili engl. *confusion matrix*. Pomoću takve matrice određujemo kvalitetu algoritma za grupiranje na nekoliko načina.

4.6.1. Čistoća grupa (engl. *purity*)

Pretpostavimo da imamo g kategorija s imenima $\{1, 2, \ldots, g\}$ i k grupa $\{X_1, X_2, \ldots, X_k\}$. S n_j^i označit ćemo broj uzoraka koji pripadaju kategoriji i, a

nalaze se u grupi X_j, a s n_j označit ćemo broj uzoraka u grupi X_j. Tada se čistoća grupa definira s

$$\frac{1}{d}\sum_{j=1}^{k}\max_i\left\{n_j^{\,i}\right\} \tag{25}$$

gdje i varira po svim imenima kategorija, a d je broj svih dokumenata. Čistoća je, prema tome, točnost klasifikacije uz pretpostavku da su svi pripadnici grupe članovi dominantne kategorije u toj grupi.

5. Alati za dubinsku analizu teksta

Postoji mnogo alata koji imaju implementirane sve korake za dubinsku analizu teksta. Jedan je od njih i SAS *Text Miner* koji će biti opisan u ovom poglavlju.

SAS je jedna od vodećih tvrtki u izradi BI (engl. *business intelligence*) programa i servisa. SAS rješenja rabe se na više od 40 000 mjesta u svijetu uključujući preko 90 % tvrtki koje se nalaze na Fortune listi 500 najuspješnijih tvrtki u svijetu. Uključivanjem alata za dubinsku analizu teksta *Text Miner* u već postojeći *Enterprise Miner*, SAS je postao prvi proizvođač BI programa koji pruža dubinsku analizu strukturiranih i nestrukturiranih podataka.

U nastavku će biti opisani svi koraci dubinske analize teksta korištenjem SAS *Text Miner* programa.

5.1. Priprema podataka za unos u SAS

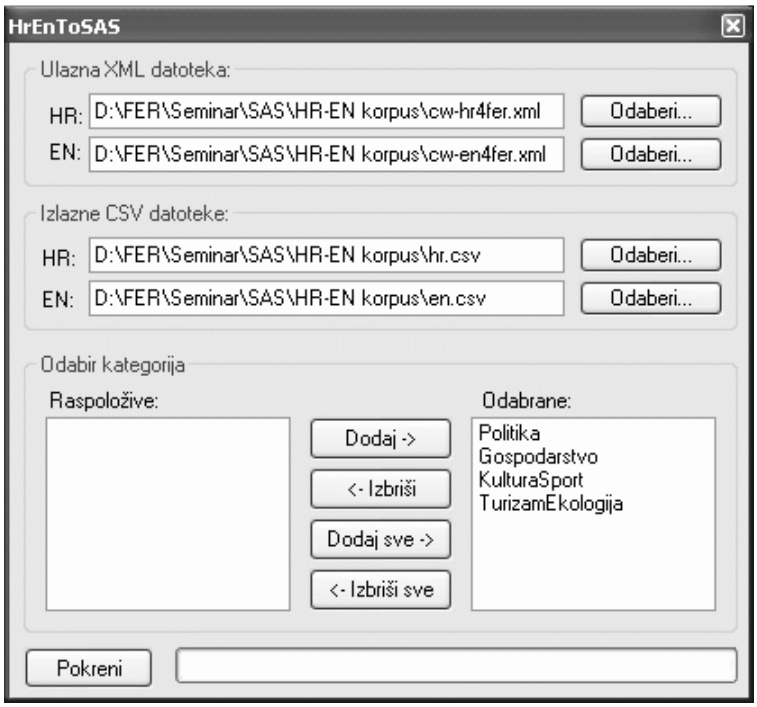
SAS *Text Miner* ima ugrađenu podršku za čitanje velikog broja različitih tipova tekstualnih dokumenata. Popis svih tipova dokumenata prikazan je u tablici (Tablica 1). Uz pomoć raznih dodatnih biblioteka moguće je ovaj popis još proširiti.

Format dokumenta	Verzija	Format dokumenta	Verzija
Adobe PDF	1.1 do 4.0	Lotus Word pro	96, 97, R9
Applix Asterix	Applix Asterix	Microsoft Excel	3, 4, 5, 97 do 2000
Applix Spread Sheet	10	Microsoft PowerPoint	4.0, 95, 97
ASCII tekst	ASCII tekst	Microsoft RTF	Sve
Corel Presentations	7.0, 8.0	Microsoft Word	1.x, 2.0, 6.0, 7.0, 8.0, 95 do 2000
Corel Quattro	7.0, 8.0	Microsoft Word for DOS	2.2 do 5.0
DCA-RTF	sc23-0758-1	Microsoft Works	1.0, 2.0, 3.0, 4.0
Framemaker Interchange Format (MIF)	5.5	Microsoft Word for MAC	4.x, 5.x, 6.x, 98
HTML	Sve	WordPerfect for DOS	5.0, 6.0
IBM DisplayWrite	1.0, 1.1	WordPerfect for MAC	2.2, 3.0
Lotus 1-2-3	2, 3, 4, 96, 97, R9	WordPerfect for Windows	7.0
Lotus AMI pro	2.0, 3.0	XYWrite	4.12

Tablica 1. Podržani formati dokumenata u SAS Text Miner-u.

Baze dokumenata koje su se rabile u eksperimentima, nalazile su se spremljene u XML datotekama. Ako je riječ o datotekama u kojima su spremljeni dokumenti na hrvatskom jeziku, tada je tekst tih dokumenata već bio morfološki normalizirani i iz njega su bile izbačene stop-riječi. Tekst u datotekama na engleskom jeziku bio je u originalnom obliku zato jer SAS *Text Miner* ima mogućnost normalizacije i uklanjanja stop-riječi za engleski jezik. Budući da na raspolaganju nije bila biblioteka koja omogućava unos XML datoteka, napravljeni su dodatni programi koji vrše konverziju dokumenata iz XML formata u CSV format[1] (engl. *Coma Separated Values*). Razlog zašto se nisu koristile samo npr. obične tekstualne datoteke jest taj što je uz tekst dokumenata bila spremljena i kategorija kojoj dokument pripada. Program koji radi konverziju u CSV format prikazan je na slici 12.

Kod programa trebamo odabrati XML datoteke s dokumentima, zatim ime CSV datoteka i kategorije koje želimo prebaciti u CSV datoteke. To je napravljeno zato da omogući obavljanje eksperimenata po želji – samo s nekim ili svim kategorijama. Nakon što kliknemo na gumb *Pokreni*, napravit ćemo konverziju.



Slika 12. Program za konverziju iz XML u CSV format.

Primjer jedne takve CSV datoteke prikazan je na slici 13. Prva linija datoteke ukazuje na njezinu strukturu. Prema prvoj liniji zaključujemo da se svaki zapis u datoteci mora sastojati od stupca `Kat` u kojem piše ime kategorije u tekstualnom obliku i zatim od četiri stupca `Politika`,

[1] tekstualna datoteka u obliku tablice u kojoj se svaki novi zapis nalazi u novom retku, a stupci su odvojeni zarezima

`Gospodarstvo`, `TurizamEkologija`, `KulturaSport` koji sadrže pohranjenu vrijednost o pripadnosti pojedinoj kategoriji u binarnom obliku (0 – ne pripada, 1 – pripada kategoriji). Posljednji stupac ima naziv `Tekst` i u njemu se nalazi cijeli tekst dokumenta.

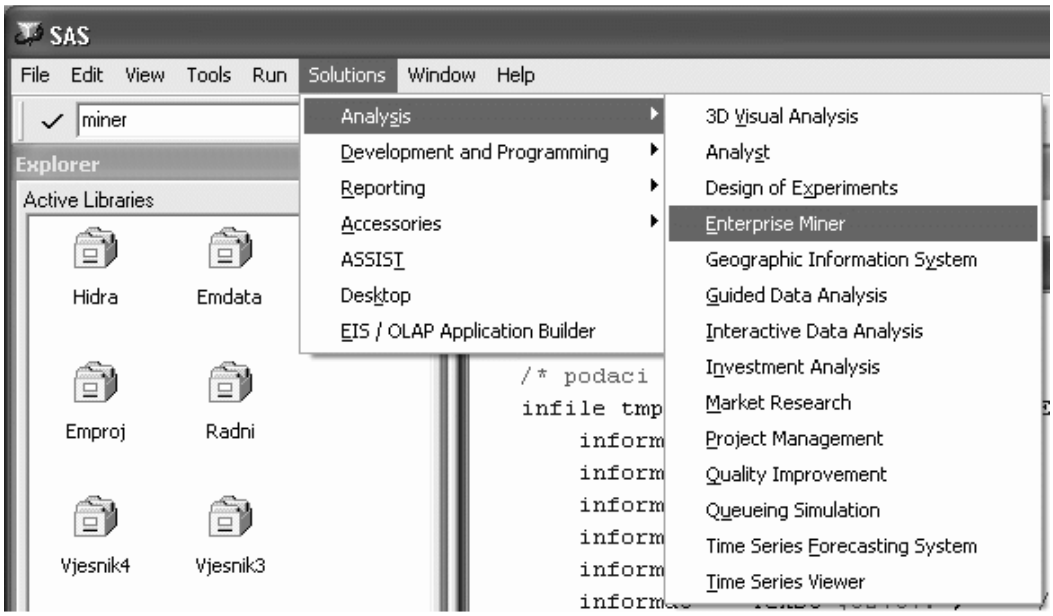
```
Kat, Politika, Gospodarstvo, TurizamEkologija, KulturaSport, Tekst
TurizamEkologija, 0, 0, 1, 0, prirodni fenomen vranskog jezera…
Politika, 1, 0, 0, 0, ukinuta uredba najmu stanova vlada odustala…
Politika, 1, 0, 0, 0, novi ministri nove osobe vladi ministar…
Politika, 1, 0, 0, 0, sastanak hrvatskog svjetskog kongresa…
Gospodarstvo, 0, 1, 0, 0, bechtel započeti gradnju ceste bregana…
KulturaSport, 0, 0, 0, 1, promocija hrvatske umjetnosti svijetu…
KulturaSport, 0, 0, 0, 1, pobjeda hrvatskih rukometaša slovenskim…
…
…
```

Slika 13. Primjer CSV datoteke pogodne za unos u SAS Text Miner.

Time je završen postupak pripreme podataka za unos u SAS *Text Miner*. Sada slijedi pokretanje SAS sustava i učitavanje ovako pripremljenih podataka.

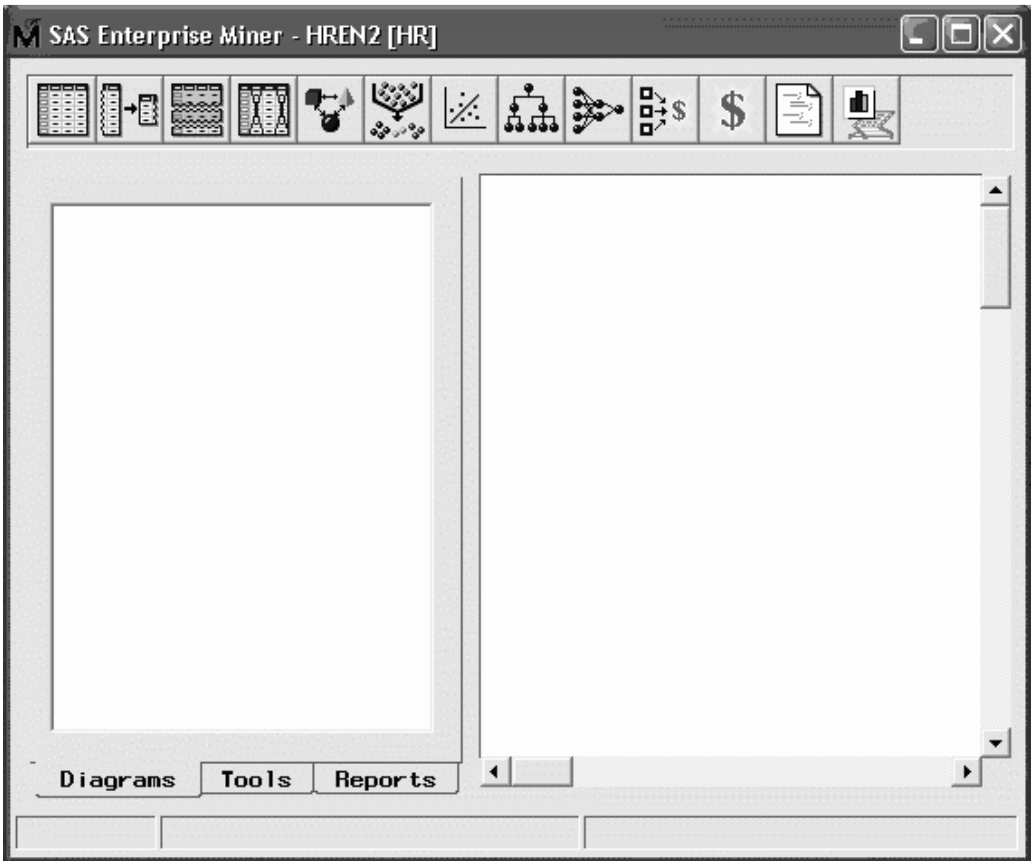
5.2. Unos podataka u SAS

Na slici 14 vidi se izgled *Solutions* izbornika u SAS sustavu. U tom se izborniku nalazi popis svih raspoloživih modula u SAS-u.



Slika 14. *Solutions* izbornik SAS sustava.

Vidi se velik broj dostupnih rješenja. Nama je zanimljiv *Analysis* dio u kojem se nalazi Enterprise Miner. Dio Enterprise Miner-a čini i SAS *Text Miner* koji se rabi za dubinsku analizu teksta. Izgled glavnog prozora *Enterprise Miner* programa prikazan je na slici 15.



Slika 15. Glavni prozor Enterprise Miner-a.

U traci s alatima nalaze se najčešće korišteni alati. Odabirom *Tools* prozora prikazuju se svi raspoloživi alati. Alat od kojeg kreće svaki dijagram u Enterprise Miner-u zove se *Input Data Source* i prvi je u traci s alatima. Taj alat zapravo predstavlja mjesto odakle će daljnji alati uzimati podatke. U njega punimo podatke iz nekog od prije stvorenih skupova podataka (engl. *dataset*). Da bismo u *Input Data Source* učitali naše dokumente, najprije moramo CSV datoteku uvesti u SAS u obliku tablice, a zatim tu tablicu povezati sa *Input Data Source* alatom. Unos podataka radi se sa programom prikazanim na slici 16. Taj se program treba upisati u *Code* prozor SAS sustava i zatim izvršiti.

Naredba `FILENAME` prima nekoliko parametara. Prvi je parametar ime (u ovom slučaju `tmpDat`) pod kojem će se u SAS-u referencirati datoteka koja je sa svojom punom stazom postavljena kao drugi parametar naredbe. Ostali parametri su `ENCODING` za koji se stavlja neki od kôdova. Za nas su zanimljivi kôdovi `latin2` (predstavlja `ISO-8859-2`) i `wlatin2` (predstavlja `Windows-1250`). Veličina pojedinog članka kojeg SAS *Text Miner* može obraditi iznosi najviše 32 767 znakova, zato se parametar `LRECL` postavlja na tu vrijednost.

Ucitavanje CW baze u DATASET v2.sas *

```
/*
    Naredba FILENAME stvara nesto slicno file handleru u c-u.
*/
FILENAME tmpDat 'D:\FER\Seminar\SAS\HR-EN korpus\hr.csv'
    ENCODING=wlatin2 RECFM=V LRECL=32767;

/*
    Naredba DATA stvara novi DATASET
*/
data HrEnKorp.HR;

/* podaci se citaju iz file handlera */
infile tmpDat delimiter = ',' MISSOVER DSD firstobs=2;
    informat    Kategorija $20. ;
    informat    Politika $1. ;
    informat    Gospodarstvo $1. ;
    informat    TurizamEkologija $1. ;
    informat    KulturaSport $1. ;
    informat    Tekst $32767. ;

    format      Kategorija $20. ;
    format      Politika $1. ;
    format      Gospodarstvo $1. ;
    format      TurizamEkologija $1. ;
    format      KulturaSport $1. ;
    format      Tekst $32767. ;

    input   Kategorija $            /* nazivi stupaca */
            Politika $
            Gospodarstvo$
            TurizamEkologija $
            KulturaSport $
            Tekst $
    ;
run;
```

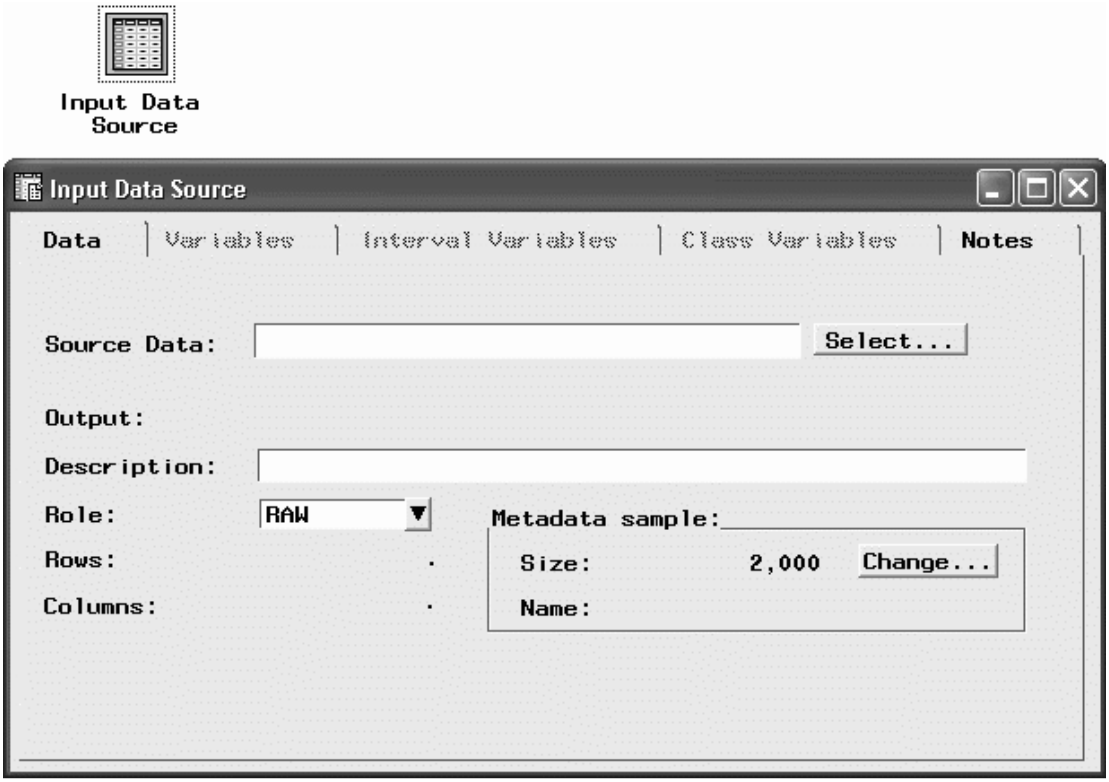
Slika 16. Program za učitavanje podataka u SAS.

Naredba **data** stvara novu tablicu u nekoj od biblioteka. U slučaju na slici biblioteka se zove `HrEnKorp`, a nova tablica koja će se u njoj stvoriti naziva se `HR`. Ako ne želite koristiti nikakvu posebnu biblioteku tada možete upisati `Work`. To je biblioteka koja uvijek postoji i koju SAS također koristi za svoj rad. Naredba `infile` govori iz koje će se datoteke napuniti novostvorena tablica. Kao parametar joj predajemo ime reference na datoteku koju smo prije otvorili (u ovom slučaju to je `tmpDat`), zatim postavimo da je `delimiter` znak zareza i da je prvi redak sa podacima drugi redak u datoteci (prvi redak je opis stupaca kao što se vidi na slici 13. Naredbe `format` i `informat` opisuju izgled tablice tj. njezine stupce zajedno sa dužinom podataka koje će se u njih spremati. Naredba `input` učitava podatke iz datoteke.

5.3. Izrada dijagrama u Enterprise Miner-u

Proces dubinske analize podataka je u Enterprise Miner-u predstavljen kao proces odabira, istraživanja, modificiranja, modeliranja podataka i ocjenjivanja rezultata (engl. *Select, Explore, Modify, Model, Assess* ili SEMMA). SEMMA proces zasniva se na dijagramu tijeka procesa (engl. *process flow diagram*) koji se može modificirati i spremati. Grafičko sučelje je dizajnirano tako da ga svatko imalo upoznat sa analizom podataka može jednostavno koristiti. Enterprise Miner sadrži skup sofisticiranih alata za analizu. Statistički alati uključuju grupiranje, samoorganizirajuće mreže (Kohonen), stabla odluke i neuronske mreže. Više o dostupnim alatima može se pročitati u sustavu pomoći.

Alati (čvorovi, engl. *nodes*) se koriste tako da se drag-and-drop tehnikom dovuku na radnu površinu, a zatim se međusobno spajaju. Prvi alat kojeg koristimo je *Input Data Source*.

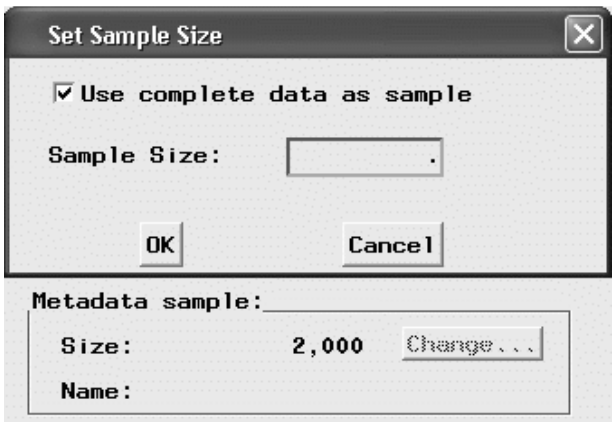


Slika 17. Input Data Source alat sa pripadajućim okvirom za podešavanje.

5.3.1. Input Data Source Node

Input Data Source je alat koji služi kao prethodnik za sve ostale alate. On sadrži podatke koje drugi alati obrađuju. Na slici 17 je prikazan *Input Data Source* alat u dijagramu zajedno sa okvirom za dijalog unutar kojeg se radi podešavanje tog alata.

Prvo što možemo učiniti je odabrati koliki broj podataka želimo analizirati. Ako se klikne na gumb *Change* koji se nalazi unutar *Metadata sample* okvira tada se može upisati broj uzoraka koje želimo analizirati ili možemo odabrati sve uzorke. Kod grupiranja obično koristimo sve uzorke zato odabiremo *Use complete data sample* (slika 18).

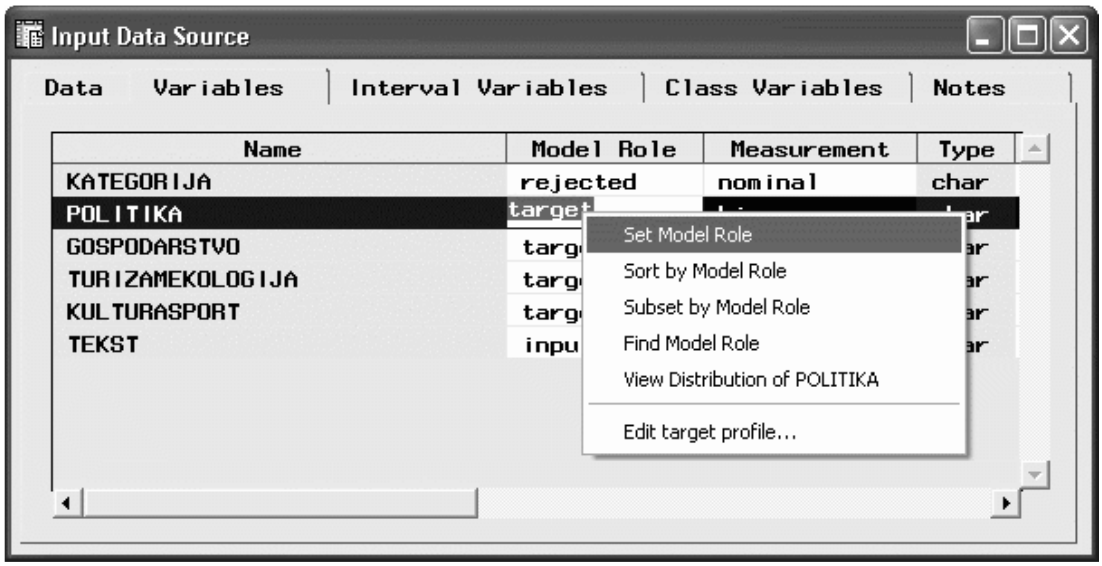


Slika 18. Odabir broja uzoraka.

Sljedeći korak je odabir podataka tj. *dataset*-a nad kojim ćemo obaviti dubinsku analizu. Nakon što pod *Source Data* dijelom odaberemo gumb *Select*… pojavit će se dijaloški okvir sa slike 19.

Slika 19. Odabir tablice sa podacima.

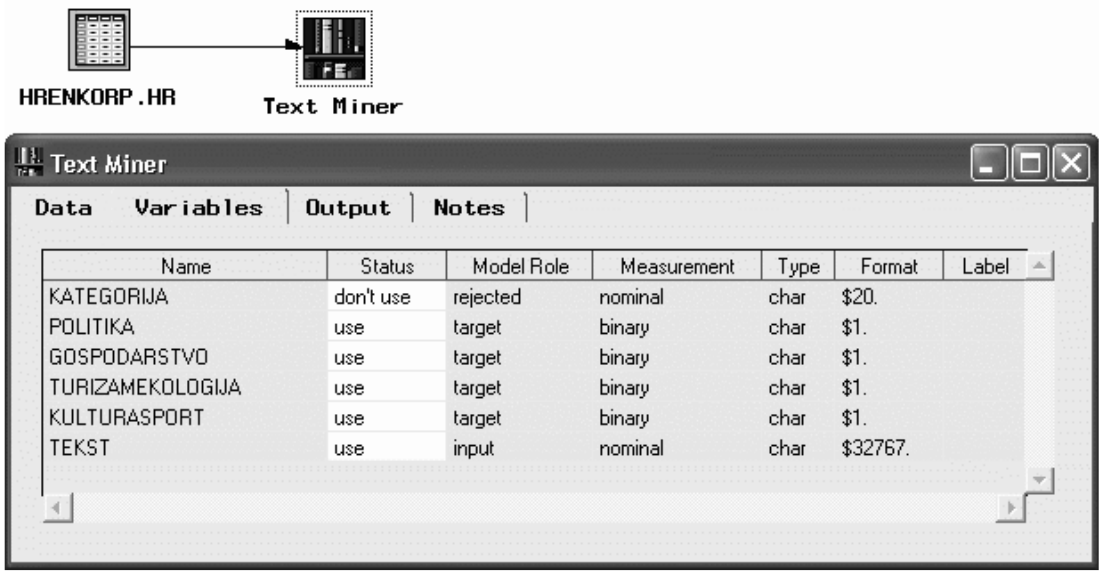
Osim toga, potrebno je postaviti ulogu svakoj od varijabli. Pod varijablom se smatra svaki stupac tablice sa podacima. Varijablama koje se ne koriste potrebno je za ulogu postaviti *rejected*, a varijablama koje se koriste za analizu (npr. tekst članaka) postavlja se uloga *input*. Binarne se varijable ne rabe za grupiranje nego obično za klasifikaciju. Njima se za ulogu postavlja *target* (slika 20).



Slika 20. Postavljanje uloge varijablama.

5.3.2. Text Miner Node

Na *Input Data Source* alat spaja se *Text Miner*. Osnovni okvir za podešavanje ovog alata sličan je kao i u svim ostalim alatima. Na njemu je potrebno odrediti koje će se varijable rabiti u postupku dubinske analize. Jedino je važno varijabli koja sadrži tekst članaka postaviti status na *use*. Prikaz spajanja *Input Data Source* alata s *Text Miner* alatom prikazan je na slici 21.

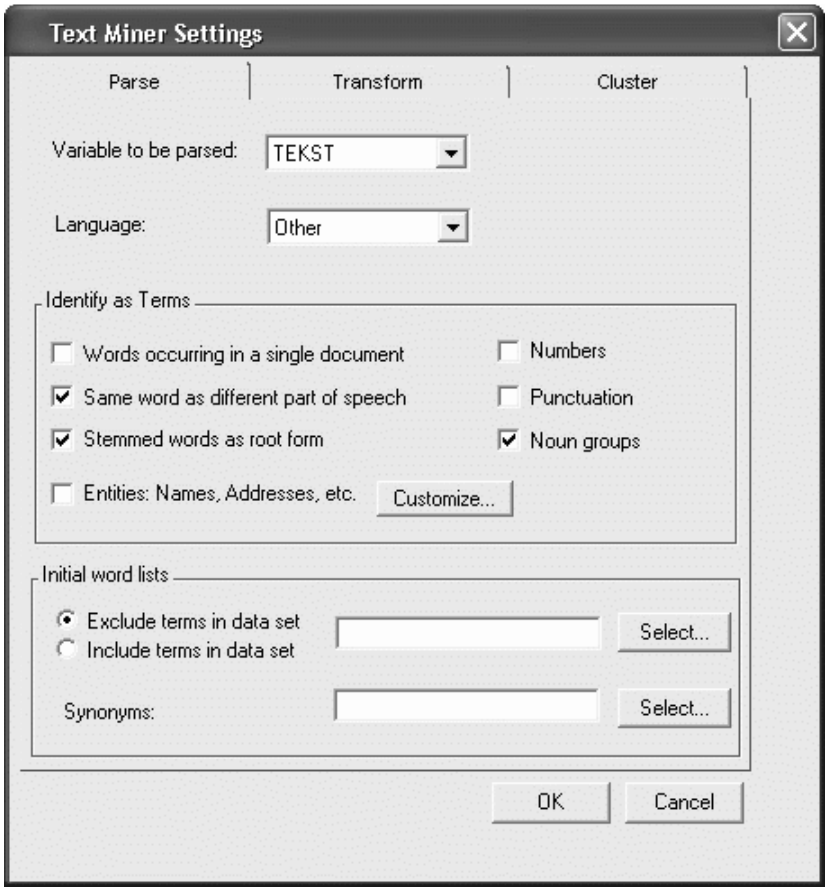


Slika 21. Text Miner alat.

Kao i većina naprednijih alata, tako i Text Miner alat ima, osim osnovnog okvira, još jedan okvir za podešavanje. On se sastoji od tri kartice: *Parse*, *Transform* i *Cluster*.

5.3.2.1. Parse kartica

Na ovoj se kartici podešavaju opcije vezane uz pretprocesiranje teksta, način čitanja riječi u tekstu i generiranje matrice riječ-dokument (za sada još bez dodjeljivanja težina riječima).



Slika 22. Dio okvira za podešavanje opcija kod parsiranja teksta.

Variable to be parsed opcijom određujemo koja će se varijabla rabiti za dubinsku analizu. Ta je varijabla obično ona koja sadrži tekst dokumenata tj. čija se uloga u prethodnim alatima u dijagramu označila kao *use* i *input*.

Language opcijom određujemo na kojem je jeziku napisan tekst spremljen u prethodno odabranoj varijabli. SAS *Text Miner* ima ugrađenu podršku za engleski, francuski, njemački te za miješane dokumente na spomenutim jezicima. Odabirom pojedinog jezika određuje se način kako će se tekst parsirati, koje će stop-riječi biti izbačene, kako će se obavljati svođenje riječi na osnovni oblik itd. Ako parsiramo dokumente koji ne pripadaju ovim trima jezicima, tada odabiremo opciju *Other*. U tom će se slučaju jezik tretirati kao engleski samo će se isključiti sve opcije vezane uz engleski jezik. Neće se obaviti nikakvo svođenje riječi na osnovni oblik niti će

se uklanjati stop-riječi. Opcija *Mixed* rabi se kada se u skupu dokumenata nalaze miješani dokumenti na engleskom, francuskom i njemačkom jeziku. Tada se automatski određuje koji dokument pripada kojem jeziku i onda se dokument parsira po pravilima tog jezika.

Grupa opcija *Identify as Terms* uključuje opcije za napredne tehnike parsiranja opisane u poglavlju 2.2. te služi za fino podešavanje postupka parsiranja.

Riječi koje se pojavljuju u samo jednom dokumentu obično se ne stavljaju u matricu riječ-dokument jer vjerojatno predstavljaju šum. No u nekim slučajevima, posebno u manjim skupovima dokumenata, može biti poželjno da se te riječi ne izbace. Ako se uključi opcija *Words occurring in a single document*, tada te riječi neće biti izbačene.

Opcija *Same words as different part of speech* uključuje označavanje riječi ovisno o njihovoj ulozi u rečenici. Na taj će se način iste riječi pojavljivati više puta u matrici riječ-dokument ako imaju različitu ulogu u rečenici (npr. imenica i pridjev).

Opcija *Stemmed words as root form* uključuje postupak svođenja riječi na njihov korijen. Taj je postupak objašnjen u poglavlju 2.2.2.2.

Kolekcija dokumenata ponekad može sadržavati izraze koji imaju poseban značaj, npr. imena ljudi, imena tvrtki, adrese, iznosi novaca itd. Kada se uključi opcija *Entities: Names, Addresses etc.*, tada se izrazi poput imena i adresa tretiraju kao jedan izraz.

Numbers i *Punctuations* opcije uključuju brojeve i posebne znakove kao izraze u matricu riječ dokument.

Ako je opcija *Noun groups* uključena, tada se za vrijeme parsiranja traže riječi koje se uvijek pojavljuju zajedno i tretiraju se kao jedan izraz (npr. „strojno učenje“, „klinička bolnica“ itd.).

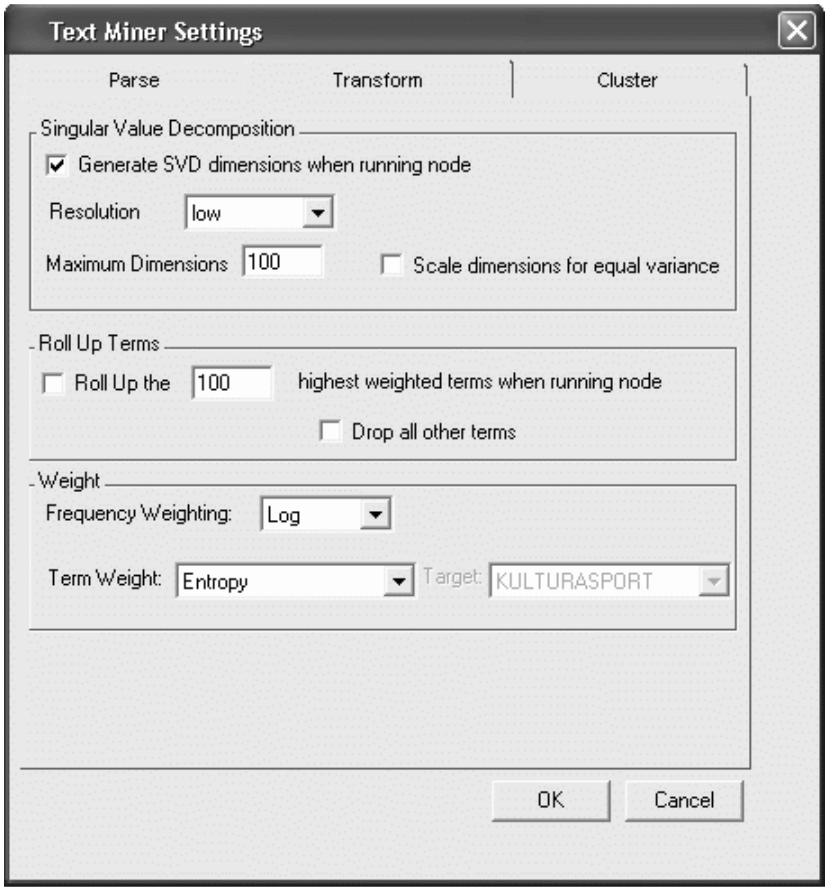
Grupa opcija *Initial word lists* uključuje ili uklanja pojedine riječi iz grupe dokumenata. Ako uključimo opciju *Exclude terms in data* set tada u nastavku trebamo odabrati tablicu sa stop-riječima. U slučaju odabira nekog od podržanih jezika ovdje će se automatski odabrati lista stop-riječi za taj jezik. Opcija *Include terms in data set* uključuje u analizu skup riječi koje su nam posebno važne. To je tzv. start-lista. Ako je ova opcija odabrana tada se iz dokumenata u analizu uzimaju samo te riječi. Opcijom *Synonyms* možemo uključiti listu sinonima u analizu. Više se o tome može pročitati u poglavlju 2.2.3.

5.3.2.2. Transform kartica

Druga kartica u okviru za podešavanje zove se *Transform*. Pomoću nje vrše se podešavanja vezana uz težinske funkcije i redukciju dimenzionalnosti. Izgled druge kartice prikazan je na slici 23.

Grupa opcija *Weight* odnosi se na težinske funkcije. Postoji mogućnost zasebnog odabira težinskih funkcija frekvencije i same riječi. Pod opcijom *Frequency weighting* moguće je odabrati *Binary*, *Log* i *None*, a pod

opcijom *Term weighting* moguće je odabrati *Entropy*, GF-IDF, IDF, *Normal*, *None*, *Chi-Squared*, *Mutual Information* i *Information Gain*. Detaljnija objašnjenja o svakoj se od njih mogu pronaći u poglavlju 2.3.2.



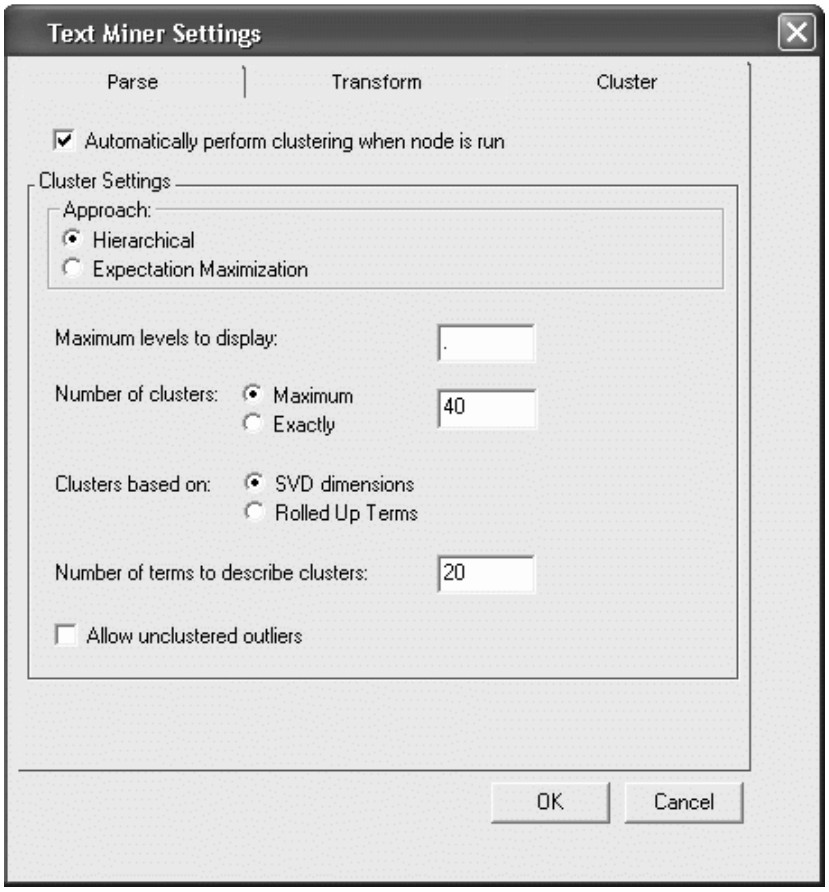
Slika 23. Dio okvira za podešavanje opcija vezanih uz težinske funkcije i redukciju dimenzionalnosti.

Preostale dvije grupe opcija odnose se na redukciju dimenzionalnosti. Jednostavnija je tehnika *Roll Up Terms* koja je objašnjena u poglavlju 3.1. Broj najviše rangiranih izraza koje želimo uzeti u analizu upisuje se u okvir za tekst *Roll Up the…* Naprednija tehnika za redukciju dimenzionalnosti jest SVD i ona je opisana u poglavlju 3.2. Opcijama za ovu tehniku moguće je podesiti rezoluciju algoritma, broj dimenzija i skaliranje. Rezolucija se može postaviti na vrijednosti *low*, *medium* i *high*. Za broj dimenzija upisuje se proizvoljan broj veći od 1. Broj stvarno izračunatih dimenzija ovisi o postavljenoj rezoluciji. On se određuje heuristički ovisno o pogrešci između dvije iteracije algoritma. Veća rezolucija uzrokuje izračunavanje većeg broja dimenzija. Ako je broj stvarno izračunatih dimenzija manji od broja upisanih dimenzija, tada algoritam nastavlja raditi sve dok se ne izračunaju preostale dimenzije. Zanimljiva je i kombinacija ova dva algoritama. Ako se uključe i Roll Up Terms i SVD, tada najprije Roll Up Terms postupak uzme l najviše rangiranih riječi, a zatim se SVD algoritmom one svedu na k dimenzija, $k<l$.

Na taj se način može uštedjeti na brzini izvođenja koraka redukcije dimenzionalnosti, a bez loših utjecaja na točnost dobivenih rezultata.

5.3.2.3. Cluster kartica

Posljednja kartica služi za podešavanje posljednjeg koraka za dubinsku analizu teksta – grupiranja (slika 24).

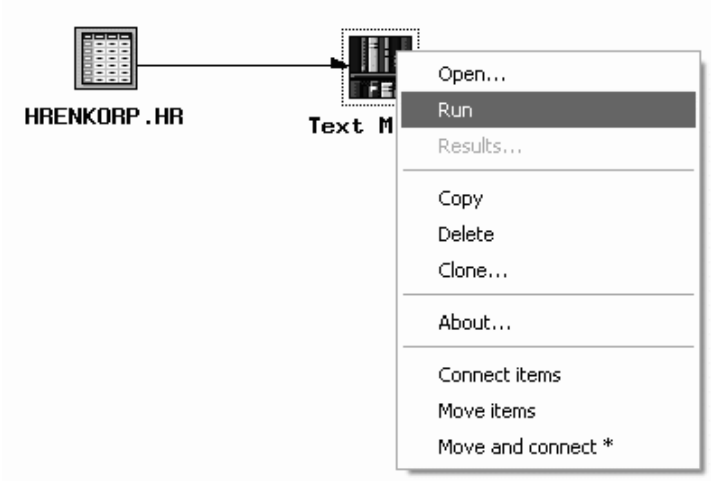


Slika 24. Dio okvira za podešavanje opcija kod grupiranja.

SAS *Text Miner* na raspolaganje daje dva algoritma grupiranja: hijerarhijski i EM. Algoritmi su objašnjeni u poglavlju 4.5. Njih biramo u grupi opcija *Approach*. Za oba algoritma vrijede opcije unutar grupe *Cluster Settings*. Pomoću opcije *Number of clusters* možemo odrediti maksimalan ili točan broj grupa (ako ga znamo) nakon kojeg će algoritam stati. Za hijerarhijsko grupiranje vrijedi opcija *Maximum levels to display* koja određuje najveću dubinu hijerarhije u stablu. Algoritmi kao ulaz mogu koristiti rezultate SVD-a ili metode *Roll Up Terms*. To se bira opcijom *Clusters based on*. Svaku grupu koju algoritam pronađe on će i opisati sa nekoliko najznačajnijih riječi. Broj tih riječi upisujemo u polje za unos *Number of terms to describe clusters*. Opcija *Allow unclustered outliers* smješta sve uzorke koji predstavljaju šum u jednu grupu pri EM grupiranju, a pri hijerarhijskom se grupiranju takvi uzorci izbacuju iz rezultata.

5.3.3. Pokretanje algoritama

Kada smo nacrtali cijeli dijagram i u svim alatima u dijagramu podesili odgovarajuće parametre, možemo pokrenuti sve algoritme. Dovoljno je na posljednji alat u dijagramu kliknuti desnom tipkom i zatim odabrati *Run* (slika 25).



Slika 25. Pokretanje cijelog postupka.

Duljina izvršavanja programa ovisi o veličini skupa podataka, postavljenim opcijama pri parsiranju, odabiru metode za redukciju dimenzionalnosti i odabiru algoritma grupiranja. Za skup od otprilike 3 500 dokumenata s odabranom SVD metodom za redukciju dimenzionalnosti od 100 dimenzija i hijerarhijskim grupiranjem taj postupak može na računalu s Intel Pentium IV procesorom brzine 2,4GHz i Hyper Threading tehnologijom te 512 MB DDR radne memorije trajati oko sat vremena.

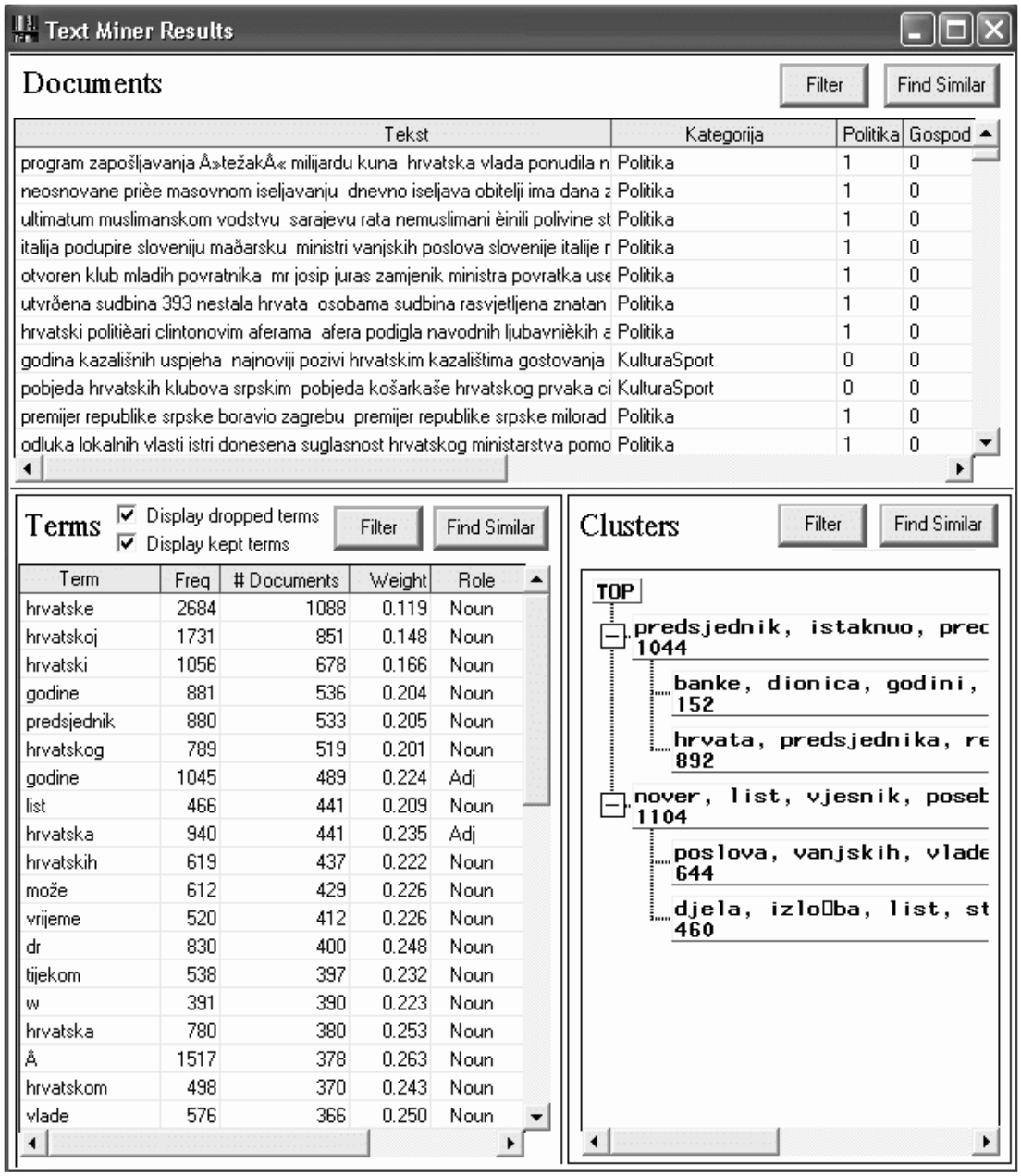
Kad program završi s radom, ponovno možemo kliknuti desnom tipkom miša na Text Miner alat i iz kontekstnog izbornika odabrati *Results*. Prikazat će se interaktivni prozor za pregledavanje rezultata koji se sastoji od tri dijela (slika 26).

U gornjem dijelu su prikazani svi obrađeni dokumenti zajedno sa svim pripadajućim varijablama. Izgled bi trebao biti isti kao i kod ulazne CSV datoteke iz koje smo učitali dokumente. Posljednji stupac (koji se na slici ne vidi) čini grupa u koju je pojedini dokument smješten. Dokumenti se mogu sortirati po svakom stupcu. Naročito korisno ih je sortirati po kategoriji ili po pridijeljenoj grupi.

U donjem lijevom dijelu nalazi se popis svih riječi koje su pronađene rijekom postupka parsiranja. Taj dio prozora zapravo predstavlja vektor sa riječima iz matrice riječ-dokument. Za svaku se riječ može vidjeti frekvencija pojavljivanja, zatim u koliko se dokumenata ta riječ pojavila, koja je njena težina izračunata odabranom težinskom funkcijom te koja je njezina uloga u rečenici. Tablica se može sortirati po bilo kojem stupcu. Korisno je obaviti

silazno sortiranje po stupcu težine tako da se na vrhu dobiju riječi sa najvećim težinskim vrijednostima.

U donjem desnom dijelu prozora nalaze se sve grupe koje je pronašao algoritam grupiranja. U slučaju hijerarhijskog grupiranja tu će biti prikazano stablo na čijim će se čvorovima nalaziti riječi koje najbolje opisuju tu grupu te broj dokumenata koji se u njoj nalaze. U slučaju EM grupiranja tu će biti prikazana tablica s trima stupcima u kojima su riječi koje najbolje opisuju grupu, broj dokumenata koji se u njoj nalaze i postotak.

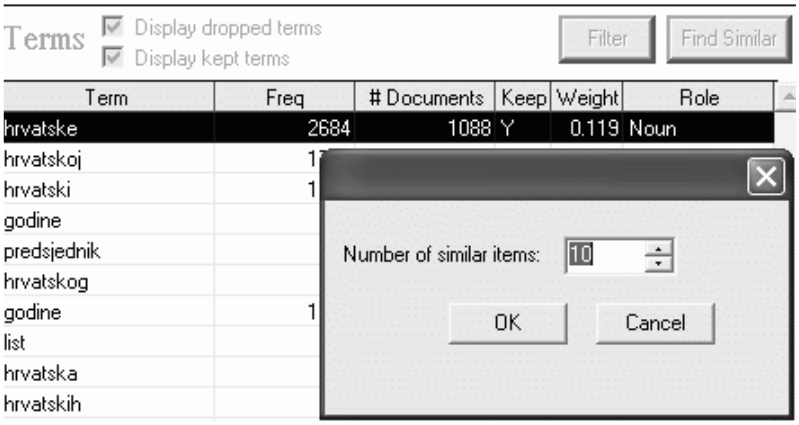


Slika 26. Interaktivni prozor za pregledavanje rezultata grupiranja.

Inicijalno sva tri dijela prozora prikazuju rezultate dubinske analize za sve dokumente, sve riječi i sve grupe koje su pronađene u skupu

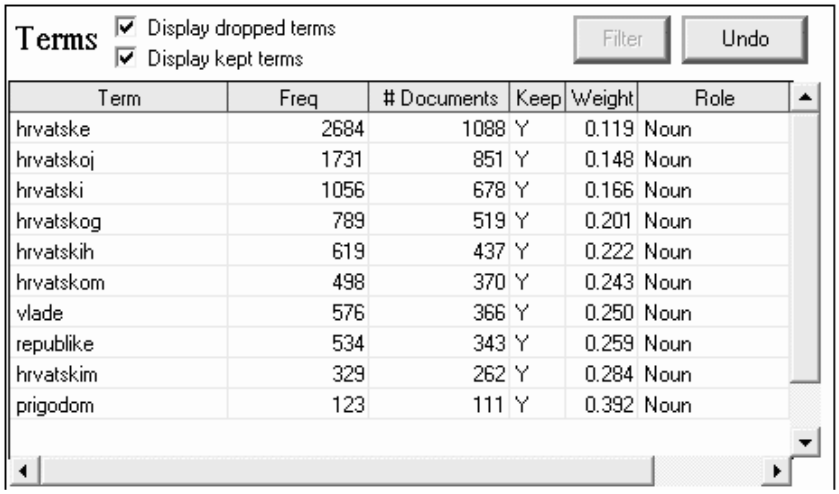
dokumenata. Ako kliknemo na gumb *Filter* u bilo kojem dijelu prozora tada možemo napraviti filtriranje sadržaja tog dijela prozora. Filtriranjem jednog dijela mijenja se sadržaj druga dva dijela prozora. Npr. filtriranjem nekoliko odabranih dokumenata sadržaj dijela prozora koji prikazuje riječi promijenit će se tako da pokazuje samo riječi koje se nalaze u odabranim dokumentima. Dio koji pokazuje grupe također će se promijeniti i pokazivat će razmještaj odabranih dokumenata u grupe. Isto vrijedi i za filtriranje riječi i grupa.

Gumb *Find Similar* omogućuje pronalaženje sličnih riječi, dokumenata i grupa. On se također nalazi u sva tri dijela prozora. U okviru za dijalog koji se pojavi kada se klikne na taj gumb potrebno je samo upisati broj sličnih riječi, dokumenata i grupa koje želimo pronaći (slika 27).



Slika 27. Traženje sličnih riječi.

Rezultat jedne takve pretrage prikazan je na slici 28. Na njoj su prikazani rezultati pretrage riječi sličnih sa riječi „hrvatske“. Iz rezultata se vidi da su sve prikazane riječi ili morfološki oblici riječi „hrvatska“ ili riječi koje se često pojavljuju uz riječ „hrvatska“ (npr. vlade).



Slika 28. Rezultat traženja riječi sličnih riječi „hrvatske“.

5.4. Primjer

Na sljedećem će primjeru biti prikazan postupak dubinske analize teksta zajedno s prednostima korištenja SVD postupka i algoritama grupiranja.

Pretpostavimo da imamo skup dokumenata prikazanih niže u tekstu. Dokumenti 1, 3 i 6 govore o bankarstvu i financijskim institucijama. Da budemo precizniji, dokumenti 3 i 6 govore o posudbi novaca od financijske institucije. Dokumenti 2, 4, 5 i 7 govore o obali rijeke. U dokumentima 8 i 9 radi se o paradi. Neki od ovih dokumenata dijele iste riječi. Primijetite da se riječi banka i obala rijeke na engleskom pišu isto – *bank*. Riječ *check* služi kao imenica u dokumentu 1 ili kao glagol u dokumentu 8. Riječ *floats* koristi se kao glagol u dokumentu 4 i kao objekt koji se pojavljuje u paradi u dokumentu 8.

- Dokument 1 – deposit the cash and check in the bank
- Dokument 2 – the river boat is on the bank
- Dokument 3 – borrow based on credit
- Dokument 4 – river boat floats up the river
- Dokument 5 – boat is by the dock near the bank
- Dokument 6 – with credit, I can borrow cash from the bank
- Dokument 7 – boat floats by dock near the river bank
- Dokument 8 – check the parade route to see the floats
- Dokument 9 – along the parade route.

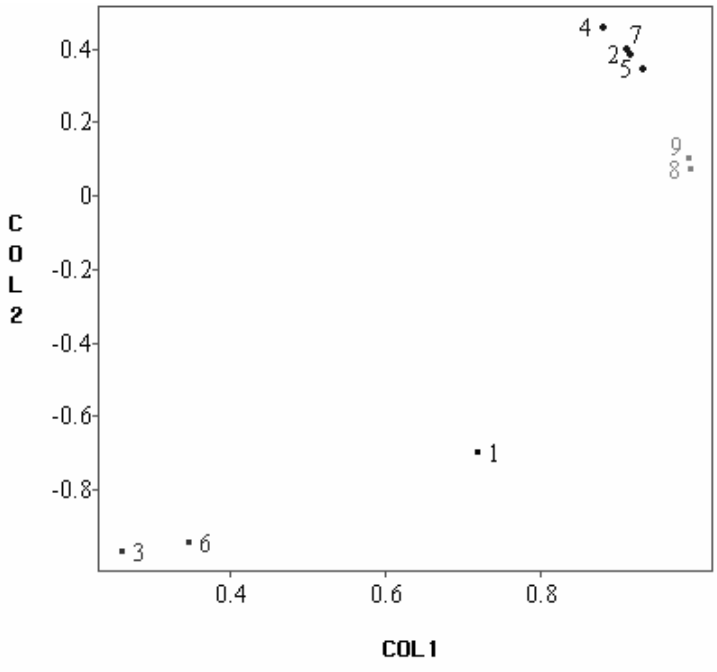
Parsiranjem skupa dokumenta stvara se matrica riječ-dokument.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
cash	1	0	0	0	0	1	0	0	0
check	1	0	0	0	0	0	0	1	0
bank	1	1	0	0	1	1	1	0	0
river	0	1	0	2	0	0	1	0	0
boat	0	1	0	1	1	0	1	0	0
borrow	0	0	1	0	0	1	0	0	0
credit	0	0	1	0	0	1	0	0	0
floats	0	0	0	1	0	0	1	1	0
dock	0	0	0	0	1	0	1	0	0
parade	0	0	0	0	0	0	0	1	1
route	0	0	0	0	0	0	0	1	1

Tablica 2. Matrica riječ-dokument s primjera.

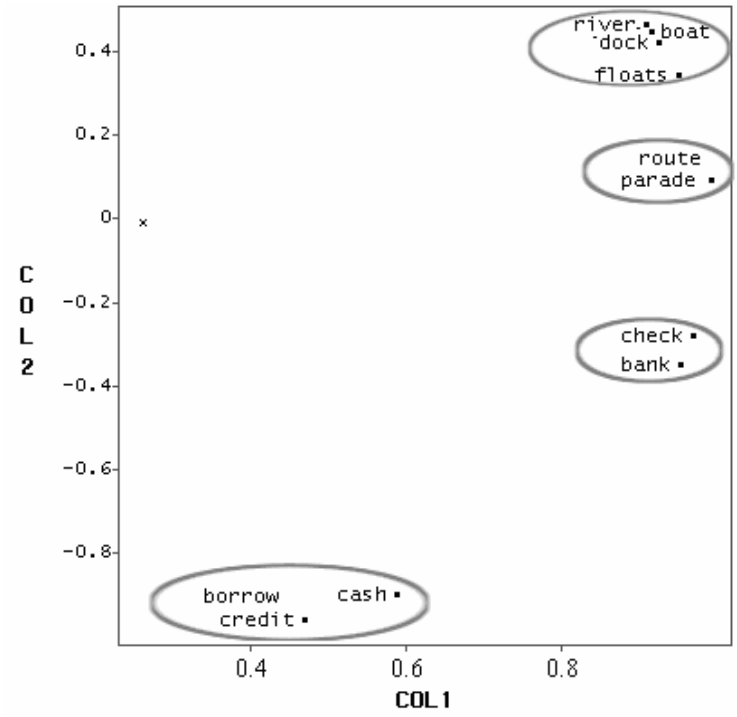
Pregledavajući matricu riječ-dokument vidimo da su dokumenti 1 i 2 sličniji nego dokumenti 1 i 3. To je zato jer oni dijele istu riječ *bank* dok dokumenti 1 i 3 nemaju niti jednu zajedničku riječ. No u stvarnosti dokumenti 1 i 2 nisu uopće slični, dok dokumenti 1 i 3 jesu. SVD postupak pomaže u uklanjaju takvih teškoća.

Ako sada primijenimo SVD algoritam na matricu riječ-dokument i projiciramo je iz 11-dimenzionalnog u dvodimenzionalni prostor dobit ćemo rezultate prikazane na slici 29.



Slika 29. Prikaz dokumenata s primjera u dvodimenzionalnom prostoru.

Dokument 1 je sada bliži dokumentu 3 nego dokumentu 2 iako sa dokumentom 3 ne dijeli nikakve zajedničke riječi. Dokument 5 je izravno povezan s dokumentima 2, 4 i 5 što je projekcija i pokazala. U reduciranom prostoru projekcija smješta slične dokumente zajedno, iako oni dijele samo nekoliko zajedničkih riječi. Na sljedećoj slici može se vidjeti prikaz izraza. Izrazi tvore četiri različite grupe.



Slika 30. Prikaz izraza s primjera u dvodimenzionalnom prostoru.

6. Eksperimenti

Na tržištu postoji mnogo alata za dubinsku analizu teksta. Svi su oni najprije prilagođeni engleskom jeziku te možda još nekim važnijim svjetskim jezicima. Dobiveni rezultati s dokumentima pisanim na tim jezicima jako su dobri. No, kako se hrvatski jezik po strukturi i morfologiji prilično razlikuje od engleskog jezika, potrebno je provjeriti iskoristivost takvih alata za dokumente na hrvatskom jeziku. Dakle, zadatak je bio usporediti rezultate grupiranja u dokumentima na hrvatskom i engleskom jeziku.

6.1. Skup podataka za testiranje

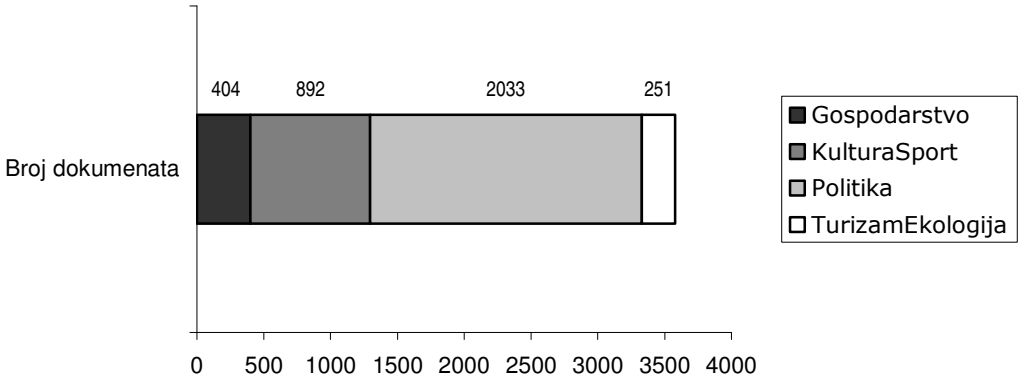
Skup podataka za testiranje čine članci iz časopisa *Croatia Weekly* koji je izlazio od 1998. do 2000. godine paralelno na hrvatskom i engleskom jeziku. Članci su raspoređeni u više kategorija koje su ovisile o mjestu na kojem se članak nalazi u časopisu. Neke od kategorija su

- unutarnja politika,
- unutarnja politika + susjedne države,
- iz tiska,
- gospodarstvo,
- turizam i ekologija,
- vanjska politika, Hrvati u svijetu,
- kultura i šport
- zadnje vijesti.

No kako su kategorije dosta izmiješane, zaključili smo da bi nešto jednostavnija podjela bila bolja. Prema tome, članci su bili raspoređeni u četiri kategorije:

- gospodarstvo (G),
- kultura i šport (KS),
- politika (P) i
- turizam i ekologija (TE).

Članci koji se nisu mogli svrstati u neku od tih kategorija bili su izbačeni. Nakon takvog odabira u skupu podataka nalazilo se 3 580 članaka na hrvatskom jeziku i isto toliko članaka na engleskom jeziku. Broj riječi u člancima na hrvatskom jeziku iznosio je 1,6 milijuna, a u člancima na engleskom 1,9 milijuna. Razdioba članaka po grupama prikazana je na slici 31.



Slika 31. Broj dokumenata po kategorijama za hrvatsko-engleski skup podataka s ukupno 3 580 članaka.

6.2. Rezultati

Još jednom treba napomenuti da u ovim eksperimentima nisu toliko bitni apsolutni rezultati. Kategorije su ionako umjetno određene pa nema prevelikog smisla tvrditi da su one apsolutno točne, a rezultati grupiranja krivi. Središte interesa bit će stavljeno na usporedbu rezultata grupiranja dokumenata na hrvatskom i engleskom jeziku. Eksperimenti su provođeni postupkom prikazanim u poglavlju 5.

6.2.1. Particijsko grupiranje

Prvi je eksperiment napravljen s EM algoritmom objašnjenim u poglavlju 4.5.2.2. Najprije su bile fiksno postavljene četiri grupe, no rezultati nisu bili zadovoljavajući. Očito je da osjetljivost algoritma nije tolika da bi mogao otkriti ovako umjetno definirane grupe. Zato su eksperimenti ponovljeni, ali ovaj put je algoritam sam odredio broj grupa. Rezultirajući broj grupa kretao se od 9 do 11 no u njima su se mogle odrediti zadane kategorije. Rezultati grupiranja zajedno sa dodanim kategorijama prikazani su u tablicama 3 i 4.

Uz pomoć ovih tablica i dodatnih podataka dobivenih grupiranjem radimo matrice sličnosti. One su prikazane u tablicama 5 i 6 za engleski i hrvatski jezik. Iz matrica sličnosti računamo čistoću grupa kao što je opisano u poglavlju 4.6.1.

Riječi koje najbolje opisuju grupu	Br. dok.	Pridijeljena kategorija
account, amount, bank, billion, business, company, economic	345	G
adriatic, century, fish, forest, island, nature, old, paint, park, preserve	382	TE
army, border, bosnia, bosnia-herzegovina, bosnian, community	585	P
best, champion, championship, coach, cup, defeat, final, game	105	KS
million, fund, company, project, sport, plan, ministry, numb, sign	432	G
camp, charge, commander, commit, court, crime, criminal, hague	167	P
bury, cemetery, grave, kill, mass, miss, victim, vukovar, war, cross	153	P
audience, award, concert, director, festival, film, list, music, play	268	KS
art, artist, artistic, author, award, book, contemporary, cultural	360	KS
animal, protect, specie, forest, natural, park, sea, nature, fish	75	TE
cooperation, democratic, election, meet, party, political, president	708	P

Tablica 3. Grupe sa ključnim riječima, brojem dokumenata u grupi te ručno pridjenutim kategorijama za članke na engleskom jeziku.

Riječi koje najbolje opisuju grupu	Br. dok.	Pridijeljena kategorija
grade, zanimati, sredstvo, namjena, osiguran, nastup, otvor, kredit	28	G
izlog, kultura, stoljeće, umjetan, listi, autor, star, knjiga, list, svijet	758	KS
festival, film, kazališan, kazalište, prikazan, redatelj, teatar, izvedba	117	KS
predsjednik, iznos, odnos, polit, vlada, poslovan, gospodar, banka	747	P
diriger, glazba, izvedba, koncert, nastup, opera, orkestar, pijanist	135	KS
stoljeće, umjetan, zagreb, nastup, autor, izlog, knjiga, djelo, listi	285	KS
bosna, međunarodni, ratni, srb, srpski, sud, zločin, polit, jugoslavija	478	P
izjava, jugoslavija, polir, poslovi, vanja, ministar, srpski, pitan, vlada	690	P
banka, financija, iznos, kredit, kuna, marka, poslovan, proizvod	342	G

Tablica 4. Grupe sa ključnim riječima, brojem dokumenata u grupi te ručno pridjenutim kategorijama za članke na hrvatskom jeziku.

	Gospodarstvo	*Ostali*
C_1	395	382

	Kultura i šport	*Ostali*
C_2	687	46

čistoća grupa = 0,8112

	Politika	*Ostali*
C_3	1575	38

	Turizam i ekologija	*Ostali*
C_4	247	210

Tablica 5. Matrica sličnosti za grupe i kategorije dokumenata na engleskom jeziku.

	Gospodarstvo	*Ostali*
C_1	272	98

	Kultura i šport	*Ostali*
C_2	827	468

čistoća grupa = 0, 7922

	Politika	*Ostali*
C_3	1737	177

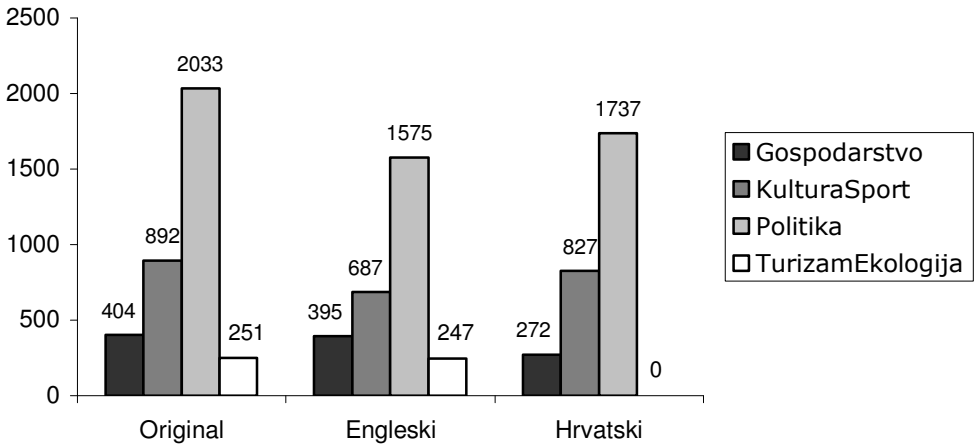
	Turizam i ekologija	*Ostali*
C_4	0	0

Tablica 6. Matrica sličnosti za grupe i kategorije dokumenata na hrvatskom jeziku.

Čistoća grupa za rezultate grupiranja članaka na engleskom jeziku iznosi 0,8112, a za hrvatski jezik 0,7922. Vidimo da razlika nije velika u korist engleskog jezika, ali promatranjem matrice sličnosti možemo doći do važnih zaključaka. Za dokumente na hrvatskom jeziku algoritam nije mogao pronaći niti jednu grupu za koju bi mogli reći da opisuje kategoriju „Turizam i ekologija“, dok je za engleske dokumente algoritam pronašao čak dvije takve grupe. Promatrajući grupu „Politika“ možemo vidjeti da je algoritam bio precizniji kada smo za eksperimente koristili dokumente na hrvatskom jeziku. Ponavljanjem eksperimenata više puta došlo se do istih rezultata. Ako sada usporedimo broj dokumenata u kategorijama „Turizam i ekologija“ i „Politika“, vidimo da je kategorija „Turizam i ekologija“ sa samo 251 dokumentom najmanja, dok je kategorija „Politika“ s 2 033 dokumenta najveća. Prema tome možemo zaključiti da je za hrvatski jezik potrebno više dokumenata po kategoriji da bi EM algoritam grupiranja mogao tu kategoriju uspješno razlučiti od drugih.

Slika 32 prikazuje usporedbu rezultata grupiranja dokumenata na hrvatskom i engleskom jeziku sa zadanom razdiobom. Kao što je već bilo rečeno, zadana razdioba je subjektivna i nije apsolutno točna, ali može poslužiti kao referentna razdioba. Iz grafa se vidi da preciznost grupiranja dokumenata na engleskom jeziku slijedi zadanu razdiobu u sve četiri kategorije. Preciznost grupiranja dokumenata na hrvatskom jeziku je dobra za kategorije s više dokumenata („Kultura i sport“ i „Politika“), dok je za

kategorije s manje dokumenata lošija (za kategoriju „Gospodarstvo“) tj. nikakva (za kategoriju „Turizam i ekologija“).



Slika 32. Usporedba preciznosti grupiranja dokumenata na hrvatskom i engleskom jeziku sa originalnom razdiobom dokumenata po kategorijama za EM algoritam grupiranja.

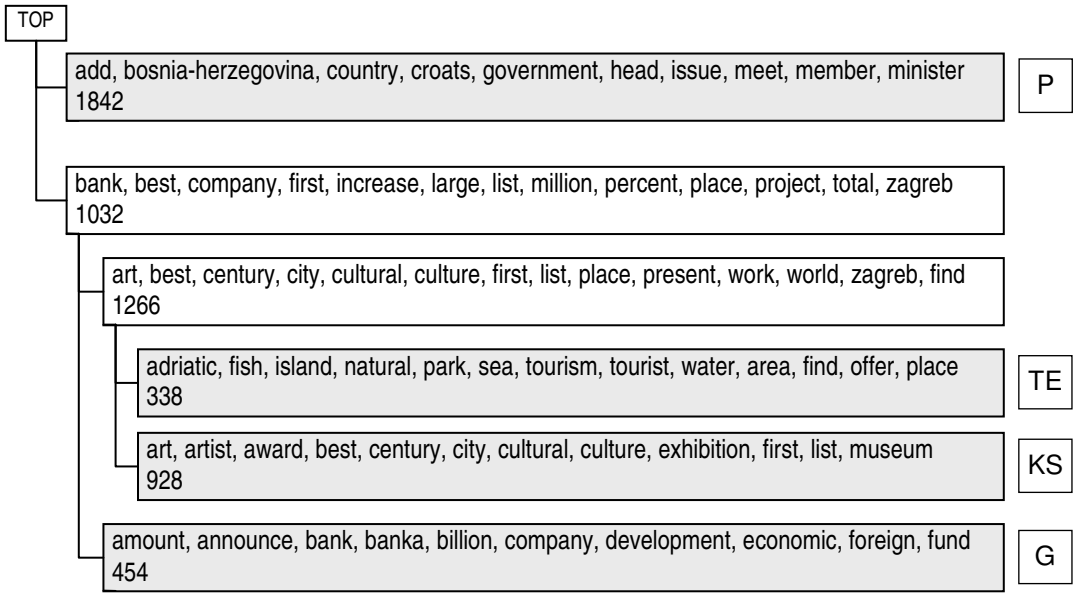
6.2.2. Hijerarhijsko grupiranje

Drugi algoritam kojeg podržava SAS *Text Miner* jest algoritam hijerarhijskog grupiranja. Više se o njemu može pročitati u poglavlju 4.5.1.1. Rezultati dobiveni ovim algoritmom prikazani su na slici 33 za dokumente na engleskom i slici 34 za dokumente na hrvatskom jeziku.

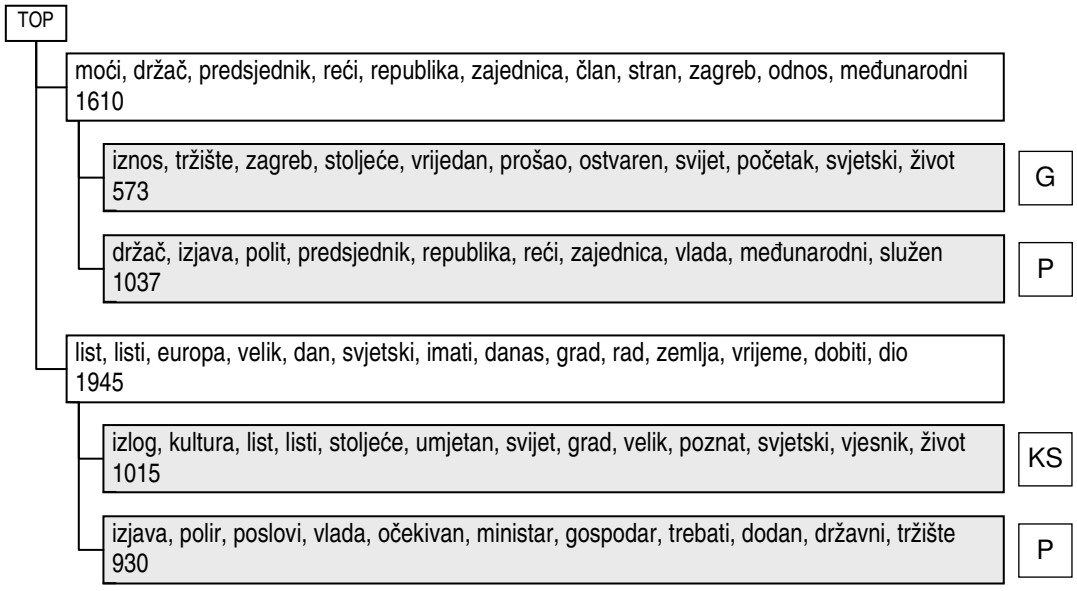
Na slikama je prikazana hijerarhija grupa koju je otkrio algoritam hijerarhijskog grupiranja. Njima su dodane i oznake kategorija u koje konačne grupe spadaju. Važno je napomenuti da samo listovi stabla predstavljaju konačne grupe. Zato su te grupe posebno obojane sivom bojom. Ako usporedimo rezultate možemo ponovno primijetiti isti nedostatak kao i pri grupiranju s EM algoritmom. Pri grupiranju dokumenata na hrvatskom jeziku algoritam ponovno nije uspio izdvojiti niti jednu grupu koju bi mogli smjestiti u kategoriju „Turizam i ekologija“. Očito je da je za hrvatski jezik potrebno nešto više dokumenata po kategoriji da bi algoritam uspio izdvojiti grupu koja odgovara toj kategoriji.

Možemo pogledati i sam oblik i izgled hijerarhije na slikama. Vidimo da je za rezultate grupiranja dokumenata na engleskom jeziku ta hijerarhija logičnija. Grupa koju povezujemo sa kategorijom „Politika“ predstavlja zasebni dio hijerarhije, a ostale tri grupe čine preostali dio hijerarhije. Grupe koje odgovaraju kategorijama „Kultura i šport“ i „Turizam i ekologija“ grupirane su zajedno što je i logično – te dvije kategorije međusobno su sličnije od kategorije „Gospodarstvo“. Za rezultate grupiranja dokumenata na hrvatskom jeziku hijerarhija je nešto drugačija. Ovdje su grupe koje vežemo uz kategorije „Politika“ i „Gospodarstvo“ smještene zajedno u jednoj grani hijerarhije, a u drugoj grani hijerarhije su zajedno smještene grupe koje odgovaraju kategorijama „Politika“ te „Kultura i šport“. Iako je gornji dio

hijerarhije logičan, donji dio hijerarhije baš i nije. Očito je da nedostaje grupa kojoj ogovara kategorija „Turizam i ekologija“ i to je vjerojatni razlog ovakvoj hijerarhiji.



Slika 33. Hijerarhija grupa s ključnim riječima, brojem dokumenata u grupi i ručno pridijeljenim kategorijama za članke na engleskom jeziku.



Slika 34. Hijerarhija grupa s ključnim riječima, brojem dokumenata u grupi i ručno pridijeljenim kategorijama za članke na hrvatskom jeziku.

Matrice sličnosti za rezultate grupiranja dokumenata na engleskom i hrvatskom jeziku prikazane su u tablicama 7 i 8.

	Gospodarstvo	*Ostali*
C_1	395	382

	Kultura i šport	*Ostali*
C_2	687	46

čistoća grupa = 0,8947

	Politika	*Ostali*
C_3	1575	38

	Turizam i ekologija	*Ostali*
C_4	247	210

Tablica 7. Matrica sličnosti za grupe i kategorije dokumenata na engleskom jeziku.

	Gospodarstvo	*Ostali*
C_1	272	98

	Kultura i šport	*Ostali*
C_2	827	468

čistoća grupa = 0, 8492

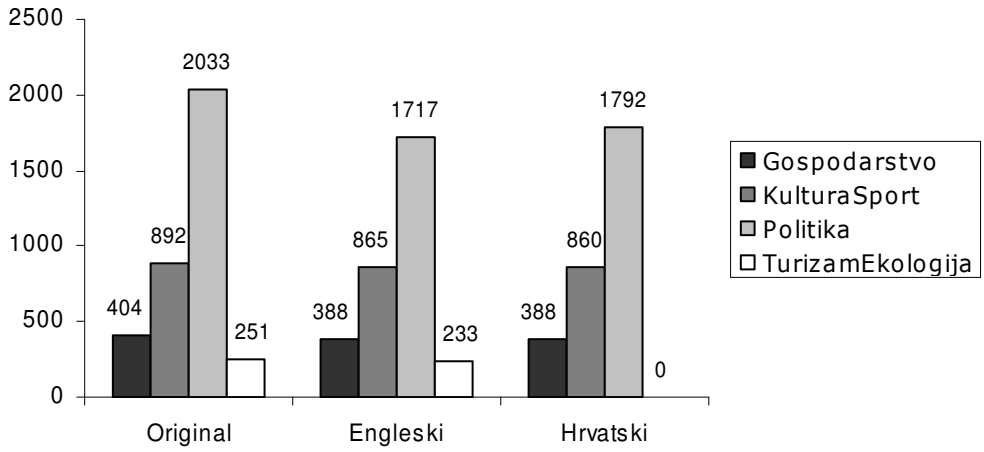
	Politika	*Ostali*
C_3	1737	177

	Turizam i ekologija	*Ostali*
C_4	0	0

Tablica 8. Matrica sličnosti za grupe i kategorije dokumenata na hrvatskom jeziku.

Ako izračunamo faktor čistoće grupa za oba rezultata, vidjet ćemo razliku i u njima. Za grupe dokumenata na engleskom jeziku čistoća iznosi 0,8947, dok je za grupe dokumenata na hrvatskom jeziku 0,8492. Usporedimo li ove rezultate s rezultatima dobivenim EM algoritmom, vidimo da su rezultati dobiveni algoritmom hijerarhijskog grupiranja bolji. U slučaju dokumenata na engleskom jeziku razlika je gotovo 10%, a kod rezultata s dokumentima na hrvatskom jeziku imamo poboljšanje od oko 5%. Valja još jednom napomenuti da su kod EM algoritma grupiranja bile dozvoljene višestruke kategorije iz kojih su se na kraju ručno stvarale četiri zadane grupe. Kod hijerarhijskog grupiranja to nije bio slučaj tako da bi se s finijim hijerarhijskim grupiranjem mogli postići još bolji apsolutni rezultati.

Već je bilo rečeno da apsolutni rezultati nisu prioritet ovih eksperimenata. Važnija je usporedba rezultata grupiranja dokumenata na engleskom i hrvatskom jeziku. Slika 35 prikazuje usporedbu rezultata sa zadanom razdiobom. Rezultati su jako dobri, kao što je faktor čistoće grupa i pokazao. Jedino što nedostaje jest jedna grupa u rezultatima grupiranja dokumenata na hrvatskom jeziku koju bismo mogli povezati s kategorijom „Turizam i ekologija“. Kod rezultata grupiranja engleskih dokumenata navedena grupa postoji i jako je dobro definirana.



Slika 35. Usporedba preciznosti grupiranja dokumenata na hrvatskom i engleskom jeziku s originalnom razdiobom dokumenata po kategorijama za algoritam hijerarhijskog grupiranja.

7. Praktična primjena grupiranja podataka

U ovom će poglavlju biti opisan jedan slučaj iz stvarnog života gdje je uporabom dubinske analize teksta u kratko vrijeme riješen problem čije bi rješavanje inače bilo dugotrajno čak i uz uporabu velikog broja ljudi.

7.1. Problem

Usklađivanjem nastavnog plana i programa Fakulteta elektrotehnike i računarstva s Bolonjskom deklaracijom trebale su se napraviti mnoge promjene. Jedna se od promjena odnosila na sadržaj i broj predmeta koji su se predavali u sklopu dosadašnjeg nastavnog plana. U sklopu tog plana postojalo je nešto više od 300 različitih predmeta što je jako velik broj. Zanimalo nas je možemo li korištenjem opisa predmeta dobiti informacije o nastavnom programu fakulteta te sličnim predmetima koji se na fakultetu predaju.

7.2. Rješenje

Jedino je znanje o nastavnom programu predmeta mogla pružiti informacija kojem zavodu pojedini predmet pripada. No kako se područja kojima se zavodi bave preklapaju, tako se i nastavni planovi pojedinih predmeta među tim zavodima preklapaju. Osim velikog broja predmeta to je još jedan razlog zašto bi ručna usporedba nastavnog plana svih predmeta bila duga i zahtjeva. Zato se iskoristilo rješenje koje pruža dubinska analiza teksta – grupiranje predmeta.

7.2.1. Ulazni podaci

Za svaki je predmet bio poznat njegov broj, naziv i opis. Opis se sastojao od nekoliko desetaka rečenica koje su bile naslovi predavanja u sklopu tog predmeta. Pretpostavka je, koja se na kraju pokazala ispravnom, bila da će se i sa tako malim brojem rečenica po predmetu moći uspješno grupirati slični predmeti.

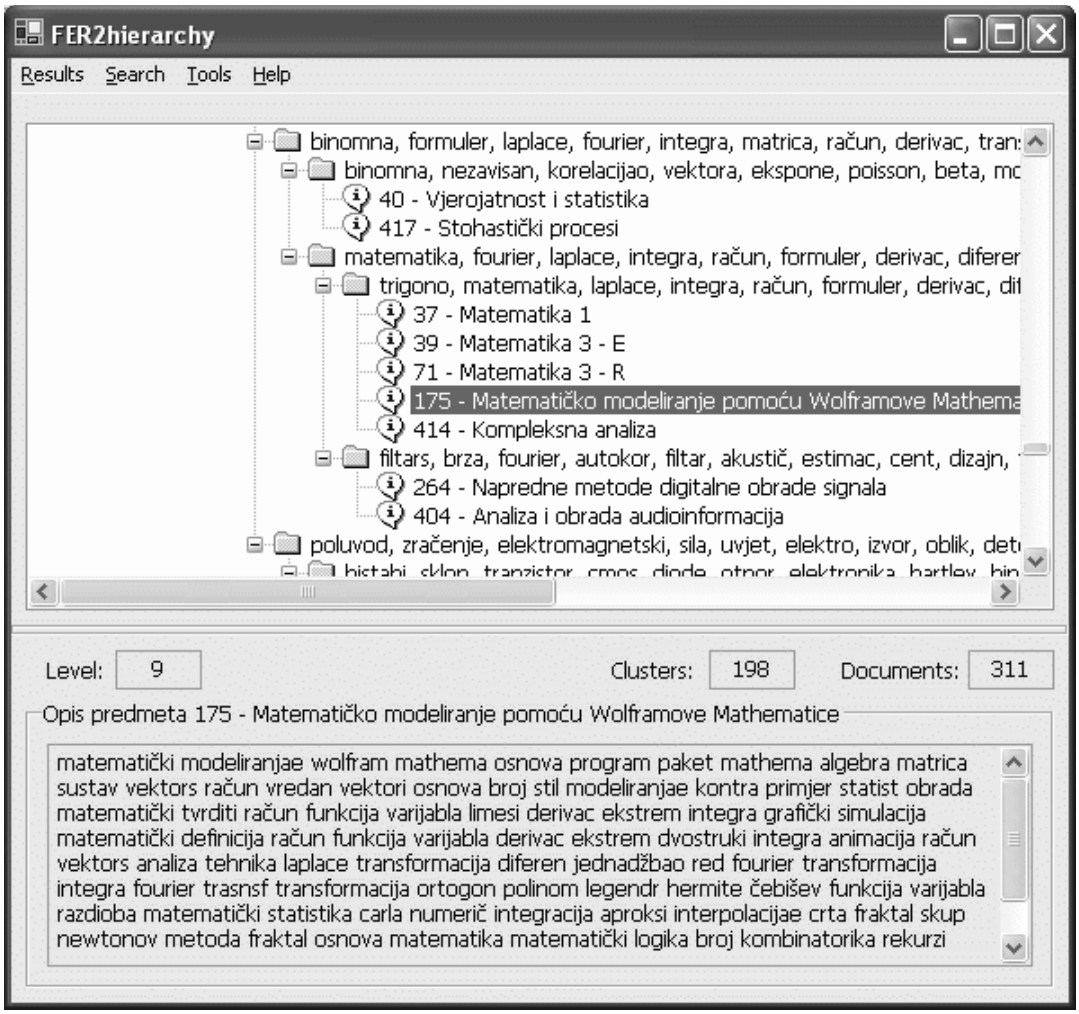
Postupkom prikazanim u prethodnim poglavljima ovog rada tekstualni opisi predmeta su bili pretprocesirani, zatim uneseni u SAS i algoritmom hijerarhijskog grupiranja grupirani sa jako finom podjelom – tristotinjak je predmeta bilo grupirano u stotinu grupa. Postupak normalizacije bio je proveden tako da je bio neutraliziran efekt flektivne morfologije, derivacijske morfologije, a zatim su bili filtrirani samo pridjevi i imenice (glagoli su bili ispušteni). Broj SVD dimenzija je bio postavljen na 100.

Rezultati grupiranja nisu mogli biti pregledavani u interaktivnom prozoru za pregledavanje rezultata grupiranja u SAS *Text Miner*-u, no zato su bili izvezeni u datoteke i napravljena je bila aplikacija za njihovo

pregledavanje. Opisi predmeta bili su dostupni na hrvatskom i na engleskom jeziku pa su zato bila napravljena oba eksperimenta. Korisniku je omogućeno da izabere na kojem će jeziku pregledavati rezultate grupiranja.

7.2.2. Aplikacija za vizualizaciju rezultata

Aplikacija je bila napravljena u cilju što lakšeg pregledavanja i pretraživanja rezultata. Izgled glavnog prozora aplikacije za vizualizaciju rezultata prikazan je na slici 36.



Slika 36. Glavni prozor aplikacije za vizualizaciju rezultata grupiranja predmeta FER2 nastavnog programa.

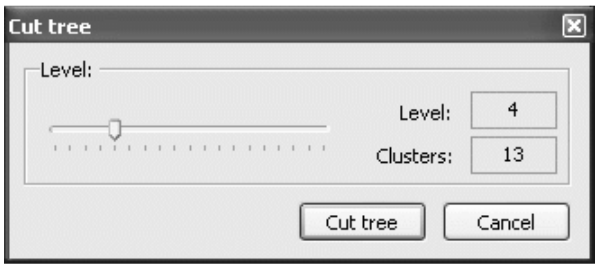
Iz izbornika *Results* mogu se odabrati rezultati grupiranja na hrvatskom ili engleskom jeziku. Nakon odabira jezika, program učitava rezultate iz datoteke i prikazuje ih u obliku stabla. Element stabla može biti grupa ili predmet. S ikonom prikazane su grupe, a s ikonom prikazani su predmeti. Grupe u sebi sadrže druge grupe i predmete, a predmeti mogu biti samo listovi stabla. Odabirom pojedine grupe ili predmeta u donjem se dijelu prozora prikazuju dodatne informacije o odabranom elementu stabla.

Za grupe se prikazuju riječi koje ih najbolje opisuju, a za predmete se prikazuje njihovo ime, broj i cijeli opis. Svaki element stabla ima svoj nivo koji se može vidjeti u okviru za tekst *Level*. Ukupan je broj grupa prikazan u okviru za tekst *Clusters*, a ukupan broj predmeta u okviru za tekst *Documents*.

Korisnici programa imaju i mogućnost pretraživanja rezultata grupiranja. U izborniku *Search* nalazi se stavka *Find…* pomoću koje se poziva okvir za pretraživanje prikazan na slici 37. Korisnici mogu pretraživati rezultate tako da u okvir za tekst *Find what* upišu broj ili naslov predmeta. Unutar grupe *Where* odabire se po čemu se žele pretraživati rezultati: po broju predmeta (*ID*) ili po broju i imenu predmeta (*ID & Name*). Pretraživanje je organizirano tako da se prikazuju svi rezultati koji zadovoljavaju uvjet pretraživanja, a ne samo prvi pronađeni rezultat. Dovoljno je samo više puta ponoviti naredbu *Find* s nepromijenjenim uvjetom za pretraživanje.

Slika 37. Okvir za pretraživanje rezultata grupiranja.

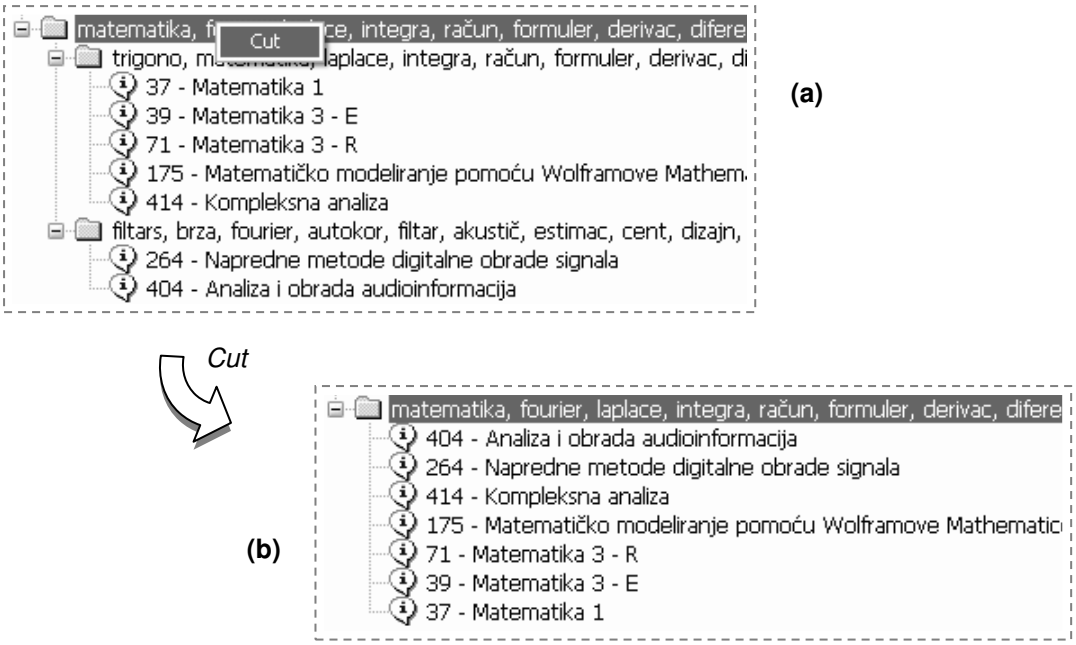
Ovakvo je detaljno razgranano stablo pogodno je za pronalaženje sličnih predmeta. No ako se želi dobiti okvirna slika o tome kakvih sve grupa predmeta ima, tada takav prikaz može biti nespretan. Zato je u programu omogućeno podrezivanje stabla na nivou kojeg zadaje korisnik. Do okvira za podrezivanje stabla koji je prikazan na slici 38 dolazi se preko *Tools* izbornika i *Cut tree…* stavke u izborniku. Okvir omogućuje podrezivanje stabla po nivoima. S pomičnom kontrolom unutar grupe *Level* odabire se nivo podrezivanja. Broj grupa koje će nastati takvim podrezivanjem prikazuje se unutar okvira za tekst *Clusters*. Odabirom nivoa i zatim naredbe *Cut tree* obavit će se podrezivanje stabla.



Slika 38. Okvir za podrezivanje stabla.

Podrezivanje samo jednog dijela stabla omogućeno je kroz kontekstni izbornik u glavnom prozoru aplikacije. Dovoljno je kliknuti desnom tipkom

miša na grupu na kojoj želimo obaviti podrezivanje i iz kontekstnog izbornika odabrati naredbu *Cut tree*. Izgled stabla prije i poslije takvog podrezivanja prikazan je na slici 39.



Slika 39. Prije (a) i nakon (b) podrezivanja jednog dijela stabla.

7.2.3. Rezultati

Hijerarhijski prikaz podataka pokazao se kao odličan odabir. Osim što hijerarhijski algoritam grupiranja ugrađen u SAS *Text Miner* može dati finiju podjelu grupa od particijskog EM algoritma, sami prikaz rezultata mnogo je razumljiviji u hijerarhijskom nego u particijskom grupiranju. S time su se složili i korisnici kojima je aplikacija puno pomogla u obavljanju posla.

8. Zaključak

Dubinska se analiza teksta pokazala jako korisnom i efikasnom metodom pri obradi velikih količina nestrukturiranih tekstualnih podataka. Velik broj znanstvenih članaka o dubinskoj analizi teksta, koji nastaju svakodnevno, pokazuje da je ovo vrlo interesantno područje koje je, unatoč relativno kratkom vremenu postojanja, dalo velike rezultate. Takvi znanstveni rezultati imali su kao posljedicu nastanak komercijalnih rješenja. Proizvođači Business Intelligence aplikacija, osim analize strukturiranih podataka, počeli su u svoja rješenja uvoditi i mogućnost analize nestrukturiranih tekstualnih podataka. Primjer je za to SAS sustav koji je bio korišten za sve eksperimente navedene u ovom radu.

Budući da tekst u dokumentima ovisi o jeziku kojim je pisan, većina je komercijalnih rješenja prilagođena svjetskim jezicima poput engleskog što bi moglo predstavljati problem za druge jezike. Ta se prilagođenost odnosi samo na korak pretprocesiranja samog teksta i upravo to omogućava efikasno rješavanje problema. Uz pretpostavku da se koraci izbacivanja stop-riječi, svođenja riječi na osnovni oblik i rješavanje problema sinonima obave prije obrade gotovim alatom, rezultati eksperimenata su pokazali da su takvi alati pogodni i za dubinsku analizu teksta pisanog i drugim jezicima, recimo hrvatskim.

Korištenje SVD algoritma za redukciju dimenzionalnosti pokazalo se kao isključivi uvjet za dobivanje bilo kakvih ozbiljnijih rezultata. Kombinacija SVD algoritma i EM algoritma grupiranja daje dobre rezultate koji mogu poslužiti za općeniti uvid u strukturu skupa dokumenata. SVD zajedno s algoritmom hijerarhijskog grupiranja čini par koji daje uvjerljivo najbolje rezultate. Osim točnosti i velike razlučivosti grupa, veliki plus hijerarhijskom grupiranju daje i način predstavljanja rezultata. Prikaz je rezultata u dendogramu pregledan, a postupkom podrezivanja moguće je prikaz takvih hijerarhijskih grupa dodatno prilagoditi zahtjevima.

Rezultati grupiranja dokumenata na hrvatskom i engleskom jeziku pokazuju da je u dokumentima na hrvatskom jeziku bilo teže otkriti kategorije s malenim broj dokumenata. Kategorije sa većim brojem dokumenata grupiraju se jednako dobro i za dokumente na hrvatskom i engleskom jeziku.

9. Literatura

[1] A. K. Jain, M. N. Murty, P. J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys, Vol 31, No. 3, pp 264-323,* (1999)

http://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf

[2] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification" (2nd ed.), *John Wiley & Sons, Inc.*, (2000)

[3] G. H. Golub, C. F. Van Loan, "Matrix Computations" (3rd ed.), *The John Hopkins University Press*, (1996)

[4] M. Berry, T. Do, G. O'Brien, V. Krishna, S. Varadhan, "SVDPACKC (Version 1.0) User's Guide", *Department of Computer Science, University of Tennessee*, (1996)

ftp://cs.utk.edu/pub/TechReports/1993/ut-cs-93-194.ps.Z

[5] SAS Institute Inc., "Getting Started with SAS® 9.1 Text Miner", *Cary, NC, SAS Institute Inc.*, (2004)

http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/em_tmgs_7693.pdf

[6] S. B. Kotsaiantis, P. E. Pintelas, "Recent Advances in Clustering: A Brief Survey", *WSEAS Transactions on Information Science and Applications, Vol. 1, No 1 (73-81)*, (2004)

http://www.cs.utsa.edu/~bylander/ cs6243/kotsiantis-clustering.pdf

[7] A. Hotho, S. Staab, A. Maedche, "Ontology-Based Text Clustering", *K unstliche Intelligenz, pp 48-54*, (2002)

http://www.aifb.uni-karlsruhe.de/WBS/Publ/2001/hothoetal.pdf

[8] A. Hotho, S. Staab, G. Stumme, "Text clustering based on background knowledge", (Technical Report), *University of Karlsruhe*, (2003)

http://www.aifb.uni-karlsruhe.de/WBS/aho/pub/hotho_etal_techreport425.pdf

[9] Cambridge University, Numerical Recipes Software, “Numerical Recipes in C: The Arth of Scientific Computing”, *Cambridge University Press*, (1992)

http://www.library.cornell.edu/nr/bookcpdf.html

[10] F. Beil, M. Ester, X. Xu, “Frequent Term-Based Text Clustering”, *In Proc. 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD)'2002*, (2002)

http://ifsc.ualr.edu/xwxu/publications/KDD02.pdf

[11] J. Šnajder, „Automatska normalizacija hrvatskog jezika“, (tehnički izvještaj), *Fakultet elektrotehnike i računarstva*, (2004)

[12] B. Mandhani, S. Joshi, K. Kummamuru, “A Matrix Density Based Algorithm to Hierarchically Co-Cluster Documents and Words”, *WWW2003 Budapest, Hungary,* (2003)

http://users.cs.dal.ca/~eem/malnis/Readings/mitacs/co-clustering/Matrix-Density-Based-Algorithm-to-Hierarchically-CoCluster-Documents-Words-Mandhani-WWW-2003.pdf

[13] B. D. Bašić, B. Bereček, A. Cvitaš, “Mining Textual Data In Croatian”, *MIPRO, Opatija*, (2005)

[14] J. Gao, J. Zhang, “Clustered SVD Strategies in Latent Semantic Indexing”, *Information Processing and Management Vol. 41, No. 5, pp 1051-1063*, (2005)

http://portal.acm.org/toc.cfm?id=1073691&type=issue