SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5672

Sustav za upravljanje i pretraživanje baze PDF dokumenata

Luka Čupić

Zagreb, svibanj 2018.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

SADRŽAJ

1. Uvod

Područje analize i pretraživanja teksta neizbježno je u današnjem svijetu tehnologije. Od internetskih tražilica koje pretražuju enormne količine podataka baziranih na zadanom upitu, osobnih pomoćnika na pametnim mobitelima koji procesiraju izgovorene riječi pa sve do analize i prepoznavanja *spam* elektroničkih poruka. Kratki osvrt na ove te mnoge druge primjene ukazuju na nepobitnu činjenicu da je pretraživanje teksta...

2. Pregled područja

Korištenje računala za povrat informacija (engl. *information retrieval*) datira čak do 1940-ih godina, daleko prije komercijalizacije računala. Očito je da je to problem...

3. Let's dive right into it...

3.1. Prikaz dokumenata

Prije *poniranja u dubine*, objasnimo prvo što u kontekstu analize i pretraživanja teksta predstavlja dokument. Neformalno dokument možemo definirati kao kolekciju riječi. Ovakva kolekcija ne mora nužno biti skup, pošto dokument može imati više ponavljanja istih riječi (ova će činjenica doći do izražaja u **poglavlju X**). No ipak, za predstavljanje dokumenata biti će korišten tzv. *model vreće riječi* (engl. *bag of words model*) kod kojeg nam nije bitna semantika samih dokumenata, pa čak niti poredak riječi, već je bitna samo činjenica pojavljuju li se riječi u određenom dokumentu, odnosno koja je učestalost njihovog pojavljivanja. Implementacija samog modela zamišljena je tako da se na početku iz svih dokumenata izvade sve riječi te potom uklone nebitne riječi (o kojima će više riječi biti kasnije) a od preostalih se riječi izgradi vektor koji će *de-facto* predstavljati dokument. Najosnovniji prikaz dokumenata jest korištenjem binarne reprezentacije: od svih Ovakav model često je korišten u području procesiranja prirodnog jezika te povrata informacija kako iz dokumenata tako iz drugih izvora tekstualnih informacija.

Da bi se dokumenti mogli predstaviti u obliku vreća riječi, potrebno je odrediti vokabular—skup svih riječi koje se nalaze u svim dokumentima promatrane kolekcije dokumenata (u daljnjem tekstu: zbirka). Iz ovako zadanog vokabulara ćemo ponajprije, ukloniti sve zaustavne riječi (engl. *stop words*)—riječi koje su učestale u nekom jeziku te su stoga nebitne za sam postupak analize teksta. Primjeri nekih zaustavnih riječi u hrvatskom jeziku su: *aha*, *nešto*, *okolo*, *zaboga*. Osim zaustavnih riječi, dodatna obrada teksta može se obaviti tzv. *stemanjem* (engl. *stemming*). Ova metoda ima zadaću svesti riječi na njihov kanonski oblik. Drugim riječima, nebitno je je li riječ napisana u jednini ili množini ili pak u kojem je padežu; bitan je samo kanonski oblik riječi. Na primjer, riječi poput *spavao* i *spavati* svesti će na *spavanje*. Nakon stvaranja vokabulara te predobrade dokumenata (izbacivanje zaustavnih riječi, stemanje) možemo krenuti s predstavljanjem dokumenata. Radi praktičnosti, najčešća metoda

predstavljanja dokumenata jest uz pomoć vektora. Najjednostavnija metoda vektorskog predstavljanja dokumenata jest binarna: za svaku riječ iz vokabulara naprosto provjerimo nalazi li se u danom dokumentu te ukoliko se nalazi, odgovarajuća komponenta vektora (indeks riječi u vokabularu) biti će 1, a u suprotnom 0. Primjerice, ukoliko imamo vokabular V = "Marko", "jako", "voli", "domaćice" te dokument D1 = "Marko", "domaćice", "domaćice", tada će odgovarajući vektor dokumenta D1 biti [1, 0, 0, 1] zbog toga što se samo prva i zadnja riječ dokumenta nalaze u vokabularu. Nadograđujući se na prethodnu metodu, dolazimo do frekvencijskog prikaza vektora. Umjesto obične binarne reprezentacije u kojoj pamtimo samo nalazi li se riječ u dokumentu ili ne, u frekvencijskom prikazu pamtimo i koliko se puta dotična riječ pojavljuje u dokumentu (komponente vektora zapravo su frekvencija (tj. broj) pojavljivanja određene riječi u dokumentu). Gledajući isti vokabular i dokument kao u prethodnom primjeru, novi vektor ovdje bi iznosio [1, 0, 0, 2] obzirom da se riječ "domaćice" u dokumentu pojavljuje dva puta. Naposlijetku dolazimo i do najčešće metode vektorskog prikaza dokumenata—TF-IDF (engl. *term frequency–inverse document frequency*). Ova metoda zasniva se na dvije intuitivne pretpostavke:

- Riječ je važnija za semantiku dokumenta što se češće u njemu pojavljuje (TF komponenta)
- Riječ je manje važna za semantiku dokumenta što se češće pojavljuje u drugim dokumentima (IDF komponenta)

TF i IDF dakle predstavljaju dvije komponente vektora kojima ćemo predstavljati dokumente. Prva komponenta je već spomenuta, frekvencija pojavljivanja riječi *w* u dokumentu *d* ($f_{w,d}$) dok je druga komponenta obrnuta frekvencija pojavljivanja riječi u cijeloj zbirci. Za riječ *w* i dokument *d*, TF i IDF komponene računaju se na sljedeći način:

$$\mathrm{tf}(t, d) = f_{t,d} \tag{3.1}$$

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{3.2}$$

Formula za TF komponentu je intuitivna i trivijalna. Formula za IDF komponentu zahtjeva kratki osvrt: riječ će biti bitnija za neki dokument što se rijeđe pojavljuje u drugim dokumentima. Ovo vidimo u formuli kao omjer ukupnog broja dokumenata (veličine zbirke) *N* te broja dokumenata koji sadrže gledanu riječ. Što je riječ više sadržana u ostalim dokumentima, omjer se smanjuje te riječ postaje manje bitna za neki dokument. Naposlijetku, cijeli se omjer logaritamski skalira kako bi se u smanjio utjecaj velikog i/ili malog broja dokumenata koji sadrže određenu riječ, na vrijednost IDF-a.

3.2. Semantička sličnost dokumenata

Nakon izgrađene vektorske reprezentacije svih dokumenata u korpusu, sljedeći korak jest samo uspoređivanje dokumenata. U sklopu ovog završnog rada, uspoređivanja dokumenata ostvaruje se na dva semantički različita načina: uspoređivanje korisničkog unosa (engl. *user input*, *query*) sa zbirkom dokumenata ili uspoređivanje dokumenta sa zbirkom dokumenata. Ova dva, naizgled različita problema, zapravo se svode na jedan: uspoređivanje kolekcije riječi sa zbirkom dokumenata. Ideja je dakle sljedeća: gleda se koliko riječi (bilo iz korisničkog unosa, bilo iz dokumenta, u daljnjem razmatranju - ulazni vektor) iz ulaznog vektora odgovara riječima iz vokabulara, tj. koliko riječi iz ulaznog vektora odgovaraju riječima iz pojedinih dokumenata u zbirci. Što je veća korespondencija određenog ulaznog vektora s vektorom pojedinog dokumenta (tj. što više riječi dijele zajedno), to kažemo da su ta dva dokumenta sličnija. Primjerice, ukoliko se u zbirci nalazi dokument o srednjevjekovnoj povijesti, a kao ulazni vektor dovedemo neku srednjevjekovnu frazu, to će ulazni vektor i taj dokument imati veću mjeru sličnosti - što nas dovodi do same definicije. Mjeru sličnosti (engl. *document similarity*) definiramo kao vrijednost na skupu pozitivnih realnih brojeva, koja ukazuje na to koliko su dva dokumenta slična — što je brojka veća, dokumenti su sličniji. U nastavku ćemo razmotriti metode mjere sličnosti korištene u ovome radu.

3.2.1. Kosinusna sličnost

Kako su dokumenti zapravo predstavljeni vektorima u više-dimenzijskom prostoru, nad takvim dokumentima (odnosno njihovim vektorima), možemo primijenjivati operacije linearne algebre, odnosno vektorske operacije. Jedna od tih operacija je upravo skalarni produkt dvaju vektora, koja za rezultat daje vrijednost kosinusa kuta između ta dva vektora. Što su dokumenti sličniji, kosinus kuta biti će bliži jedinici, odnosno što su dokumenti različitiji, kosinus kuta biti će bliži nuli. Ovo saznanje možemo iskoristiti u izgradnji sljedećeg modela sličnosti dokumenata. Neka je v_{d_i} vektorska reprezentacija (binarna, frekvencijska ili TF-IDF) dokumenta d_i. Tada se sličnost dvaju dokumenata mjeri kao:

$$similarity(d_i, d_j) = \frac{v_{d_i} \cdot v_{d_j}}{||v_{d_i}|| \cdot ||v_{d_j}||} \tag{3.3}$$

Pošto se skalarni produkt dva vektora svodi na sumu umnožaka pripadajućih komponenti (prva s prvom, druga s drugom itd.), ovo intuitivno možemo zamisliti tako da naprosto zbrajamo "korespondencije" odgovarajućih riječi. Naime, ako se riječ nalazi u oba dokumenta, tada će taj umnožak biti pozitivan te će se pridodati mjeri sličnosti,

odnosno povećati ju. Ako neka riječ ne postoji u dokumentu, tada će taj umnožak biti nula pa će automatski sličnost biti manja.

3.2.2. Okapi BM25

BM25 još je jedna funkcija rangiranja koja, kao i funkcija rangiranja po kosinusnoj sličnosti, ima za zadaću rangirati dokumente po relevantnosti određenom korisničkom upitu (odnosno nekom dokumentu). Za dokument Q, koji sadrži riječi q_1,...,q_n, BM25 mjera sličnosti nekog dokumenta D iz zbirke računa se kao:

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}, \tag{3.4}$$

gdje je $f(q_i, D)$ frekvencija od q_i u dokumentu D, $|D|$ je broj riječi u dokumentu D, a avgdl je prosječan broj riječi u dokumentima iz zbirke. k_1 i b slobodni su parametri koji se uglavnom uzimaju kao $k_1 \in [1.2, 2.0]$ te $b = 0.75$. IDF(q_i) je IDF vrijednost komponente q_i.

4. Programsko rješenje

Cilj ovog rada jest istražiti već ranije spomenute metode analize i pretraživanja teksta. Kako je implementacija programskog rješenja specifična za dokumente tipa PDF, takve dokumente prvo treba preprocesirati kako bi bili spremni za obradu. Programske potpora kao implementacija problema ovog završnog rada napisana je u programskom jeziku Java. Razlog ovakvog odabira leži u tome što je Java popularan objektno orijentirani jezik po čemu je idealan za rješavanje problema poput DOCUMENT RETREIVAL-a zbog svoje nativne podrške apstraktnih kolekcija podataka, podrške raznih biblioteka i sl. U svrhu preprocesiranja PDF dokumenata, koristi se Apache PDFBox koji omogućava brzo i jednostavno izvlačenje teksta iz PDF dokumenata. Kao algoritam za stemanje riječi koristi se poznati 'Portland Stemming Algorithm' čije su implementacije javno dostupne u većini popularnijih programskih jezika, pa tako i u Javi.

5. Zaključak

Zaključak.

Sustav za upravljanje i pretraživanje baze PDF dokumenata

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.