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 predstavlja dokument. Neformalno dokument možemo definirati kao kolekciju riječi. Ovakva kolekcija riječi ne mora nužno biti skup, pošto dokument može imati više ponavljanja istih riječi; ova će činjenica doći do izražaja u **poglavlju X**. Umjesto toga, pretpostavljamo da se dokument sastoji od tzv. *vreće riječi* (engl. *bag of words*) kod koje nam nije bitna semantika samog dokumenta, pa čak niti poredak riječi, već je bitna samo činjenica da se riječi pojavljuju u dokumentu, odnosno učestalost njihovog pojavljivanja. 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. 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). 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 broja dokumenata i/ili malog broja dokumenata koji sadrže određenu riječ, na vrijednost IDF-a.

3.2. Semantička sličnost dokumenata

Nakon izgrađene vektorske reprezentacije dokumenata, sljedeći korak jest samo uspoređivanje dokumenata. Uspoređivanje se može ostvariti na dva različita načina: uspoređivanje korisničkog unosa (engl. *user input*, *query*) sa zbirkom dokumenata ili uspoređivanje dokumenata međusobno.

3.2.1. Semantička sličnost dokumenata

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco...

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.