Unapređivanje LSI metode za sumiranje teksta

Apstrakt:

Veoma je lako izgubiti se u velikoj količini teksta, automatsko pronalaženje ključnih podataka može pomoći u lakšem snalaženju i savladavanju teksta. Ovaj rad se bavi unapređivanjem LSI metoda koja koristi razlaganje na singularne vrednosti kako bi izdvojila najbitnije rečenice. Unapređenja se vrše promenom matrice iz matrice broja reči po rečenici u tf-idf matricu, promenom formule po kojej se tekst deli na paragrafe i korišćenjem lematizacije.

Uvod:

Ljudi se svakodnevno sreću sa velikom količinom informacija u obliku teksta. Vrlo često nisu sve informacije u nekom tekstu bitne ili ne znamo da li nas taj tekst zanima dovoljno da uložimo više vremena u njegovo čitanje. Ovaj problem može da se reši time što umesto celog teksta dobijemo skraćenu verziju sa ključnim rečenicama[1]. Jedan od načina je sumiranje korišćenjem LSI[1] (Latent semantic indexing) metode. LSI radi tako što napravi korelacionu matricu reči i rečenica. Svaka kolona predstavlja jednu rečenicu, a svako polje u toj koloni broj pojavljivanja neke reči u toj rečenici. Nakon toga se na toj matrici radi redukcija dimenzionalnosti korišćenjem matematičke metode SVD, dekompozicija na singularne vrednosti. Singularne vrednosti su koreni sopstvenih vrednosti matrice X.T*X, gde je X matrica koja se rastavlja uz pomoć SVD-a. Razlog za redukciju dimenzionalnosti je prebacivanje podataka u prostor definisan manjim brojem baznih vektora kako bi bilo lakše da se nađu veze između podataka. Ovaj rad se bavi unapređivanjem LSI metode i testira unapređenje na tekstovima koji su na srpskom jeziku. Metod je nadogradjen korakom obrade podataka koji se sprovodi pre upotrebe LSI metode, lematizovanje reči, korišćenjem tf-idf (term frequency inverse document frequency) metrike umesto brojanja reči. Unapređivanje sadrži i korak nakon LSI metode koji se bavi drugačijom podelom na paragrafe.

metode:

LSI:

Prvi korak je formiranje matrice od teksta. Svaka kolona je jedna rečenica, a svako polje u koloni je broj ponavljanja neke reči u toj rečenici. Nakon toga se na datoj matrici koristi SVD (Singular value decomposition, dekompozicija na singularne vrednosti). Rezultat SVD-a su tri matrice T_0, S_0 i D_0 takve da $X=T_0*S_0*D_0.T$. Ako je originalna matrica X dimenzija nxm T_0 će biti dimenzija nxn S_0 će biti dimenzija nxm, a D_0 će biti dimenzija mxm. Matrice T_0 i D_0 su ortogonalne $T_0*T_0.T = I, D_0*D_0.T = I$. Matrica S_0 je dijagonalna matrica kojoj su vrednosti na dijagonali opadajuće. Pošto je svaka vrsta posle m-te u matrici S_0 nula onda je i svaka kolona posle m-te u matrici T_0 beskorisna u proizvodu jer će se svaki član iz tih kolona množiti sa nulom u matričnom proizvodu. Drugi korak je da se matrici S_0 uklone sve vrste posle m-te i

matirci T_0 sve kolone posle m-te. Uklanjanjem poslednje kolone ili vrste je kad se od neke matrice dobije nova kojoj je manja druga ili prva dimenzija, ako je bila dimenzije (n,m) postaće dimenzije (n,m-1) ili (n-1,m), svi elementi u matrici ostaju isti i izbacuju se samo poslednja kolona ili vrsta. Nakon ovog koraka proizvod ove tri matrice je i dalje originalna matrica X. Sledeći korak je redukcija dimenzionalnosti. Redukcija dimenzionalnosti se radi tako što se sa matrice S_0 dalje skidaju po jedna vrsta i kolona sve dok je odnos zbira elemenata na dijagonali nakon skidanja i zbira svih elemenata na dijagonali pre skidanja veći od neke odabrane vrednosti.

$$\frac{\sum_{i}^{m-k} S_0(i,i)}{\sum_{i}^{m} S_0(i,i)} > \alpha \quad (1)$$

U formuli (1) k je broj vrsta i kolona koje treba da se skinu sa matrice S_0, a α je prag u odnosu na koji se bira k. Sa matrice T_0 se skine isti broj poslednjih kolona, a sa matrice D_0 se skine isti broj poslednjih vrsta. Nakon ovog koraka dobijamo matrice T, S i D. Ove tri matrice su redom dimenzija nxh, hxh i hxm, gde je h=m-broj kolona/vrsta skinutih sa matrice S_0. Nova matrica $\hat{X}$ = T*S*D.T je istih dimenzija kao matrica X ali ima manji rang. Iz ove matrice možemo da dobijemo matricu sličnosti rečenica. Matrica sličnosti rečenica je $\hat{X}.T * \hat{X}$ = D*S*S*D.T. Sličnost između rečenice j i rečenice i se gleda u polju sa koordinatama (j,i). Od dijagonale koja je tačno iznad glavne (0,1)->(n-1, n) napravimo niz, u tom nizu se nalaze sličnosti susednih rečenica. Za svaki lokalni minimum m u tom nizu se nađe prva leva veća vrednost Lv i prva veća desna vrednost Dv. Lokalni minimum m je element u nizu gde i element pre njega i element posle njega imaju veću vrednost od njega samog. Onda se izračuna dubina po formuli (2)

$$D = (Lv+Dv)/(2*m)-1 \quad (2)$$

Svuda gde je dubina veća od neke određene vrednosti σ se postavi granica za paragraf. Parametar σ se bira tako što se izvuče njegova najoptimalnija vrednost korišćenjem tekstova za koje je poznat broj paragrafa, uzimanjem aritmetičke sredine vrednosti koje najpreciznije dele tekst na paragrafe. Pretpostavi se da će dubina biti najveća baš na mestima gde treba da bude podela na paragrafe.

$$\sigma = \frac{depth[n-1]+depth[n-2]}{2} \quad (3)$$

U jednčini (3) je formula po kojoj se računaσ, depth je niz sortiranih dubina koje su izračunate na nekom tekstu, a n broj paragrafa. Ovo izračunavanje se vrši više puta za tekstove sa različitim brojem paragrafa, na kraju se za vrednost σ uzima aritmetička sredina svih rezultata. Dalje se za svaku rečenicu računa njena ukupna sličnost, saberu se sličnosti te rečenice sa svakom.

$$sim_k = \sum_{i}^{m} sim(k,i) \quad (4)$$

Gde je k indeks rečenice za koju se računa ukupna sličnost, sim(k,i) je vrednost polja (k,i) u matrici sličnosti $\hat{X}.T * \hat{X}$. Na samom kraju se bira najboljih n% rečenica iz svakog paragrafa, dužina sumiranog teksta je n% dužine originalnog teksta.

LSImproved:

Prvi nivo unapređenja je korišćenje tfidf metrike umesto samo broja reči. Predpostavka je da će ovo poboljšati rezultate jer nemaju sve reči istu vrednost za tekst. Ako se koristi samo broj reči za formiranje matrice X svaka reč je podjednako bitna, a korišćenjem tfidf metrike reči koje se često ponavljanju poput veznika i nekih dopunskih reči imaju manju vrednost što nam odgovara jer su manje relevantne za samo značenje rečenice.

Drugi nivo unapređenja je lematizacija. Pošto reči u srpskom jeziku mogu da imaju puno oblika, padeži za imenice ili vremena za glagole, neke dve rečenice mogu da sadrže istu reč u drugom obliku ali da metod koji se koristi u referentnom radu to ni ne primeti. Lematizacijom se rešava taj problem i zbog toga će sve rečenice koje sadrže iste reči u drugim oblicima imati veću međusobnu sličnost nego pre lematizacije, što je i potrebno. Za lematizaciju se koristi biblioteka "classla".

Treći nivo unapređenja je promena funkcije koja deli tekst po paragrafima. Nova funkcija umesto da u nizu sličnosti susednih rečenica upoređuje lokalni minimum sa najbližim vrećim vrednostima upoređuje ga sa najbližim lokalnim maksimumima. Lv i Dv su najbliži lokani maksimumi, vrednosti elemenata niza koji imaju veću vrednost od oba susedna elementa. Oznaka za lokalni minimum ostaje m.

$$D = (Lv+Dv)/(1-2*m) \quad (4)$$

Pošto je vrednost lokalnog minimuma često nula ili jako blizu nuli formula je dodatno promenjena.

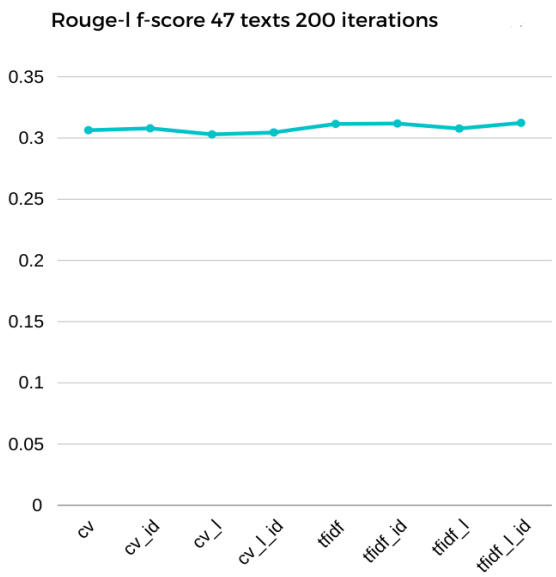
Rezultati:

Testiranje koda je izvršeno na tekstovima koji se sastoje od više spojenih apstrakata.

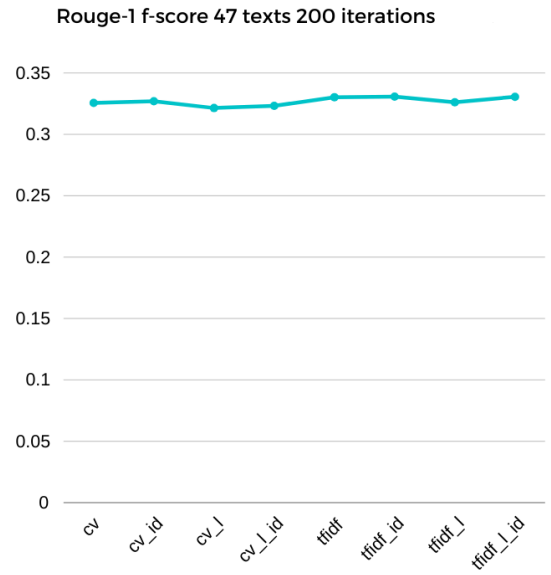
cv->count vectorizer

_id->korišćena je nova funkcija za izračunavanje dubine

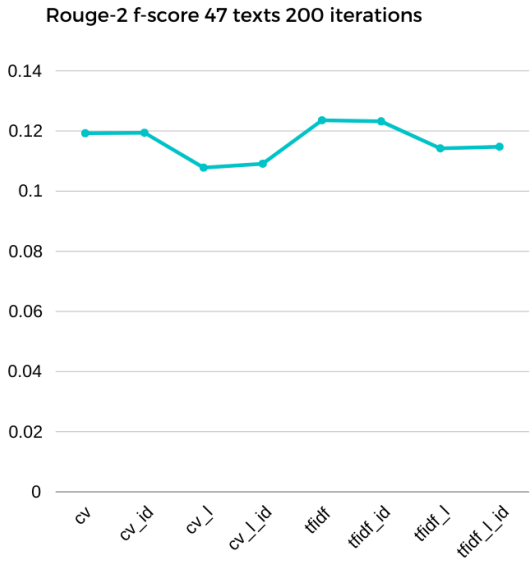
_l->korišćena je lematizacija



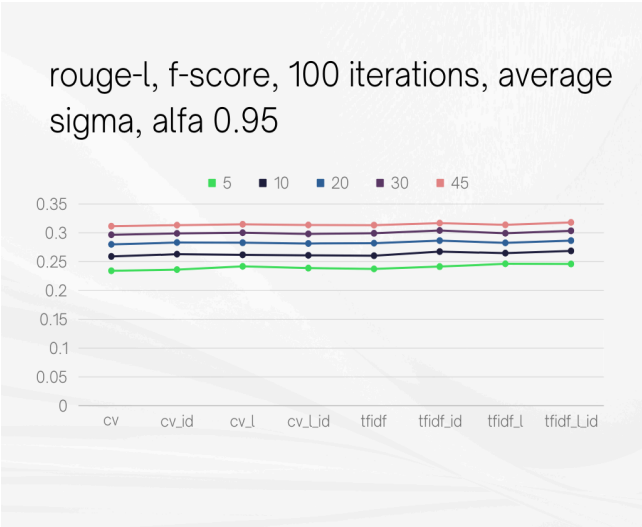
Vrednosti za σ su 1 i 0.4 za staru i novu funkciju izračunavanja dubine redom.



Vrednosti za σ su 1 i 0.4 za staru i novu funkciju izračunavanja dubine redom.



Vrednosti za σ su 1 i 0.4 za staru i novu funkciju izračunavanja dubine redom.



Ovde je sigma dobijeno korišćenjem metode navedene ranije

.

Reference:

[1] Dongmei Ai · Yuchao Zheng · Dezheng Zhang, Automatic text summarization based on latent semantic indexing