Klasifikacija tekstova temeljena na skupovima riječi

Sadržaj

1. UVOD

Model zbirki značajki se koristi u strojnom učenju. Koristi se u Naive Bayes spam filtering...

2. MODEL ZBIRKI ZNAČAJKI

Model zbirki značajki je način izvlačenja značajki iz nekog sadržaja s ciljem sažimanja i oblikovanja tog sadržaja. U kontekstu ovog rada značajke su riječi te njihova frekvencija pojavljivanja, a sadržaj je tekst. Model ne očuvaje redoslijed riječi što bi se moglo zaključiti iz njegovog engleskog naziva: „Bag of words model“ - riječi su „smještene u vreću“, tj. gubi se njihov poredak. Umjesto pojedinačnih riječi može se brojati i učestalost skupa susjednih riječi, takozvanih tokena ili n-torki. Tokeni i njihov broj ponavljanja za svaki pojedinačni tekst spremljeni su u strukturu podataka zvanu rječnik, gdje su ključevi tokeni a vrijednosti broj ponavljanja tog tokena. Kod modela zbirki značajki taj rječnik se zove vektor(primjer vektora slika).

```
tekst = "Marko je išao igrati nogomet. Išao je i David."

vektor = {'marko': 1, 'je': 2, 'išao': 2, 'igrati': 1, 'nogomet': 1, 'i': 1, 'david': 1}
```

3. PROGRAM

Cilj programa je sa što većom preciznosti klasificirati uneseni tekst, tj. odrediti kojoj od zadanih tema taj tekst najviše odgovara. To radi tako da uspoređuje riječi unesenog teksta sa riječima već poznatih tekstova kojima program unaprijed zna temu. Kroz par distinktivnih metoda klasifikacije pokušano je maksimizirati preciznost.

Za stvaranje vektora vokabulara (korpus?) sam koristio internet članke dnevnog časopisa „24sata“. U početku je bilo planirano da program može primiti članke iz više različitih časopisa, no bilo je teško ukloniti irelevantne dijelove web-stranice jer svaki časopis ima svoj dizajn „HTML“ koda. Jer ti irelevantni dijelovi web-stranice sadrže riječi/rečenice koje nisu povezane s kontekstom samog članka, bilo ih je potrebno ukloniti kako ne bi uništili vektor te kategorije. Primjeri takvih dijelova su izbornici na vrhu s nazivima kategorija, nepovezani predloženi članci i reklame duž te informacije o kompaniji na dnu web-stranice. U slučaju da se dizajn „HTML“ koda članka časopisa „24sata“ promijeni ili se članak unese s nekog drugog časopisa, tipa „Novi list“, sav tekst koji se vidi na stranici bez obzira na povezanost s temom članka, će biti unesen u vektor. To posebno dolazi kod izražaja kod metoda koje su usredotočene na tokene čija je učestalost pojavljivanja manja.

Program razlikuje šest kategorija tema članka: filmovi/serije, glazba, gospodarstvo, politika, sport i tehnologija. Svaka kategorija ima svoju tekstualnu datoteku koja sadrži 20? poveznica koje su ručno unesene gledajući kako je koji članak kategoriziran na web-stranici časopisa.

~~Program klasificira temu zadanog teksta (ručno unesenog ili dohvaćenog putem poveznice) u šest različitih kategorija~~

Poveznice, bile one unesene od strane korisnika ili uzete iz tekstualnih datoteka kategorija, se obrađuju tako da se prvo preuzme „HTML“ kod web-stranice kojoj ta poveznica vodi. Iz tog koda se uklanjaju svi znakovi unutar „<style>“ i „<script>“ „HTML“ oznaki (eng. tag) te pronađeni zakomentirani kod. Također uklanaju se i oznake „<a>“, „“, „<b>“, „<i>“, „<sup>“, „“, „ “, ali ne i tekst unutar njih. (kako su obrađeni HTML kodovi 24sata?). Zamijenjeni su i „ “... znakovi. Rezultat je čitljivi tekst isključivo vezan uz samu temu članka. Takvom tekstu ili ako korisnik za klasifikaciju nije unio poveznicu nego tekst se sva slova pretvaraju u mala slova te su rečenice podjeljene u svoje redove, koristeći interpukcijske znakove kao separatore.

To se dalje obrađuje ovisno o metodi klasifikacije koja se koristi. Iz tih metoda nastaju vektori te ako se radi o vektorima vokabulara iste kategorije oni se povezuju unijom(primjer slika).

Načini uspoređivanja baznih vektora i testnih vektora, tj. način određivanja preciznosti je .... „ifidf“(poveznica prema poglavlju) ima svoj način jer je ...

Nakon izračuna preciznosti klasifikacije, automatski se izradi tekstualna datoteka(slika tekstualne datoteke) koja sadrži koja su preklapanja i koliko ih je – za svaku kategoriju. Pregledavanjem sadržaja datoteke može se zaključiti zašto program odluči da je jedna kategorija zastupljenija od druge, te će biti korištena za objašnjavanje dobivenih rezultata kroz cijeli rad.

Jer je korišten programski jezik Python, ovaj cijeli proces je nešto sporiji kad bi ga usporedili s nekim drugim, ali zato ima puno jednostavniju sintaksu te je lakše rukovati s *string-ovima*.

4. METODE KLASIFIKACIJE

U nastavku su navedene metode s kojima je pokušano povećati preciznost klasifikacije. Za uspredbu preciznosti metoda klasifikacije svakom metodom je klasificiran po jedan hrvatski Wikipedija članak iz svake od ukupno šest kategorija te su rezultati prikazani u tablicama. Za kategoriju sport je izabran članak Luke Modrića, za gospodarstvo „Turizam u Hrvatskoj“, za politiku članak Zorana Milanovića...(poveznice za te članke)

4.1. N-torke

N-torka je konačan niz n objekata, a ovdje ona predstavlja niz susjednih riječi s ciljem očuvanja dodatnih informacija iz članka. Najjednostavnija metoda je 1-torka na koje se i otale metode nadograđuju ~~Uklonjene su sve „riječi“ koje se ne sastoje isključivo od znakova abecede.~~ Za n-torke pretpostavka je da će što je veći *n* biti manja preciznost zbog jako malog broja matches.

1-torka je u prijevodu samo jedna riječ i predstavlja najjednostavniju i potencijalno najgoru? od metoda klasifikacije.

(objašnjenje tablice)

(tablica sa rezultatima)

2-torka, dvojka

(objašnjenje tablice)

(tablica sa rezultatima)

3-torka, trojka

(objašnjenje tablice)

(tablica sa rezultatima)

4-torka, četvorka?

(objašnjenje tablice)

(tablica sa rezultatima)

Preostale metode se oslanjaju na 1-torku, a n-torke se opet pojavljuju kod kombinacija metoda(poveznica na poglavlje?)

4.2. Uklanjanje nebitnih riječi

Pod nebitnim riječima smatraju se riječi iz kojih ne bih mogli razaznati o čemu se piše u nekom tekstu, tj. koja je njegova tema. Za uklanjanje takvih riječi napravljena je tekstualna datoteku koja sadrži što više zamjenica, brojeva, priloga, prijedloga, veznika, čestica i usklika. Tekstualna datoteka je ručno pregledana te su izostavljene riječi koje bi ipak mogle imati kontekstualnu važnost. Jedan primjer je riječ „oko“ koja je i prijedlog ali i imenica. Pretpostavka je da ova metode neće imali preveliki utjecaj na preciznost klasifikacije jer će dva dovoljno duga teksta različitih tema vjerojatno imati slične „nebitne riječi“, koje bi se u vektorima „poništile“.

(objašnjenje tablice)

(tablica sa rezultatima)

4.3. Korijenski algoritmi

Ova metoda pokušava povezati riječi koje nemaju isti oblik, jer su u drugom padežu ili glagolskom vremenu, ali imaju isto značenje. Mnogi od pronađenih korijenskih algoritama su fokusirani na engleski jezik te nisam našao dovoljno materijala da bi ih mogao prilagoditi hrvatskome jeziku.

4.3.1. Prefiks-sufiks algoritam

Uklanjanje sufiksa je puno jednostavnije u engleskom jeziku zbog njihovog malenog broja, uvelike zbog nedostatka padeža te načina oblikovanja glagolskih vremena(ok?, usporedi s hrvatskim i ostalim jezicima). Sufiksi imenica, pridjeva i glagola skupljeni su u jednu tekstualnu datoteku te ako su pronađeni na kraju bilo koje riječi, uklonjeni su. Na isti način, lista prefiksa

pronađena na Wikipediji(izvor), prebačena je u svoju tekstualnu datoteku. Prefiksi i sufiksi su sortirani po duljini kako bi se osiguralo da se prvo ukloni najduži mogući dio riječi. (primjer?)

(objašnjenje tablice)

(tablica sa rezultatima)

Kod sufiks-prefiks algoritma neophodno je prvo uklanjanje nebitnih riječi zbog njihove kratke duljine. Tokeni stvoreni iz veznika pa, te, ni, ali... postaju potpuno beznačajni kad im se uklone sufiksi a, e te i – česti sufiksi deklinacije većine imenica. Iako uklanjanje nebitnih riječi pomaže u preciznosti klasifikacije, ne rješava u potpunosti problem preagresivnog uklanjanja sufiksa.

(objašnjenje tablice)

(tablica sa rezultatima)

Sufiks „ama“ se dodaje na korijen imenica ženskog roda u dativu, lokativu i instrumentalu množine (e-sklonidba) te ga napravljena tekstualna datoteka sufiksa sadrži. Problem stvaraju riječi koje završavaju na „ama“ ali nisu u navedenim padežima. Mama, jama, slama, tama i dama su riječi u nominativu jednine te uklanjanjem sufiksa „ama“ dobivamo jednoslovne tokene koji su potpuno neupotrebljivi. Ovo je samo jedan od mnogih primjera preagresivnog uklanjanja sufiksa. Način na koji bi se ovaj problem mogao rješiti je da se uklanjanje sufiksa dopusti jedino ako je duljina sufiksa (broj slova sufiksa) manja ili jednaka polovici duljine cijele riječi (broja slova cijele riječi). Dodatno, „korijenovanje“ bi se mogao zabraniti ako je token koji nastane prekratak – u sljedećoj tablici(poveznica do tablice) prikazani su rezultati klasifikacije token je prekratak ako je duljine tri slova ili manje. U sljedećoj tablici(pointer) prikazani su rezultati klasifikacije

(objašnjenje tablice)

(tablica sa rezultatima)

4.3.2. nenazvani korijenski algoritam (koristeći bazu podataka hrvatskih imenica i glagola)

cro-dict opis

(objašnjenje tablice)

(tablica sa rezultatima)

„Wječnik“ (poveznica) je internet rječnik zasnovan na dobrovoljnim dodacima i izmjenama korisnika te je „sestrinski projekt Wikipedije“(citat). Sadrži više od deset tisuća riječi (citat) – većinom imenice. Velik broj web-stranica tih riječi uopće ne sadrži njihovu deklinaciju te uspoređujući to s toliko (izvor) riječi koju hrvatski jezik zapravo ima. Iako taj rječnik sadrži samo frakciju pravog broja rječi ne bi ga bilo loše implemenirati. Problem je što...

4.4. „tf-idf“

što je...

Jer metoda „tf-idf“ prebacuje važnost s tokena koji se ponavljaju puno na tokene koji se ponavljaju malo nebitne riječi koje se nalaze u headeru i footeru će poremetiti preciznost...

(objašnjenje tablice)

(tablica sa rezultatima)

5. KOMBINACIJE METODA

(najbolje kombinacije te njihove tablice s rezultatima)

6. ZAKLJUČAK