Klasifikacija tekstova temeljena

na skupovima riječi

PRILOG 2

ORIGINAL ZADATKA

POTPISANA IZJAVA

ZAHVALA

Sadržaj

[1. UVOD 6](#_Toc144143844)

[2. MODEL ZBIRKI ZNAČAJKI 7](#_Toc144143845)

[3. PROGRAM 8](#_Toc144143846)

[4. OBRADA DOKUMENTA 10](#_Toc144143847)

[4.1. Izdvajanje sadržaja članka 10](#_Toc144143848)

[4.2. Obrada preostalog HTML koda 12](#_Toc144143849)

[5. METODE STVARANJA N-TORKI 13](#_Toc144143850)

[5.1. Uklanjanje nebitnih riječi 13](#_Toc144143851)

[5.2. Automatsko određivanje korijena riječi 14](#_Toc144143852)

[5.2.1. Prefiks-sufiks algoritam 14](#_Toc144143853)

[5.2.2. Određivanje korijena riječi korištenjem baze podataka 15](#_Toc144143854)

[6. BODOVANJE N-TORKI I IZRAČUN PRECIZNOSTI 17](#_Toc144143855)

[6.1. Binarno bodovanje 18](#_Toc144143856)

[6.2. Broj ponavljanja 19](#_Toc144143857)

[6.3. „tf-idf“ 20](#_Toc144143858)

[7. REZULTATI (USPOREDBE PRECIZNOSTI) 21](#_Toc144143859)

[7.4. Obične n-torke 21](#_Toc144143860)

[8. ZAKLJUČAK 28](#_Toc144143861)

1. UVOD

„Bag of words model“ - prevedeno „model temeljen na zbirkama riječi“ ili opširnije „model zbirki značajki“ se koristi u obradi prirodnog jezika i kod sustava za povrat informacija [1]. Često je rješenje problema klasifikacije dokumenta - pridodavanja nekakve kategorije tekstu na temelju njegovog sadržaja [2]. Klasifikacija dokumenata se koristi, među ostalima, u filtriranju neželjene e-pošte, jezične identifikacije, bržeg i lakšeg procesiranja upitnika korisničke podrške. Naprimjer, ako su u primljenoj e-mail poruci česti izrazi poput „laka zarada“, „brzo mršavljenje“ ili „prestanak pušenja“ vrlo je vjerojatno da je sadržaj te e-mail poruke prijevara. Na sličan način, ako se u upitniku korisnika pojavljuju riječi „prijava“, „lozinka“, „zaboravljena“... sistem može automatski poslati odgovor koji sadrži upute za ponovno postavljanje lozinke.

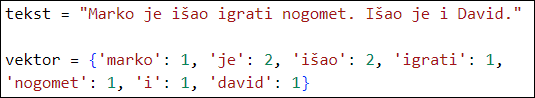
~~Na osnovu ovog modela napravljeni su modeli koji se mogu koristiti i u računalnom vidu kod obrade slika [3]. Umjesto riječi, značajke su objekti na slici koji su nastali povezivanjem piksela sličnih boja, uzoraka koji se ponavljaju ili oblika koji su nastali prateći linije između različitih boja. Razmatranjem broja pronađenih objekata i njihovih relativnih položaja na slici, klasifikator može odrediti sadržaj slike. Neke od uporaba računalnog vida su prepoznavanje okoliša kod video nadzora (Slika 1.1.) i autonomnih vozila, automatska inspekcija kvalitete proizvoda u procesu proizvodnje te skeniranje tekstova u fizičkom obliku za spremanje u digitalnom obliku [4].~~

~~~~

*~~Slika 1.1. Detekcija objekata koristeći računalni vid [5]~~*

1. JEZIČNI MODEL TEMELJEN NA ZBIRKAMA RIJEČI

Jezični model temeljen na zbirkama riječi je način izvlačenja značajki iz nekog sadržaja s ciljem sažimanja i klasificiranja tog sadržaja [6]. U kontekstu ovog rada značajke su riječi i vrijednost koja na neki način predstavlja njihovu brojnost, 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 n-torki ili *tokena*. N-torke za svaki pojedinačni tekst su spremljene u strukturu podataka zvanu rječnik, gdje je ključ n-torka, a vrijednost je iznos pridodan toj n-torki ovisno o metodi bodovanja. Kod modela zbirki značajki taj rječnik se zove vektor (Slika 2.1.).



*Slika 2.1. Vektor nastao iz teksta, gdje su riječi bodovane po njihovom ponavljanju*

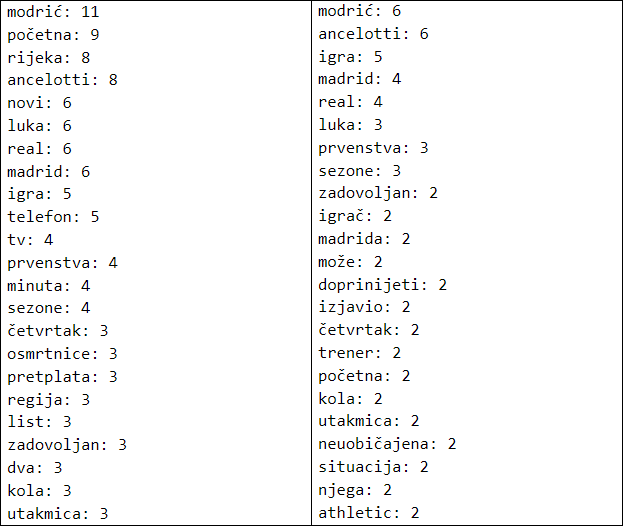
Postoje različite metode ignoriranja irelevantnih n-torki te spajanja sličnih n-torki uklanjanjem prefiksa i sufiksa ili korištenjem skupa podataka gdje su slične riječi već povezane. Također postoje i različiti načini za bodovanje n-torki koji su izračunati iz njihovog broja ponavljanja u tekstu. Metode iz tih dviju kategorija će biti objašnjene te će kasnije biti kombinirane i testirane s ciljem stvaranja najpreciznijeg klasifikatora.

1. OBRADA ČLANKA

Za stvaranje vektora klasifikatora korišteni su članci portala 24sata [8] i Novi list [9]. Za korisnikov unos kod klasifikacije članka je predviđeno korištenje hrvatskih članaka Wikipedije [10]. U početku je bilo planirano da program može primiti članke s više website, no bilo je teško ukloniti irelevantne dijelove webpage jer svaki website ima svoj dizajn HTML koda. Zbog tog razloga i jer je kod njih odvajanje teksta članka bilo nešto jednostavnije uzeti su 24sata, Novi list i Wikipedija.

* 1. Izdvajanje sadržaja članka

Jer irelevantni dijelovi webpage sadrže riječi/rečenice koje nisu povezane s kontekstom samog članka, bilo ih je potrebno ukloniti kako ne bi poremetili klasifikaciju. To posebno dolazi do izražaja kod metoda bodovanja koje su usredotočene na n-torke čija je učestalost pojavljivanja manja. Primjeri takvih dijelova su izbornici na vrhu s nazivima kategorija, nepovezani predloženi članci i reklame duž i informacije o vlasniku na dnu webpage. U lijevoj listi riječi na Slici 4.1. vidi se zašto je izdvajač teksta članka potreban. Riječi poput „novi“, „telefon“, „osmrtnice“ i „pretplata“ nemaju veze sa samim člankom nego se odnose na irelevantne elemente webpage.



*Slika 4.1. Najčešće riječi (isključujući veznike) sportskog članka portala „Novi list“ bez (lijevo) i sa izdvajačem sadržaja (desno)*

Ako se radi o HTML kodu webpage 24sata, Novog lista ili Wikipedije traži se *<div>* oznaka sa jedinstvenom *class* ili *id* vrijednosti koja je ručno pronađena. U slučaju promjene dizajna ili nekih pojedinosti HTML koda webpage, ovaj izdvajač sadržaja neće moći pronaći tekst članka, stoga nije dugoročno riješenje. Unutar parova oznaka, *<div>* i *</div>*, traže se *<p>* oznake koje označavaju paragrafe. Sva tri website u svoje webpage stavljaju tekst vezan uz članak unutar tih oznaka. Između *<p>* i *</p>* parova još uvijek nije čisti tekst nego postoje razne oznake za njegovo modificiranje: podebljano, nakoso, podcrtano... U sljedećem potpoglavlju opisano je kako se HTML kod između „paragraf“ oznaka dalje obrađuje.

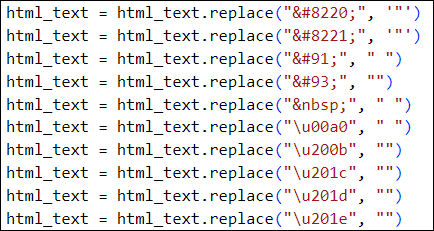
* 1. Obrada preostalog HTML koda

Ako poveznica koja se želi obraditi nije sa 24sata, Novog lista ili Wikipedije, kod joj se preuzima te se prvo uklanjanju *<style>* i *<script>* oznake i svi znakovi između njih te pronađeni komentari (komentari u kontekstu koda). Kao i u paragrafu prije, preostaje tekst sa raznim modifikatorima.

HTML kodu sa modifikatorima uklanjaju se oznake *<a>*, *<span>*, *<b>*, *<i>*, *<sup>*, *<strong>*, *<div>*, ali ne i tekst unutar njih. U HTML jeziku postoje različiti kodovi/*string*-ovi koji zamijenjuju znakove koji bi mogli poremetiti interpretaciju koda. Takvi kodovi su ručno pronađeni prolazeći kroz članke klasifikatora i nađeni su im pripadajući ispravni znakovi. Na Slici 4.2. su prikazani neki od tih kodova te znakovi s kojima su zamijenjeni koristeći Python metodu *replace()* koja se poziva na *string* objektima. Prvi parametar je *string* koji se traži, a drugi parametar je *string* s kojim se prvi zamijenjuje.

Rezultat je čitljivi tekst isključivo vezan uz temu članka. Takvom tekstu se sva slova pretvaraju u mala slova te su rečenice podjeljene u svoje redove, koristeći interpukcijske znakove kao separatore. Svaki red predstavlja „jednostavnu“ rečenicu iz koje se lako stvaraju n‑torke. Proces je isti ako je korisnik unio svoj proizvoljni tekst – sva se slova pretvaraju u mala te se rečenice odvajaju.

AKO JE PROMENJEN ČLANAK KLASIFIKATORA TU SE SPREMA I KAO STO JE PRIJE SPOMENUTO PRVO KLASIFICIRANJE CE DUZE POTRAJAT?? OVO MOZDA POTPUNO IGNORIRAT OVISI O BROJU STRANICA



*Slika 4.2. Zamjena HTML kodova sa ispravnim znakovima*

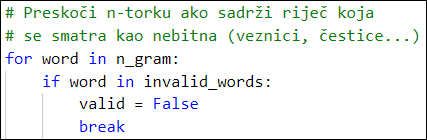
1. METODE STVARANJA N-TORKI

N-torka je konačan niz *n* objekata, a ovdje ona predstavlja niz susjednih riječi. s ciljem očuvanja dodatnih informacija KOJIH - IZRISI AKO IMA DOVOLJNO STRANICA iz članka. Čine osnovu vektora klasifikatora i testnih vektora. Čisti tekst iz kraja prošlog poglavlja, nakon što je u potpunosti obrađen, se dijeli na n-torke. Svaka n-torka koja se u potpunosti ne sastoji od znakova abecede je odbačena. Za n-torke pretpostavka je da će za velike *n* biti lošija preciznost klasifikatora jer su manje šanse da će se isti skup većeg broja susjednih riječi ponoviti. (SLIKA 3-torke?)

U nastavku su navedene metode s kojima je pokušano ili ukloniti nebitne n-torke ili spojiti n-torke koje nemaju isti oblik, ali imaju isto značenje.

* 1. Uklanjanje nebitnih riječi

Pod nebitnim riječima smatraju se one riječi iz kojih se ne bih moglo razaznati o čemu se piše u nekom tekstu, tj. koja je njegova tema. Za uklanjanje takvih riječi napravljena je tekstualna datoteka 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 toga 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 klasifikatora „poništile“. Ako je u n-torci pronađena nebitna riječ, n-torka se preskače (Slika 5.1.).



*Slika 5.1. Izbacivanje n-torke ako sadrži nebitnu riječ („invalid\_words“ je lista koja sadrži sve nebitne riječi)*

* 1. Automatsko određivanje korijena riječi

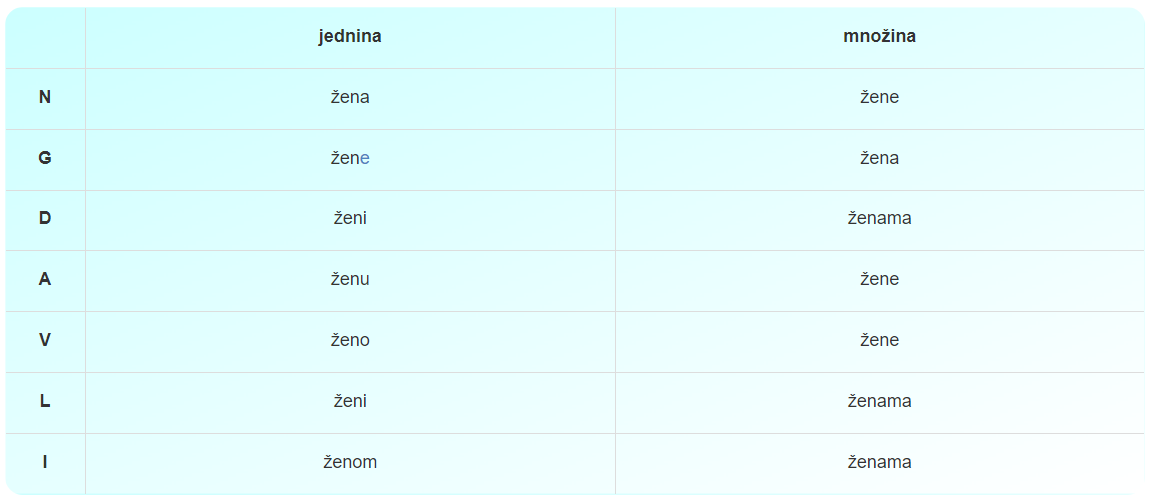
Ova metoda pokušava algoritmima povezati riječi koje nemaju isti oblik, jer su u drugom padežu ili glagolskom vremenu, ali imaju isto značenje, tj. predstavljaju isti pojam. Mnogi od pronađenih algoritama su fokusirani na engleski jezik te ih je bilo vrlo složeno ili nemoguće prilagoditi hrvatskome jeziku.

* + 1. Prefiks-sufiks algoritam

Uspoređujući hrvatsku i englesku gramatiku, može se primijetiti da bi uklanjanje sufiksa u hrvatskom bilo puno složenije. U usporedbi s drugim jezicima hrvatski ima velik broj različitih sufiksa, uvelike zbog postojanja padežaUklanjanje 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)AKO IMA DOVOLJNO TEKSTA FUCK THIS. Sufiksi imenica, pridjeva i glagola prikupljeni su s website „Hrvatske školske gramatike“ [11] u jednu tekstualnu datoteku i ako su pronađeni na kraju bilo koje riječi, uklonjeni su. Na isti način uklonjeni su i prefiksi čija je lista pronađena na Wikipediji [12]. Prefiksi i sufiksi su sortirani po duljini kako bi se osiguralo da se ukloni najduži mogući dio riječi. (SLIKA AKO TREBA MJESTA?). Prefiksi i sufiksi su pronađeni koristeći Python metode *startswith(prefix)* i *endswith(suffix)* te uklonjeni metodama *removeprefix(prefix)* i *removesuffix(suffix)* koje se pozivaju na *string* objektima.

Kod prefiks-sufiks algoritama skoro je neophodno uklanjanje nebitnih riječi zbog njihove kratke duljine. N-torke stvorene iz veznika „pa“, „te“, „ni“, „ali“... postaju potpuno beznačajne kad im se uklone sufiksi „a“, „e“ te „i“ – česti sufiksi deklinacije većine imenica. Iako uklanjanje nebitnih riječi pomaže u preciznosti klasifikacije, ne rješava problem preagresivnog uklanjanja sufiksa.

Sufiks „ama“ se dodaje na korijen imenica ženskog roda u dativu, lokativu i instrumentalu množine (e-sklonidba, Slika 5.2.) 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“ dobiju se jednoslovne n-torke koje su potpuno neupotrebljive. Ovo je samo jedan od mnogih primjera preagresivnog uklanjanja sufiksa. Način na koji bi se ovaj problem mogao rješiti je da se uklanjanje prefiksa/sufiksa dopusti jedino ako je njihova duljina (broj slova) manja ili jednaka polovici duljine cijele riječi (broja slova cijele riječi). Dodatno, uklanjanje prefiksa/sufiksa bi se mogao zabraniti ako je n-torka koja nastane prekratka ~~– 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~~



*Slika 5.2. E-sklonidba imenica [11]*

* + 1. Određivanje korijena riječi korištenjem skupa podataka

Umjesto da se ručno pokušavaju ukloniti prefiksi i sufiksi, lakše bi bilo pronaći postojeći skup podataka koji sadrži sve moguće oblike svih riječi uz njihov osnovni oblik.

Wječnik [13] je internet rječnik zasnovan na dobrovoljnim dodacima i izmjenama korisnika te je „sestrinski projekt Wikipedije“. Sadrži više od 10 000 hrvatskih riječi [13] – većinom imenica. Uspoređujući to s 400 000 riječi [14](upitni izvor?) koju hrvatski jezik zapravo ima, činjenicu da velik broj website-a tih riječi uopće ne sadrži njihovu deklinaciju i da su tablice deklinacija formatirane nekonzistentno, ovaj skup podataka se ne čini kao dobar kandidat za implementaciju. Pored Wječnik-a pronađen je i Github repozitorij [15] koji sadrži tekstualnu datoteku sa velikim brojem hrvatskih riječi uključujući njihove deklinacije, no taj skup podataka također pati od problema loše formatiranosti.

croDict [16] je website koji služi kao prevoditelj riječi s engleskog i njemačkog na hrvatski. Osim toga sadrži i popise imenica te glagola na tim jezicima, uključujući i njihove deklinacije, tj. konjugacije. U polje za pretraživanje se upiše riječ kojoj se traži osnovni oblik i izabere se je li imenica ili glagol. Jer je potrebno da osnovni oblik riječi dohvati program, nije moguće koristiti se elementima sučelja webpage. Srećom, osim polja za pretraživanje i opcije za vrstu riječi, informacije o riječi moguće je definirati unutar poveznice (Slika 5.3.). Jer program ne zna odrediti vrstu riječi potrebno je isprobati obje opcije. U slučaju da traženu riječ ne prepoznaje, na vrhu webpage se ispiše: „Der gesuchte Begriff konnte nicht gefunden werden“, što je prevedno s njemačkog: „Traženi pojam nije pronađen“. Nakon slanja zahtjeva za HTML kodom i njegovog primitka, ako program nije pronašao upozorenje o nepronalasku, pokušava dohvatiti riječ iznad tablica koja predstavlja osnovni oblik unesene riječi – nominativ jednine za imenice te infinitiv za glagole (Slika 5.4.).

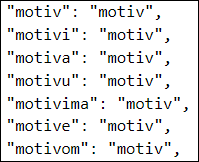


*Slika 5.3. Poveznica web-stranice s označenim riječima (lijevo – vrsta riječi (imenica/glagol), desno – riječ kojoj tražimo osnovni oblik) koje se izmijenjuju za dobitak osnovnog oblika riječi*



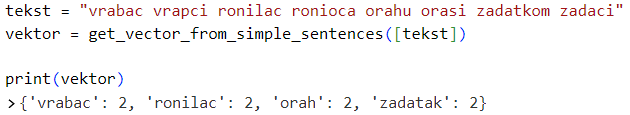
*Slika 5.4. Isječak webpagea poveznice sa Slike 5.3., osnovni oblik riječi ispisan na vrhu*

Kako bi se olakšalo opterećenje croDict poslužitelju, umanjila šansa zabrane pristupa websiteu zbog prevelikog broja zahtjeva te ubrzala klasifikacija, odlučeno je da će se rezultati upita webpageu spremiti lokalno u *json* datoteku, gdje je ključ riječ s kojom pretražujemo, a vrijednost je osnovni oblike te riječi (Slika 5.5.). Sada, ako se ponovno traži osnovni oblik iste riječi, neće se ponovno slati zahtjev nego će se uzeti iz rječnika. Za još veće ubrzanje programa i umanjenje broja zahtjeva, umjesto da program spremi samo riječ s kojom je pretraživano i njen osnovni oblik, preuzme i spremi sve oblike riječi te ih poveže.



*Slika 5.5. Isječak „json“ datoteke koji prikazuje sve oblike riječi „motiv“ spremljene u rječnik*

Rezultat je metoda koja može spojiti sve oblike velikog broja imenica i glagola, čak i onih sa nepravilnim deklinacijama/konjugacijama. Na Slici 5.6. se vidi definirani „tekst“ *string* koji sadrži imenice nepravilnih deklinacija te ispis vektora tog teksta koji je stvoren funkcijom *get\_vector\_from\_simple\_sentences()* (koja u pozadini koristi skup podataka stvoren iz croDict-a).



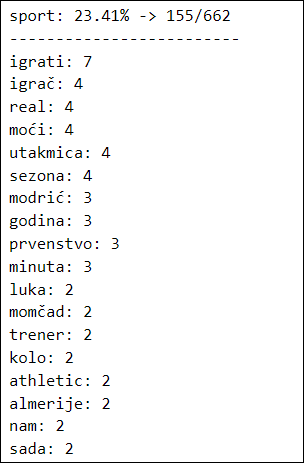
*Slika 5.6. Primjer kako se korištenjem croDict skupa podataka povezuju različiti oblici riječi*

Ako za zadanu riječ website nije mogao pronaći valjani osnovni oblik, ta riječ je lokalno spremljena u tekstualnu datoteku, kako bi se spriječilo ponovno slanje zahtjeva stranici. Nepronađene riječi još uvijek mogu biti unesene u vektor te će u kasnijem poglavlju utjecaj njihovog „postojanja“ na preciznost klasifikacije biti testiran.(OVO TESTIRAT)

1. BODOVANJE N-TORKI I IZRAČUN PRECIZNOSTI

Nakon što su sve n-torke teksta definirane, treba im pridodati nekakvu vrijednost. Ta vrijednost može biti koliko su puta ponovljenje, njihova relativna frekvencija s obzirom na količinu n-torki teksta ili nešto složenije poput metode „tf-idf“ koja ignorira riječi koje su prečeste – poput veznika [6]. MOŽDA SPONENI BOOLEAN NA KRATKO

Nakon svakog upita korisnika i izračuna preciznosti klasifikacije, automatski se izradi tekstualna datoteka „rezultat.txt“ (Slika 6.1.) koja sadrži sve kategorije/teme sortirane po vjerojatnosti i njihove n-torke koje se preklapaju s testiranim tekstom uz pripadajuće bodove, po kojima su i sortirane. Pregledavanjem sadržaja datoteke može se zaključiti zašto program odluči da je vjerojatnost jedne kategorije veća od druge, te će biti često korištena za objašnjavanje dobivenih rezultata.



*Slika 6.1. Isječak s početka tekstualne datoteke koja prikazuje opširniji rezultat klasifikacije uz vrijednosti(bodove) svake n-torke (ovdje je to broj ponavljanja) svih kategorije*

* 1. Binarno bodovanje

Najjednostavnija metoda zapravo niti ne boduje n-torke nego samo bilježi njihovu postojanost. Vektori nisu potrebni jer bi vrijednost svake n-torke (KOJA SE PONAVLJA?) bila jedan, pa su zato spremljene u obične liste (ALI DA SE NE PONAVLJAJU). Liste n-torki članaka iste kategorije povežu se unijom(BEZ PONAVLJANJA). Iz testiranog teksta uzme se svaka različita n-torka i ako postoji u listi n-torki neke kategorije, zbroju preklapanja s tom kategorijom se dodaje jedan. To se ponovi za svaku kategoriju te se za izračun vjerojatnosti kategorije njen broj preklapanja podijeli sa ukupnim zbrojem preklapanja svih kategorija. Formula 6.1 prikazuje taj izračun zapisan matematički, gdje predstavlja listu jedinstvenih n‑torki kategorije *k,* a listu jedinstvenih n-torki testiranog teksta.

(6.1)

* 1. Broj ponavljanja

U ovoj metodi bodovanja, vektori tekstova se stvaraju tako da se n-torki pridodaje vrijednost koja označava broj njezinog ponavljanje u tekstu(slika?). Kada se stvaraju vektori kategorija (klasifikatora), broj ponavljanja istih n-torki vektora članaka iste kategorije se jednostavno zbroji.

Tijekom uspoređivanja vektora kategorija s testnim vektorom s ciljem izračuna vjerojatnosti, gledaju se samo n-torke koje postoje u oba vektora, odnosno u vektoru kategorije i u testnom vektoru. Kako bi se lakše izračunala vjerojatnost, za svaku kategoriju napravi se rezultantni vektor koji sadrži samo one n-torke koje postoje i u vektoru kategorije i u testnom vektoru. Za određenu n-torku rezultantnog vektora uspoređuju se vrijednosti unutar vektora kategorije i testnog vektora, te se uzima ona koja je manja. Dobivena vrijednost unosi se u rezultantni vektor za određenu n-torku. Vjerojatnost da testirani dokument pripada nekoj kategoriji računa se dijeljenjem zbroja vrijednosti svih n-torki rezultantnog vektora trenutne kategorije sa zbrojem vrijednosti svih rezultantnih vektora svake kategorije. U formuli 6.2 vidi se izračun vjerojatnosti (u postotku) da testirani dokument pripada kategoriji *k*, gdje K predstavlja skup svih kategorija, vektor kategorije, testni vektor i *token* n-torku.

(6.2)

* 1. „tf-idf“

„Term frequency–inverse document frequency“ ili skraćeno „tf-idf“ je mjera koja pokazuje koliko je neka n-torka važna

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...

Ovoj metodi nije potrebno uklanjanje nebitnih rijeci

* + 1. Učestalost n-torke

Učestalost n-torke („tf“) je VRLO SLIČNA OVOME relativna učestalost ponavljanja n‑torke s obzirom na ukupni broj n-torki teksta u kojem se nalazi.

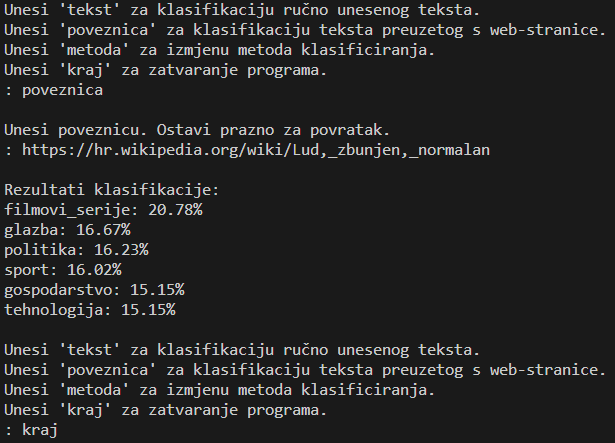
1. PROGRAMSKA OPREMA

Cilj programa je sa što većom preciznosti klasificirati uneseni tekst, tj. odrediti kojoj od definiranih kategorija taj tekst najviše odgovara. To radi tako da uspoređuje riječi poznatih tekstova kojima su teme već određene s riječima teksta kojega unese korisnik. Kroz par distinktivnih metoda pokušano je maksimizirati preciznost klasifikacije.

Program razlikuje šest kategorija/tema teksta: filmovi/serije, glazba, gospodarstvo, politika, sport i tehnologija. Kategorije su izabrane tako da se uklone moguća preklapanja tema, s ciljem da se preciznost može maksimizirati. Svaka kategorija ima svoju tekstualnu datoteku koja sadrži 20 poveznica članaka internet izdanja dnevnih novina – portala, koje su ručno unesene s obzirom kako su kategorizirane na website.

S Github repozitorija [7] program se može preuzeti i pokrenuti na računalima operacijskog sustava Windows.

Pri otvaranju programa dobiju se upute za upis jedne od četiri riječi (Slika 3.1.). Te riječi su „tekst“, „poveznica“, „metoda“ i „kraj“. Unosom „tekst“ korisniku je omogućen unos proizvoljnog teksta kojemu želi odrediti kategoriju. Poželjno je da ovaj tekst bude što duži jer pomaže u preciznosti klasifikacije(ZAŠTO??) jer su veće šanse da će doći preklapanja. Umjesto proizvoljnog teksta, moguće je unijeti poveznicu bilo kakvog internet dokumenta/članka. Ipak, preporuča se unos poveznica članaka portala „24sata“ ili „Novi list“ te hrvatske „Wikipedije“. Razlog zašto se ovo preporuča je naveden u poglavlju nakon. Nakon unosa proizvoljnog teksta ili poveznice, pritiskom na tipku „enter“, ispisuju se teme uz njihove vjerojatnosti u obliku postotka, prema kojima su i sortirane. Unosom „metoda“ dobiju se upute za izmjenu metoda izračuna vjerojatnosti. Jer su početne postavke metoda izabrane na temelju zaključka ovog rada, trebale bi dati najbolje rezultate. Unosom riječi „kraj“ program se zatvori.



*Slika 3.1. Prozor programa sa zahtjevom klasifikacije*

Korišten je programski jezik „Python“ zbog lakšeg rukovanja sa *string* objektima, ali i općenito jednostavnije sintakse. Za klasifikaciju teksta s oko 5500 riječi programu je potrebno u prosjeku *~0.27* sekundi. Korištenjem najsloženijih metoda to vrijeme poraste na *~0.32* sekunde. ~~U slučaju da korisnik izmjeni ili doda poveznice iz kojih se grade vektori klasifikatora, prvo klasificiranje će trajati nešto duže zbog preuzimanja, obrade i spremanja novih članaka.KRIVO NE RADI OVAKO NEGO KAD ZAPOCNE O ovome će biti objašnjeno više u sljedećem poglavlju. MŽDA I NETREBA , MAKNUT????(KASNIJE OBJASNI DA SE SPREMAJU CLANCI PRVI PUT).~~

Iz članaka jedne od šest kategorija, koji su preuzeti putem spremljenih poveznica, stvoreni su vektori koji su međusobno povezani, tvoreći vektor te kategorije. Drugo ime za vektore svih kategorija je vektori klasifikatora. Testirani tekst je onaj tekst koji se želi klasificirati, bio on ručno unesen ili izdvojen iz webpage unesene poveznice. Vektor koji je nastao iz testiranog teksta se zove testni vektor.

1. REZULTATI (USPOREDBE PRECIZNOSTI)

Za uspredbu preciznosti metoda klasifikacije svakom metodom je klasificiran po jedan dovoljno velik hrvatski Wikipedija članak iz svake od ukupno šest kategorija te su rezultati prikazani u tablicama. Redovi predstavljaju unesene članke, a stupci su kategorije članaka. Vrijednosti unutar ćelija su vjerojatnosti u postotcima da je neki članak neke kategorije. Vjerojatnosti za točnu kategoriju su podebljane (tvore dijagonalu). Boja pozadine svake ćelija je zatamnjena ovisno o unesenoj vjerojatnosti kako bi se lakše moglo vidjeti koji su postotci veći. Za kategoriju filmovi/serije je uzet članak serije „Lud, zbunjen, normalan“, za glazbu članak Olivera Dragojevića, za gospodarstvo članak o inflaciji, za politiku članak predsjednika Zorana Milanovića, za sport članak Luke Modrića i za tehnologiju članak o računalima.

Zbog broja metoda stvaranja n-torki i metoda njihovog bodovanja, sve kombinacije nisu testirane u ovome radu. Izabrati će se metoda bodovanja s obzirom koji bude imao najveću preciznost u prvih par tablica.

* 1. Obične n-torke

Tablice 7.1., 7.2., 7.3., i 7.4. prikazuju preciznosti klasifikacija 1-torki, 2-torki, 3-torki i 4-torki koje su binarno bodovane. Nikakve metode povezivanja sličnih n-torki nisu korištene.

*Tablica 7.1. Rezultati klasifikacije korištenjem 1-torki koje su binarno bodovane*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **20,63%** | 17,86% | 14,68% | 16,27% | 15,48% | 15,08% |
| Oliver Dragojević | 16,67% | **24,28%** | 14,95% | 14,13% | 14,86% | 15,13% |
| Inflacija | 12,73% | 16,50% | **22,24%** | 16,08% | 13,15% | 19,30% |
| Zoran Milanović | 16,24% | 17,52% | 17,36% | **18,31%** | 14,33% | 16,24% |
| Luka Modrić | 15,58% | 18,25% | 14,88% | 14,83% | **20,88%** | 15,58% |
| Računalo | 14,48% | 18,17% | 17,51% | 15,47% | 12,03% | **22,34%** |

*Tablica 7.2. Rezultati klasifikacije korištenjem 2-torki koje su binarno bodovane*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **21,79%** | 21,79% | 12,82% | 16,67% | 15,38% | 11,54% |
| Oliver Dragojević | 16,25% | **23,14%** | 16,53% | 12,67% | 14,88% | 16,53% |
| Inflacija | 7,74% | 14,29% | **27,98%** | 16,67% | 11,90% | 21,43% |
| Zoran Milanović | 18,89% | 17,22% | 18,33% | **18,89%** | 12,78% | 13,89% |
| Luka Modrić | 14,96% | 18,90% | 14,85% | 13,50% | **23,17%** | 14,62% |
| Računalo | 12,86% | 21,12% | 18,69% | 13,11% | 11,65% | **22,57%** |

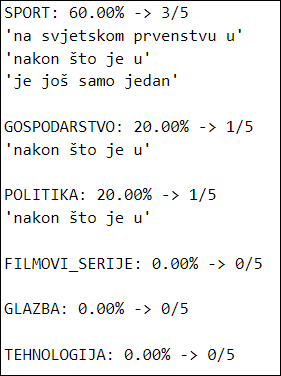
*Tablica 7.3. Rezultati klasifikacije korištenjem 3-torki koje su binarno bodovane*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **14,29%** | 28,57% | 28,57% | 14,29% | 0,00% | 14,29% |
| Oliver Dragojević | 9,09% | **36,36%** | 27,27% | 9,09% | 18,18% | 0,00% |
| Inflacija | 7,69% | 23,08% | **23,08%** | 15,38% | 7,69% | 23,08% |
| Zoran Milanović | 21,43% | 14,29% | 7,14% | **21,43%** | 21,43% | 14,29% |
| Luka Modrić | 12,66% | 22,78% | 16,46% | 11,39% | **26,58%** | 10,13% |
| Računalo | 16,67% | 20,83% | 8,33% | 12,50% | 8,33% | **33,33%** |

*Tablica 7.4. Rezultati klasifikacije korištenjem 4-torki koje su binarno bodovane*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **0,00%** | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| Oliver Dragojević | 0,00% | **0,00%** | 0,00% | 0,00% | 0,00% | 0,00% |
| Inflacija | 0,00% | 0,00% | **0,00%** | 0,00% | 0,00% | 0,00% |
| Zoran Milanović | 0,00% | 0,00% | 0,00% | **0,00%** | 100,00% | 0,00% |
| Luka Modrić | 0,00% | 0,00% | 20,00% | 20,00% | **60,00%** | 0,00% |
| Računalo | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | **0,00%** |

Može se primijetiti da korištenjem najprimitivnijeg bodovanja n-torki, bez uklanjanja nebitnih riječi i bez automatskog povezivanja sličnih riječi, klasifikator za 1-torke točno pogađa teme svih članaka, iako nije posve siguran. Najveću vjerojatnost od *24,28%* dobiva Oliver Dragojević za kategoriju glazba. POTENCIALNO JER JE BIO U ČLANKU KLASIFIKATORA(SLIKA 1-torke) I ZATO 3-torka JE TAKO VELIKA ZA NJEGA Kod 2-torki točno pogađa tri članka, a za članke „Lud, zbunjen, normalan“ i „Zoran Milanović“ klasifikator najviše vjerojatnosti daje dvjema kategorijama. Kod 3-torke dobivamo uglavnom krive rezultate. Zbog jako maloga preklapanja 4-torki (Slika 7.1.), vjerojatnosti u Tablici 7.4. su jako nepouzdane. Ipak Wikipedija članak Luke Modrića je točno svrstan kao „sport“. Može se vidjeti da od svih 4‑torki koje se preklapaju, samo jedna („na svjetskom prvenstvu u“) zapravo ima veze sa svojom kategorijom (sportom).



*Slika 7.1. 4-torke koje se nalaze u Wikipedija članku Luke Modrića i u člancima kategorija klasifikatora*

U sljedeće četiri tablice (7.5., 7.6., 7.7., 7.8.) mogu se vidjeti rezultati klasifikatora za iste n-torke, ali koje su bodovane po njihovom ponavljanju. Ovdje također nisu korištene metode uklanjanja nebitnih n-torki niti metode njihovog povezivanja.

*Tablica 7.5. Rezultati klasifikacije korištenjem 1-torki koje su bodovane po broju ponavljanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **20,78%** | 16,67% | 15,15% | 16,23% | 16,02% | 15,15% |
| Oliver Dragojević | 16,67% | **20,45%** | 15,98% | 15,28% | 15,87% | 15,76% |
| Inflacija | 13,65% | 15,98% | **21,41%** | 16,24% | 14,29% | 18,43% |
| Zoran Milanović | 16,12% | 16,91% | 17,03% | **18,24%** | 15,58% | 16,12% |
| Luka Modrić | 15,45% | 17,54% | 16,04% | 15,44% | **19,00%** | 16,52% |
| Računalo | 14,93% | 17,49% | 17,20% | 16,30% | 14,08% | **20,00%** |

*Tablica 7.6. Rezultati klasifikacije korištenjem 2-torki koje su bodovane po broju ponavljanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **23,75%** | 21,25% | 12,50% | 16,25% | 15,00% | 11,25% |
| Oliver Dragojević | 16,35% | **23,56%** | 16,11% | 12,50% | 15,38% | 16,11% |
| Inflacija | 8,56% | 14,44% | **27,81%** | 16,04% | 11,23% | 21,93% |
| Zoran Milanović | 18,53% | 16,81% | 18,10% | **19,40%** | 12,93% | 14,22% |
| Luka Modrić | 15,18% | 18,45% | 15,32% | 13,51% | **22,22%** | 15,32% |
| Računalo | 13,39% | 20,89% | 18,05% | 13,18% | 11,97% | **22,52%** |

*Tablica 7.7. Rezultati klasifikacije korištenjem 3-torki koje su bodovane po broju ponavljanja*

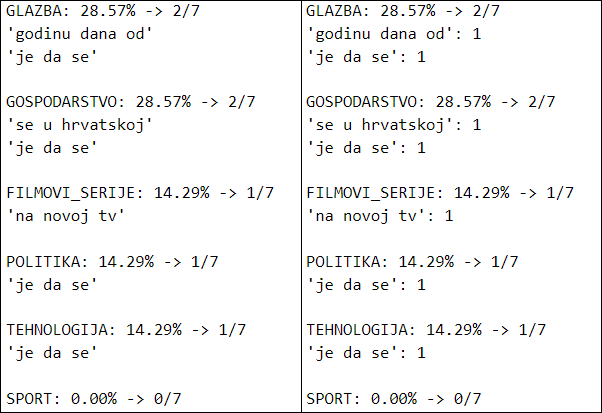
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **14,29%** | 28,57% | 28,57% | 14,29% | 0,00% | 14,29% |
| Oliver Dragojević | 9,09% | **36,36%** | 27,27% | 9,09% | 18,18% | 0,00% |
| Inflacija | 6,67% | 20,00% | **26,67%** | 13,33% | 6,67% | 26,67% |
| Zoran Milanović | 20,00% | 13,33% | 6,67% | **26,67%** | 20,00% | 13,33% |
| Luka Modrić | 10,64% | 19,15% | 18,09% | 13,83% | **29,79%** | 8,51% |
| Računalo | 17,86% | 17,86% | 7,14% | 17,86% | 10,71% | **28,57%** |

*Tablica 7.8. Rezultati klasifikacije korištenjem 4-torki koje su bodovane po broju ponavljanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **0,00%** | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| Oliver Dragojević | 0,00% | **0,00%** | 0,00% | 0,00% | 0,00% | 0,00% |
| Inflacija | 0,00% | 0,00% | **0,00%** | 0,00% | 0,00% | 0,00% |
| Zoran Milanović | 0,00% | 0,00% | 0,00% | **0,00%** | 100,00% | 0,00% |
| Luka Modrić | 0,00% | 0,00% | 28,57% | 14,29% | **57,14%** | 0,00% |
| Računalo | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | **0,00%** |

Uspoređivanjem vjerojatnosti Tablice 7.1. i Tablice 7.5. (1-torke), može se vidjeti da korištenjem metode brojanja ponavljanja n-torki, klasifikator još uvijek točno određuje teme članaka, no u tu točnu vjerojatnost je nesigurniji za *~1,52* postotna poena u prosjeku. prosjek je izračunat tako DA JE Razlika vjerojatnosti točne kategorije. Te se razlike zbroj i podijele se sa brojem članaka. Tablica 7.2. i Tablica 7.6. pokazuju da u je u 2-torkama bodovanje brojem ponavljanja bolje nego binarno bodovanje jer svakom članku točno određuje temu te je u tu točnu vjerojatnost sigurniji za *~0,29* postotna poena u prosjeku. 3-torke (Tablica 7.3., Tablica 7.7.) za članke „Lud, zbunjen, normalan“ i „Oliver Dragojević“ imaju identične vjerojatnosti za svaku kategoriju. Na slici 7.2. se vidi zašto su vjerojatnosti iste – 3-torke se ponavljaju samo jednom pa preslikavaju binarno bodovanje. Brojanjem ponavljanja n-torki, kategorija članaka Zorana Milanovića je ovdje točno određena, dok je u binarnom bodovanju članak imao iste najveće vjerojatnosti za kategorije politika i filmovi/serije. POSTOTAK 3-TORKE. Tablica 7.8. pokazuje kako 4-torke opet imaju jako nepouzdane rezultate, pa će biti preskočene dok se ne dođe do „pametnijih“ metoda?.

Ako se ignoriraju 4-torke, zbrajanjem prosjeka promjene vjerojatnosti dobije se da je ukupni rast sigurnosti jednak *~0.03* postotna poena. Gledajući to i činjenicu da je većem broju članaka točno određena tema, može se zaključiti da je u ovom okužrenju brojanje ponavljanja n-torki preciznije.



*Slika 7.2. 3-torke Wikipedija članka „Lud, zbunjen, normalan“ koristeći binarno bodovanje (lijevo) i bodovanje brojem ponavljanja (desno)*

1. ZAKLJUČAK

~~Može se zaključiti da kompleksnost metoda nije proporcionalna sa preciznočću klasifikatora, bar u okruženju u kojem se nalazi~~

~~Iako se ne vidi izravno iz rezultata, bilo bi dobro da su vektori klasifikatora opsirniji, tj da je vise članaka i da su duzi~~

**LITERATURA**

[1] „Bag-of-words model“, s interneta, <https://en.wikipedia.org/wiki/Bag-of-words_model>, 20. kolovoza 2023.

[2] „Document Classification With Machine Learning: Computer Vision, OCR, NLP, and Other Techniques“, s interneta, <https://www.altexsoft.com/blog/document-classification>, 19. kolovoza 2023.

[3] „Bag-of-words model in computer vision“, s interneta, <https://en.wikipedia.org/wiki/Bag-of-words_model_in_computer_vision>, 14. kolovoza 2023.

[4] „Computer vision“, s interneta, <https://en.wikipedia.org/wiki/Computer_vision>, 21. kolovoza 2023.

[5] „Security Camera & Video Surveillance Blog“, s interneta, <https://videos.cctvcamerapros.com/v/car-detection-ai-security-camera.html>, 24. kolovoza 2023.

[6] „A Gentle Introduction to the Bag-of-Words Model“, s interneta, <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>, 14. kolovoza 2023.

[7] „bag\_of\_words\_model“, s interneta, <https://github.com/aZina0/bag_of_words_model>, 23. kolovoza 2023.

[8] „24sata“, s interneta, <https://www.24sata.hr/>, 15. kolovoza 2023.

[9] „Novi list“, s interneta, <https://www.novilist.hr/>, 20. kolovoza 2023.

[10] „Wikipedija“, s interneta, <https://hr.wikipedia.org/wiki/Glavna_stranica>, 16. kolovoza 2023.

[11] „Hrvatska školska gramatika“, s interneta, <http://gramatika.hr/>, 17. kolovoza 2023.

[12] „Hrvatski prefiksi“, s interneta, <https://hr.wikipedia.org/wiki/Hrvatski_prefiksi>, 27. srpnja 2023.

[13] „Wječnik“, s interneta, <https://hr.wiktionary.org/wiki/Glavna_stranica>, 15. kolovoza 2023.

[14] „Hrvatski ima 400. 000 riječi“, s interneta, <https://www.jutarnji.hr/naslovnica/hrvatski-ima-400.-000-rijeci-4001484>, 20. kolovoza 2023.

[15] „rjecnik-hrvatskih-jezika“, s interneta, <https://github.com/gigaly/rjecnik-hrvatskih-jezika>, 16. kolovoza 2023.

[16] „croDict“, s interneta, <https://www.crodict.hr/>, 17. kolovoza 2023.