SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Prijediplomski sveučilišni studij računarstva

Završni rad

**KLASIFIKACIJA TEKSTOVA TEMELJENA**

**NA SKUPOVIMA RIJEČI**

Rijeka, rujan 2023. Ante Kozina

0069088428

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Prijediplomski sveučilišni studij računarstva

Završni rad

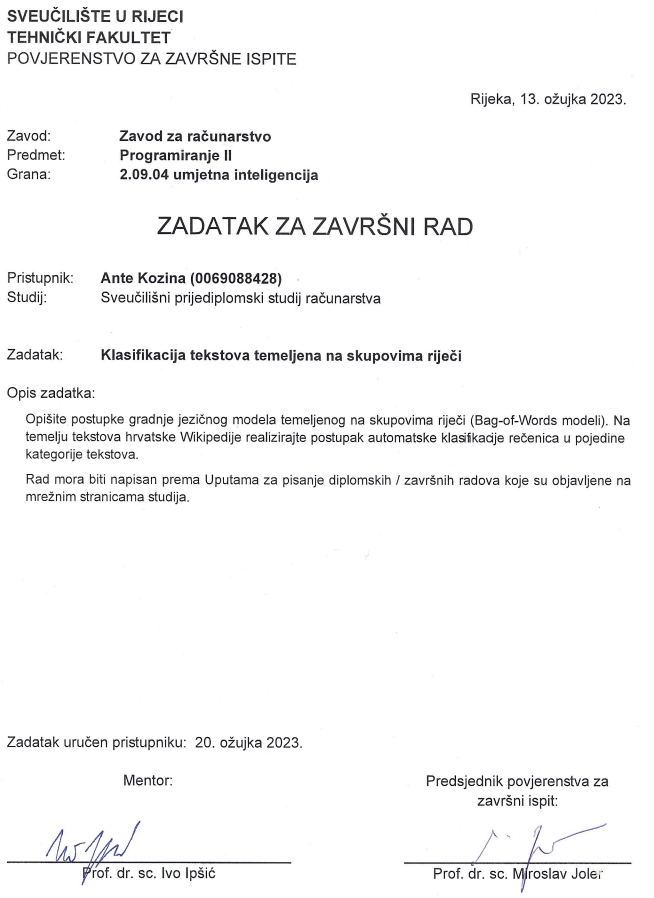
**KLASIFIKACIJA TEKSTOVA TEMELJENA**

**NA SKUPOVIMA RIJEČI**

Mentor: Prof. dr. sc. Ivo Ipšić

Rijeka, rujan 2023. Ante Kozina

0069088428



POTPISANA IZJAVA

ZAHVALA

Zahvalio bih svojem mentoru prof. dr. sc. Ivi Ipšiću na predloženoj temi i na velikoj pomoći kod realizacije završnog rada.

Također, zahvalio bih svojoj obitelji i prijateljima na njihovoj strpljivosti i za rečenice koje su smislili kako bi mogao testirati klasifikator.

Sadržaj

[1. UVOD 7](#_Toc144592455)

[2. JEZIČNI MODEL TEMELJEN NA ZBIRKAMA RIJEČI 8](#_Toc144592456)

[3. KLASIFIKATOR 9](#_Toc144592457)

[3.1. Izdvajanje sadržaja članka 9](#_Toc144592458)

[3.2. Obrada preostalog HTML koda 11](#_Toc144592459)

[4. METODE ODABIRA I POVEZIVANJA N-TORKI 12](#_Toc144592460)

[4.1. Uklanjanje nebitnih riječi 12](#_Toc144592461)

[4.2. Automatsko određivanje korijena riječi 13](#_Toc144592462)

[4.2.1. Prefiks-sufiks algoritam 13](#_Toc144592463)

[4.2.2. Određivanje korijena riječi korištenjem skupa podataka 14](#_Toc144592464)

[5. METODE IZRAČUNA VJEROJATNOSTI 17](#_Toc144592465)

[5.1. Binarno bodovanje 17](#_Toc144592466)

[5.2. Broj ponavljanja 17](#_Toc144592467)

[5.3. „tf-idf“ 18](#_Toc144592468)

[6. PROGRAMSKA OPREMA 20](#_Toc144592469)

[7. REZULTATI 23](#_Toc144592470)

[7.4. Neobrađene n-torke 23](#_Toc144592471)

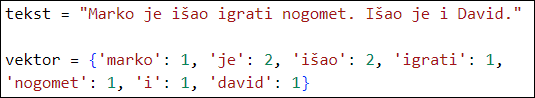
[8. ZAKLJUČAK 30](#_Toc144592472)

1. UVOD

„Bag of words model“ - prevedeno „model temeljen na zbirkama riječi“ 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.

1. JEZIČNI MODEL TEMELJEN NA ZBIRKAMA RIJEČI

Jezični model temeljen na zbirkama riječi je način izvlačenja riječi iz nekog sadržaja s ciljem sažimanja i klasificiranja tog sadržaja [6]. 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 dokument 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 temeljenog na zbirkama riječi taj rječnik se zove vektor (slika 2.1.).



*Slika 2.1. Vektor nastao iz dokumenta, 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 različite metode za pregledavanje i usporedbu broja ponavljanja n-torki unutar njihovih vektora. Metode iz tih dviju kategorija će biti objašnjene te će kasnije biti kombinirane i testirane s ciljem stvaranja najpreciznijeg klasifikatora.

1. KLASIFIKATOR

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.

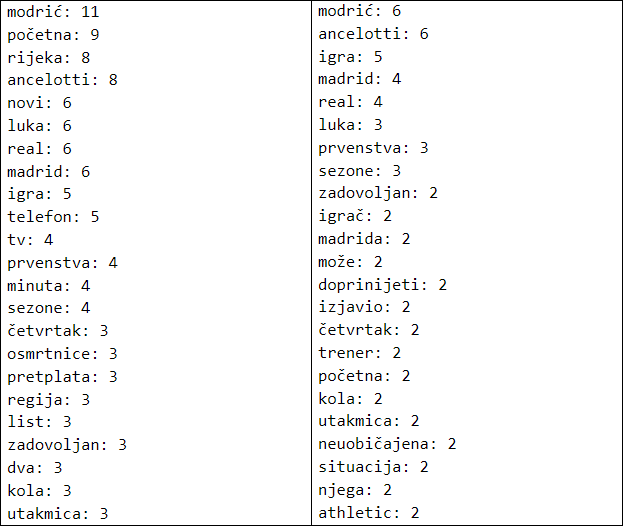
Program razlikuje šest kategorija/tema teksta: filmovi/serije, glazba, gospodarstvo, politika, sport i tehnologija. Kategorije su izabrane tako da se uklone moguća preklapanja tema, s ciljem da se preciznost može maksimizirati. Svaka kategorija ima svoju tekstualnu datoteku koja sadrži 20 poveznica članaka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

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 dokument je zajednički naziv za tekst koji se želi klasificirati, bio on ručno unesen ili izdvojen iz web-stranica unesenih poveznica.

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 web-stranica jer svaki website ima svoj dizajn HTML koda. Zbog tog razloga i jer je kod njih izdvajanje teksta članka bilo nešto jednostavnije uzeti su 24sata, Novi list i Wikipedija.

* 1. Izdvajanje sadržaja članka

Jer irelevantni dijelovi web-stranica 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 web-stranice. U lijevoj listi riječi na slici 3.1. 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 web-stranice.



*Slika 3.1. Najčešće riječi (isključujući veznike) sportskog članka portala „Novi list“ bez (lijevo) i sa izdvajačem sadržaja (desno)*

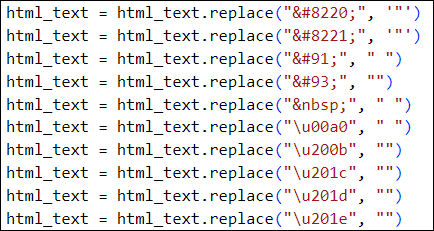
Ako se radi o HTML kodu 24sata, Novog lista ili Wikipedije traži se *<div>* oznaka sa jedinstvenom *class* ili *id* vrijednosti koja je ručno pronađena/izabrana. U slučaju promjene dizajna ili nekih pojedinosti HTML koda web-stranice, ovaj izdvajač sadržaja neće moći pronaći tekst članka, stoga nije dugoročno riješenje. Unutar parova oznaka, *<div>* i *</div>*, traže se *<p>* oznake koje označavaju paragrafe. Sva tri website u svoje web-stranice 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 3.2. 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.



*Slika 3.2. Zamjena HTML kodova sa ispravnim znakovima*

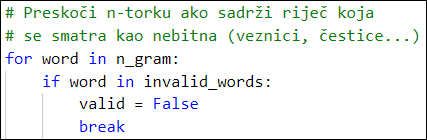
1. METODE ODABIRA I POVEZIVANJA N-TORKI

N-torka je konačan niz *n* objekata, a ovdje ona predstavlja niz *n* susjednih riječi. 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.

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 (pronađeni na [11]). 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 4.1.).



*Slika 4.1. Izbacivanje n-torke ako sadrži nebitnu riječ („invalid\_words“ je lista koja sadrži sve nebitne riječi)*

* 1. Automatsko određivanje korijena riječi

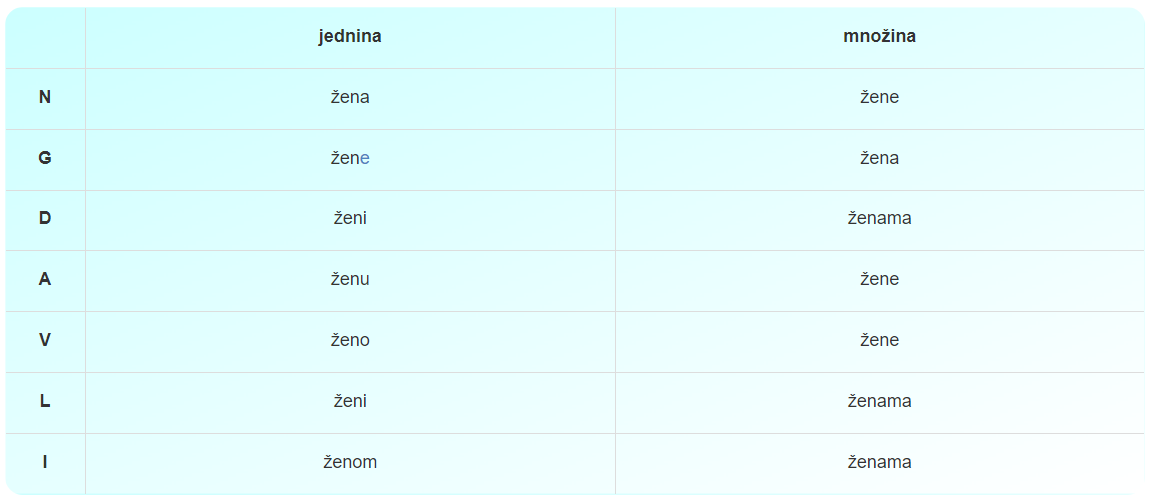
Ova metoda pokušava algoritmima povezati riječi koje nemaju isti oblik, jer su u drugom padežu ili glagolskom vremenu, ali imaju isto značenje, tj. predstavljaju isti pojam. ~~Mnogi od pronađenih algoritama su fokusirani na engleski jezik te ih je bilo vrlo složeno ili nemoguće prilagoditi hrvatskome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

* + 1. Prefiks-sufiks algoritam

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. 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 4.2.) 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 duljina rezultata (broj slova) nije premala. „“ može biti neka DEFINIRANA duljina ili RELATIVNA S OBZIROM NA DUILJINU PREFISKA/SUFISKA – DODAJ OVO U STEMMING



*Slika 4.2. 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 web-stranice. Srećom, osim polja za pretraživanje i opcije za vrstu riječi, informacije o riječi moguće je definirati unutar poveznice (slika 4.3.). Jer program ne zna odrediti vrstu riječi potrebno je isprobati obje opcije. U slučaju da traženu riječ ne prepoznaje, na vrhu web-stranice 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 4.4.).

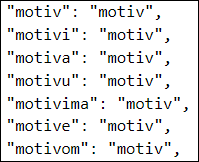


*Slika 4.3. Poveznica web-stranice s označenim riječima (lijevo – vrsta riječi (imenica/glagol), desno – riječ kojoj tražimo osnovni oblik) koje se izmijenjuju za dobitak osnovnog oblika riječi*



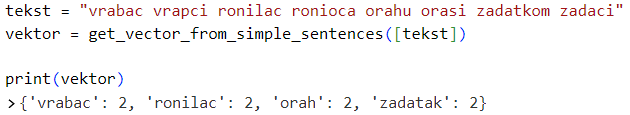
*Slika 4.4. Isječak web-stranicaa poveznice sa Slike 5.3., osnovni oblik riječi ispisan na vrhu*

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 web-stranici spremiti lokalno u *json* datoteku, gdje je ključ riječ s kojom pretražujemo, a vrijednost je osnovni oblike te riječi (slika 4.5.). 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 4.5. 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 4.6. 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 4.6. 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 kasnijemHOCE LI?? poglavlju utjecaj njihovog „postojanja“ na preciznost klasifikacije biti testiran.(OVO TESTIRAT)

1. METODE IZRAČUNA VJEROJATNOSTI

Nakon što su vektori svih dokumenata definirani, sadržeći n-torke tih dokumenata i njihov broj ponavljanja, putem njih je potrebno izračunati vjerojatnosti da je testirani dokument neke kategorije (teme). Načini na koje su vektori klasifikatora i vektor testiranog dokumenta uspoređeni i obrađeni te različite formule izračuna preciznosti su navedene u nastavku.

* 1. Binarno bodovanje

Najjednostavniju metodu izračuna preciznosti zapravo niti ne zanima broj ponavljanja n‑torke nego samo njeno postojanje, otkuda i ime metode. Stoga, u vektoru dokumenta, broj ponavljanja možemo zamjeniti jedinicom. Za svaku kategoriju napravljen je skup koji sadrži jedinstvene n-torke koje se pojavljuju i u vektoru te kategorije i u vektoru testiranog dokumenta. Za izračun vjerojatnosti da je testirani dokument neke kategorije, veličina skupa kategorije podjeljenja je sa zbrojem veličina skupova svih kategorija. Formula 5.1 prikazuje izračun te vjerojatnosti () u obliku postotka.

(5.1)

gdje je:

*k* kategorija

*K* skup svih kategorija

skup jedinstvenih n-torki kategorije *k*

skup jedinstvenih n‑torki testiranog dokumenta

* 1. Broj ponavljanja

Kod ove metode nije samo važno da n-torka postoji u dokumentima, nego je važan i njen broj ponavljanja. Kao i prije, za svaku kategoriju napravljen je skup koji sadrži jedinstvene n-torke koje se pojavljuju i u vektoru te kategorije i u vektoru testiranog dokumenta, no sada je potrebno zadržati njihove količine ponavljanja. Za određenu n-torku skupa uspoređuju se količine ponavljanja unutar vektora kategorije i vektora testiranog dokumenta, te se uzima ona koja je manja. Zbrojem tih manjih vrijednosti svih n-torki skupa kategorije dobivamo „rezultat“ kategorije. Vjerojatnost da testirani dokument pripada nekoj kategoriji računa se dijeljenjem „rezultata“ te kategorije sa zbrojem „rezultata“ svih kategorija. U formuli 5.2 vidi se izračun vjerojatnosti (u obliku postotku) da testirani dokument pripada kategoriji *k*.

(5.2)

gdje je:

*t* n-torka

*k* kategorija

*K* skup svih kategorija

skup jedinstvenih n-torki kategorije *k*

skup jedinstvenih n‑torki testiranog dokumenta

broj ponavljanja n-torke *t* u kategoriji *k*

broj ponavljanja n-torke *t* u testiranom dokumentu

* 1. „tf-idf“

„Term frequency–inverse document frequency“ ili skraćeno *tf-idf* mjera je koja pokazuje koliko je neka n-torka važna u skupu dokumenata, a u kontekstu ovog rada, koliko je važna u kategoriji, tj. skupu dokumenata kategorije [6, 17]. Računa se umnoškom učestalosti n-torke (*tf*) u skupu dokumenata kategorije i inverznom učestalosti dokumenta n-torke (*idf*) u istom tom skupu dokumenata kategorije. Jednadžba 5.3 prikazuje taj umnožak, gdje *t* predstavlja n-torku, a *k* kategoriju.

(5.3)

gdje je:

učestalost n-torke *t* u kategoriji *k*

inverzna učestalost dokumenta n-torke *t* u kategoriji *k*

Jer metoda *tf-idf* prebacuje važnost s n-torki koje se ponavljaju puno na n-torke koje se ponavljaju malo, riječi web-stranice koje nisu vezane uz članak, poput reklama, mogu poremetiti vektore klasifikatora i time i preciznost. Zbog istog tog razloga, ova bi metoda trebala moći smanjiti važnost nebitnih riječi koje se često ponavljaju, poput veznika i čestica, umanjujući potrebu za korištenjem već definirane metode za uklanjanje nebitnih riječi.

Učestalost n-torke (*tf*) relativna je učestalost ponavljanja n‑torke s obzirom na ukupni broj n-torki dokumenta u kojem se nalazi. Frekvenciji se izračuna logaritam baze 10 kako bi se umanjila važnost n-torki koje se ponavljaju puno puta [17]. U formuli 5.4 se vidi izračun učestalosti n-torke *t* u dokumentu ili skupu dokumenata (ako je uvrštena kategorija) *d*.

(5.4)

gdje je:

skup jedinstvenih n-torki dokumenta *d*

broj ponavljanja n-torke *t* u dokumentu *d*

Inverzna učestalost dokumenta (*idf*) n-torke odražava udio dokumenata u skupu svih dokumenata kategorije koji sadrže tu n-torku. Iako u kontekstu ovog rada to nije potrebno napraviti zbog malog broja dokumenata kategorije, i ovdje se također preporuča izračun logaritma izraza. Formula za njen izračun je sljedeća (5.5):

(5.5)

gdje je:

broj dokumenata kategorije *k*

broj dokumenata kategorije *k* koje sadrže n-torku *t*

Za izračun koliko je neka n-torka testiranog dokumenta bitna u kategoriji, uzima se njena važnost u skupu dokumenata kategorije (*tfidf*). Za povećanje izražaja važnosti n-torke, njen *tfidf* iznos pomnožimo s njenom relativnom učestalosti (*tf*) u testiranom dokumentu. Te „važnosti“ svih n-torki kategorije se zbroje, tvoreći „rezultat“ kategorije. Taj rezultat se, kao i kod drugih metoda, normalizira – podijeli se sa zbrojem rezultata svih kategorija. Konačno, izračun vjerojatnosti (u postotku) da je testirani dokument kategorije *k* je prikazan sljedećom jednadžbom (5.6):

(5.6)

gdje je:

*t* n-torka

testirani dokument

*k* kategorija

*K* skup svih kategorija

skup jedinstvenih n-torki kategorije *k*

skup jedinstvenih n‑torki testiranog dokumenta

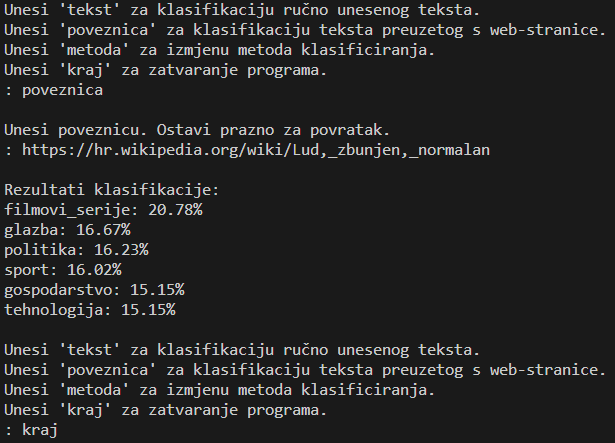
učestalost n-torke *t* u dokumentu *d*

*tf* učestalost n-torke - inverzna učestalost dokumenta n-torke *t* u kategoriji *k*

1. PROGRAMSKA OPREMA

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

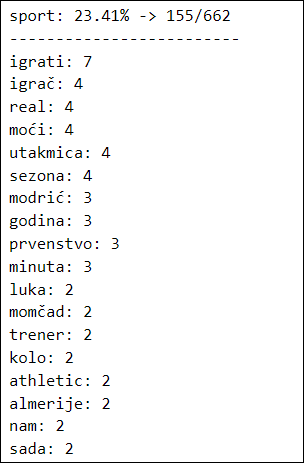
Pri otvaranju programa dobiju se upute za upis jedne od četiri riječi (slika 6.1.). Te riječi su „tekst“, „poveznica“, „metoda“ i „kraj“. Unosom „tekst“ korisniku je omogućen unos proizvoljnog teksta kojemu želi odrediti kategoriju. Poželjno je da ovaj tekst bude što duži jer pomaže u preciznosti klasifikacije jer su šanse za preklapanje n-torki veće. 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 6.1. 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.

Nakon svakog upita korisnika i izračuna preciznosti klasifikacije, automatski se izradi tekstualna datoteka „rezultat.txt“ (slika 6.2.) koja sadrži sve kategorije/teme sortirane po vjerojatnosti i njihove n-torke koje se preklapaju s testiranim dokumentom 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 6.2. Isječak s početka tekstualne datoteke koja prikazuje opširniji rezultat klasifikacije uz broj ponavljanja n-torki svih kategorije*

1. REZULTATI

Za usporedbu efikasnosti metoda kojima je pokušano maksimizirati preciznost klasifikatora, svakom metodom je klasificiran po jedan dovoljno velik hrvatski Wikipedija članak iz svake od ukupno šest kategorija te su rezultati prikazani u tablicama. 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. Neobrađene 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 i binarnog bodovanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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 i binarnog bodovanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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%** |

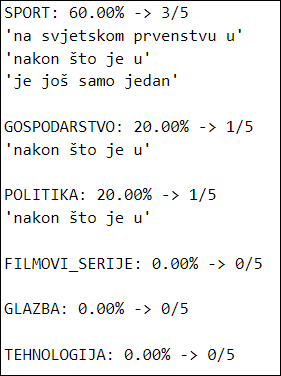
*Tablica 7.3. Rezultati klasifikacije korištenjem 3-torki i binarnog bodovanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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 i binarnog bodovanja*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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*

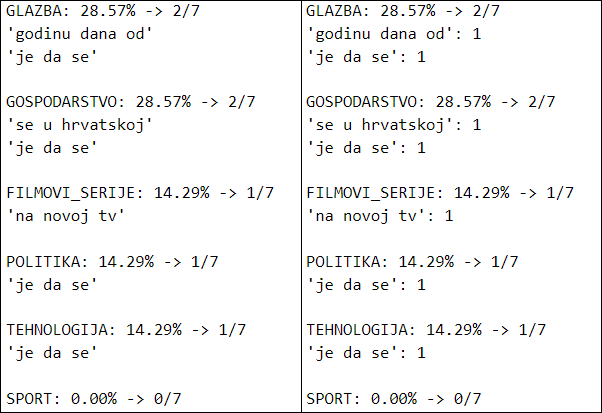
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **14,29%** | 28,57% | 28,57% | 14,29% | 0,00% | 14,29% |
| Oliver Dragojević | 9,09% | **36,36%** | 27,27% | 9,09% | 18,18% | 0,00% |
| 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)*

Sljedeće tablice KOJE prikazuju 1-torke, 2-torke i 3-torke bodovane „tf-idf“ mjerom.

*Tablica 7.9. Rezultati klasifikacije korištenjem 1-torki i „tf-idf“ metode*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **28,19%** | 10,23% | 13,38% | 16,08% | 17,55% | 14,57% |
| Oliver Dragojević | 20,74% | **15,77%** | 14,65% | 15,18% | 18,99% | 14,67% |
| Inflacija | 14,97% | 10,87% | **23,41%** | 16,59% | 15,86% | 18,30% |
| Zoran Milanović | 18,66% | 11,80% | 15,39% | **21,43%** | 18,45% | 14,28% |
| Luka Modrić | 19,37% | 12,30% | 14,10% | 15,10% | **24,63%** | 14,51% |
| Računalo | 18,33% | 11,99% | 17,19% | 15,98% | 17,12% | **19,39%** |

*Tablica 7.10. Rezultati klasifikacije korištenjem 2-torki i „tf-idf“ metode*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **27,95%** | 12,64% | 11,80% | 21,30% | 16,32% | 10,00% |
| Oliver Dragojević | 21,13% | **17,94%** | 15,48% | 12,76% | 19,10% | 13,60% |
| Inflacija | 12,35% | 13,01% | **23,21%** | 17,53% | 14,17% | 19,74% |
| Zoran Milanović | 24,06% | 12,72% | 16,12% | **20,46%** | 14,58% | 12,06% |
| Luka Modrić | 19,53% | 13,33% | 14,13% | 15,44% | **25,20%** | 12,37% |
| Računalo | 17,07% | 15,39% | 17,92% | 16,87% | 14,58% | **18,17%** |

*Tablica 7.11. Rezultati klasifikacije korištenjem 3-torki i „tf-idf“ metode*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | filmovi/ serije | glazba | gospodarstvo | politika | sport | tehnologija |
| Lud, zbunjen, normalan | **35,25%** | 20,27% | 15,10% | 22,73% | 0,00% | 6,64% |
| Oliver Dragojević | 12,20% | **26,50%** | 23,31% | 10,15% | 27,84% | 0,00% |
| Inflacija | 6,62% | 16,30% | **26,52%** | 11,01% | 5,95% | 33,59% |
| Zoran Milanović | 17,08% | 7,84% | 9,73% | **34,25%** | 18,10% | 12,99% |
| Luka Modrić | 13,43% | 8,88% | 19,93% | 16,35% | **34,85%** | 6,57% |
| Računalo | 25,25% | 12,70% | 4,06% | 20,05% | 16,14% | **21,79%** |

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

Spoemnuti su problemi obrađivanja članaka

**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.

[17] „Speech and Language Processing“, s interneta, <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>, 30. kolovoza 2023.