Računalna analiza sentimenta

Šikač, Patrik Noah

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Patrik Noah Šikač

RAČUNALNA ANALIZA SENTIMENTA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Patrik Noah Šikač

Matični broj: 0016130318

Studij: Informacijski sustavi

RAČUNALNA ANALIZA SENTIMENTA

ZAVRŠNI RAD

Mentor :

Dr. sc. Okreša Đurić Bogdan

Varaždin, rujan 2020.

Patrik Noah Šikač

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj rad proučava koncept analize sentimenta u kontekstu primjene metoda umjetne inteligencije. Opisani su osnovni pojmovi koji predstavljaju temelj računalne analize sentimenta, a to su umjetna inteligencija, strojno učenje i rudarenje podataka. Osim samog pojma analize sentimenta, definirane su razine na kojima se odvija analiza sentimenta. Nakon toga, obrađeni su primjeri korištenja računalne analize sentimenta u praksi te su pomoću tih primjera opisane prepreke s kojima se stručnjaci, odnosno, programeri susreću i njihova potencijalna rješenja. Opisane su dvije vrste sustava za računalnu analizu sentimenta: sustav temeljen na pravilima i automatski sustav. Za dotične sustave opisane su metode kojima su oni realizirani i utvrđene njihove prednosti i mane. Praktični dio rada uključuje primjenu implementacija sustava realiziranih različitim kombinacijama metoda za realizaciju sustava za analizu sentimenta na izabranim skupovima podataka te analizu dobivenih rezultata.

Ključne riječi: umjetna inteligencija, strojno učenje, analiza sentimenta, sentiment, rudarenje mišljenja, rudarenje podataka, klasifikacija, vektorizacija

Sadržaj

1. Uvod

Unutar industrije informacijske i komunikacijske tehnologije (u daljnjem tekstu ICT), sve se veća važnost postavlja na razvoj umjetne inteligencije. Najpopularnija primjena umjetne inteligencije trenutno je analiza podataka i strojno učenje koju već primjenjuju industrijski giganti poput Amazona i Facebooka, no postoje još i druge primjene poput manipulacija videa, generiranje slika, tekstova i glazbe, prepoznavanje lica, itd. Samo iz ovih par primjera je neporecivo da potencijal umjetne inteligencije nema granica, a ova grana industrije koja se razvija već desetljećima nije još uvijek dostigla svoj vrhunac.

Jedna od najzanimljivijih uporaba umjetne inteligencije jest interakcija sa upravo onim aspektom čovjeka koji strojevi ne razumiju, a to su emocije. Ovaj rad se bavi upravo takvom vrstom umjetne inteligencije koja služi za analizu emocija iz teksta, a zove se računalna analiza sentimenta. Računalna analiza sentimenta (engl. Sentiment Analysis) jest znanstvena disciplina koja uz pomoć strojnog učenja i rudarenja podataka zaključuje koje se emocije kriju iza teksta. Ova vrsta analize ima vrlo širok spektar primjene, od poslovne pa sve do edukacijske i potencijal za rast joj je vrlo velik. Motivacija za pisanje ovog rada proizlazi iz fascinantnosti ovog područja i želje za proučavanjem primjene i potencijala ove opširne teme. Cilj ovog završnoga rada jest pobliže opisati što je računalna analiza sentimenta, kako ona funkcionira, gdje se točno primjenjuje. Opisane su razine na kojima se može odvijati analiza, prepreke koje se pojavljuju i potencijalna rješenja te su na kraju prikazane primjene dvije vrste analize na stvarnim primjerima.

2. Metode i tehnike rada

Za izradu ovog rada korištene su sljedeće metode:

- Induktivna metoda – kako bi se došlo do zaključka o računalnoj analizi sentimenta, prvo je potrebno analizirati sve pojedine dijelove koji ju čine.
- Deduktivna metoda – zbog opširnosti ovog područja nije moguće analizirati sve primjene ove tehnologije, no moguće je zaključiti koja će zapažanja vrijediti u ostalim situacijama.
- Metoda analize – koriste se za raščlambu analize sentimenta na cjeline kako bi se mogli detaljno razraditi i objasniti, npr. razine analize, vrste analize, vrste vektorizacije, itd.
- Metoda klasifikacije – služe za podjelu pojedinih pojmova radi lakšeg objašnjavanja i razumijevanja.
- Metoda modeliranja – kako bi se pokazala primjena računalne analize sentimenta na stvarnim primjerima, potrebno je izraditi modele koji će vraćati razumljive i mjerljive podatke.

Tehnologije koje su korištene za izradu ovog rada:

- PyCharm Community Edition 2020.2 – razvojno okruženje za rad u programskom jeziku Pythonu.
- draw.io – alat za izradu grafikona

3. Što je analiza sentimenta?

U ovome se poglavlju obrađuje pojam računalne analize sentimenta, počevši od definicije strojnog učenja, rudarenja podataka i umjetne inteligencije jer ti pojmovi su usko povezani sa temom ovog završnog rada. Kako bi dobro razumjeli računalnu analizu sentimenta, moraju se prvo razumjeti aspekti koji ju čine. Nakon definicije ta tri osnovna pojma, napravljen je detaljan pogled u samu računalnu analizu sentimenta. Bitno je shvatiti kako ona funkcionira i koje su njezine koristi i primjene u stvarnome svijetu.

3.1. Umjetna inteligencija, strojno učenje i rudarenje podataka

Za umjetnu inteligenciju, također poznatu kao i strojna inteligencija, vežu su razne definicije koje prioretiziraju drugačije aspekte umjente inteligencije. Može se reći da je umjetna inteligencija svaki program koji obavlja funkcije koje se asociraju sa ljudskom inteligencijom jednako ili bolje od čovjeka. Russel [1] definira četiri pristupa definiranja umjetne inteligencije: pristup ljudskog ponašanja, ljudskog razmišljanja, racionalnog razmišljanja i racionalnog ponašanja.

Pristup ljudskog ponašanja nalaže da se umjetna inteligencija svojim ponašanjem ne smije razlikovati od čovjeka. To se testira Turingovim testom u kojemu ljudski ispitivač ispituje računalo skupom različith pisanih pitanja na koje računalo odgovara. Ako ispitivač ne primjeti da je ispitanik računalo, test je uspješan. Međutim, stručnjaci ne stavljaju prioritet na ovaj pristup jer im je u cilju saznati što se skriva iza inteligencije, a ne izraditi primjerak iste. Taj je dio više pokriven pristupom ljudskog razmišljanja. Naime, ovim se pristupom pokušava izraditi kognitivni model koji će se svojim ulazima i izlazima podudarati s odgovarajućim ljudskim ponašanjima. Može se reći da se ovim pristupom želi izraditi program koji prati ljudski način razmišljanja. Pristup racionalnog razmišljanja, tj. "zakona mišljenja" temelji se na rješavanju problema pomoću logike. Dakle, ovim bi se pristupom mogao, u principu, riješiti svaki problem zapisan u logičkoj notaciji. Međutim, proces zapisivanja problema u formalnu logičku notaciju nije jednostavno za svaki problem. Nadalje, rješavanje složenih problema može istrošiti veliki dio računalnih resursa. Posljednji pristup je pristup racionalnog ponašanja, a postiže se uporabom racionalnih agenata. Računalni agenti obavljaju svoje funkcije autonomno, promatraju svoje okruženje, prilagođavaju se promjenama te kreiraju i prate vlastite ciljeve. Svrha racionalnih agenata je da izvrše zadani zadatak na najbolji mogući način.

Pojam umjetne inteligencije često je vezan za sve strojeve ili računala koji su u stanju obavljati neke kognitivne funkcije koje se asociraju sa ljudskim umom, poput čitanja, učenja i rješavanja raznih problema. Umjetna inteligencija kao znanstvena disciplina osnovana je 1955. godine te je kroz godine imala svoje uspone i padove [2] . Neki stručnjaci smatraju da se već nalazimo u „zlatnome dobu“ umjetne inteligencije. Bez obzira je li to istina ili nije, činjenica je da se razvoju umjetne inteligencije posvećuje više pažnje nego ikad prije u povijesti ove discipline.

Strojno učenje je grana umjetne inteligencije koja se bavi izučavanjem algoritama koji

se poboljšavaju kroz iskustvo, tj. „učenje“ [3]. Algoritmi za strojno učenje izrađuju matematički model prema podacima za treniranje, što im služi za donošenje odluka ili predviđanje bez ikakvih eksplicitnih uputa. Kod jednostavnih programa lagano je zadati računalu točne upute kojima bi došlo do rješenja, npr. jednostavno grananje, no kod izrade kompleksnijih programa gdje je izrada algoritama presložena i za same programere, kao kod prepoznavanja uzoraka, bolje je izraditi algoritam kojim će se računalo podučavati za rješavanje takvih problema. Temeljni faktor za uspješno strojno učenje su skupovi podataka. Što više skupova podataka računalo obradi algoritmom za podučavanje, to su bolje šanse da algoritam bude precizniji. Strojno učenje možemo podijeliti na više vrsta: nadzirano učenje (engl. supervised learning), nenadzirano učenje (engl. unsupervised learning), polu-nadzirano učenje (engl. semisupervised learning) te ojačano učenje (engl. reinforced learning) .

Nadzirano učenje radi na principu treniranja računala pomoću setova podataka koji imaju određenu ulaznu vrijednost i izlaznu vrijednost [4]. Na primjer, za ulazne vrijednosti možemo postaviti podatke o učenicima u srednjoj školi te izlazne vrijednosti definiramo kao „1“ za one učenike koji su upisali fakultet i „0“ za one koji nisu nastavili školovanje na fakultetu. Računalo pomoću tih označenih podataka izrađuje popis pravila kojima povezuje ulaze sa izlazima. Na temelju tih pravila bismo mogli predviđati koji će učenici upisati fakultet, a koji ne. Tehnike koje se primjenjuju kod nadziranog učenja su metoda potpornih vektora, linearna regresija, logistička regresija i naivni Bayesovi klasifikatori [5].

Nenadzirano učenje, međutim, radi na način da se računalu zadaju setovi podataka koji imaju samo ulazne vrijednosti, tj. nemaju definirane „odgovore“. Računalo u tom slučaju mora samostalno, tj. bez ljudskog nadzora, izvući informacije iz tih podataka prepoznavanjem uzoraka u tim podacima [6]. Dakle, može se reći da nenadzirano učenje služi kako bi se prepoznala dosad nepoznata struktura unutar skupa podataka i to funkcionira na način da pomoću raznih tehnika grupiranja, računalo prepoznaje neke aspekte po kojima su podaci slični, odnosno različiti te kako međusobno utječu jedni na druge. Neke od tehnika koje se koriste za ovu vrstu učenja su klasterizacija, otkrivanje anomalija i neuralne [5]. Po prijašnjem primjeru, algoritam će pronaći neke zajedničke karakteristike pojedinih učenika te ih grupirati po tome, npr. broj članova obitelji. Zbog toga je najbolja primjena nenadziranog učenja otkrivanje novih informacija unutar nekog skupa podataka, dok se nadzirano učenje više primjenjuje za predviđanje znanja .

Polu-nadzirano učenje ima istu svrhu kao nadzirano učenje i za sam proces učenja koriste se i označeni i neoznačeni setovi podatka [7]. Obično se prvo trenira sa manjom količinom označenih setova podataka te se nakon toga nastavlja proces učenja sa dosta većim brojem neoznačenih setova podataka. Ovaj način strojnog učenja se primarno upotrebljava zbog smanjivanja troškova koji se pojavljuje kod označavanja podataka te se tim procesom izbjegava. Visok trošak označavanja podataka pojavljuje se kod podataka za čiju obradu je potreban vješt stručnjak (npr. opisivanje audio segmenata) ili pak uključuje fizičku eksperimentaciju (npr. kreiranje trodimenzionalne strukture proteina) .

Ojačano učenje je vrsta strojnog učenja koja se često primjenjuje u robotici, industriji videoigara i navigaciji. Algoritam funkcionira na način da računalni program mora obaviti neki

zadatak (npr. uspješno navigirati kroz nivo video igre) sa obećanjem „nagrade“ [8]. Time će računalo izvršavati zadatak sa ciljem postizanja optimalnog rezultata. Ovaj pristup učenja razlikuje se od nadziranog učenja po tome što ne treba označene skupove podataka niti eksplicitno ispravljanje nezadovoljavajućih radnji, već je usredotočenost na pronalaženju ravnoteže između istraživanja, tj. isprobavanja novih metoda i eksploatacije već postojećeg znanja.

Rudarenje podataka (engl. Data Mining) je oblik analize podataka koji se postiže umjetnom inteligencijom i rudarenjem podataka, a odnosi se na izvlačenje informacija iz sortiranih, organiziranih ili grupiranih setova podataka [9]. Rudarenje podataka se izvodi na velikom broju podataka s ciljem dobivanja nekog znanja ili informacija i najčešće se primjenjuje u poslovnome svijetu, npr. za donošenje odluka na temelju dobivenih saznanja. Podaci na kojima se vrši rudarenje možda na prvi pogled ne sadrže neke važne informacije, ali kada se obrade može se vidjeti neka korelacija između svih podataka, kao što su informacije o kupcima, tržištu, itd. Rudarenjem podataka možemo izraditi modele kojima otkrivamo znanje ili ga pak predviđamo na temelju dobivenog modela.

Sami proces rudarenja se može svesti na tri koraka: pretprocesiranje podataka, rudarenje podataka i potvrda rezultata. Pretprocesiranje se odnosi na sastavljanje skupa podataka i čišćenje istog od „šumova“. Šum u skupu podataka može se definirati kao pojavljivanje nepoželjnih instanci podataka unutar skupa koje mogu znatno narušiti točnost modela koji se kreira u procesu rudarenja. Zhu i Wu [10] razlikuju dvije vrste šuma: atributni šum i klasni šum. Atributne šumove čine greške koje se pojavljuju pri zadavanju vrijednosti atributa entiteta, a to mogu biti netočne, nepostojeće i nepotpune vrijednosti atributa. Postoje dva izvora klasnih šumova: proturiječni unosi i pogrešne klasifikacije. Proturiječni unosi su svi unosi koji se pojavljuju više puta u skupu podataka, no imaju drugačije oznake. Pogrešnom klasifikacijom smatra se svaka instanca koja je označena netočnom klasom.

Nakon toga se obavlja samo rudarenje koje se u većini slučajeva dijeli na šest koraka: prepoznavanje anomalija, učenje pravila asocijacija, klasterizacija, klasifikacija, regresija i sažimanje. Prepoznavanje anomalija jest proučavanje zapisa podataka koji „strše“ od ostatka skupa podataka te zaključivanje radi li se o anomaliji ili o grešci. Učenjem pravila asocijacija pronalaze se poveznice između nekih podataka. Klasterizacija je definiranje grupa podataka „klastera“ koji su po nekim aspektima slični, recimo po godini rođenja. Klasifikacija je proces generalizacije strukture kako bi bila bolje prilagođena novim podacima. Regresija je kreiranje funkcija kojom pokušavamo što bliže prikazati model podataka kako bi se odredio odnos unutar skupa podataka. Sažimanje se odnosi na kompaktni prikaz skupa podataka koji često uključuje vizualni prikaz rezultata i generiranje izvješća.

3.2. Definicija analize sentimenta

Računalna analiza sentimenta, poznata još kao „rudarenje mišljenja“ jest područje rudarenja podataka kojima se analiziraju ljudska mišljenja, sentimenti, stavovi i emocije prema entitetima poput proizvoda, usluga, organizacija, pojedinaca, pitanja, događaja, tema i njihovih atributa [11, str. 1]. Rudarenje mišljenja obuhvaća opširno područje izazova od kojih će neki

biti opisani u ovome radu.

Iako obrada prirodnog teksta i lingvistika imaju dugu povijest u području informatike tek se ozbiljnija istraživanja na temu analize sentimenta prvi put pojavljuju 2000. godine, no sami izraz analize sentimenta najvjerojatnije prvi put spominju Nasukawa i Yi 2003., dok se izraz rudarenje mišljenje prvi put spominje u radu Dave-a i ostalih [11, str. 1]. Od tad je računalna analiza sentimenta se razvila u vrlo aktivno područje istraživanja. Razlog tome se krije u činjenici da je primjena rudarenja mišljenja opširna; praktički svugdje gdje se koristi prirodni jezik može se primijeniti na neki način računalna analiza sentimenta. Međutim, najbitniji razlog tolikog razvoja ovog područja je zbog porasta uporabe komercijalnih aplikacija. Naravno, veliku ulogu u porastu popularnosti rudarenja mišljenja ima sami rast korištenja društvenih mreža, gdje se može naći dosad nikad u povijesti viđena količina subjektivnih podataka. Bez tih podataka, dosta istraživanja ne bi bilo izvedivo. Stoga nije ni čudo da je brzi rast analize sentimenta direktno povezan sa rastom društvenih mreža.

3.3. Razine analize sentimenta

Tri glavne razine na kojima su se provodila istraživanja o analizi sentimenta su [11, str. 4]:

- Analiza na razini dokumenta
- Analiza na razini rečenice
- Aspektna analiza sentimenta

Cilj analize na razini dokumenta je saznati je li sentiment cijelog dokumenta pozitivan ili negativan. Na primjer, takav dokument može biti recenzija nekog proizvoda ili usluge. Kada bi pročitali recenziju, mogli bi dobiti dojam je li proizvod ili usluga dobar ili loš. Međutim, pretpostavka koja se veže za analizu na razini dokumenta je da se sentimenti pronađeni u dokumentu odnose samo na jedan entitet te da postoji samo jedan autor dokumenta. U suprotnome više nije moguće primijeniti analizu na dokumentnoj razini, već aspektnu analizu. Za ovu vrstu analize pretežito se primjenjuju metode koje rade na principu nadziranog strojnog učenja, iako postoje poneke metode koje funkcioniraju na temelju nenadziranog učenja.

Analiza na razini rečenice blisko je vezana s klasifikacijom subjektivnosti i njome se određuje ima pojedina rečenica pozitivan, negativan ili neutralan sentiment. Može se smatrati detaljnijom od analize na razini dokumenta jer pobliže analizira objekte na koje se odnosi sentiment. Kod ove razine analize susrećemo se s problemom kojim se bavi klasifikacija subjektivnosti, a to je razlika između subjektivnosti i sentimenta. Naime, rečenice se dijele na subjektivne i objektivne gdje subjektivne rečenice iskazuju neko osobno mišljenje ili poglede, dok objektivne rečenice iskazuju činjenične informacije. Komplikacija se pojavljuje zbog činjenice što objektivne rečenice mogu iskazivati sentiment, ali i subjektivne rečenice ne moraju iskazivati sentiment. Drugi problem se pojavljuje kod složenih rečenica koje mogu sadržavati više entiteta ili aspekata entiteta za koje može biti vezano više sentimenata. Još jedna prepreka kojom se susrećemo kod analize rečenica su komparativne rečenice. U rečenici „Zima

je bolja od ljeta.“ vidimo da postoji sentiment, no zbog toga što se radi o komparativnoj rečenici nije jasno je li sentiment pozitivan, negativan ili neutralan. Također, u rečenicama se mogu nalaziti riječi koje iskazuju suprotni sentiment od same rečenice. Većina se navedenih problema rješava na aspektno razini analize koja ide još više u dubinu.

Kao što je napomenuto u prijašnjem odlomku, aspektna se analiza provodi na još dubljoj razini od dokumentne i rečenične analize. Razlog tome je što u prijašnje dvije razine nije moguće dokučiti što točno se točno ljudima sviđa ili ne sviđa. Za razliku od promatranja jezičnih konstrukata (rečenice, dokumenti, odlomci, fraze, itd.), predmet promatranja na aspektnoj razini jesu sama mišljenja [11, str. 49]. Metoda se temelji na ideji da se mišljenje sastoji od sentimenta koji može biti pozitivan ili negativan i objekta mišljenja. Na taj način će se analizom dobiti precizniji rezultat, tj. biti će vidljivo na što se točno mišljenje odnosi. U rečenici „Volim ovaj automobil, iako troši dosta goriva“ može se prepoznati da općenito prevladava pozitivan sentiment, iako postoji i unutar nje negativan sentiment, zbog čega rečenica nije u potpunosti pozitivna. Sentiment vezan za glavni entitet, tj. automobil je pozitivan, dok je za sporedni entitet, potrošnja goriva, vezan negativan entitet. Dakle, cilj aspektne analize je prepoznati entitete i njihove aspekte ako postoje, a zatim pronaći sentimente vezane za njih. Premda je rudarenje mišljenja na aspektnoj razini preciznije i mnogo detaljnije od analize na razini dokumenta i rečenice, baš je zbog tih razloga kompleksnije i izazovnije za implementirati od te dvije razine koje su same po sebi već složene.

4. Primjeri primjene rudarenja mišljenjima

Mišljenja su jedan od najvažnijih utjecaja na ljudsko ponašanje te su zbog toga ključni u skoro svim ljudskim aktivnostima [12]. Također je poznato da su ljudi socijalna bića i kao takvi, prije nego što donesemo neku odluku, često tražimo ostale njihovo mišljenje. Tvrtke i organizacije koje žele uspješno poslovati traže mišljenja svojih kupaca o proizvodima i uslugama koje nude. Prije kupnje nekog proizvoda, potrošače zanimaju mišljenja ostalih korisnika koji su kupili taj proizvod kako bi dobili bolji dojam kakav je; za vrijeme političkih izbora građani će proučiti kakvo mišljenje imaju ostali sugrađani u vezi političkih kandidata i stranaka. Prije su pojedinci, kada su trebali mišljenje u vezi nečega, pitali svoje poznanike ili obitelj za njihovo stajalište u vezi entiteta, a kada su organizacije i poduzeća trebali proučiti mišljena potrošača o njima, oni bi provodili upitnike, istraživanja, fokusne grupe itd. Proučavanje stajališta potrošača i javnosti je toliko važno da postoje tvrtke koje se bave marketingom, javnim odnosima i vođenjem političkih kampanja .

Međutim, naglim rastom društvenih mreža, pojavila se mogućnost pregledavanja raznih mišljenja preko recenzija, blogova, foruma, blogova, objava po raznim društvenim mrežama, itd. Ako neka organizacija ili tvrtka želi saznati kakvo je mišljenje javnosti o njima, dovoljno je sakupiti sve te podatke sa društvenih mreža, profiltrirati ih te na kraju iz njih izvući sentimente. Postoji mnogo raznih primjena analize sentimenta, no za potrebe rada izabrane su tek samo neke, a to su:

- Proučavanje tržišta
- Povratne informacije korisnika
- Predviđanje političkih izbora

4.1. Proučavanje tržišta

Analiza sentimenta je iznimno korisna kada je potrebno proučiti stanje tržišta ili pak za proučavanje konkurencije. Pomoću nje, moguće je pratiti vlastiti brand na društvenim mrežama i ostalim Web mjestima gdje se mogu pronaći stajališta o vlastitome brandu i uspoređivati ga sa konkurencijom te time saznati kakvo je stanje tržišta [12]. Jedan od ključnih elemenata u vezi praćenju trendova jest vrijeme. To znači da je sakupljanjem podataka kroz određeno razdoblje i generiranjem izvješća koja sadrže sažet prikaz informacija o tržištu moguće dobiti bolji i precizniji prikaz stanja tržišta, a samim time i mogućnost predviđanja budućih trendova kako bi poslovanje bilo što efektivnije .

4.2. Povratne informacije korisnika

Kada organizaciju ili tvrtku zanimaju dojmovi o proizvodima ili uslugama koje nude ili pak mišljenje o samome brandu, mogu se poslužiti analizom sentimenta i izvući najbitnije informacije [12]. Recimo da tvrtka koju promatramo je aviokompanija i zanima ju kakav je općeniti

sentiment prema uslugama prijevoza koje oni nude. Proces analize povratnih informacija korisnika može izgledati tako da se nakon prikupljanja podataka u obliku objava sa društvenih mreža ili recenzija o letovima odvija aktivnost same analize sentimenta. Primjerice, u jednoj recenziji korisnik usluge prijevoza kaže sljedeće: „Let je bio ugodan, posluga je bila uljudna, no hrana koju nude mi se nije svidjela.“ Gledajući recenziju, mogu se razaznati tri entiteta: let, posluga i hrana. Glavni entitet je let, dok su posluga i hrana aspekti leta. Za let je vezan pridjev ugodan, za poslugu pridjev ugodna, a hrana je objekt koji se osobi nije svidio. Kada bi se ponovio ovaj proces milijun puta, mogli bi dobiti dobru sliku o usluzi leta te prepoznati koji su elementi dobro odrađeni, a koji nisu i gdje ima mjesta za poboljšanje. Ovakav pristup analizi također je i primjer aspektne analize sentimenta koja je obrađena u poglavlju 3.3.

4.3. Predviđanje političkih izbora

Opće je poznato da su društvene mreže odlična platforma za vođenje političkih kampanja i većina je političara toga svjesna. Razlog tome je naravno što je broj korisnika velik, ali i činjenica da su društvene mreže savršene za kreiranje vlastitih socijalnih mjehurića koji su ispunjeni personaliziranim sadržajem. To znači da korisnici mogu pratiti sadržaj točno onih političara i političkih stranki koje oni žele. Zbog toga su provedena mnoga istraživanja na povezanosti društvenih mreža sa političkim kampanjama te postoji li mogućnost iskorištavanja podataka sa društvenih mreža za predviđanje ishoda izbora. Jedno od takvih istraživanja proveli su Tumasjan, Andranik i sur. [13].

Istraživanje se temeljilo na saveznim izborima u Njemačkoj koji su se održali 27. rujna 2009., a platforma koja je bila odabrana za promatranje Twitter. Cilj istraživanja je bio saznati može li se iz objava sa Twittera dobiti informacije o javnom mišljenju i političkom krajoliku u off-line svijetu. Za potrebe istraživanja korišteno je 100 000 tvitova (objava na Twitteru) koji su bili objavljeni u narednih par tjedana prije samih izbora. Iako analiza sentimenta Twittera nije bila proučavana u tome razdoblju, postojala su razna istraživanja samog Twittera i korisnika koji ga koriste. Primijećeno je da iako na Twitteru postoji ogromna količina sadržaja i šuma, većina korisnika pozorno prati sadržaj onih računa na koje su pretplaćeni .

Svi tvitovi koji su bili prikupljeni za istraživanje spominjali su barem jednu od šest glavnih stranaka tadašnjih izbora ili pak političare koji su pripadali tim strankama. Sama analiza sentimenta provedena u LIWC2007 (Linguistic Inquiry and Word Count), softverom za analizu teksta razvijenog za procjenu emocionalnih, kognitivnih, i strukturnih komponenti uzoraka teksta koristeći psihometrički potvrđeni unutarnji rječnik.

Rezultati analize pokazali su da korisnici često prosljeđuju politički sadržaj svojim pratiocima. Istraživanjem je pronađena poveznica između broja tvitova povezanih sa strankom, tj. pažnje koju dobiva neka politička stranka i rezultata izbora 2009. godine. Naime, prosječna apsolutna pogreška (engl. mean absolute error) između postotka pažnje na Twitteru i konačnih rezultata je 1.65%, dok su rezultati šestero instituta za istraživanje političkih kampanja u prosjeku varirali između 1.1% i 1.7%. U zaključku, vidljivo je da se broj spominjanja može smatrati vjerodostojnim pokazateljem rezultata udjela glasova političkih izbora. Nadalje, broj spominja-

nja (engl. mentions) drugih političkih stranaka se pokazala kao dobra naznaka ideoloških veza ili pak vjerojatnosti kreiranja političkih koalicija između stranaka.

5. Prepreke u izvođenju analize sentimenta

Iako je područje analize sentimenta znatno napredovalo od svojih začetaka 2000-tih godina te je danas u širokoj primjeni i dalje postoje brojne prepreke i problemi zbog kojih analiza sentimenta još nije na ljudskoj razini. Neki su problemi koji se pojavljuju pri implementaciji metoda rudarenja mišljenjem navedeni već u prethodnim poglavljima. Velik broj komplikacija koje se pojavljuju jezične je prirode i računala koja su primarno rađena za obavljanje logičkih i matematičkih funkcija nemaju sposobnost i finoću rada s prirodnim jezicima poput ljudi.

Jedna od težih prepreka je postizanje analize sentimenta na razinama veće granuliranosti (dokument, poglavlje, odlomak, rečenica, itd.). Analiza na većem stupnju granuliranosti postaje sve složenija jer se moraju uzimati u obzir dodatni detalji. Na primjer, analiza dokumenta, koja je složena sama po sebi, jednostavnija je od aspektne analize sentimenta, zbog toga što se na aspektnoj razini definiraju aspekti nekog entiteta uz sami entitet, čega nema na razini dokumenta. Ova problematika biti će više objašnjena u sljedećem poglavlju.

Sljedeća problematika je prilično zanimljiva, a radi se o prepoznavanju stajališta autora teksta. Naime, automatska detekcija stajališta jedan je od novijih koncepata za koji su prvi put 2016. godine Mohammad, Sobhani i Kiritchenko [14] kreirali skup podataka objava sa Twittera koje su bile označene sentimentom i stajalištem. Program bi automatski trebao uočiti je li stajalište autora prema objektu promatranja pozitivno, negativno ili neutralno, tj. je li autor za, protiv ili indiferentan prema predmetu analiziranja. Ova vrsta detekcije je zanimljiva jer ima potencijala za širok spektar uporabe, npr. u područjima pretraživanja informacija, sažimanju teksta i logičkih posljedica. Moglo bi se, u stvari, tvrditi da otkrivanje stava donosi komplementarne informacije u analizi sentimenta, jer je češće čitatelju važniji stav autora prema određenim predmetima mišljenja od informacije je li autor, tj. govornik ljut ili sretan. Ljudima nije problem prepoznati stav autora, dok je to prilično zahtjevan pothvat za računala. Za uspješno prepoznavanje stava, automatski sustavi moraju često identificirati relevantne informacije koje se ne nalaze u tekstu. Taj proces je potpomognut aspektnom analizom sentimenta.

Analiza sentimenta se pretežito koristi kako bi se odredila polarnost teksta, dakle, moguće je detektirati pozitivan ili negativan sentiment, no kad bi se dodala još dimenzija, intenzitet, sustav bi mogao raspoznavati nijanse između pozitivnih osjećaja (sreća, ljubaznost, smirenost) i negativnih osjećaja (panika, mržnja, ljubomora). Russel je opisao upravo takav model, poznat pod nazivom cirkulacijski model [14]. Međutim, ne postoje neki značajni napreci u izgradnji automatskih sustava koji prepoznaju intenzitet osjećaja, tj. uzbuđenje. Prepoznavanje suptilnijih aspekata poput ovih bi potencijalno moglo pronaći veliku primjenu u industriji videoigara, javnom zdravstvu, vojnoj inteligenciji i poslovanju s klijentima gdje postoji potreba za razlikovanjem između pozitivnih osjećaja poput uzbuđenosti i sabranosti te negativnih osjećaja kao što su tuga i ljutnja.

Veliki problem pri analizi sentimenta predstavljaju jezični izrazi i stilske figure. Jezični izrazi nisu kompozitni, tj. njihovo značenje se ne može izvući iz njihovih sastavnica, već se mora promatrati cijeli izraz. Najpoznatiji i najzanimljiviji izrazi u analizi sentimenta su ironija i sarkazam. U kontekstu rudarenju mišljenja, sarkastični i ironični izrazi imaju pozitivan tekst, ali

negativno mišljenje. Također vrijedi i suprotno. Upravo je to razlog zašto i najbolji sustavi imaju problema sa prepoznavanjem sarkazma i ironije. Napravljena su mnoga istraživanja u ovome području i zaključeno je da su ironični izrazi često negativnog sentimenta. Nadalje, Tsur i suradnici su 2010. godine tvrdili su da se sarkastične rečenice često pojavljuju u skupu sa ostalim sarkastičnim rečenicama. Gonzalez-Ibanez i suradnici [11, str. 44] u svome su svome radu proučavali analizu sentimenta u kontekstu objava s Twittera s ciljem raspoznavanja sarkastičnih i nesarkastičnih izjava koje izražavaju pozitivne, odnosno negativne sentimente. Model se temeljio na nadziranom učenju te metodi potpornih vektora i logističke regresije. Pokusni rezultati su pokazali da je problem detekcije sarkazma i ironije prilično izazovan. Naime, najviši postignuti postotak točnosti zabilježen je bio svega 57%; time se može zaključiti da je potrebno još dosta rada kako bi se usavršilo ovo područje do zadovoljavajuće razine.

Analiza sentimenta primjenjiva na više domena možda je jedan od najsloženijih problema kojima se trenutno susreću stručnjaci ovog područja. Vrijedi činjenica da je rudarenje mišljenja jako ovisno o domeni. Kada se metoda koja je koristila podatke za treniranje jedne domene primjeni na drugoj domeni, često dolazi do znatnih gubitaka u preciznosti algoritma. Razlog je jezična barijera; u jednoj domeni neki pozitivan izraz može u drugoj domeni predstavljati nešto negativno. Recimo da treniramo model skupom podataka koji se sastoji od recenzija usisavača, a nakon toga model testiramo na skupu podataka kojeg čine recenzije zvučnika. Tada bi, na primjer, glasnoća predstavljala kod zvučnika predstavljala nešto pozitivno, dok kod usisavača glasnoća je nepoželjna i zbog toga će bi sentiment trebao biti negativan. Ideja je da se razvije metoda koja bi funkcionirala u svim domenama bez gubitaka u performansama. Međutim, trenutno je prilično komplicirano kreirati model koji ima dobre performanse u više domena, pogotovo ako model upotrebljava nenadzirano učenje. U svome istraživačkome radu, Al-Moslmi, Tareq i sur. [15] zaključuju da ne postoji dobar skup podataka koji se može iskoristiti da se istrenira model koji bi radio prihvatljivo na više domena. Također navode nekoliko potencijalnih rješenja za više-domensku analizu. Prvo koje pokazuje potencijal je strojno učenje s više pogleda. Radi na principu spajanja više polu-nadziranih metoda učenja kako bi se dobili različiti pogledi nad istim podacima. Kao drugo potencijalno rješenje izdvajaju modele koji mogu naučiti značajke koje ne ovise o domeni iz više različitih izvora, tj. domena. Treće rješenje koje bi se moglo pokazati obećavajućim su modeli razvijani na način da su sposobni prilagoditi i koristiti leksikon subjektivnosti neovisan o domeni širenjem polariteta (pozitivan ili negativan sentiment) kroz pivot, umjesto od izvora do ciljane domene. Četvrto, navode kako postoji prostor za daljnji razvoj modela koji koriste nove ugrađene riječi i tehnike prijenosa značajki. Posljednje ponuđeno rješenje je izrada modela uz pomoć dubok učenja u kombinaciji sa raznim tehnikama.

Posljednja prepreka koja se obrađuje u ovome poglavlju je jezična podrška. Dva opće prihvaćena pristupa obradi prirodnog jezika su simbolički i statistički. Ukratko, simbolički se pristup temelji na postojanju unaprijed definirane gramatike koja se koristi za analizu podataka. Gramatika je ispunjena raznim poznatim gramatičkim uzorcima i njihovim značenjem koji se zatim pretražuju u obrađenim podacima. Međutim, zbog ograničenja simboličkog pristupa, istraživači su se okrenuli drugom pristupu, statističkom pristupu. Statistički pristup temelji se na strojnom učenju i rudarenju podataka. Glavna značajka ovog pristupa je postojanje korpusa

koji su, u stvari, skupovi tekstova čije riječi mogu biti označene jezičnim oznakama (npr. imenica, glagol, pridjev) ili klasificirane po konceptima (npr. osoba, zgrada, itd.) i algoritam za učenje koji pomoću korpusa i neoznačenih podataka uči nove koncepte. Iako se statistički pristup mijenjao kroz godine, svejedno je ostao temelj za sve moderne NLP algoritme. Međutim, problem je što najveću jezičnu podršku imaju upravo oni jezici koji imaju veliki broj korisnika, npr. engleski i kineski, dok jezici koji nisu toliko zastupljeni, poput jezika s područja središnje i istočne Europe, zaostaju u tome području jer potrebni alati za analizu sentimenta ne postoje ili nisu na adekvatnom stupnju razvijenosti [16]. Na sljedećim se grafikama može vidjeti razina tehnološke podržanosti za govorne i tekstualne resurse te analizu teksta za trideset europskih jezika.

Slika 1: Podrška tekstualnih i govornih resursa za 30 europskih država [16, str. 56]

Odlična podrška	Dobra podrška	Osrednja podrška	Djelomična podrška	Slaba podrška/ nema podrške
	Engleski	Francuski	Baskijski	Hrvatski
		Nizozemski	Bugarski	Estonski
		Njemački	Češki	Islandski
		Španjolski	Danski	Irski
		Talijanski	Finski	Latvijski
			Galicijski	Litvanski
			Grčki	Malteški
			Mađarski	
			Norveški	
			Poljski	
			Portugalski	
			Rumunjski	
			Slovenski	
			Slovački	
			Srpski	

Slika 2: Podrška resursa analize teksta za 30 europskih država [16, str. 56]

6. Kako se izvodi analiza sentimenta?

U ovom se poglavlju obrađuju načini izvođenja analize sentimenta kako bi bilo jasnije što se zapravo događa iza kulisa analize sentimenta. Analiziraju se prednosti i mane svake pojedine metode, njihove značajke i procese koji se odvijaju kako bi se analiza izvela. Algoritmi za analizu sentimenta se dijele na dvije glavne vrste, a to su:

- sustavi temeljeni na pravilima
- automatski sustavi

6.1. Sustavi temeljeni na pravilima

Sustavi temeljeni na pravilima (engl. rule-based systems) su sustavi koji analiziraju tekst prateći određeni set pravila koje je zadao sami programer [12]. Ovi sustavi su podosta jednostavniji od automatskih sustava jer automatski sustavi primjenjuju strojno učenje. Pravila koja prate ovi sustavi mogu biti prilično raznolika, ali većina funkcionira na sličan način. Prvi preporučeni korak je predobrada podataka. Predobrada služi kako bi se uredio tekst kako bi se uklonile riječi koje ne pridonose analizi, tzv. zaustavne riječi (engl. stop words). Te riječi su najčešće riječi nekog jezika i baš se zbog svoje učestalosti odstranjuju iz dokumenta jer ne pridonose značajno analizi, a u nekim slučajevima čak i čine nepotreban šum. Ako tekst sadrži neke specijalne znakove koji ne pridonose analizi, kao što su HTML elementi i slično, moraju se također ukloniti iz teksta. Daljnji procesi koji se mogu primjenjivati su korjenovanje (engl. stemming), tokenizacija (engl. tokenization), označavanje dijelova govora (engl. part-of-speech tagging) te kreiranje leksikona [16].

Korjenovanje je jedan od načina izvođenja procesa koji se u području obrade prirodnog teksta naziva normalizacija. Normalizacija je proces vraćanja riječi u svoj kanonski oblik, tj. osnovni oblik. To znači da će se u hrvatskom jeziku, imenice vraćati u nominativ, glagoli u infinitiv, itd [17]. Normalizacija je korisna, a neki bi rekli, čak i nužna jer se tim procesom grupiraju riječi istog značenja, a različitog oblika. Bez normalizacije, riječi „Mariju“, „Mariji“ i „Marija“ tretirale bi se kao različite riječi, iako bi bilo poželjnije da se tretiraju kao ista. Osim korjenovanja, druga popularna tehnika normalizacije je lematizacija.

Kao što je navedeno u prošlom odlomku, normalizacija vraća riječ u svoj osnovni oblik ili korijen. Proces korjenovanja radi na način da se sa riječi uklanjaju uobičajeni sufiksi i prefiksi. Međutim, procesom korjenovanja moguće je dobiti riječi koje nisu prave riječi. Npr. kada bi korjenovali riječi „trouble“, „troubled“ i „troubling“, za rezultat bi dobili riječ „troubl“ koja nije zapravo riječ, ali svejedno služi kako bi povezala ove tri riječi. Kako bi se izbjegli takvi rezultati, koristi se drugi način normalizacije, lematizacija. Lematizacija, za razliku od korjenovanja, radi pomoću korpusa koji sadrži sve oblike riječi i njihov korijen. Na taj se način može dobiti osnovni oblik riječi koji zapravo postoji u rječniku određenog jezika. Iako se lematizacija čini kao očito rješenje za normalizaciju, postoji problem, a to su performanse. Naime, kako lematizacija za dobivanje osnovnog oblika riječi koristi korpus, potrebno je određeno vrijeme za pronalaženje

riječi unutar korpusa što rezultira gubitkom u performansama. Kako onda izabrati metodu za normalizaciju? Za program gdje je ključna brzina, preporuča se korištenje korjenovanja, dok je u programima gdje je bitna preciznost, bolja opcija odabrati lematizaciju.

Tokenizacija je proces razdvajanja teksta na riječi, tj. tokene. Proces radi na način da se na mjestima gdje se pojavljuju određeni simboli poput razmaka, točaka i zareza odvajaju riječi od teksta i dodaju u polje. Pomoću primjera će biti prikazano kako točno to funkcionira.

Ako imamo rečenicu „Danas je lijep i sunčan dan“ i na njoj primijenimo proces tokenizacije, izlaz će onda izgledati otprilike ovako: „[’Danas’, ’je’, ’lijep’, ’i’, ’sunčan’, ’dan’].

Označavanje dijelova govora je proces u kojemu računalo iterira kroz tekst te daje oznake kojoj vrsti riječi (npr. imenica, pridjev, glagol) pripada određena riječ. U analizi sentimenta je označavanje dijelova govora iznimno korisno jer je pomoću tih informacija moguće doći do novih zaključaka.

Nakon što se izvrši pretprocesiranje teksta, pomoću leksikona koji sadrži popis pozitivnih (npr. dobar, najbolji, sretan) i negativnih riječi (npr. loš, najlošiji, tužan) nekog jezika uspoređuje se svaka riječ teksta. Ako riječ pripada skupu pozitivnih riječi, brojač pozitivnih riječi povećava se za jedan, no ako se riječ pojavljuje u leksikonu negativnih riječi, brojač negativnih riječi se povećava. Međutim, ako se radi o neutralnoj riječi, brojač neutralnih riječi se povećava. Nakon što algoritam završi iteraciju kroz tekst, provjerava se stanje brojača. Sentiment teksta se smatra pozitivnim ako je broj pozitivnih riječi veći od broja negativnih i neutralnih, negativnim ako je broj negativnih riječi veći od broja pozitivnih i neutralnih te neutralan ako je broj neutralnih riječi najveći.

Analiza sentimenta pomoću sustava temeljenog na pravilima ima svoje prednosti i mane. Prednost je jednostavnost algoritma jer osim pretprocesiranja teksta, sami algoritam za određivanje sentimenta je prilično jednostavan i jasan. Međutim, zbog te jednostavnosti se može reći i da je sustav podosta naivan. Naime, kako se sentiment određuje isključivo po leksikonu pozitivnih i negativnih riječi, program će u nekim slučajevima krivo tumačiti sentiment jer su jezici složeni i nije moguće odrediti sentiment po individualnim riječima. Na primjer, rečenicama koje sadrže riječ „ali“ često se može promijeniti sentiment iz negativnog u pozitivni i obratno, što je prekomplicirano za ovaj sustav. Teoretski je moguće izgraditi sustav temeljen na pravilima koji graniči svojom preciznošću automatskim sustavima, ali taj bi sustav bio vrlo složen i vjerojatno imao loše performanse.

6.2. Automatski sustavi

Kada pričamo o modernoj analizi sentimenta uglavnom se misli na analizu koja radi automatskim pristupom, također poznatim kao statistički pristup [18]. Statistički pristup radi na principu primjene statistike i strojnog učenja. Strojno učenje se koristi kako bi se obradila velika količina teksta iz koje se program uči kako bi raspoznao sentimente u riječima, rečenicama, odlomcima, itd. Statističkim metodama koje se primjenjuju u strojnom učenju kreiraju se modeli pomoću kojih se određuje imaju li riječi, odnosno izrazi, pozitivan ili negativan sentiment. Automatski sustavi najčešće se izrađuju nadziranim strojnim učenjem, no postoje neki modeli

koji su izgrađeni polu-nadziranim i nenadziranim pristupima.

6.2.1. Vektorizacija teksta

Prvi korak u automatskom pristupu je pretprocesiranje teksta kao i kod pristupa temeljenom na pravilima. Kao i u prijašnjem pristupu, vrijede sve metode za uređivanje i filtriranje teksta kako bi se nastavila obrada podataka. Rezultat pretprocesiranja teksta je uređeni skup podataka, tj. korpus.

Nakon obrade skupa podataka, sljedeći korak je pretvaranje teksta u oblik koji računalo razumije. Naime, tekst nije računalu razumljiv podatak i zbog toga ga je potrebno pretvoriti u nešto što će razumjeti i raspoznavati. U tu svrhu se koriste vektori, a proces se naziva vektorizacija teksta. Modeli koji se često primjenjuju u tu svrhu su „Bag-of-words“, TF-IDF i Word2Vec, no za potrebe ovog rada, obrađena su samo prva dva modela.

6.2.1.1. „Bag-of-words“ model

Ovaj model prilično je popularan zbog svoje jednostavnosti i fleksibilnosti, ali i svoje moćnosti. Naziv dobiva iz svojeg svojstva da se sve informacije o redoslijedu i strukturi riječi unutar dokumenta gube; bit cijelog modela je bilježenje koliko se puta neka riječ pojavljuje u dokumentu neovisno o lokaciji. Najveći značaj se daje onoj riječi koja se najviše pojavljuje u dokumentu, dok se najmanji značaj daje riječi koja se najrjeđe pojavljuje. Može se reći da na taj način se riječi „izvlače iz vreće“ kada se pojavi potreba te iz toga proizlazi naziv „vreća riječi“. Model je pojašnjen konkretnim modelom radi boljeg shvaćanja načina rada [19]–[21].

Prvi korak vektorizacije pomoću ovog modela je skupljanje podataka. Izvor podataka može biti rečenica, pjesma, dokument, knjiga pa čak i cijela biblioteka knjiga. Radi primjera korištena su dva odlomka pjesme Antuna Branka Šimića „Opomena“:

„Čovječe pazi

da ne ideš malen

ispod zvijezda!

Pusti

da cijelog tebe prođe

blaga svjetlost zvijezda!“

Nakon toga, sve riječi se pretvaraju u tokene i kreira se rječnik koji sadrži sve riječi dokumenta. U ovom će slučaju rječnik biti popunjen sljedećim riječima: čovječe, pazi, da, ne, ideš, malen, ispod, zvijezda, pusti, cijelog, tebe, prođe, blaga, svjetlost. Ukupan broj riječi koje se nalaze u rječniku je 14 te se rječnik može prikazati vektorom koji se može koristiti za reprezentaciju dokumenta pomoću vektora. Ako se prvi odlomak promatra kao zaseban dokument onda će njegov vektorski oblik biti [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0], dok će drugi odlomak biti prikazan vektorom [0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1]. Kao što se može vidjeti, vektor je trenutno relativno malen, no kada se vektorizacija primjeni na nekom

većem skupu podataka, vektor može postati iznimno velik. Takvi vektori se često sastoje od mnogo nula pa se nazivaju rijetkim vektorima (engl. sparse vectors); zbog njih se povećava potrošnja računalnih resursa i memorije pa je u interesu na neki način smanjiti veličinu rječnika. Rječnik se može smanjiti u fazi pretprocesiranja podataka pomoću normalizacije, ignoriranja interpunkcijskih znakova i uklanjanjem riječi koje se često pojavljuju u dokumentima poput nekih veznika, priloga, itd.

Poboljšanje performansi modela može se postići grupiranjem riječi, tj. uporabom n-gramova. Na taj se način može dobiti bolji kontekst iz dokumenta, ali i smanjiti rječnik. Riječi koje stoje samostalno zovemo unigrami, dok su parovi riječi bigrami i sve grupacije od više riječi n-grami. Npr. prvi odlomak kreirao bi sljedeće bigrame: čovječe pazi, pazi da, da ne, ne ideš, ideš malen, malen ispod, ispod zvijezda. U analizi sentimenta su n-gramovi iznimno korisni jer je moguće preciznije odrediti sentiment ako se uoči da se određena kombinacija riječi često pojavljuje u dokumentima pozitivnog, odnosno, negativnog sentimenta.

Iako je ovaj model jednostavan i fleksibilan, najveća mana su mu performanse. Naime, za svaku novu riječ vokabular se povećava, a time se povećava i vektor vokabulara što rezultira „rijetkom matricom“ koja uvelike ruši performanse modela. Također je važno zamijetiti da model vreće riječi ne sadrži nikakve informacije o gramatici u rečenicama niti redoslijedu riječi unutar teksta [21].

6.2.1.2. TF-IDF

TF – IDF je oznaka za učestalost izraza (engl. term frequency) i inverzna frekvencija dokumenta (engl. inverse document frequency) te predstavlja statistiku o važnosti pojedinih riječi ili izraza u skupu podataka [22].Iz ove definicije već je moguće primijetiti napredak naspram modela vreće riječi jer se za razliku od same učestalosti pojavljivanja riječi promatra i važnost. TF – IDF model ima široku uporabu u rudarenju teksta, pronalaženju podataka i korisničkom modeliranju.

Učestalost izraza se može računati na više načina; moguće je koristiti sami broj pojavljivanja u skupu podataka $f(t, d)$ no postoje elegantnije metode. Jedna od njih je logaritamski skalirana učestalost izražena sljedećom formulom:

$$tf(t, d) = log(f(t, d) + 1)$$

Drugi izraz koji se mora računati je inverzna frekvencija dokumenta (IDF). IDF se koristi kako bi se odredilo koliko informacija pruža riječ ili izraz, npr. je li učestala ili rijetka unutar skupa podataka, tj. korpusa. Računa se tako da se logaritamski skalira inverzni razlomak dokumenata koji sadrže riječ ili izraz. To znači da se ukupni broj dokumenata dijeli sa brojem dokumenata koji sadrže traženu riječ ili izraz te se nakon toga koeficijent logaritmira u bazi deset. Formula je prikazana na sljedeći način:

$$idf(t, D) = \log \frac{N}{|\, d \in D : t \in d \,|}$$

N predstavlja ukupni broj dokumenata u skupu podataka $N = |D|$

$d \in D : t \in d$ predstavlja broj dokumenata gdje se izraz t pojavljuje. Ako se izraz ne pojavljuje u korpusu, nazivnik će biti nula što nije dopušteno stanje tako da se često izraz još zbraja s 1: $1 + |\, d \in D : t \in d\,|$.

TF – IDF računa se množenjem učestalosti izraza i inverzne frekvencije dokumenta:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

Visoka vrijednost u TF – IDF dobiva se visokom frekventnosti unutar dokumenta i niskom frekventnosti dokumenata koji sadrže izraz; na taj se način filtriraju izrazi koji su učestali u dokumentima. Kako je omjer unutar IDF logaritamske funkcije uvijek veći ili jednak 1, vrijednost IDF-a uvijek je veća ili jednaka 0. Što se izraz češće pojavljuje u dokumentima, to se više omjer unutar logaritma približava 1, a time se IDF i TF-IDF sve više približavaju 0 [22].

6.2.2. Klasifikacijski algoritmi

Nakon primjene jedne od metode vektorizacije teksta, sljedeći korak u automatskoj analizi sentimenta je klasificiranje skupa podataka pomoću klasifikacijskih algoritama. U statistici, klasifikacija je postupak identificiranja kojoj kategoriji u skupu kategorija pripada promatrani entitet i to na temelju skupa podataka koji služi za treniranje algoritma gdje sve instance imaju određenu kategoriju [23]. Primjer iz prakse može biti svrstavanje e-maila u klasu „spam“ ili „ne-spam“, dok u kontekstu analize sentimenta može biti pozitivan ili negativan sentiment. Klasifikacija predstavlja primjer računalnog prepoznavanja uzoraka. U kontekstu strojnog učenja, klasifikacija se smatra primjenom nadziranog učenja. Odgovarajuća metoda kod nenadziranog učenja je klasterizacija, postupak gdje se podaci grupiraju u kategorije na temelju neke sličnosti.

Često se pojedinačni entiteti promatraju u skupu mjerljivih svojstava, koja mogu biti kategorična (npr. krvne grupe), ordinalna (npr. veliko, srednje, malo), nominalna (npr. broj pojava riječi u tekstu) i kontinuirana (realni brojevi poput visine). Algoritme koji provode klasifikaciju nazivamo klasifikatorima. Postoji mnogo vrsta klasifikatora, no za potrebe ovog završnog rada obrađena su samo dva klasifikatora koji se koriste pri demonstraciji u poglavlju 7.2., a to su multinomijalni Naivni Bayesov klasifikator i linearna metoda potpornih vektora (engl. Linear Support Vector Machines).

6.2.2.1. Multinomijalni naivni Bayesov klasifikator

Naivni Bayesovi klasifikatori pripadaju obitelji jednostavnih probalističkih klasifikatora koji se temelje na Bayesovom teoremu i „naivnoj“ pretpostavci o nezavisnosti između značajki te se smatraju jednim od najjednostavnijih modela Bayesove mreže [24]. Unatoč tome što su jednostavni, Bayesovi klasifikatori su dokazano brzi, učinkoviti i precizni, zbog čega vrlo dobro funkcioniraju za rješavanje problema u području obrade prirodnog jezika. Da su Naivni Bayesovi klasifikatori probalistički znači da se pomoću njih može izračunati vjerojatnost vrijednosti svojstva određenog teksta [25].

Bayesov teorem, još poznat i kao Bayesov zakon, opisuje vjerojatnost događaja temeljem prethodnih saznanja koja su vezana uz sami događaj [26]. Na primjer, ako postoji saznanje da se vjerojatnost pojavljivanja zdravstvenih poteškoća povećava s čovjekovom dobi, Bayesovim se teoremom može odrediti vjerojatnost puno preciznije ako se zna dob osobe, nego ako se pretpostavlja da je osoba prosječne dobi stanovništva.

Matematički izraz Bayesovog teorema glasi:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad P(B) \neq 0$$

A i B predstavljaju događaje, a $P(A|B)$ predstavlja vjerojatnost da se dogodi A, u slučaju da je B istinit. $P(B|A)$ predstavlja vjerojatnost događanja B ako je A istinita tvrdnja. Na kraju, $P(A)$ i $P(B)$ predstavljaju vjerojatnosti da se dogodi A, odnosno B; također su poznate kao granične vjerojatnosti.

Naivni Bayes jednostavna je tehnika za konstrukciju klasifikatora, modela koji dodjeljuju oznake klasa promatranim instancama, koje su predstavljene kao svojstveni vektori nekog modela, pri čemu se oznake klase preuzimaju iz određenog skupa. Međutim, ne postoji samo jedan algoritam za treniranje takvih klasifikatora, već postoji obitelj klasifikatora koji rade na temelju jednog principa: svi naivni Bayesovi klasifikatori uzimaju kao pretpostavku da je vrijednost određenog svojstva neovisna od vrijednosti bilo kojeg drugog svojstvo, s obzirom na promatranu instancu. Na primjer, voće se može smatrati narančom ako je okruglo, narančaste boje i promjera 8 centimetara. Naivni Bayesov klasifikator će smatrati da svako ovo svojstvo neovisno pridonosi vjerojatnosti da je određeno voće naranča, unatoč mogućim povezanostima između svih tih svojstava.

Multinomijalni naivni Bayesov klasifikator, uzorci (svojstveni vektori) predstavljaju učestalosti u kojima su određeni događaji generirani multinomom $(p_1, \ldots, p_n)$ gdje p_i predstavlja vjerojatnost da se događaj i dogodi. U slučaju da postoji više klasa događaji su generirani sa K multinoma. Tada, svojstveni vektor $x = (x_1, \ldots, x_n)$ predstavlja histogram, gdje x_i broji koliko je puta događaj i primijećen u određenoj instanci. Ovaj se model najčešće koristi za klasificiranje dokumenata; pojavljivanje riječi unutar pojedinog dokumenta smatra se događajem. Vjerojatnost opažanja histograma x zadana je sljedećim izrazom:

$$p(x|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{xi}$$

Multinomijalni naivni Bayesov klasifikator postaje linearan kada se izrazi u logaritamskom prostoru:

$$\begin{aligned} \log p(x|C_k) &\propto \log(p(C_k) \prod_{i=1} {p_{ki}}^{x_i}) \\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki} \\ &= b + w_k^\top x \end{aligned}$$

gdje je $b = \log p(C_k)$, a $w_{ki} = \log p_{ki}$

Ako se određena klasa i svojstveni vektor nikada ne pojave zajedno u skupu podataka za treniranje, tada će procjena temeljena na učestalosti biti jednaka nuli, zato što je procijenjena vjerojatnost izravno proporcionalna broju pojavljivanja vrijednosti karakteristike. Tu nastaje problem jer će kao posljedica množenja svih vjerojatnosti biti brisanje svih informacija o ostalim vjerojatnostima. Rješenje za taj problem je uvođenje ispravljanja vrijednosti pomoću konstante tako da niti u kojem slučaju vrijednost vjerojatnosti bude točno nula. Ovaj se način reguliranja naivnog Bayesovog klasifikatora naziva Lindstoneovo zaglađivanje ili Laplaceovo zaglađivanje ako je vrijednost konstante jednaka jedan. Zaglađivanje se izražava na sljedeći način:

$$\widehat{\theta} = \frac{x_i + \alpha}{N + \alpha d}, \quad (i = 1, \ldots, d)$$

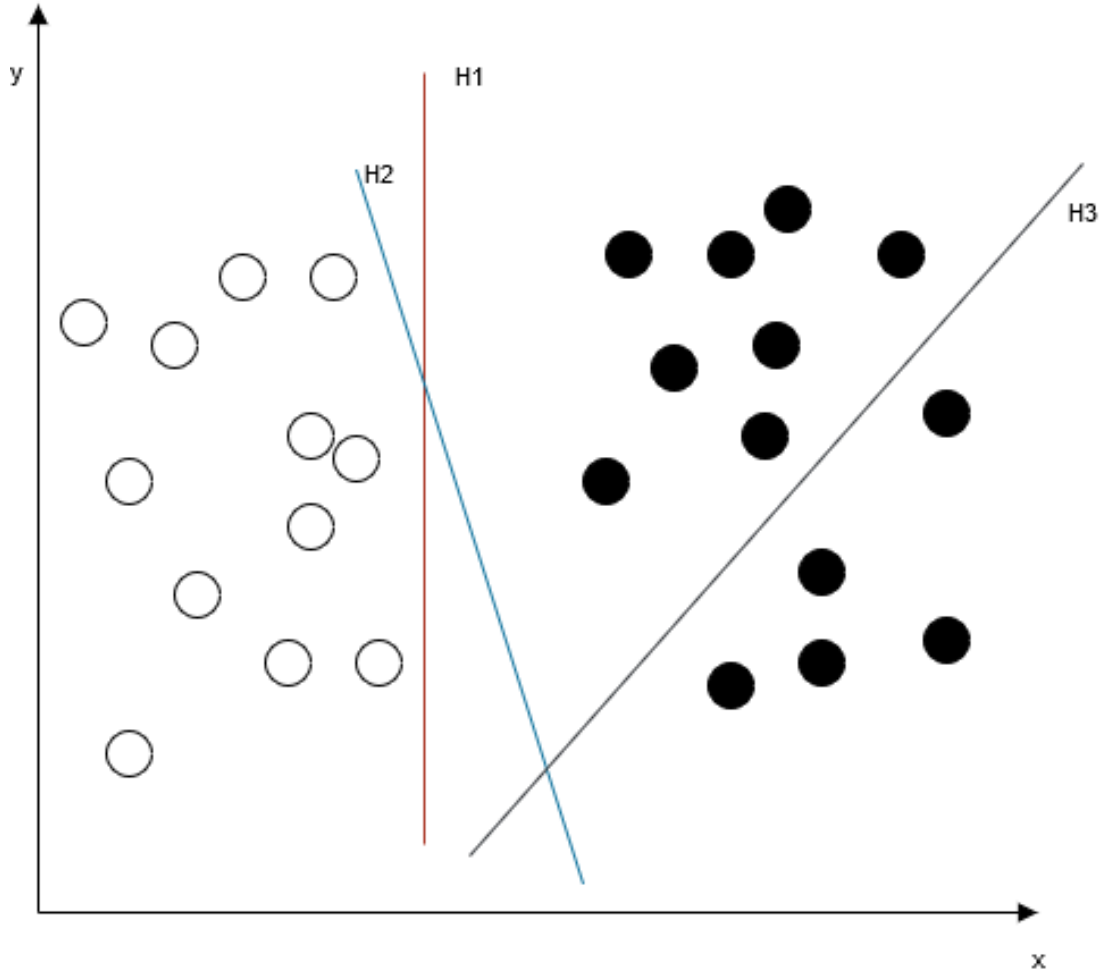
Svojstveni vektor $x = \langle x_1, x_2, \ldots, x_d \rangle$ iz multinomijalne distribucije ima N pokušaja, a konstanta je predstavljena simbolom α. Kada je $\alpha = 0$, tada ne postoji zaglađivanje, a $\alpha = 1$ kada se prati Laplaceovo pravilo nasljeđivanja.

6.2.2.2. Linearna metoda potpornih vektora

Metoda potpornih vektora (engl. support-vector machines, SVM) jedan je od najrobusnijih modela predviđanja koja se koristi u nadziranom učenju za potrebe klasifikacije i analize regresije. SVM građena je kako bi klasificirala dvije grupe podataka na način da se pomoću označenog skupa podataka za treniranje algoritma izgradi model koji nove instance po njihovim značajkama ili u jednu ili drugu klasu [27]. Iako je rađen da raspoznaje između samo dvije skupine podataka, SVM se može koristiti za klasifikaciju više klasa. Tada radi po principu jedna klasa – ostale klase, dok inače radi na principu jedna klasa – druga klasa. Za razliku od naivnog Bayesovog klasifikatora, SVM klasifikator nije probalistički, već linearni binarni klasifikator, iako postoje neke metode koje koriste SVM kao probalistički klasifikator.

SVM model svaku instancu podatka postavlja u n-dimenzionalni prostor, gdje n predstavlja broj značajki podatka (npr. ime, prezime, dob, itd.). Kako dokumenti sadrže samo tekst, za značajke se koriste učestalosti riječi u dokumentu kod metode vreće riječi, odnosno njihova važnost u slučaju TF-IDF. Nakon što se svi podaci postave u prostoru kao n-dimenzionalni vektori, potrebno je na neki način napraviti podjelu između dvije zadane klase, tj. kategorije. Rješenje za taj problem je razdvajanje klasa pomoću konstrukcije hiperravnina ili skupa hiperravnina unutar prostora s velikim ili beskonačnim brojem dimenzija [28]. Hiperravnina predstavlja vektorski potprostor koji je za jednu dimenziju manji od vektorskog prostora kojemu pripada, npr. u trodimenzionalnom prostoru će hiperravnina biti dvodimenzionalna. Postoji bezbroj mogućih hiperravnina koje se mogu konstruirati kako bi se razdvojile dvije klase pa kako izabrati najbolju. Najbolja hiperravnina je ona koja razdvaja dvije najbliže točke suprotnih klasa na takav način da je razmak, tj. margina između njih maksimalna. Razlog tome je što veća margina rezultira manjom pogreškom generalizacije klasifikatora. Međutim, postoje situacije u kojima nije moguće konstruirati linearnu hiperravninu jer se točke klasa preklapaju. Tada se koristi „jezgreni trik“ koji uvelike olakšava tu radnju. Naime, kako bi nelinearno podijelili klase potrebno je prikazati

točke u višoj dimenziji nego što već jesu. To se postiže uz pomoć skalarnih vektora kojima se povećavaju performanse i jezgrene funkcije.



Slika 3: Prikaz određivanja margine u dvodimenzionalnom prostoru [27]

Prema slici 3 može se vidjeti prikaz podataka dvaju klasa u dvodimenzionalnome prostoru te određivanje hiperravnine (u ovom slučaju pravca) na temelju što veće margine. Kako je već prije u tekstu navedeno, moguće je konstruirati mnogo hiperravnina koje bi mogle podijeliti dvije grupe podataka, no optimalna je ona koja radi što veći razmak između dvije najbliže točke različitih klasa. Na primjeru je moguće vidjeti da je od ponuđenih hiperravnina najbolja H1 jer je tada margina najveća.

7. Izrada sustava za analizu sentimenta

U ovome se poglavlju primjenjuje svo znanje koje je obrađeno kroz završni rad.Prikazuje se izrada sustava analize sentimenta temeljenog na pravilima i automatski sustav. Za sustav temeljen na pravilima koristi se biblioteka Reldi (engl. Regional Linguistic Data Initiative) koja sadrži naredbe za rad sa hrvatskim jezikom koja će, uz leksikon pozitivnih i negativnih riječi, poslužiti za analizu rečenica na hrvatskome jeziku. Nakon toga, objašnjava se programski kod za automatski sustav analize sentimenta. Opisuje se pretprocesiranje teksta, za kojim slijedi vektorizacija teksta i na kraju izrada klasifikatora te analiza rezultata. Cijeli je kod dostupan na Github-u [1].

7.1. Sustav temeljen na pravilima

Kao što je ranije spomenuto, ovaj sustav radi na hrvatskome jeziku te služi za analizu jednostavnijih rečenica, makar se može koristiti i na nešto većim tekstovima. Osim Reldi, za potrebe ovog algoritma koristile su se biblioteke „json“ i „re“. Iz naziva se može zaključiti da se biblioteka json koristi za rad sa JSON datotekama, dok re služi za rad sa regularnim izrazima, no njihova će točna svrha biti kasnije objašnjena.

U opisu sustava temeljnim na pravilima spomenuto je kako mogu raditi na principu brojača pozitivnih i negativnih riječi i upravo tako će raditi i ovaj sustav. U isječku koda 7.1. nalazi se početna konfiguracija programa gdje se dodaju potrebne biblioteke, postavljaju početne vrijednosti za brojače pozitivnih i negativnih riječi. Prije same analize sentimenta, tekst se lemitizira i sprema u JSON obliku radi lakšeg iteriranja po tekstu. Također se uključuju rječnici pozitivnih i negativnih riječi koji služe za usporedbu pojedinih riječi.

Isječak koda 7.1: Inicijalizacija i postavljanje početnih vrijednosti varijabli

```
import json, re
from reldi.tagger import Tagger
positive = 0
negative = 0

t=Tagger('hr')
t.authorize('my_username','my_password')

data = json.loads(t.tagLemmatise(u'Ovaj␣program␣dobro␣radi'.encode('utf8')))['lemmas
    ']['lemma']

dict_positive = open('positive_words_hr.txt','r', encoding='utf8')
words_positive = dict_positive.read()
dict_negative = open('negative_words_hr.txt','r', encoding='utf8')
words_negative = dict_negative.read()
```

Isječak koda 7.2. prikazuje iteraciju kroz lemitizirani tekst. Svaka riječ se ispisuje radi boljeg praćenja rada programa, a zatim se pomoću regularnog izraza pretražuje leksikon pozi-

[1] Poveznica za Github repozitorij - https://github.com/psikac/zavrsni_rad

tivnih riječi na način da se traži isključivo trenutna riječ u leksikonu jer se, u suprotnom, može priznati dio riječi kao sama riječ koja se traži (npr. „što“ unutar „poštovanje“). Ako se riječ nalazi unutar leksikona pozitivnih riječi, brojač pozitivnih riječi se povećava za jedan, no ako se ne nalazi, tada se pretražuje leksikon negativnih riječi gdje vrijedi isto načelo: u slučaju da je riječ unutar leksikona negativnih riječi, odgovarajući brojač se povećava. Nakon prolaska kroz sve riječi unutar rečenice, evaluira se stanje brojača; u slučaju da je pozitivni brojač veći od negativnog, sentiment je pozitivan. Međutim, ako je negativni brojač veći od pozitivnog, sentiment je negativan. Treća mogućnost je da brojači imaju jednaku vrijednost, a tada se sentiment smatra neutralnim.

Isječak koda 7.2: Algoritam za analizu sentimenta pomoću pravila

```
for element in data:
    text = element['text']
    print(text)
    re_text = r'\b' + text + r'\b'
    if re.search(re_text, words_positive):
        positive = positive + 1
    elif re.search(re_text, words_negative):
        negative = negative + 1

print(f'Pozitivne rijeci: {positive} Negativne rijeci: {negative}')
if positive > negative:
    print("Recenica ima pozitivan sentiment.")
elif negative > positive:
    print("Recenica ima negativan sentiment.")
else:
    print("Recenica ima neutralan sentiment.")
```

Probne rečenice koje su korištene su sljedeće:

- "Ovaj program dobro radi."
- „Ne osjećam se dobro.“
- "Sutra je petak."
- "Jučer je bio najgori dan u mom životu."

Analiza prve rečenice daje naznaku da se radi o pozitivnom sentimentu što je točno. Međutim, u drugoj se rečenici pojavljuje problem; program tvrdi da je rečenica pozitivna, iako se iz konteksta može zaključiti da se radi o negativnom sentimentu. Odavde se može zaključiti da nije dovoljna analiza pojedinih riječi, već je bitno i cjelokupno značenje rečenice. Treća rečenica očekivano se ocjenjuje kao neutralna rečenica, dok se u četvrtoj korektno prepoznaje negativan sentiment.

Zaključak je da je ovaj pristup analizi prihvatljiv kada rečenice sadrže karakteristično pozitivne, odnosno negativne riječi, no kada se analiziraju složenije rečenice, preciznost algoritma znatno pada jer broj zadanih riječi nije dovoljan kako bi se precizno odredio sentiment. Veliki problem predstavljaju negacije pozitivnih riječi, suprotni veznici i sami leksikon u kojemu

se riječi poput jeseni smatraju negativnima. Poboljšani rad algoritma mogao bi se postići u kombinaciji sa strojnim učenjem kojim bi se neki od ovih problema mogli ispraviti.

7.2. Automatski sustav

Domena izabrana za izradu automatskog sustava za računalnu analizu sentimenta su recenzije filmova. Skup podataka preuzet je sa Kaggle-a [2] i sadrži pedeset tisuća recenzija na engleskome jeziku i na svakoj je recenziji označen pozitivan, odnosno negativan sentiment. Za proces vektorizacije odabrane su metode vreće riječi i TF-IDF jer su jednostavne, resursno nezahtjevne, ali precizne. Modeli su izrađeni pomoću multinomijalnog naivnog Bayesovog klasifikatora i metode suprotnih vektora. One su također izabrane zbog svojih performansi i jednostavnosti. Automatski sustav je, dakle, realiziran u četiri kombinacije algoritama čiji se rad evaluira i međusobno uspoređuje; time se analizom rada pojedine kombinacije dolazi do zaključka koja kombinacija dalje najbolje rezultate. Kako bi se implementirao sustav, praćene su upute jedne implementacije pronađene na Kaggle-u [3].

Za implementaciju sustava za rudarenje mišljenjima korištene su sljedeće biblioteke:

- Pandas – biblioteka za rad s dokumentima i tekstom.
- Natural Language Toolkit (nltk) – skup biblioteka koje služe za statističku i simboličnu obradu prirodnog teksta.
- skicit-learn (sklearn) - biblioteka za strojno učenje bazirana na NumPy i SciPy bibliotekama koja sadrži razne algoritme klasifikacije, regresije i klasterizacije.
- Beautiful Soup - Python paket za prolaženje kroz HTML i XML dokumente.

U isječku programskog koda 7.3. može se vidjeti podjela skupa podataka na dio za treniranje i testiranje. Uobičajeno je da se većina podataka odvaja za svrhe treniranja modela, dok se ostatak koristi za ispitivanje performansi. Od pedeset tisuća recenzija, četvrtina njih je nasumično odabrana za testiranje, dok je ostatak uporabljen za treniranje modela. Oznake recenzija su također promijenjene tako da oznaku „positive“ zamijeni broj „1“, a „negative“ sa brojem „0“. Odsječak završava pozivanjem funkcije za pretprocesiranje podataka:

Isječak koda 7.3: Podjela skupa podataka [29]

```
train_reviews, test_reviews, train_sentiments, test_sentiments = train_test_split(
    dataset['review'], dataset['sentiment'], test_size=0.25, random_state=42)
corpus_train = []
corpus_test = []

train_sentiments = (train_sentiments.replace({'positive': 1, 'negative': 0})).values
test_sentiments  = (test_sentiments.replace({'positive': 1, 'negative': 0})).values
```

[2] Poveznica na skup podataka - https://kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[3] Poveznica na uputstva za implementaciju sustava - https://kaggle.com/subhamoybhaduri/approaches-of-nlp-and-sentiment-classification

```
corpus_train = CleanUpData(train_reviews)
corpus_test = CleanUpData(test_reviews)
```

Prvi korak implementacije algoritma je izrada funkcije za pretprocesiranje teksta. Naime, kao što je ranije spomenuto, bolje se performanse postižu kada se iz teksta uklone nepotrebni simboli, normalizira tekst i uklone zaustavne riječi; svaka od ovih radnji se čini kao sitnica, no kada se iskoriste sve zajedno na skupu podataka od pedeset tisuća instanci, razlika je znatna. Nakon tokenizacije i obrade, tekst se ponovno spaja u cjelinu i nastavlja se obrada. U nastavku se nalazi isječak programskog koda 7.4. koji sadrži funkcije potrebne za pretprocesiranje teksta:

Isječak koda 7.4: Čišćenje skupa podataka [29]

```
def CleanUpData(data):
    clean_data = []
    for i in range(data.shape[0]):
        print(i,"/",data.shape[0])
        soup = BeautifulSoup(data.iloc[i], "html.parser")
        entry = soup.get_text()
        entry = re.sub('\[[^]]*\]', '␣', entry)
        entry = re.sub('[^a-zA-Z]', '␣', entry)
        entry = entry.lower()
        entry = entry.split()
        entry = [word for word in entry if not word in set (stopwords.words('english
            '))]
        lem = WordNetLemmatizer()
        entry = [lem.lemmatize(word, pos="v") for word in entry]
        entry = '␣'.join(entry)
        clean_data.append(entry)
    return clean_data
```

Posljednji korak analize je postavljanje parametara algoritama za vektorizaciju i klasifikaciju te uporaba istih i na kraju ispis rezultata. U isječku koda 7.5. je prikazan samo postupak za kombinaciju TF-IDF i linearne metode potpornih vektora jer se kod ne razlikuje dovoljno da bi postojala potreba pokazivati ostale isječke. Svi algoritmi za vektorizaciju imali su postavljen parametar da kreiraju vokabulare skup sa unigramima i bigramima kako bi točnost modela bila viša. Korištenje trigrama rezultiralo bi još većom točnošću, ali računalo na kojem je rađen završni rad nije imalo dovoljnu radnu memoriju kako bi obradilo toliku količinu podataka. Završetkom vektorizacije i izrade modela za analizu sentimenta, obavlja se funkcija predviđanja u kojoj se pomoću testnog skupa podataka evaluira rad modela. Nakon toga se ispisuju rezultati analize i usporedba nekoliko modelom određenih oznaka sa stvarnim oznakama recenzije.

Isječak koda 7.5: Izrada modela za analizu sentimenta [29]

```
tfidf_vec = TfidfVectorizer(ngram_range=(1, 2))

tfidf_vec_train = tfidf_vec.fit_transform(corpus_train)
tfidf_vec_test = tfidf_vec.transform(corpus_test)
linear_svc_tfidf = LinearSVC(C=0.5, random_state=42)
linear_svc_tfidf.fit(tfidf_vec_train, train_sentiments)
```

```
predict_svc_tfidf = linear_svc_tfidf.predict(tfidf_vec_test)

print("Classification Report: \n", classification_report(test_sentiments,
    predict_svc_tfidf,target_names=['Negative','Positive']))
print("Confusion Matrix: \n", confusion_matrix(test_sentiments, predict_svc_tfidf))
print("Accuracy: \n", accuracy_score(test_sentiments, predict_svc_tfidf))

test_result = pd.concat([dataset_predict, test_actual_label,test_predicted_label],
    axis=1)
print(test_result.head())
```

U tablici 1 prikazani su rezultati ispitivanja svih modela. Kriteriji ocjenjivanja za modele su: točnost, preciznost, odziv, F1 ocjena i broj pogođenih nasumičnih recenzija iz skupa podataka. Točnost predstavlja općenitu ocjenu izvedenu iz svih formalnih parametara ocjenjivanja. Preciznost se definira kao omjer točno pogođenih pozitivnih predviđanja [30], [31]. Zadana je formulom:

$$Preciznost = \frac{Pravi\ pozitivi}{Pravi\ pozitivi + Lazni\ pozitivi}$$

Odzivom se računa koliko je stvarno pozitivnih predviđanja točno identificirano. Izraz kojim se računa je:

$$Odziv = \frac{Pravi\ pozitivi}{Pravi\ pozitivi + Lazni\ negativi}$$

F1 ocjena služi kako bi izbalansirala preciznost i odziv, a formula joj je sljedeća:

$$F1\ ocjena = 2\frac{Preciznost * Odziv}{Preciznost + Odziv}$$

Tablica 1: Prikaz rezultata testiranja modela za analizu sentimenta

Kombinacija	Točnost	Preciznost	Odziv	F1 ocjena	Pogođeno testova (od 5)
TF-IDF i LSCV	0,90792	0,91	0,91	0,91	4
CV i LSCV	0,89848	0,90	0,90	0,90	4
TF-IDF i MNB	0,88688	0,89	0,89	0,89	5
CV i MNB	0,88344	0,89	0,89	0,89	5

Maksimalna točnosto modela je oko 90%, što je prilično dobar rezultat. Međutim, vjerojatno bi preciznost bila još veća da se za analizu koristio leksikon načinjen od trigrama, ali kako je već spomenuto ranije, previše resursa se troši na taj način. To je razlog zašto se većina takvih aktivnosti odrađuje na serverima, a ne na osobnim računalima. Iz rezultata vidi se da se najveća preciznost modela postiže kombinacijom TF-IDF metodom vektorizacije i linearne metode potpornih vektora, a zatim kombinacijom vreća riječi i linearna metoda potpornih vektora te se na začelju nalaze modeli koji koriste naivni Bayesov klasifikator. Iz toga se može

zaključiti da točnost modela dosta ovisi o metodi vektorizacije koja se koriti na tekstu. Drugi zaključak iz ove demonstracije je da se bolje performanse postižu sa linearnom metodom potpornih vektora. Međutim, zanimljivo je da se pri ispitivanju modela sa pet nasumičnih recenzija bolji rezultati postižu kod modela koji koriste multinomijalni naivni Bayesov model.

8. Zaključak

Za realizaciju ovog rada korištene su razne metode od kojih su neke metode dedukcije, indukcije i modeliranja. Tehnologije korištene za izradu rada su PyCharm razvojno okruženje u kojemu su napisani praktični dijelovi završnog rada, dok je draw.io korišten za izradu grafikona.

Ovim radom zaključeno je da je računalna analiza sentimenta, još poznata kao rudarenje mišljenja, znanstveno područje koje spaja umjetnu inteligenciju, strojno učenje i rudarenje podataka, a njime je cilj prepoznati sentiment koji se krije iza nekog teksta. Definirane su tri razine analize sentimenta: analiza na razini dokumenta, rečenice i aspektna analiza sentimenta. Područje primjene rudarenja mišljenja jako je široko; varira od one u poslovnom svijetu (npr. praćenje trendova, proučavanje tržišta, evaluacija povratnih informacija korisnika) pa sve do primjene u politici (predviđanje predsjedničkih izbora, praćenje sentimenta stanovništva, itd.). Iako je potencijal ove tehnologije iznimno velik, postoje razne prepreke koje joj narušavaju performanse, a neki od njih su prepoznavanje negacija i stilskih figura, loši rezultati u drugim domenama i loša jezična podrška alata.

Definirane su dvije vrste sustava za analizu sentimenta, sustavi temeljeni na pravilima i automatski sustavi. Nakon toga opisan je način realiziranja tih sustava te metode vektorizacije i klasifikacije koje se koriste pri analizi. Na kraju su obje vrsta sustava implementirane i njihov rad ispitan. Zaključeno je da sustavi temeljeni na pravilima imaju manju preciznost od automatskih sustava jer dosta ovise o leksikonima koje koriste. Automatski sustavi su imali najbolje performanse kada su radili u kombinaciji sa TF-IDF metodom vektorizacije i linearnom metodom potpornih vektora koja se koristila kao klasifikator. Zamijećeno je da veliku ulogu u točnosti algoritma ima odabrani klasifikacijski algoritam, a zatim algoritam koji služi za vektorizaciju.

Popis literature

[1] S. J. Russell i P. Norvig, *Artificial Intelligence: A Modern Approach*, 3. izdanje, S. Russell i P. Norvig, ur., serija Prentice Hall Series in Artificial Intelligence. New Jersey, USA: Prentice Hall, 2010, 1132 **pagetotals**, ISBN: 978-0-13-604259-4.

[2] Wikipedia suradnici, *Artificial intelligence — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/w/index.php?title=Artificial_intelligence&oldid=977168879 (pogledano 8. 2. 2020).

[3] ——, *Machine learning — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=976690424 (pogledano 8. 6. 2020).

[4] ——, *Supervised learning — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/w/index.php?title=Supervised_learning&oldid=973316580 (pogledano 8. 6. 2020).

[5] A. I. SAS: Analytics i D. Managment, *Machine Learning, What it is and why it matters*, SAS: Analytics, Artifical Intelligence i Data Managment. adresa: https://www.sas.com/en_us/insights/analytics/machine-learning.html (pogledano 8. 6. 2020).

[6] Wikipedia suradnici, *Unupervised learning — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/wiki/Unsupervised_learning (pogledano 8. 6. 2020).

[7] ——, *Semi-supervised learning — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/wiki/Semi-supervised_learning (pogledano 8. 6. 2020).

[8] ——, *Reinforcement learning — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/wiki/Reinforcement_learning (pogledano 8. 6. 2020).

[9] ——, *Data mining — Wikipedia, The Free Encyclopedia*, 2020. adresa: https://en.wikipedia.org/wiki/Data_mining (pogledano 8. 6. 2020).

[10] X. Zhu i X. Wu, „Class Noise vs. Attribute Noise: A Quantitative Study", en, *Artificial Intelligence Review*, sv. 22, br. 3, str. 177–210, studeni 2004, ISSN: 0269-2821, 1573-7462. DOI: 10.1007/s10462-004-0751-8. (pogledano 7. 9. 2020).

[11] B. Liu, „Sentiment Analysis and Opinion Mining”, en, *Synthesis Lectures on Human Language Technologies*, sv. 5, br. 1, str. 1–167, svibanj 2012, ISSN: 1947-4040, 1947-4059. DOI: 10.2200/S00416ED1V01Y201204HLT016. (pogledano 23. 8. 2020).

[12] MonkeyLearn, *Sentiment analysis*, MonkeyLearn. adresa: https://monkeylearn.com/sentiment-analysis/ (pogledano 8. 10. 2020).

[13] A. Tumasjan, T. O. Sprenger, P. G. Sandner i I. M. Welpe, „Election Forecasts With Twitter: How 140 Characters Reflect the Political Landscape”, en, *Social Science Computer Review*, sv. 29, br. 4, str. 402–418, studeni 2011, ISSN: 0894-4393, 1552-8286. DOI: 10.1177/0894439310386557. (pogledano 23. 8. 2020).

[14] S. M. Mohammad, „Challenges in Sentiment Analysis”, *A Practical Guide to Sentiment Analysis*, E. Cambria, D. Das, S. Bandyopadhyay i A. Feraco, ur., sv. 5, Cham: Springer International Publishing, 2017, str. 61–83, ISBN: 9783319553924 9783319553948. DOI: 10.1007/978-3-319-55394-8_4. (pogledano 25. 8. 2020).

[15] T. Al-Moslmi, N. Omar, S. Abdullah i M. Albared, „Approaches to Cross-Domain Sentiment Analysis: A Systematic Literature Review”, *IEEE Access*, sv. 5, str. 16 173–16 192, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2690342. adresa: http://ieeexplore.ieee.org/document/7891035/ (pogledano 13. 8. 2020).

[16] M. Schatten, J. Ševa i B. Okreša Đurić, „Synesketch: An Introduction to Social Semantic Web Mining & Big Data Analytics for Political Attitudes and Mentalities Research”, sv. 4, str. 40–62, siječanj 2015, ISSN: 2285 – 4916, 2285 – 4916.

[17] J. Brownlee, *A Gentle Introduction to the Bag-of-Words Model*, en-US, listopad 2017. adresa: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (pogledano 25. 8. 2020).

[18] U. Krcadinac, P. Pasquier, J. Jovanovic i V. Devedzic, „Synesketch: An Open Source Library for Sentence-Based Emotion Recognition”, *IEEE Transactions on Affective Computing*, sv. 4, br. 3, str. 312–325, srpanj 2013, ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2013.18. (pogledano 25. 8. 2020).

[19] R. Stecanella, *The Beginner's Guide to Text Vectorization*, en-US, MonkeyLearn, rujan 2017. adresa: https://monkeylearn.com/blog/beginners-guide-text-vectorization/ (pogledano 25. 8. 2020).

[20] P. Pantola, *Natural Language Processing: Text Data Vectorization*, en, lipanj 2018. adresa: https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7 (pogledano 25. 8. 2020).

[21] P. Huilgol, *BoW Model and TF-IDF For Creating Feature From Text*, veljača 2020. adresa: https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/ (pogledano 25. 8. 2020).

[22] Wikipedia suradnici, *tf–idf*, en, Page Version ID: 962443067, lipanj 2020. adresa: https://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=962443067 (pogledano 25. 8. 2020).

[23] ——, *Statistical classification*, en, Page Version ID: 963809210, lipanj 2020. adresa: https://en.wikipedia.org/w/index.php?title=Statistical_classification&oldid=963809210 (pogledano 25. 8. 2020).

[24] ——, *Naive Bayes classifier — Wikipedia, The Free Encyclopedia*, en, Page Version ID: 974500501, kolovoz 2020. adresa: https://en.wikipedia.org/w/index.php?title=Naive_Bayes_classifier&oldid=974500501 (pogledano 25. 8. 2020).

[25] B. Stecanella, *A practical explanation of a Naive Bayes classifier*, en-US, svibanj 2017. adresa: https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/ (pogledano 25. 8. 2020).

[26] Wikipedia suradnici, *Bayes' theorem — Wikipedia, The Free Encyclopedia*, en, Page Version ID: 972729352, kolovoz 2020. adresa: https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=972729352 (pogledano 25. 8. 2020).

[27] ——, *Support vector machine — Wikipedia, The Free Encyclopedia*, en, Page Version ID: 973189035, kolovoz 2020. adresa: https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=973189035 (pogledano 25. 8. 2020).

[28] B. Stecanella, *An Introduction to Support Vector Machines (SVM)*, en-US, lipanj 2017. adresa: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/ (pogledano 25. 8. 2020).

[29] *Approaches of NLP and Sentiment Classification*, en, 2020. adresa: https://kaggle.com/subhamoybhaduri/approaches-of-nlp-and-sentiment-classification (pogledano 11. 9. 2020).

[30] Google Developers, *Classification: Precision and Recall | Machine Learning Crash Course*, en. adresa: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall (pogledano 25. 8. 2020).

[31] K. P. Shung, *Accuracy, Precision, Recall or F1?*, en, travanj 2020. adresa: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 (pogledano 25. 8. 2020).

Popis slika

Popis tablica

Popis isječaka kodova