SOFTVERSKI ALAT ZA ANALIZU SENTIMENTA NAD RECENZIJAMA

SOFTWARE TOOL FOR SENTIMENT ANALYSIS OF REVIEWS

Dražen Drašković, *IEEE member*, Milica Mitrić, Jelica Cincović, Boško Nikolić
Univerzitet u Beogradu - Elektrotehnički fakultet

Sadržaj *– Ovaj istraživački rad bavi se analizom sentimenta recenzija na engleskom jeziku sa platforme Amazon. Skup podataka koji je prikupljen ima više od 560 hiljada recenzija, koje uključuje informacije o proizvodima i korisnicima, ocenama i tekstualni pregled kritike. U istraživanju je korišćeno 12 tradicionalnih modela (Naivni Bajesov model, logistička regresija i metod potpornih vektora, uz vektorizaciju tipa Bag of words, Bag of N-grams, i Term Frequency-Inverse Document Frequency), kao i 6 modela dubokog učenja koji koriste konvolucione i rekurentne neuralne mreže.*

Abstract *– This research paper deals with the sentiment analysis of reviews in English from the Amazon platform. The collected data set has more than 560 thousand reviews, which includes product and user information, ratings and a text overview of the review. The research used 12 traditional models (Naive Bayesian model, logistic regression and support vector machine, with Bag of words, Bag of N-grams, and Term Frequency-Inverse Document Frequency vectorization), as well as 6 deep learning models using convolutional and recurrent neural networks.*

Ključne reči*- analiza sentimenta, Naivni Bajes, logistička regresija, metod potpornih vektora, Bag of words, Bag of N-grams, Term Frequency-Inverse Document Frequency.*

1. UVOD

Razvojem interneta menja se način na koji ljudi izražavaju i traže mišljenja. U današnje vreme postoji veliki broj veb strana na kojima korisnici mogu izraziti svoje stavove, kao što su društvene mreže, portali, forumi, blogovi i slično. Broj korisnika koji ovakve podatke kreiraju i koriste je u stalnom porastu, čime se povećava i količina dostupnih podataka za analizu. Ovi korisnički podaci mogu se iskoristiti u različite svrhe, od analize sentimenta rečenice, preko filtera neželjene elektronske pošte, do prepoznavanja autora teksta, njegovog pola, godišta, itd. [1]

U ovom radu, autori se bave analizom sentimenta, odnosno klasifikacijom teksta. Ova vrsta analize postaje sve popularnija u mnogim domenima: politici, zdravstvu, proizvodnji potrošačkih proizvoda, i drugim. Na ovaj način su se predviđali rezultati izbora ili položaj proizvoda na tržištu. Ovo istraživanje bavi se analizom sentimenta u internet trgovini. Amazon je jedan od giganata internet trgovine koji ljudi svakodnevno koriste i gde mogu da pronađu recenzije o proizvodu koji žele da kupe. Recenzije su korisne i za prodavce, jer im pomažu da bolje razumeju potrošače i njihove potrebe. Analiza sentimenta je odigrala ključnu ulogu u mašinskom ocenjivanju proizvoda na osnovu komentara korisnika, a zbog toga što daje uvid u stavove ljudi, postala je i jedan od najmoćnijih alata marketinga. Obrada sentimenta, ne samo medijskih objava, već svih oblika povratnih reakcija (komentari, forumi) prevazilazi nivo osnovnog. Koristeći analizu sentimenta, kompanije mogu da prate, ne samo šta ljudi kažu o brendu, već i da otkriju ono što potrošači žele i očekuju.

Zadatak analize sentimenta može da se modelira kao problem klasifikacije, pri čemu se na ulaz klasifikatora dovodi tekst, a izlaz je kategorija kojoj tekst pripada, npr. pozitivan ili negativan. Cilj ovog istraživačkog rada je razvoj modela mašinskog učenja koji može što preciznije da predvidi da li recenzija nosi pozitivnu ili negativnu konotaciju.

2. OPIS ISTRAŽIVANJA

U ovom istraživanju se upoređuju tradicionalne metode mašinskog učenja - Naivni Bajesov model, logistička regresija i metoda potpornih vektora, kao i modeli dubokog učenja. Kreirani modeli se porede u pogledu efikasnosti (vreme izvršavanja) i tačnosti.

A. KORIŠĆENI ALGORITMI I TEHNIKE

Logistička regresija je algoritam mašinskog učenja koji se koristi u problemima klasifikacije, npr. da li je poruka e-pošte neželjena (*spam*) ili nije. Osnovna verzija ovog algoritma služi za binarnu klasifikaciju. Najčešće se jedna klasa označava sa y=0, a druga sa y=1, ali je tu problem ukoliko funkcija na izlazu da vrednost blisku 0.5, pošto klasifikator nije tada siguran u svoju odluku. Zato se za hipotezu usvaja logistička (sigmoid) funkcija:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Naivni Bajesov algoritam ima dve faze. Prva faza je faza učenja, u kojoj se algoritmu daju podaci sa oznakom klase kojoj pripadaju. U ovom delu algoritam računa broj pojavljivanja elemenata svake klase, pa se na osnovu toga izračunaju verovatnoće svake klase. Druga faza je faza predviđanja klase. Podatak treba svrstati u klasu koja je za njega najverovatnija, tj. ima najveću aposteriornu verovatnoću. Ovako odlučivanje se naziva Maximum a posteriori (MAP). Naivni Bajesov model polazi od dve „naivne“ pretpostavke nad atributima. Pretpostalja da su sva obeležja podjednako važna. To znači da poznavanje vrednosti jedne odlike ništa ne govori o vrednosti druge odlike. U praksi Naivni Bajesov klasifikator često ostvaruje dobre rezultate, jer za pravilnu klasifikaciju nije neophodno poznavanje tačnih vrednosti p(y|x).

Metod potpornih vektora spada u diksriminativne modele. Ovaj model je posebno pogodan za skupove podataka sa velikim brojem atributa. To je linearni

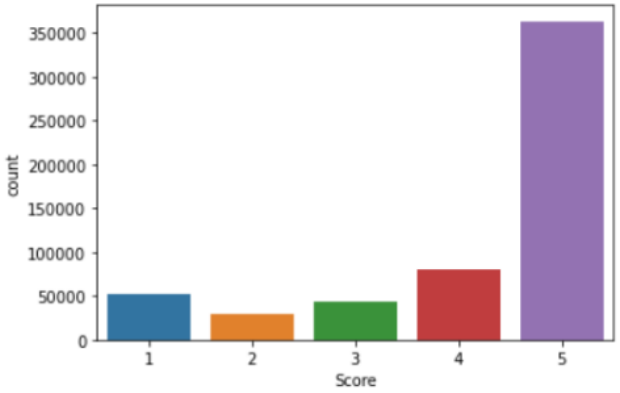
klasifikator koji pronalazi hiperravan koja razdvaja dve klase. Polazi od pretpostavke da su podaci linearno separabilni i tada je moguće pronaći beskonačno mnogo hiperravni koje su u stanju da izvrše razdvajanje podataka iz skupa za obučavanje. Cilj metode potpornih vektora je dobijanje što veće geometrijske margine na obučavajućem skupu. Na ovaj način se maksimizira rastojanje primera iz obučavajućeg skupa od separacione prave, odnosno minimizira se verovatnoća greške na obučavajućem skupu.

Konvoluciona neuralna mreža (CNN) se sastoji od jednog ulaznog, jednog izlaznog i jednog ili više skrivenih slojeva. Kod CNN specifični su konvolucioni slojevi i slojevi sažimanja. Ove mreže najčešće kreću sa jednim ili više konvolucionih slojeva, sledi sloj sažimanja, pa ponovo konvolucioni sloj, i tako nekoliko puta. Arhitektura konvolucionih neuralnih mreža se pokazala veoma dobro u radu sa slikama i prepoznavanju karakteristika sa istih. Svaki konvolucioni sloj se sastoji od filtera koji sadrži težine. Filteri su najčešće manjih prostornih dimenzija od ulaza, ali su jednake dubine kao i ulaz. Na primer, ako se na ulazu konvolucionog sloja nađe slika sa tri komponente (RGB) i veličine 25x25px, filter može biti matrica težina veličine 5x5. Potrebno je imati tri takve matrice, po jednu za svaku komponentu, tj. boju. Filter se onda konvoluira sa matricom i kao rezultat se dobija dvodimenzionalna aktivaciona mapa koja predstavlja odziv filtera na svakoj prostornoj poziciji. Mreža će naučiti težine unutar filtera kako bi se „aktivirala“ na mestima gde prepoznaje određena slikovna svojstva, npr. ivice. Izlaz sloja će biti sve aktivacione mape, odnosno više dvodimenzionalnih matrica, gde svaka predstavlja jednu dubinu izlaza. Sloj sažimanja se najčešće koristi nakon nekoliko konvolucionih slojeva sa ciljem smanjivanja rezolucije mapi. Osim smanjivanja rezolucije, slojevi sažimanja povećavaju prostornu invarijantnost neuralne mreže.

Rekurentne neuralne mreže su posebna vrsta neuralnih mreža koje ne prosleđuju informacije iz jednog sloja isključivo u naredni, već u sebi mogu da sadrže cikluse. U rekurentnim neuralnim mrežama izlaz iz nekog neurona može ući u neki ciklus i kasnije opet postati ulaz istog tog neurona i tako se mešati sa nekim kasnijim ulazima. Jedan od primera je razvoj mreže za prevod rečenica. Potrebno je da za svaku reč odrediti moguća značenja i onda u kombinaciji sa ostatkom rečenice odrediti kontekst posmatrane reči. Ukoliko bi pokušalo da se koristi *feedforward* mreža, ona bi kao ulaz u jednom trenutku uzimala samo jednu reč. Ne bi mogla da pamti i kombinuje sa ostatkom rečenice, jer ona nema mogućnost pamćenja. Za ovakve probleme koriste se rekurentne mreže, jer one uz pomoć petlji uvek zadržavaju deo informacije u sebi. Često se radi lakšeg praćenja događaja u rekurentnim mrežama, ona „razvija kroz vreme“ i tako se dobija nešto slično poznatim mrežama bez povratnog prenosa samo što se svaki blok odvija u jednoj jedinici vremena.

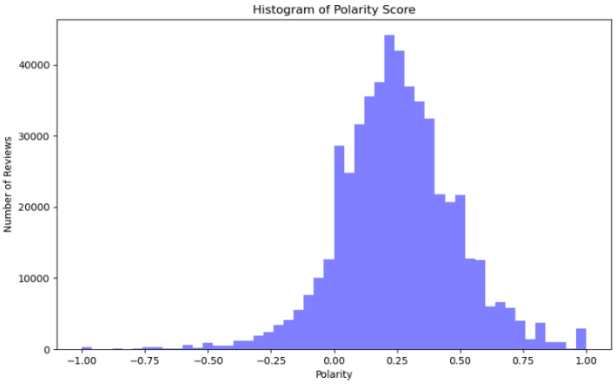
B. SKUP PODATAKA

Baza podataka uključuje 568 454 recenzije prikupljene u rasponu od 1999. godine do 2012. godine [2]. Recenzije uključuju informacije o proizvodima i korisnicima, ocene i tekstualni pregled kritike. Na slici 1, prikazana je raspodela recenzija u zavisnosti od ocene. Može se zaključiti da je reč o nebalansiranom skupu podataka, jer sadrži mnogo više recenzija sa višim ocenama.



Slika 1. Raspodela broja recenzija po ocenama

Za analizu baze podataka potrebno je takođe obratiti pažnju na polaritet. Polaritet je ceo broj u opsegu [-1,1], gde 1 označava pozitivnu recenziju, a -1 negativnu. Pozitivne recenzije imaju veći polaritet. Slika 2 potvrđuje da baza sadrži nebalansirani skup podataka, jer je veći broj kritika sa desne strane grafikona.



Slika 2. Raspodela polariteta

C. OBRADA PODATAKA

Preprocesiranje podataka obuhvata čišćenje podataka i izdavajanje karakteristika. Postavlja se pitanje da li se izdvajanje karakteristika vrši pre ili posle čišćenja podataka. Oba pristupa dobro funkcionišu. Međutim, u procesu čišćenja podataka gube se neke karakteristike, pa ih je potrebno izvući pre samog čišćenja. S druge strane, neke funkcije imaju smisla samo kada se izdvoje nakon čišćenja podataka. Stoga je potrebno i izdvajanje karakteristika nakon čišćenja podataka.

U prvoj fazi izdvojene su karakteristike koje je nemoguće dobiti nakon čišćenja podataka: broj zaustavnih reči, broj znakova interpunkcije, broj hashtag znakova, broj numeričkih karaktera i broj reči napisanih velikim slovom.

Druga faza obrade, odnosno čišćenje podataka, podrazumevala je tekst napisan malim slovom, uklanjanje znakova interpunkcije, uklanjanje zaustavnih reči, definisanje novih zaustavnih reči na osnovu 20 reči koje se najčešće pojavljuju u recenzijama i njihovo uklanjanje, i proveru pravopisa. Na slici 3 prikazan je spisak od 20

reči koje se najčešće pojavljuju, odakle su izdvojene i uklonjene reči „br“, „get“ i „also“.

```
br          264689
like        251864
good        195348
one         172306
taste       166572
great       163560
coffee      160180
product     146439
flavor      142442
tea         133094
love        126635
food        123770
would       123356
get         108169
really      100414
dont         95555
much         91906
also         86099
little       83197
use          82743
dtype: int64
```

Slika 3. Spisak reči koje se najčešće pojavljuju

Neke karakteristike, kao što su broj reči, broj karaktera i prosečna dužina reči, izvučeni su tek u trećoj fazi, nakon čišćenja teksta. Podaci izvučeni predobradom prikazani su na slici 4.

Text	stopwords	punctuation	hastags	numerics	upper	word_count	char_count	avg_word
november downloaded livestrong app began maki...	90	25	0	0	19	135	900	6.0
im normally fan bigelow green teas blueberry f...	11	13	0	0	1	19	144	6.6
popin cooking fun enjoyed arrived days early ...	17	16	0	1	1	26	155	5.2
beware sent one bottle contacted amazon said ...	0	12	0	0	37	20	126	5.6
sorry say coffee ok didnt buy emerils kcups bi...	8	10	0	1	3	16	76	4.4

Slika 4. Izdvojene karakteristike

3. RAZVOJ MODELA

Pre konstruisanja bilo kakvog klasifikatora, podatke je potrebno podeliti na skup za obučavanje i skup za testiranje. Za obučavanje je korišćeno 75% podataka, a 25% za testiranje performansi klasifikatora. Uvedena je pretpostavka da recenzija ima pozitivan sentiment ukoliko je proizvod dobio ocenu 4 ili 5, a ako ima ocenu 1 ili 2, sentiment je negativan. Ocena 3 kod proizvoda je zanemarena u ovom istraživanju, jer se smatralo da tada recenzija ne nosi ni pozitivan ni negativan sentiment, već je neutralna.

Kako na ulaze klasifikatora nije moguće dovoditi reči, potrebno je bilo uvesti numeričku reprezentaciju reči pre procesa obuvačanja. Prvi korak ka tome bio je proces tokenizacije, odnosno segmentacije. Tokenizacijom se grupa rečenica pretvara u token, odnosno razdvajaju se podaci u manje grupe [3]. Tokenizacija je urađena korišćenjem funkcije *word_tokenize( )* biblioteke *nltk*, u programskom jeziku *Python*.

Sledeći korak je vektorizacija. Najjednostavniji oblik vektorizacije je *one hot encoding* [4]. Podrazumeva stvaranje jako velike matrice sa kolonom za svaku reč u korpusu, gde je korpus skup svih recenzija. Zatim se svaka recenzija transformiše u jedan red koji sadrži elemente 0 i 1, pri čemu 1 znači da se reč u korpusu koja odgovara toj koloni pojavljuje u recenziji koja se posmatra. Elementi ove matrice će biti uglavnom nule.

Bag of words je osnovni model koji se koristi u obradi prirodnih jezika. Ovaj naziv je dobio jer ne vodi računa o redosledu reči u rečenici, već govori samo da li je reč prisutna ili nije. Neka je dat primer sa nekoliko rečenica:

„There used to be Stone Age”
„There used to be Bronze Age”
„There used to be Iron Age”
„There was Age of Revolution”
„Now it is Digital Age”

Prvo se formira rečnik svih reči u rečenicima. On se sastoji od reči: [There, was, to, be, used, Stone, Bronze, Iron, Revolution, Digital, Age, of, Now, it, is]. Vektorska konverzija poslednje rečenice „Now it is Digital Age” bila bi: [0,0,0,0,0,0,0,0,0,1,1,0,1,1,1], gde je 1 oznaka da reč iz vektora postoji u rečenici, a 0 oznaka da reč iz vektora ne postoji u rečenici. Ovaj pristup je unigram, jer uzima u obzir samo jednu reč odjednom. Slično tome postoje bigram koji uzima u obzir dve reči odjednom (There used, Used to, to be, be Stone, Stone age), i n-gram koji uzima u obzir n reči odjednom.

Term Frequency-Inverse Document Frequency (skr. TF-IDF) govori o važnosti reči u korpusu ili skupu podataka. TF-IDF sadrži dva koncepta *Term Frequency* (skr. TF) i *Inverse Document Frequency* (skr. IDF). *Term Frequency* ili učestalost pojma definiše koliko se često neka reč pojavljuje u dokumentu ili korpusu. Kako svaka rečenica nije iste dužine, moguće je da se reč pojavljuje više puta u dugoj rečenici. Uzimajući to u obzir TF bi moglo da se računa po formuli:

$$TF = \frac{Number\ of\ times\ a\ word\ appears\ in\ the\ document}{Total\ number\ of\ words\ in\ the\ document}$$

Ukoliko u određenom dokumentu postoji učestalost neke reči od nekoliko miliona, da bi se izbegli veliki brojevi, dodaje se logaritam, pa formula za učestalost pojavljivanja se može zapisati ovako:

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & tf_{t,d} > 0 \\ 0 \end{cases}$$

Inverse Document frequency (skr. IDF) je još jedan koncept koji se koristi za utvrđivanje važnosti reči. Zasnovan je na činjenici da su ređe reči informativnije i važnije. IDF je predstavljen formulom:

$$IDF = \log_{10} \frac{Number\ of\ documents}{Number\ of\ documents\ in\ which\ the\ word\ appear}$$

Nakon transformacije podataka u oblik pogodan za modeliranje, može se započeti sa izgradnjom klasifikatora. Korišćena je biblioteka za mašinsko učenje

sklearn. Naučene modele je zatim bilo neophodno serijalizovati i sačuvati serijski format u datoteku. *Pickle* je standardni način serijalizacije objekata u programskom jeziku *Python*. Modeli su čuvani u fajlu naredbom pickle.dump(). Kasnije je ovaj fajl mogao da se učita, da bi se model deserijalizovao i koristio za pravljenje novih predikcija.

Za kreiranje neuralnih mreža koje se koriste za analizu sentimenta recenzije koristila se biblioteka *Keras*. *Keras* nudi *Embedding* sloj koji se može iskoristiti za neuralne mreže koje rade nad tekstualnim podacima. On zahteva da svi ulazni podaci budu celobrojni, tako da je svaka reč predstavljena jedinstvenim celim brojem. Ovaj korak pripreme se takođe mogao uraditi koristeći biblioteku Keras, tako da se odmah nakon učitavanja podataka izvršio proces tokenizacije. Svakoj reči se dodeljuje ceo broj i taj broj se stavlja u listu.

Pre treniranja neuralne mreže potrebno je pretvoriti svaku reč u *word embedding*. To su vrste numeričke reprezentacije reči koje omogućavaju algoritmima da uče. One vode računa o kontekstu reči u odnosu na druge reči u rečenici. Ova transformacija dovodi do toga da se reči sa sličnim značenjem grupišu bliže jedna drugoj, a različite reči se nalaze dalje u hiperravni. Za potrebe ovog rada korišćene su već postojeće reprezentacije reči *Word2vec* i *FastText*. Dakle, svaka reč se pretvara u odgovarajuću numeričku reprezentaciju i dovodi na ulaz neuralne mreže.

Word2vec je tehnika dubokog učenja sa dvoslojnom neuralnom mrežom, koja na ulazu ima ogroman skup podataka (Google podatke) i pretvara ga u vektorski prostor. *Word2vec* postavlja reč u prostor obeležja, tako da je lokacija svake reči određena njenim značenjem. Reči sličnog značenja su zajedno grupisane.

FastText je visoko efikasna biblioteka za predstavljanje i klasifikaciju teksta koju je objavio Facebook AI Research (FAIR) tim 2016. godine. Slično kao *Word2Vec*, i *FastText* razmatra sličnost reči, ali razmatra podreči koristeći Bag of n-grams. Na primer, reč train se deli na „tra“, „rai“ i „ain“. Na ovaj način je predstavljanje reči otpornije na pravopisne greške. Takođe, *FastText* mnogo bolje obrađuje nepoznate reči, jer duge reči raščlani u podreči koje se mogu pojaviti u još nekim rečima, što mu daje bolji kontekst.

Embedding sloj je inicijalizovan slučajnim težinama i naučiće težine za sve reči u skupu za treniranje. Nabrajaju se sve jedinstvene reči i locira se odgovarajući vektor u *Word2Vec* ili *FastText* embedding-u. Svaki embedding sloj mora imati tri argumenta. Prvi argument je veličina rečnika u tekstualnim podacima. Na primer, ako su podaci kodirani na vrednosti između 0 i 10, tada bi veličina rečnika bila 11 reči. Drugi argument je veličina vektorskog prostora u koji će reči biti ugrađene. Definiše veličinu izlaznih vektora iz ovog sloja za svaku reč. Treći argument je dužina ulaza.

Za problem analize sentimenta trenirane su konvoluciona, rekurentna i konvoluciona rekurentna neuralna mreža. Cela arhitektura modela konvolucione neuralne mreže može se podeliti na dva dela, blokove CNN slojeva (Convolution1D, MaxPool1D) i potpuno povezane slojeve. Analiza sentimenta je problem binarne klasifikacije, tj. problem može da se formuliše kao predviđanje verovatnoće da primer pripada jednoj klasi. Klasama mogu da se dodele vrednosti 0 i 1. Ovo određuje strukturu izlaznog sloja koji će se sastojati od jednog čvora sa sigmoidnom funkcijom aktivacije. Treniranje neuralne mreže je problem optimizacije, jer se težine ne mogu tačno analitički odrediti. Zbog toga, mora da se specificira funkcija koja će se optimizovati. U ovom slučaju izabrana je binarna kros-entropija, odnosno logaritamski gubitak. Za njenu minimizaciju, umesto stohastičkog gradijentnog spusta koristiće se ugrađeni metod optimizers.Adam(). Dropout se dodaje kako bi se sprečilo preobučavanje neuralne mreže. Dropout se odnosi na ignorisanje jedinica (tj. neurona) tokom faze obučavanja i ima smisla koristiti ga jedino u potpuno povezanim slojevima. Ova operacija podrazumeva nuliranje kolona u matrici težina i ekvivalentna je netreniranju ili „ispadanju“ neurona. Na ovaj način se povećava broj iteracija potrebnih za konvergenciju, ali se smanjuje vreme treniranja u svakoj epohi. Na sličan način se može definisati rekurentna neuralna mreža [5]. Razlika je ta što umesto konvolucionih slojeva dodajemo LSTM sloj.

4. REZULTATI ISTRAŽIVANJA

U ovom radu upoređene su tradicionalne i metode dubokog učenja za analizu sentimenta. Kreirano je 12 tradicionalnih modela: Naivni Bajesov model, model sa logističkom regresijom, i model koji primenjuje metodu potpornih vektora, uz vektorizaciju tipa *Bag of words*, *Bag of N-grams* i *TFIDF*. Rezultati ovih modela, koji obuhvataju tačnost modela i vreme treniranja, prikazani su u tabelama 1, 2 i 3. Takođe, kreirane su konvoluciona i rekurentna neuralna mreža uz dve vrste ugrađivanja reči – *Word2Vec* i *FastText*. Rezultati tih modela prikazani su u tabelama 4 i 5, a u tabeli 6 prikazan je rezultat hibridnog modela.

Na osnovu dobijenih rezultata, najbolje performanse ostvario je klasifikator zasnovan na metodi potpornih vektora. Takođe, vidi se da visoku tačnost imaju i konvolucione neuralne mreže, iako se najčešće koriste kada su elementi skupa za obučavanje slike. Rekurentne neuralne mreže prepoznaju obrasce kroz vreme, dok konvolucione neuralne mreže uče da prepoznaju obrasce u prostoru. Za zadatke u kojima je dužina teksta važna, npr. odgovaranje na pitanja ili prevod, ima više smisla koristiti rekurentne neuralne mreže. Za konvolucione neuralne mreže, najprirodniji su zadaci klasifikacije, u koje spada i analiza sentimenta [6]. Operacija konvolucije gubi informacije o lokalnom redosledu reči, ali svaki filter detektuje posebne obrasce, npr. „I hate“, „very good“, itd. i konvolucione neuralne mreže ih prepoznaju u rečenici bez obzira na njihov položaj. Konvolucione neuralne mreže su mnogo brže od rekurentnih. Takođe su i robusnije, jer se često dešava da rekurentna neuralna mreža divergira.

Tabela 1. Rezultati za Naivni Bajesov model

Tip vektorizacije	Tačnost	Vreme treniranja
Bag of words	0.8699227852149678	2.981224298477173
Bag of n-grams	0.8877339668905438	5.232130527496338
Bag of words TFIDF	0.9051187158282694	3.208014726638794
Bag of n-grams TFIDF	0.9135255326601788	7.957592248916626

Tabela 2. Rezultati za model sa logističkom regresijom

Tip vektorizacije	Tačnost	Vreme treniranja
Bag of words	0.9124137615936401	7.925859212875366
Bag of n-grams	0.9210870988867059	11.669291973114014
Bag of words TFIDF	0.9131676337551972	7.397418737411499
Bag of n-grams TFIDF	0.925343811394892	13.535844087600708

Tabela 3. Rezultati modela sa metodom potpornih vektora

Tip vektorizacije	Tačnost	Vreme treniranja
Bag of words	0.9161907372717443	56.23175835609436
Bag of n-grams	0.9210870988867059	304.0738010406494
Bag of words TFIDF	0.9150789662052056	31.03028416633606
Bag of n-grams TFIDF	0.9353725956046969	288.0533971786499

Tabela 4. Rezultati za model zasnovan na konvolucionoj neuralnoj mreži

Tip ugrađivanja reči	*Tačnost*
Word2vec	*0.8995770061775357*
Fast Text	*0.9007562155746934*

Tabela 5. Rezultati za model zasnovan na rekurentnoj neuralnoj mreži

Tip ugrađivanja reči	*Tačnost*
Word2vec	*0.9008880740087033*
Fast Text	*0.916921883083902*

Tabela 6. Rezultati za model zasnovan na konvolucionoj rekurentnoj neuralnoj mreži

Tip ugrađivanja reči	*Tačnost*
Word2vec	*0.8996226529929896*
Fast Text	*0.9007486077721311*

Problem nebalansiranog skupa podataka može prouzrokovati neželjeno ponašanje klasifikatora kojim se favorizuje klasa sa više podataka na trening skupu. U sledećem primeru, iako je ukupan sentiment recenzije negativan, mašina bi klasifikovala recenziju kao pozitivnu zbog broja pozitivnih reči koje recenzija sadrži:

„This film sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."

Ovo se naziva koncept suprotstavljenih očekivanja. Čest je u analizi recenzija i prepoznali su ga Pang(2002) i Turney(2002) koji su uočili da „celina nije nužno suma delova". Jedan od problema je i implicitni sentiment, koji se obično prenosi kroz neke neutralne reči, što otežava procenu njegovog polariteta. Na primer, rečenica poput „Item as described", koja se često pojavljuje u pozitivnim recenzijama, se sastoji samo od neutralnih reči.

5. ZAKLJUČAK

Obrada prirodnog jezika postaje sve važnija grana veštačke inteligencije. Kroz mnogobrojne primene kao što su mašinsko prevođenje, borba protiv neželjene elektronske pošte i analiza sentimenta spaja lingvistiku i veštačku inteligenciju. U ovom istraživanju, korišćenjem programskog jezika Python i biblioteka za mašinsko učenje, prikazana je jedna implementacija alata za analizu sentimenta zasnovana na različitim vrstama algoritama. Korišćeni su Naivni Bajesov model, logistička regresija, metod potpornih vektora, konvolucione i rekurentne neuralne mreže. Zaključeno je da određene tehnike poput Bag of n-grams i TF-IDF poboljšavaju performanse klasifikatora. Na bazi podataka recenzija, najveću tačnost postigao je klasifikator zasnovan na metodi potpornih vektora (sa TF-IDF) od 93.53%, a svi algoritmi imali su izuzetno visoku tačnost od preko 90%. Dalje istraživanje će biti usmereno na rešavanje problema nebalansiranog skupa podataka, detekciju ironije i sarkazma.

ZAHVALNICA

Ovaj rad je rezultat istraživanja na projektu *AVANTES* (*Advancing Novel Textual Similarity-based Solutions in Software Development*) koji je finansiran od strane Fonda za nauku Republike Srbije, u okviru Programa za razvoj projekata iz oblasti veštačke inteligencije. Autori se zahvaljuju na finansijskoj podršci.

LITERATURA

[1] Tan, W., Wang, X. (2018) „Sentiment Analysis for Amazon Reviews", Stanford University.

[2] McAuley, J.J., Leskovec, J. (2013) „From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews", 22nd international conference on World Wide Web, Rio De Janeiro, Brazil.

[3] Pankjead, S. (2018) „Sentiment classification on Amazon reviews using machine learning approaches", KTH Royal Institute of Technology in Stockholm.

[4] Kim, Y. (2014) „Convolutional Neural Networks for Sentence Classification", Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 810-819.

[5] Shrestha, N., Nasoz, F. (2019) „Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings", International Journal on Soft Computing, Artificial Intelligence and Applications, Dostupno na: arxiv.org/ftp/arxiv/papers/1904/1904.04096.pdf

[6] Haque, T., Saber, N., Shah, F. (2018) „Sentiment Analysis on Large Scale Amazon Product Reviews", International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand.