SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 7025

Strojno učenje za analizu sentimenta u mikroblogovima

Ivan Križanić

Zagreb, lipanj 2020.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

ZAHVALA

SADRŽAJ

1. Uvod

Mikroblogovi su danas jedan od najčešće korištenih i najčešće proučavanih oblika komunikacije na internetu. Pronalaze se na iznimno popularnim društvenim mrežama, kao što su Twitter i Facebook, koje broje milijune korisnika diljem svijeta. Ljudi ih objavljuju u stvarnom vremenu, izražavajući svoje osjećaje, stavove i razmišljanja o svakodnevnom životu. Mnogi događaji i pojave u svijetu dobro su popraćeni reakcijama na društvenim mrežama, stoga je korisno proučavati velike skupove objava kao izvor stajališta, preferenci, osjećaja i mnogih drugih svojstava koja se daju izvući iz teksta objave.

Ovaj se rad bazira na mikroblogovima društvene platforme Twitter. Takozvani Tweetovi, mikroblogovi platforme Twitter, kratke su poruke sačinjene od najviše 140 znakova. Prvi je objavljen 2005. godine, a dvije godine kasnije dnevno se objavljivalo 5000 mikroblogova. Po zadnjim poznatim podatcima taj broj iznosi preko 500 milijuna objava dnevno (Salman Aslam, 2020). Radi se o iznimno velikom broju podataka koji kao skup mogu nositi korisne informacije, stoga ne čudi da postoje tvrtke koje u ponudi imaju analizu mikroblogova s Twittera i drugih društvenih platformi (*brandmentions.com, mention.com*). Povratna informacija korisnika vrijedan je resurs kojim se tvrtke mogu opskrbiti, stoga analiza društvenih platformi ima velik ekonomski i društveni značaj. Obradi tako velikog broja podataka pristupa se tehnikama strojnog učenja, a konkretno područje koje se primjenjuje za ovakve zadatke naziva se obrada prirodnog jezika, ili još preciznije, analiza sentimenta.

Rad se bavi problemom klasifikacije mikroblogova na one pozitivnog, neutralnog i negativnog sentimenta. Zadatak odgovara podzadatku A, četvrtog zadatka na natjecanju *Semeval 2017*, koji je u vrijeme održavanja privukao 39 timova iz cijelog svijeta. Ta je godina bila peta u nizu na kojoj se pojavio isti zadatak, što pokazuje interes zajednice za problem analize sentimenta. U sklopu zadatka napravljene su dvije implementacije modela za klasifikaciju. Jedna pripada standardnom strojnom učenju i temelji se na SVM-modelu s linearnom jezgrom, a druga pripada području dubokog učenja i temelji se na povratnoj neuronskoj mreži (engl. *Reccurent neural network*, RNN) s arhitektu-

rom ćelije s dugoročnom memorijom (engl. *Long short-term memory*, LSTM).

Rad je strukturno podijeljen na sljedeći način. U prvom se dijelu nalazi osvrt na radove koji su utjecali na ovaj rad, odnosno glavne izvore koji su bili motivacija za implementacije oba pristupa. U drugom dijelu osvrće se na implementaciju u ovom radu. Prvo se objašnjava odabir pristupa, a zatim i modeli korišteni u pristupima. Također su objašnjenje tehnologije korištene u obradi ulaznih podataka, te konačno i izrada značajki korištenih u treniranju modela. Treći dio rada opisuje podatke koji su korišteni i implementaciji, te prolazi kroz eksperimente i rezultate oba pristupa, da bi ih konačno i usporedio. Na kraju rada nalazi se zaključak i osvrt na moguća poboljšanja implementacije.

2. Srodni radovi

Na temu analize sentimenta napisano je mnogo radova, a velik broj bavi se upravo mikroblogovima s društvenih mreža i to vrlo često Twitterom. Uz to, u sklopu natjecanja *Semeval* neki natjecatelji objavljuju i rad u kojem se osvrću na svoju implementaciju rješenja. Stoga je dostupno puno informacija koje se mogu iskoristiti za vlastitu implementaciju, ali je istovremeno i otežano implementirati neviđeno rješenje. Najbolji rezultat implementacije u ovom radu ima točnost od 64.24%, što odgovara 10. mjestu na ljestvici predanih implementacija natjecanja *Semeval 2017* (Sara Rosenthal, Noura Farra, Preslav Nakov, 2017). Prvo mjesto s točnošću od 68.1% podijelila su dva tima: *DataStories* i *BB_twtr*. Upravo je tim *BB_twtr* zaslužan za aktualan *state-of-the-art* model u području analize sentimenta mikroblogova. Njihova trenutna implementacija hvali se da ostvaruje *F1-score* u iznosu od 68.5%.

U sljedećih nekoliko odlomaka osvrće se na radove koji su služili kao izvor metoda i ostalih informacija koje su korištene u izradi ove implementacije.

2.1. Rad tima *BB_twtr* - najuspješniji model današnjice

Prvi u nizu radova na koje se treba osvrnuti jest rad pobjednika natjecanja *Semeval 2017*, a ujedno i aktualni *state-of-the-art* model u području analize sentimenta mikroblogova. Radi se o radu *BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs* (Cliche, 2017). Poblemu su pristupili tehnikama dubokog učenja. Prva faza rada bavi se izradom vektora riječi koji su dalje korišteni u treniranju CNN i LSTM modela mreža. Eksperimentirali su s tri različite tehnike izrade vektora riječi (*Word2Vec, FastText, GloVe*). U drugoj su fazi nenadziranim učenjem razdijelili sentiment na negativan i pozitivan, jer je prije toga sentiment polariteta u vektorima bio vrlo slab. U trećoj su fazi provodili nadzirano učenje koristeći podatke s natjecanja i model izgrađen od 10 CNN i 10 RNN mreža sa LSTM arhitekturom koje koriste različit broj epoha za treniranje i različite vektore riječi. U podzadatku A postigli su točnost od 68.1%, a model su koristili i u ostala 4 podzadatka natjecanje te su u svim

zadatcima ostvarili najbolji rezultat.

Budući da navode CNN i RNN (sa LSTM arhitekturom) modele mreža kao najbolje u području analize sentimenta, u implementaciji ovog rada upotrebljava se RNN model mreže sa LSTM arhitekturom kako bi se upoznalo s njegovim mogućnostima. Umjesto izrade vektora riječi iz velikog skupa mikroblogova, koriste se gotovi vektori iz biblioteke *Spacy* koja koristi vektore izrađene metodom *Word2Vec*. Pri tome se gube prednosti posebnih značajki koje su karakteristične za jezik mikroblogova, a koje bi se mogle pokazati u vektorima nastalim na temelju mikroblogova, ali pristup je jednostavniji i štedi znatnu količinu računalne obrade koja bi bila potrebna za izradu vlastitih vektora.

2.2. Rad tima *DataStories*

U podzadatku A natjecanja *Semeval 2017*, zadatka 4, prvo mjesto dijelila su dva tima, ali tim *DataStories* imao je niži *F1-score*. Svoj su pristup opisali u radu *DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis* (Baziotis et al., 2017). S obzirom na to da su prethodnih godina ostvarili slabije rezultate, dok su timovi koji su koristili pristup dubokog učenja pretežno zauzeli pozicije na vrhu, *DataStories* tim odlučio je skrenuti pažnju s klasičnog strojnog učenja na duboko učenje. Rad su podijelili na dva osnovna koraka: obradu teksta i treniranje modela. Za obradu teksta implementirali su vlastite funkcije koje su primjenjive u općoj upotrebi, ali su usmjerene na obradu mikroblogova s Twittera. Za izradu vektora riječi koristili su 330 milijuna neoznačenih mikroblogova na engleskom jeziku. Na vokabularu od 660 tisuća riječi koristili su *GloVe* metodu izrade vektora. U obradi teksta koristili su vlastiti tokenizator koji je prilagođen Twitteru i posjeduje mogućnost izvlačenja raznih elemenata poput datuma, valuta, emotikona i sličnih sadržaja. Za razliku od njih, u svojoj implementaciji koristim implementaciju tokenizatora iz biblioteke *SpaCy* jer je pristupačna i široko korištena. U daljnjoj su obradi primijenili standardne postupke pročišćavanja teksta koji se koriste u obradi prirodnog jezika.

Osvrnuli su se na konvolucijske mreže (CNN) i naglasili problematiku gubitka informacije o poretku riječi prilikom uporabe istih. Iz tog su razloga preferirali RNN s arhitekturom ćelije s dugoročnom memorijom (LSTM). U svojoj su implementaciji koristili dvoslojni dvosmjerni model s mehanizmom za pozornost koji pospješuje prepoznavanje korisnih težina. U LSTM sloju modela koristili su 150 neurona i trenirali s podskupovima od 128 podataka. U testiranju su naveli kako mehanizam pozornosti doprinosi rezultatu za 0.04% te stoga nije implementiran u modele ovog rada.

2.3. Rad tima *TakeLab* - pristup klasičnim strojnim učenjem

Za razliku od velikog broja ekipa na natjecanju, tim *TakeLab* odlučio se za pristup klasičnim metodama strojnog učenja. Koristili su skup ručno izrađenih značajki i trenirali na SVM-modelu s linearnom jezgrom. Kao značajke koriste *Tf-Idf* i gotove vektore riječi, ali i neke specifične značajke poput leksikona pozitivnih i negativnih riječi te posebnu značajku po kojoj je rad dobio ime: "*Nedavne smrti i moć nostalgije*", odnosno originalni engleski naziv *Recent Deaths and the Power of Nostalgia in Sentiment Analysis in Twitter* (Lozić et al., 2017). Značajka se temelji na činjenici da je sentiment mikroblogova koji spominju nedavno preminule ljude pretežno pozitivan jer se ljudi obično prisjećaju pozitivnih stvari vezanih za pokojnika. Također su iskoristili svojstva nostalgije koja upućuju na pretežno pozitivan sentiment prilikom spominjanja pojmova i pojava iz prijašnjih vremena. Za značajne ljude kreirali su značajke koje opisuju osobe s atributima svojstvenima njihovoj društvenoj ulozi, a za pojmove kojima je često pridjeljenje nekakva ocjena, npr. filmovi, igrice, glazba i slično, napravili su značajke koje donose informacije o uspješnosti i popularnosti pojma. Svojim SVM-modelom ostvarili su solidan plasman u nekoliko zadataka, a u zadatku kojime se bavi ovaj rad ostvarili su 16. mjesto.

Zbog prisutnosti metode u njihovim radom, a i mnogim drugima koji koriste ručno izrađene značajke, u ovoj je implementaciji iskorišten leksikon pozitivnih i negativnih riječi i SVM-model. Naprednije i inovativne značajke koje čine njihov rad posebnim nisu implementirane u model ovog rada.

3. Implementacija

3.1. O zadatku 4 natjecanja *Semeval 2017* i analizi sentimenta u mikroblogovima

Verzija zadatka s kojim se ovaj rad bavi peta je u nizu na natjecanju *Semeval*. Kao i svih prijašnjih godina, zadatak je bio poprilično popularan i privukao 48 timova koji su sudjelovali u različitim podzadacima. U zadatku se kroz godine pojavilo nekoliko podzadatka kao što su ocjena pripadnosti sentimenta mikrobloga određenoj temi i skaliranje pripadnosti na skali od 1 do 5. Osnovna verzija zadatka bavi se klasifikacijom mikroblogova u tri razine polariteta, preciznije u mikroblogove pozitivnog, neutralnog i negativnog sentimenta.

Analiza sentimenta u tekstovima kao što su mikroblogovi s društvenih mreža donosi razne poteškoće s kojima se ne mora nositi kada je riječ o standardnijim oblicima teksta. Problematične karakteristike mikroblogova su niska ograničenost broja znakova koja uzrokuje sažet izraz, ali i povećava uporabu kolokvijalnih izraza, skraćenica i raznih suvremenih novotvorenica koje bismo mogli okarakterizirati kao *slang*. Prisutni su i razni elementi koji ne pripadaju prirodnom jeziku kao što su emotikoni, hiperlinkovi i razne oznake kao npr. oznaka korisničkog imena koja ima oblik *"@user"*. Hiperlinkovi u takvim kratkim tekstovima često nose velik teret značenja, odnosno često tek uz informaciju o sadržaju na koji hiperlink pokazuje možemo pravilno ocijeniti sentiment same poruke. Problem je i u pravopisu, korisnici često mijenjaju riječi radi postizanje vizualnog ili nekog drugog efekta, pa tako možemo naići na tvorevine poput: "ŁoŁ", "ca$h", "©ool", i slične koje bi bilo poželjno prepoznati i pretvoriti u smislene riječi ili kratice. Također je prisutno nizanje istog slova u riječima poput "*cool*" koje možemo pronaći u obliku kao što je "*cooool*" ili negaciji *"no"* kojoj se često nadodaje zadnje slovo *"o"*. Takvim je riječima također poželjno ukloniti suvišne znakove kako bi se pronašle u rječniku, ali treba imati na umu da takvo ponavljanje znakova nosi značenje u sebi, a koje bismo klasičnim ispravljanjem pravopisa izgubili.

Korisno bi bilo prepoznati i skrivene riječi kao što su *"F**k"*. *"S**t"*, *"N***a"* jer su to često riječi koje mogu znatno utjecati na sentiment objave, no to nije tako jednostavan zadatak zbog raznih metoda kojima se takve, često proste riječi, pokušavaju ukomponirati u tekstove.

Kada se odmakne od početne obrade teksta nailazi se na nove poteškoće kao što su korištenje sarkazma i učestalost ciničnog tona koji u potpunosti mijenjaju polaritet sentimenta, a koje je vrlo teško prepoznati iz perspektive modela. Ograničenost duljine poruke posljedično donosi manjkavost izraza koji se često bolje razumiju ako se posjeduje znanje o svijetu i vremenu u kojem su napisani, a ne samo o jeziku i značenju istog, što je još jedno svojstvo koje je vrlo teško ostvarivo modelima strojnog učenja.

3.2. Odabir metoda i pristupa

Korišteni pristupi spadaju u metode strojnog učenja. Strojno se učenje dijeli na tri osnovne vrste: (1) nadzirano strojno učenje, (2) nenadzirano strojno učenje i (3) podržano strojno učenje. Razlika proizlazi iz prisutnosti oznaka podataka na kojima se vrši učenje, odnosno ako su podatci korišteni u npr. klasifikaciji označeni s oznakama klase kojoj pripadaju, onda je riječ o nadziranom strojnom učenju, dok se o nenadziranom strojnom učenju radi ako su oznake klasa odsutne tijekom cijelog procesa učenja. U ovom se radu koristi samo varijanta nadziranog strojnog učenja jer su svi podatci označeni.

U rješavanju problema koriste se dva različita pristupa kako bi se osvijestilo o prednostima i manama jednog i drugog. Prvi pristup pripada klasičnim metodama strojnog učenja i temelji se na vlastoručnoj izradi značajki i upotrebi SVM-modela. Drugi pristup pripada grani strojnog učenja koja se naziva duboko učenje i temelji se na značajkama nastalima od vektora riječi i LSTM inačici modela povratne neuronske mreže.

3.2.1. Klasično strojno učenje - SVM-model

Strojevi potpornih vektora (engl. *Support-vector machine*) diskriminativni su modeli korišteni u nadziranom strojnom učenju, a koriste se u rješavanju klasifikacijskih i regresijskih problema (Jan Šnajder, 2014). U klasifikaciji se izvorno koriste za binarnu klasifikaciju, stoga implementacija u ovom radu koristi posebnu modifikaciju na koju će se osvrnuti naknadno. SVM rješava problem proizvoljnosti hipoteze uvođenjem kriterija maksimalne margine (engl. *maximum margin*). Naziv dolazi od takozvanih

potpornih vektora koji su kombinacija odabranih vektora iz skupa za učenje, a koji omogućuju prikaz hiperravnine kod modela. Proširenje učinkovitosti SVM-modela postiže se korištenjem jezgrenih funkcija postupkom trika jezgre (engl. *kernel trick*). SVM-model je linearan:

$$h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0,$$

a za nelinearnost se može upotrijebiti preslikavanje ϕ. Uz pretpostavku linearne odvojivosti i uzimajući da vrijedi $y \in \{-1, +1\}$ može se konstatirati da vrijedi:

$$\forall(\mathbf{x}^{(i)}, y^{(i)}) \in D : y^{(i)} h(\mathbf{x}^{(i)}) \geq 0.$$

Riječima rečeno može se tvrditi da za svaki par vektora značajki i oznake klase vektora postoji predikcija $h(\mathbf{x})$ koja je istog predznaka kao oznaka klase, odnosno umnožak predikcije i oznake klase uvijek je veći ili jednak 0. Ako su primjeri linearno odvojivi, postoji beskonačan broj rješenja koji zadovoljavaju navedeni izraz. Traži se rješenje maksimalne margine, što ima smisla jer je u interesu što bolje odvojiti klase primjera. Po definiciji je margina jednaka najmanjoj udaljenosti između $\mathbf{x}$ i hiperravnine, a cilj je pronaći za koju vrijednost $\mathbf{x}$ i w_0 margina ima maksimalan iznos, što vodi do sljedeće formule za izračun margine:

$$\underset{\mathbf{w},w_0}{\operatorname{argmax}} \left\{ \frac{1}{|\mathbf{w}|} \min_i \left\{ y^{(i)} \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}\left(\mathbf{x}^{(i)}\right) + w_0 \right) \right\} \right\}.$$

Ovom je formulom teško riješiti optimizacijski problem, pa se stoga problem oblikuje u problem konveksne optimizacije. Uzimajući pretpostavku da za $x^{(i)}$ koji je najbliži margini vrijedi:

$$y^{(i)} \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}\left(\mathbf{x}^{(i)}\right) + w_0 \right) = .1$$

Zbog toga mora vrijediti da za $\forall(\mathbf{x}^{(i)}, y^{(i)}) \in D$ vrijedi:

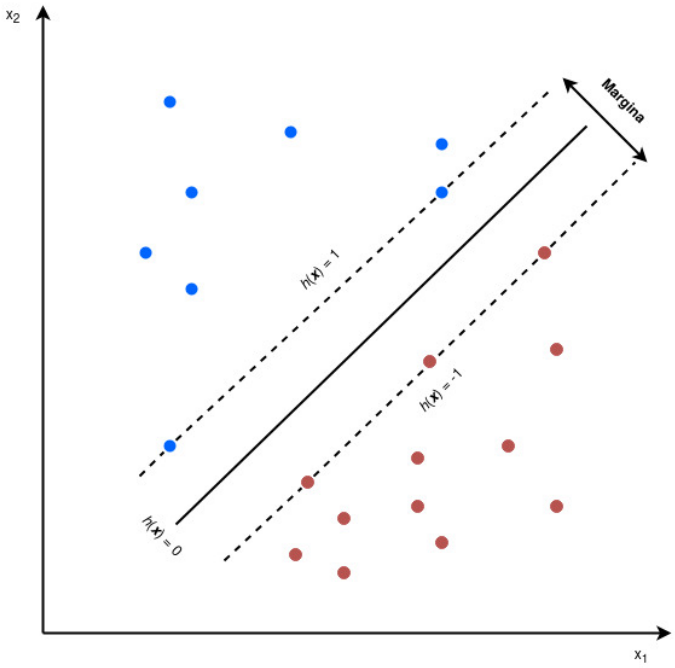
$$y^{(i)} \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}\left(\mathbf{x}^{(i)}\right) + w_0 \right) \geqslant 1, \quad n = 1, \ldots, N$$

Za primjere za koje vrijedi jednakost kažemo da su ograničenja aktivna, dok za ostale kažemo da su ograničenja neaktivna. Maksimizirana margina ima barem dva aktivna ograničenja, što se može vidjeti na slici 3.1. Širina maksimalne margine iznosi $\frac{2}{||\mathbf{w}||}$, pa se problem optimizacije može svesti na maksimizaciju izraza:

$$\underset{\mathbf{w},w_0}{\operatorname{argmax}} \frac{1}{||\mathbf{w}||}.$$

Kako bi se za optimizaciju mogla primijeniti metoda Lagrangeovih multiplikatora, izraz se piše kao:

$$\underset{\mathbf{w},w_0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2,$$



Slika 3.1: Maksimalna margina

jer minimum $||\mathbf{w}||^2$ jednak je maksimumu $\frac{1}{||\mathbf{w}||}$.
Time se problem svodi na problem kvadratno ograničenog kvadratnog programiranja (engl. *quadratically constrained quadratic programming)* i može se riješiti primjenom Lagrangeovog multiplikatora. Konačan izraz nastao kombiniranjem ciljne funkcije i uvjeta je sljedeća Lagrangeova funkcija:

$$L\left(\mathbf{w}, w_0, \boldsymbol{\alpha}\right) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left\{ y^{(i)} \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi} \left(\mathbf{x}^{(i)} \right) + w_0 \right) - 1 \right\}.$$

U daljnje postupke optimizacije ovaj rad ne ulazi, ali bitno je napomenuti da je rezultat optimizacije N-dimenzionalni vektor parametra α te da se klasifikacija neviđenog primjera vrši računanjem sljedećeg izraza, odnosno određujući njegov predznak:

$$h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) + w_0 = \sum_{i=1}^{N} \alpha_i y^{(i)} \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi} \left(\mathbf{x}^{(i)} \right) + w_0.$$

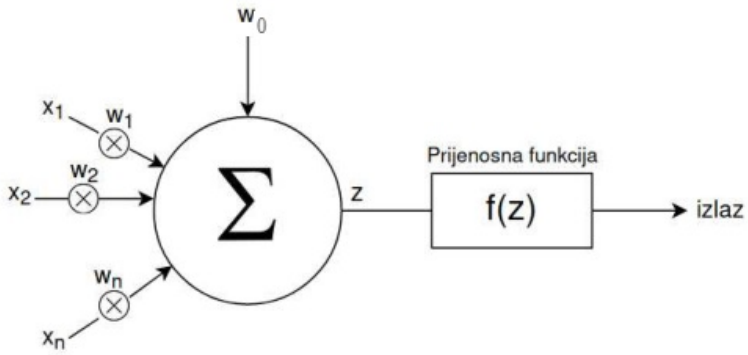
Problem višeklasne klasifikacije

Standardnom upotrebom SVM-model radi binarnu klasifikaciju, pa je za višeklasnu klasifikaciju $(K > 2)$ potrebno koristiti posebne metode. Osnovna ideja izvedbe jest modificirati problem višeklasne klasifikacije u više problema binarne klasifikacije. Postoji nekoliko načina na koji se ostvaruje željena modifikacija, a implementacija

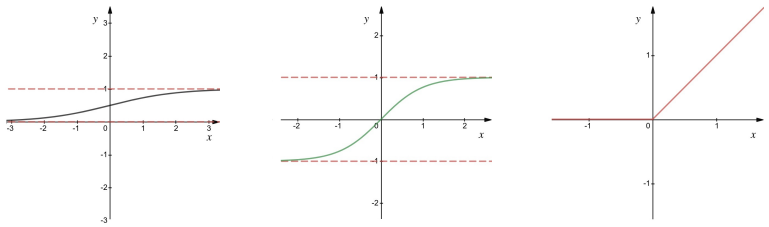
rješenja ma koju se rad osvrće koristi metodu jedan-naspram-ostali (engl. *one-vs-rest*) koja je zadana metoda knjižnice *scikit-learn*. Princip se temelji na svođenju problema na $K - 1$ problema binarne klasifikacije, gdje je K broj klasa koje početni problem može klasificirati. Tada svaki od $K - 1$ binarnih klasifikatora h_i odjeljuje klasu C_i od svih preostalih klasa. Problem pristupa javlja se ako više klasifikatora klasificira primjer kao pozitivan za svoju klasu, jer tada nije moguće jednoznačno odrediti klasu kojoj primjer pripada.

3.2.2. Duboko učenje

Duboko učenje grana je strojnog učenja čiji se modeli temelje na neuronskim mrežama (Čupić, 2016). Postoji mnogo modela i varijacija, a implementacija u ovom radu koristi povratnu neuronsku mrežu (engl. *Recurrent neural network*) s arhitekturom ćelije s dugoročnom memorijom (engl. *Long short-term memory*). Ostali poznati modeli neuronskih mreža su konvolucijske neuronske mreže (engl. *Convolutional neural network*), duboke neuronske mreže (engl. *Deep neural network*), duboka probabilistička mreža (engl. *Deep belief network*). Općenito, neuronske mreže složeni su sustavi čija je osnovna gradivna jedinica neuron čiji je zadatak propuštati težinsku sumu kroz prijenosnu funkciju određujući tako izlaznu vrijednost. Ulazi u neuron množe se s težinskim funkcijama $w_{1..n}$ te se sumiraju uz dodatak pomaka (engl. *bias*) w_0 daju vrijednost z koja propuštanjem kroz prijenosnu funkciju daje izlaz $f(z)$ iz neurona. Primjer neurona prikazan je na slici 3.2.



Slika 3.2: Umjetni neuron



(a) sigmoidalna funkcija **(b)** tanh(x) **(c)** *ReLU*

Slika 3.3: Aktivacijske funkcije

Prijenosne funkcije

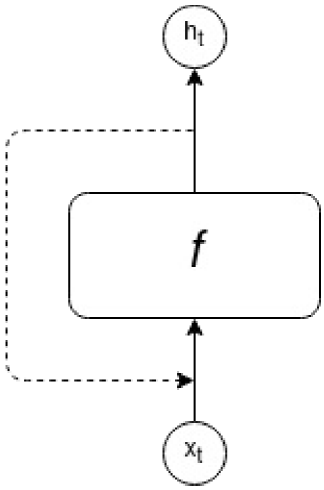
Propusne funkcije mogu biti razne, a najčešće su sljedeće:

- binarna step funkcija $f(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$
- sigmoidalna funkcija $\sigma(x) = \frac{1}{1+e^{-x}}$
- ReLU funkcija $max(0, x)$
- tanges hiperbolički funkcija $\tanh(x)$

Linearno ponašanje nije karakteristično za prirodne pojave, pa stoga linearne prijenosne funkcije nisu pogodne za upotrebu u područjima poput računalnog vida ili obrade prirodnog jezika. Nelinearne funkcije omogućuju bolju procjenu prirodnih fenomena kao što jest jezik. Najkorištenije nelinearne funkcije su sigmoidalna funkcija, tangens funkcija i *ReLU* funkcija. Sigmoidalna funkcija (slika 3.3(a)) najčešće je korištena funkcija izlaznog sloja kada se radi o binarnoj klasifikaciji jer skuplja sve vrijednosti na interval $[0, 1]$. Slična tome je i hiperbolička tangens funkcija (slika 3.3(b)) koja radi na intervalu $[-1, 1]$, ali zbog svoje derivacije pruža snažniji gradijent i možemo ju smatrati superiornijom u odnosu na sigmoidalnu funkciju (Ng). Funkcija koja je najčešće prisutna u LSTM arhitekturi RNN-a je *ReLU* prijenosna funkcija (slika 3.3(c)) koja svim negativnim vrijednostima pridaje vrijednost 0, dok pozitivne vrijednosti slijede linearnu funkciju. Postoji varijacija te aktivacijske funkcije po imenu *Leaky ReLU* koja za negativne vrijednosti slijedi funkciju $f(x) = 0.01x$, a za pozitivne se ponaša isto kao *ReLu* i slijedi funkciju $f(x) = x$ ili eventualno $f(x) = kx$.

Povratne neuronske mreže

Arhitektura mreže koja je implementirana u radu je povratna neuronska mreža (RNN) koja pripada porodici slojevitih unaprijednih neuronskih mreža. To znači da u sebi nema cikluse, odnosno ulazi neurona ne ovise o izlazima neurona koji se nalaze u



Slika 3.4: Princip ponašanja povratne neuronske mreže

dubljim slojevima. Karakteristika povratne neuronske mreže prisutnost je memorije. To im omogućuje obradu sekvencionalnih ulaza, odnosno mreže takvih arhitektura uzimaju u obzir poredak ulaznih podataka i time stvaraju povezanost između ulaza. Princip rada može se objasniti uz pomoć prikaza na slici 3.4. Za ulaz x_0 u prvom koraku vrijedi da je ulaz jednak x_0, a za svaki sljedeći ulaz vrijedi da je jednak x_t+h_{t-1}, odnosno kombinaciji izlaza iz prethodnog koraka i trenutnog ulaza iz skupa ulaznih podataka. Problem koji se javlja u RNN-arhitekturi jest takozvani gubitak gradijenta, odnosno događa se da vrijednost u neuronu postaje toliko beznačajna da je daljnje treniranje gotovo onemogućeno (Pascanu et al., 2013). Tome se doskače korištenjem ćelije s dugoročnom memorijom (LSTM).

Arhitektura ćelije s dugoročnom memorijom

Arhitektura ćelije s dugoročnom memorijom (LSTM) prikazana je na slici 3.5. Sastoji se od triju ulaznih vrata:

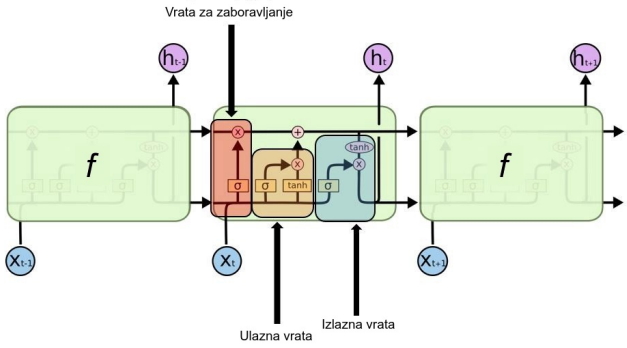
- ulazna vrata
- vrata za zaboravljanje
- izlazna vrata

Ulazna vrata odgovorna su za odabir vrijednosti koje će modificirati stanje u memoriji. Sastoje se od sigmoidalne i tangens hiperbolične funkcije. Sigmoidalna funkcija odgovorna je za odabir vrijednosti koje će sudjelovati u modifikaciji memorije, a tangens hiperbolični odgovoran je za pridjeljivanje odgovarajuće težine ulazu. Vrata za zaboravljanje odgovorna su za prebiranje vrijednosti iz prethodne iteracije i ulaznog

podatka i također funkcioniraju na temelju sigmoidalne funkcije. Izlazna vrata određuju izlaz koristeći ulazni podatak i stanje u memoriji, a kada je riječ o prijenosnim funkcijama izvedba im je jednaka ulaznim vratima. Funkcionalnost LSTM arhitekture zapisana jednadžbama je sljedeća:

$$\begin{aligned} f_t &= \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ i_t &= \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \\ o_t &= \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ \tilde{c}_t &= \sigma_h\left(W_c x_t + U_c h_{t-1} + b_c\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\ h_t &= o_t \circ \sigma_h\left(c_t\right) \end{aligned},$$

gdje je x_t vektor ulaznih podataka, f_t aktivacijski vektor vrata za zaborav, o_t aktivacijski vektor izlaznih vrata, i_t aktivacijski vektor ulaznih vrata, h_t izlazni vektor LSTM-a, $\tilde{c}_t$ aktivacijski vektor ulaza u ćeliju, c_t vektor stanja ćelije i konačno W, U i b matrice težina i pomaka koje je potrebno naučiti tijekom treniranja.



Slika 3.5: LSTM arhitektura

3.3. Obrada ulaznih podataka

Zbog prirode mikroblogova ulazni su podatci prošarani raznim značajkama koje je potrebno ukloniti ili preinačiti. Pri stvaranju podataka pogodnih za izradu značajki nastale su dvije vrste podataka.

Prva je jednostavna i služi izradi značajki temeljenih na frekvencijama riječi ili vektorima riječi. Dobivena je tako što je prvo provedena zamjena emotikona s njihovom

jezičnom reprezentacijom u smislu da je npr. ":-)" pretvoreno u *"happy"*. Zatim je provedeno uklanjanje svih nejezičnih elemenata kao što su hiperlinkovi, korisnička imena, emotikoni, brojevi i slično. Mikroblogovi su rastavljeni na riječi koristeći biblioteku *Spacy* te je nad dobivenim skupom jezičnih elemenata proveden postupak ispravljanja pravopisa i prepoznavanje žargona. Uklonjene su riječi koje ne doprinose značenju. Takve se riječi nazivaju zaustavne riječi (engl. *stop-words*) i uklonjene su koristeći biblioteku za obradu prirodnog jezika *Natural Language Toolkit (NLTK)* (Bird et al., 2009). Na preostalim riječima proveden je postupak lematiziranja, odnosno pretvaranja riječi iz izvedenog oblika u njen korijenski oblik, takozvanu lemu. Za postupak lematiziranja korišten je `WordNetLemmatizer` iz spomenute biblioteke NLTK.
Druga vrsta sastoji se od mikroblogova koji su označeni koristeći alat koji je napravio spomenuti tim *DataStories* s natjecanja *Semeval* (Baziotis et al., 2017). U mikroblogu se ovim postupkom označavaju elementi poput hiperlinkova, cenzuriranih riječi, brojeva, riječi napisanih velikim slovima, korisničkih imena i slično. Takve su podatci ostavljeni u obliku teksta, odnosno nisu razlomljeni na manje elemente, jer su korišteni za prebrojavanje prisutnosti spomenutih elemenata.

3.4. Značajke

S obzirom na to da se paralelno koriste dva pristupa izrade i treninga modela, izrada i korištenje značajki također je podijeljena u dva smjera. Izrada značajki za klasično strojno učenje znatno je opsežniji i kreativniji proces nego izrada istih za pristup dubokim učenjem. Moglo bi se reći da je srž klasičnog pristupa upravo u izradi značajki jer se modeli sami po sebi ne mogu značajno konfigurirati, pa rezultat najviše ovisi o onome što mu se na ulazu pruži. Kod dubokog učenja postoje jednostavni standardni pristupi koji su ponekad gotovo mandatorni.

3.4.1. Značajke u klasičnom pristupu

Značajke u ovom pristupu čine okosnicu uspjeha modela, pa je stoga izradi posvećen znatan udio vremena. Značajke se mogu grupirati u tri kategorije:

- brojanje riječi i vektori riječi
- polaritet i sentiment riječi
- brojanje prisutnosti elemenata

Brojanje riječi i vektori riječi

U ovoj se kategoriji nalaze dvije vrste značajki koje se međusobno isključuju, odnosno ne koriste se istovremeno. Prva značajka je uobičajena kao početni uzorak značajki koji se koristi za treniranje osnovnog modela (engl. *baseline model*) kao referenca za daljnje eksperimente. Radi se se o metodi vreće riječi (engl. *Bag-of-Words*), odnosno preciznije o primjeni mjere učestalosti riječi *TF-IDF* (engl. *term frequency-inverse document frequency*). Mjera se definira na sljedeći način: potrebno je definirati dvije zasebne statističke mjere – mjeru učestalosti izraza (*tf*) i inverznu učestalost u dokumentima (*idf*). Prvu mjeru koja označava učestalost pojave riječi računamo na sljedeći način:

$$\mathrm{tf}(t, d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

gdje t predstavlja izraz, d predstavlja dokument, odnosno skup svih izraza u promatranom tekstu, $f_{t,d}$ predstavlja broj pojavljivanja izraza u tekstu, a brojnik predstavlja najveći broj pojavljivanja nekog izraza u tekstu. Ova se normalizirana verzija učestalosti pojave izraza koristi radi sprječavanja pristranosti prema velikim tekstovima. Druga mjera koju određujemo je inverzna učestalost izraza u dokumentu i računa se na sljedeći način:

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

gdje D predstavlja skup svih tekstova na kojima računamo učestalost, a nazivnik razlomka predstavlja broj pojavljivanja izraza u tekstu, dok je N ukupan broj dokumenata u skupu D. Konačna mjera jednaka je umnošku dviju prethodno izračunatih mjera, odnosno:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

U implementaciji rješenja koristim gotovu metodu iz knjižnice *scikit-learn*. Korištenjem nastalih značajki u SVM-modelu dobio sam točnost od 41.22%, što je korektan osnovni model s obzirom na točnost nasumičnog odabira koja iznosi 33.3%.

Druga značajka koju sam uveo, a koja pripada ovoj kategoriji, temelji se na vektorima riječi iz knjižnice *Spacy*. Radi se o vektorima s 300 dimenzija nastalima primjenom *Word2vec* (Mikolov et al., 2015) metode. *Word2vec* je model plitke neuronske mreže s dva sloja koji treniranjem pokušava rekonstruirati lingvističko značenje riječi. Kao ulaz koristi vrlo velik skup tekstualnih podataka koji mogu biti raznog porijekla, kao

npr. članci *Wkipedie*, objave na društvenim mrežama, primjerci elektroničke pošte itd. Kao rezultat nastaju višedimenzionalni vektori čije se dimenzije obično kreću između 100 i 1000 dimenzija. Vektori su u prostoru smješteni tako da su vektori riječi bliskog znanja prostorno bliski jedan drugome.
Koristeći prethodno izrađene informacije proizašle iz metode *tf-idf* kodirao sam ulazne informacije vektorima riječi tako da sam izračunao prosječnu vrijednost vektora svih riječi koje se pojavljuju u mikroblogu, a za broj značajki u tako nastalom vektoru odabrao sam vrijednost broja riječi u najduljem mikroblogu. Poboljšanje nastalo korištenjem ovih značajki značajno je povećalo uspješnost modela koji je nakon treniranja imao točnost od 60.9%

Polaritet i sentiment riječi

S obzirom na to da je zadatak klasifikacija s obzirom na polaritet mikrobloga, bilo je nužno dotaknuti se polariteta i sentimenta samih riječi. Za to su iskorištena dva leksikona. Prvi od njih je leksikon ocjena riječi koji sadrži ocjene po atributima zadovoljstva, uzbuđenosti i dominantnosti po imenu *Affective Norms for English Words* (Bradley i Lang, 1999). Prva verzija sastojala se od nešto više od tisući riječi, no 2013. godine proširena je na 14000 (Warriner et al., 2013). Pri implementaciji rješenja korišten je pristup temeljen na pristupu koji je prisutan u repozitoriju *dwzhou/SentimentAnalysis* (Doris Zhou, 2020). Drugi leksikon koji je korišten je zapravo običan popis pozitivnih i negativnih riječi (Hu, 2006). Dodavanjem značajki dobivenih korištenjem leksikona podigao sam točnost modela za 1%, odnosno postigao točnost od 61.9%.

Brojanje prisutnosti elemenata

U ovoj se kategoriji nalaze značajke nastale brojanjem ili promatranjem prisutnosti raznih elemenata u mikroblogovima. Elementi čija je prisutnost označena zastavicama 0 ili 1 su: *e-mail* adrese, hiperlinkovi, znakovi valute, datumi, telefonski brojevi itd. Za neke se elemente bilježio točan broj pojavljivanja u mikroblogu, a izrazi za koje je bilježena ta informaciju su: izrazi s nizanjem znakova (npr. *"cooool"*), izrazi napisani velikim slovima, cenzurirani izrazi (npr. *"F****), broj ponovljenih riječi, pojave takozvanih *hashtagova* i broj uskličnika. Kao dodatna značajka nadodan je i ukupan broj riječi u rečenici. Dodavanjem ovih značajki ostvaren je porast točnosti od 0.4%, odnosno postignuta je konačna točnost SVM-modela od 62.32%.

3.4.2. Značajke u dubokom učenju

Izrada značajki korištenih u modelu dubokog učenja znatno je jednostavnija. Potrebno je izgraditi vokabular riječi koje se pojavljuju u skupu podataka i svakoj riječi u vokabularu pridijeliti redni broj koji će služiti kao oznaka riječi. Zatim je potrebno izgraditi matricu vektora riječi (engl. *embedding matrix*) koja se sastoji od onoliko stupaca koliko vektor riječi ima dimenzija, što je u slučaju ove implementacije 300 dimenzija. U redovima su po rednim brojevima iz vokabulara kodirane riječi odgovarajućim vektorima riječi. Model tijekom inicijalizacije prima matricu vektora. Pomoću izrađenog vokabulara vrši se kodiranje sadržaja mikroblogova, odnosno umjesto tekstualnog sadržaja mikroblogovi postaju vektori brojeva koji predstavljaju redni broj riječi u vokabularu. Radi konzistentnosti dimenzija odabrana je maksimalna veličina vektora koja odgovara najvećem vektoru u skupu podataka za treniranje, a svi manji vektori nadopunjavaju se nulama do željene veličine. Takav skup vektora predaje se modelu kao ulaz u treningu i u evaluaciji modela.

4. Provedba eksperimenata

Ovo poglavlje opisuje proces provođenja eksperimenata i osvrće se na postignute rezultate. Detaljno opisuje karakteristike implementacija i uspoređuje učinke izmjena modela koje su se činile tijekom eksperimentiranja, kao i podešavanje hiperparametara.

4.1. Podatci

Skup podataka za trening sastoji se od 49491 mikroblogova, dok se skup podataka za testiranje sastoji od 12258 mikroblogova. Podatci su odmah podijeljeni na one za treniranje i one za testiranje jer je sam skup podataka proizašao iz natjecanja *Semeval 2017*, pa su korišteni originalni podatci za odgovarajuće faze natjecanja, tako da su rezultati postignuti na testnom skupu podataka mjerodavni onima koji su dobiveni kao rezultati natjecanja. Podatci se strukturno sastoje od identifikacijskih brojeva objava, oznake polariteta objave i teksta objave. Identifikacijske oznake izbačene su prilikom učitavanja jer niti jedna značajka ne proizlazi iz njih. Primjer jedne originalne objave:

```
"(OFF TOPIC) - there is only 3 episodes on the first disk of #Dexter.
Please hurry, @netflix with the 2nd #fitblog"
```

Ti su podatci obradom poprimili oblik prikladan izvlačenju značajki, pa je prethodno spomenuta objava pretvorena u dvije vrste podataka. Prva vrsta jest popis riječi koje se nalaze u objavi, a koje ne pripadaju zaustavnim riječima (enlg. *stop-words*), a druga vrsta je tekst koji sadrži oznake bitnih elemenata i svojstava objave. Primjer obje vrste podataka:

```
['topic', 'episode', 'first', 'disk', 'dexter', 'please', 'hurry',
'fit', 'web', 'log']
```

```
( off topic ) - there is only episodes on the first disk of dexter.
please hurry , with the 2 nd fit blog .
```

Sastav

Što se udjela podataka tiče, vidljiva je razlika u odnosu na skup podataka za treniranje i skup za testiranje. Mikroblogova neutralnog polariteta ima podjednako mnogo, ali u skupu za trening ima više nego dvostruku više pozitivno označenih mikroblogova nego negativnih, dok u skupu za testiranje ima 50% više mikroblogova s negativnom oznakom. Precizni podatci o broju i udjelima mikroblogova vidljivi su u tablici 4.1.

	pozitivni	neutralni	negativni
Podatci za treniranje	19652 (39.64%)	22195 (44.78%)	7723 (15.58%)
Podatci za testiranje	2375 (19.33%)	5937 (48.33%)	3972 (32.33%)

Tablica 4.1: Zastupljenost polariteta mikroblogova u podatcima

4.2. Eksperimentiranje

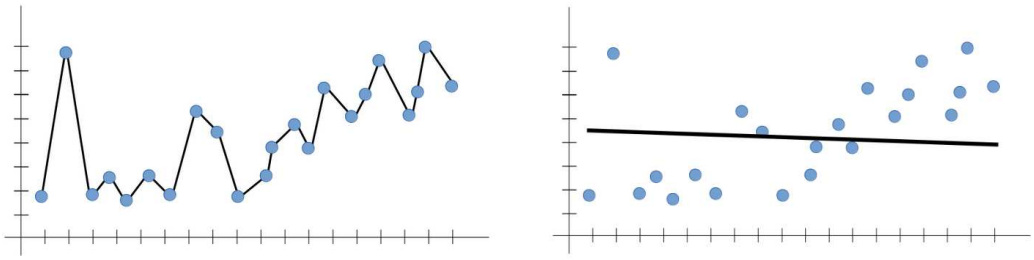
Metrika korištena u obje implementacije zasniva se na prosječnom odazivu (engl. *Average recall*) svake od tri klase u koje su se mikroblogovi trebali klasificirati. Formalno definirano metrika je sljedeća:

$$rezultat = \frac{(R^P + R^N + R^U)}{3},$$

gdje su pribrojnici brojnika redom jednaki točnosti prepoznavanja pozitivnih, negativnih i neutralnih primjera. Korištena metrika odabrana je zato što je korištena na posljednjoj godini natjecanja *Semeval* i zato što se bolje ponaša kod nebalansiranog skupa podataka kakav je skup na kojemu se vrši treniranje i testiranje u ovoj implementaciji.

Problem prenaučenosti i nedovoljne naučenosti

Prilikom eksperimentiranja bitno je izbjeći dvije najčešće greške u treniranju modela, prenaučenost (engl. *overfitting*) i nedovoljnu naučenost (engl. *underfitting*). Kod prenaučenosti model pretjerano dobro klasificira primjerke iz skupa za učenje, odnosno pretjerano prilagođava funkciju predviđanja podatcima koji su mu dostupni što negativno utječe na moć generalizacije modela. Takav model imat će pretjerano visoku



(a) Prenaučenost **(b)** Nedovoljna naučenost

Slika 4.1: Greške u treniranju modela (Al-Masri, 2019)

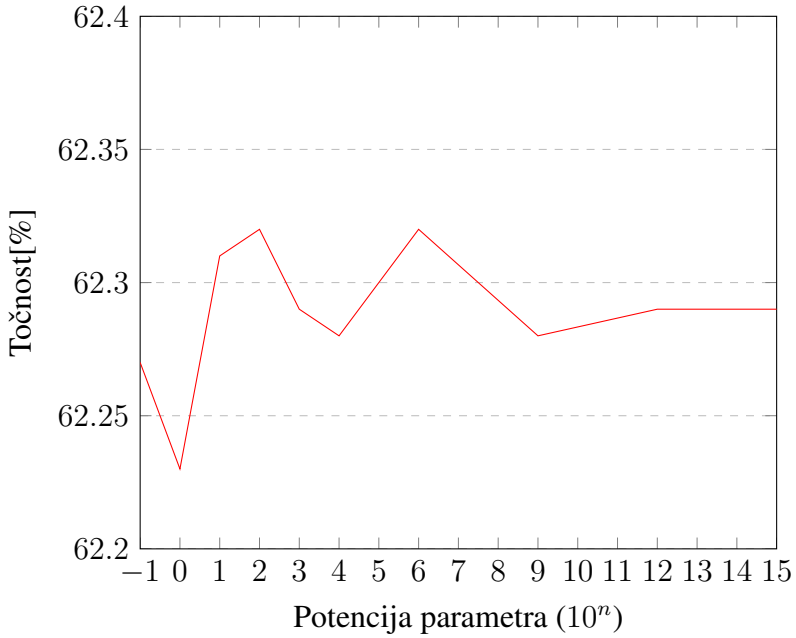
točnost na skupu za treniranje i znatno lošiju točnost na skupu za testiranje. S druge strane, nedovoljno treniranje modela je situacija u kojoj model nije dovoljno prilagodio funkciju predviđanja skupu ulaznih podataka i na taj je način napravio funkciju koja nedovoljno dobro generalizira primjerke iz skupa za treniranje, ali i posljedično loše klasificira i primjerke iz skupa za testiranje. Kako bi se takve greške izbjegle potrebno je nadzirati proces učenja modela. Prilikom svake epohe u treningu neuronske mreže mjeri se pogreška na podatcima za treniranje i podatcima za validaciju. Cilj je prekinuti treniranje u trenutku kada se greška na podatcima prestane smanjivati, odnosno prije nego počne rasti. Ako se trening prekine puno prije nego greška na skupu za validaciju prestane padati model će biti nedovoljno naučen. Ako se pak dopusti treniranje nakon što greška na skupu za validaciju počne rasti, model će biti prenaučen i sposobnost generaliziranja na neviđenim primjerima bit će mu smanjena.

4.2.1. SVM-model

Svi eksperimenti u implementaciji metodama klasičnog strojnog učenja rade se na SVM-modelu iz knjižnice *scikit-learn*. Budući da su dostupni odvojeni podatci za treniranje i testiranje, ne vrši se križna validacija (engl. *cross-validation*), već se treniranje i testiranje odvija na zadanim podatcima. Jedini hiperparametar koji se može konfigurirati prilikom treniranja SVM-modela s linearnom jezgrom jest parametar c koji utječe na širinu maksimalne margine prilikom razdvajanja klasa u podatcima. Odabir manje vrijednosti hiperparametra c utječe na povećanje širine maksimalne margine, ali kao posljedicu ostavlja dio podataka pogrešno razdijeljen. Odabirom veće vrijednosti smanjuje se širina maksimalne margine hiperravnine, ali zato manje podataka biva krivo razdijeljeno tijekom treniranja. Vrijednosti hiperparametra c koje ima smisla probati dolaze iz iznimno širokog intervala, stoga se u ovoj implementaciji vrši

Značajke	Rezultat
Tf-Idf (*Bag-of-Words*)	41.22%
Spacy vektori riječi	60.90%
Spacy vektori riječi + *ANEW* + polaritet riječi	61.91%
Spacy vektori riječi + *ANEW* + polaritet riječi + značajke prebrojavanja	62.32%

Tablica 4.2: Rezultati eksperimenata SVM-modela



Slika 4.2: Utjecaj hiperparemetra c na točnost SVM-modela

treniranje za vrijednosti hiperparametra od 10^{-7} do 10^{15} povećavajući vrijednost parametra 10 ili 100 puta svakom iteracijom. Početni se model temelji na metodi vreće riječi (engl. *Bag-of-words*), odnosno na primjeni metode *Tf-Idf*. Najgori mogući model nasumičnog pogađanja imao bi točnost od 33.32%, a početni model ima točnost od 41.22% što ga čini korektnim početnim modelom. S obzirom na to da je razvoj značaka opisan u ranijim poglavljima, a jedini hiperparametar je c, rezultati eksperimenata sa značajkama opisani su na tablici 4.2, a utjecaj hiperparametra c na točnost prikazan je slikom 4.2. Model postiže identičnu točnost s vrijednostima hiperparametra 10^2 i 10^6 koja iznosi 62.32%. Vrijednosti manje od 10^{-1} nisu prikazane na grafu, iako je proveden eksperiment i s njima, ali rezultat je znatno lošiji.

4.2.2. LSTM-model

Eksperimentiranje u implementaciji koja koristi metode dubokog učenja znatno je većeg opsega nego kod klasičnog strojnog učenja. LSTM arhitektura RNN-modela ima velik broj parametara koji se mogu konfigurirati, pa je stoga skup mogućih postava eksperimenata prevelik da bi se u realnom vremenu isprobao. Hiperparametri koji utječu na točnost predviđanja koju utrenirana neuronska mreža ostvaruje su sljedeći: broj neurona u LSTM sloju, veličina mini-serije (engl. *mini-batch*), broj LSTM slojeva, odabir optimizacijske i propusne funkcije, udio neurona koji se napuštaju (engl. *dropout*), stopa učenja (engl. *learning rate*), podrezivanje gradijenta (engl. *gradient-clipping*) (Reimers i Gurevych, 2017). Svi hiperparametri ne pridonose jednako poboljšanju točnosti, a zbog prevelikog broja kombinacija parametara, eksperimenti su provođeni u raznim etapama kako bi se što bolje precizirao skup eksperimenata koji donose najbolje rješenje. Početni model odabran kao referenca za daljnje eksperimente jednoslojna je povratna neuronska mreža s arhitekturom LSTM koja broji 30 neurona trenirana do najboljeg mogućeg rezultata s veličinom mini-serije od 2048 uzoraka i bez ikakvih dodatnih hiperparametara. Zanimljivost rezultata te mreže jest to što je točnost predviđanja pozitivnih i neutralnih primjera visoka (65% i više), a točnost predviđanja negativnih primjera iznosi puno nižih 20%.

Jednosmjerna i dvosmjerna varijanta LSTM arhitekture

Dodatna značajka koju je potrebno odabrati je vrsta LSTM arhitekture. Postoji jednosmjerna (engl. *undirectional*) i dvosmjerna (engl. *bidirectional*) arhitektura. Nadogradnja na jednosmjernu arhitekturu prilično je jednostavna. U jednosmjernoj arhitekturi za izračun vrijednosti koristi se prethodni izlaz i trenutna ulazna vrijednost, što bi se moglo definirati kao *pogled u prošlost*. Kod dvosmjerne arhitekture konačna vrijednost računa se kao kombinacija vrijednosti dobivene u prolazu u jednom i u drugom smjeru. U glavnini eksperimenata korištena je verzija s dvosmjernim prolazom i korištenjem iste ostvaren je i najbolji rezultat.

Optimizacijska funkcija i stopa učenja

Najkorištenija optimizacijska funkcija u LSTM arhitekturi povratne neuronske mreže je *Adam* (Kingma i Ba, 2014). Temelji se na stohastičkom gradijentom spustu (engl. *Stohastic Gradient Descent*, SGD) koji predstavlja osnovni optimizacijski algoritam u dubokom učenju. Za razliku od standardnog pristupa koji unaprijed određuje kako će se stopa učenja mijenjati, *Adam* koristi adaptivnu stopu učenja, odnosno korigira ju u

iteracijama učenja. Upravo je po adaptivnoj stopi učenja i dobio ime (engl. *adaptive moment estimation*). U mnogim situacijama pokazao se kao dobar odabir optimizacijske funkcije, a zbog svoje brzine pogodan je za eksperimente, stoga implementacija u ovom radu koristi isključivo *Adam* optimizacijsku funkciju. Formalno definirano proces osvježavanja parametara je sljedeći:

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1) \nabla_w L^{(t)}$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1 - \beta_2) \left(\nabla_w L^{(t)}\right)^2$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1 - \beta_1^{t+1}}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1 - \beta_2^{t+1}}$$

$$(w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon},$$

gdje je L^t funkcija gubitka, a w^t parametri u iteraciji t, ϵ je mala konstanta koja sprječava dijeljenje s nulom, a β_1 i β_2 su faktori zaboravljanja gradijenta.

Broj neurona u sloju i broj LSTM slojeva

Dodavanjem slojeva neuronskim mrežama povećava se mogućnost apstrakcije i pronalaženja povezanosti u sadržaju. To je osobito korisno u obradi prirodnog jezika, jer jezične povezanosti ponekad su na visokoj razini apstrakcije. Kada je u pitanju primjena LSTM arhitekture, više slojeva ni u kojem slučaju ne garantira poboljšanje modela. Nizom testiranja utvrđeno je da su dva sloja optimalna u ostvarenju rješenja zadatka kojim se ovaj rad bavi. Kada su u pitanju veličine slojeva, odnosno broj neurona u slojevima, najbolji je rezultat ostvaren korištenjem samo 30 neurona, a vrlo sličan rezultat ostvaren je i s $50, 75$ i 150 neurona, što opet pokazuje da veći broj nije nužno garancija boljeg rješenja.

Napuštanje neurona

Osnovna zadaća tehnike napuštanja neurona (engl. *dropout*) je sprječavanje prenaučenosti modela. Princip je rada jednostavan, sa zadanom konstantom $\alpha \in [0, 1)$ određuje se udio neurona koji će biti napušteni, a odabir neurona koji se napuštaju prepušten je nasumičnom odabiru. Većina najboljih rezultata ostvarena je korištenjem $\alpha = 0.5$, a jedan primjer porasta maksimalne točnosti modela povećanjem parametra α vidljiv je u tablici 4.3.

α	Najbolji rezultat
0	62.32%
0.25	63.33%
0.5	**63.99%**

Tablica 4.3: Utjecaj hiperparametra α na točnost LSTM-modela

Veličina mini-serije	Najbolji rezultat
16	61.35%
32	**64.24**%
64	63.99%
128	63.04%
258	63.36%

Tablica 4.4: Utjecaj veličine mini-serije na točnost LSTM-modela

Veličina mini-serije

Veličina mini-serije (engl. *mini-batch*) predstavlja broj elemenata u podskupu podataka na kojem model iterativno trenira prilagođavajući svakom iteracijom težine. Smanjivanjem veličine mini-serije značajno se usporava treniranje, ali provođenjem eksperimenata pokazalo se da su manje veličine mini-serije pogodnije za ovu vrstu problema. Iako je broj eksperimenata nedovoljan da bi se izvukao valjan zaključak, može se pretpostaviti da je optimalan raspon veličine mini-serije u intervalu između 32 i 64. Rezultati u tablici 4.4 ostvareni su različitim kombinacijama hiperparametara, a najbolji rezultat za svaku veličinu mini-serije zapisan je u tablici. Eksperimenti su provedeni i na većim veličinama mini-serije, ali zbog uočene tendencije da je manja veličina pogodnija za treniranje glavnina eksperimenata provedena je s veličinama iz intervala $[16, 256]$.

Funkcija gubitka

Važna komponenta modela je funkcija gubitka (engl. *loss function*). Standardni odabir za problem višekategorijske klasifikacije je funkcija kategoričkog gubitka unakrsne entropije (engl. *categorical cross-entropy loss*). Koristi se kada je primjer potrebno klasificirati u jedinstvenu klasu, odnosno kada je moguća samo jedna točna klasifika-

cija. Formula za izračun gubitka je:

$$L(l, p) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(l_{ij} * \log\left(p_{ij}\right)\right),$$

gdje je l oznaka klase, a p predikcija. Izračun se provodi nad binarnim vektorom oznaka klasa koji se sastoji od jedne jedinice i $k-1$ nula, gdje je k broj klasa u koje se vrši predikcija. Takvo se kodiranje naziva jedinično vruće kodiranje (engl. *one-hot encoding*). Slikovito rečeno funkcija uspoređuje vektor predviđanja s binarnim vektorom oznaka.

Podrezivanje gradijenta

Primjena podrezivanja gradijenta nije se pokazala korisna, dostatan broj eksperimenata pokazao je da gotovo nema nikakav utjecaj na točnost, a ukoliko je i postojala mala razlika, ona nije uvijek bila niti u pozitivnom niti u negativnom smjeru, stoga podrezivanju gradijenta u većini eksperimenata nije pridana pozornost.

Programska implementacija

Implementacija modela ostvarena je korištenjem knjižnice *Keras* (Chollet et al., 2015) koja kao osnovu koristi knjižnicu *Tensorflow* (Abadi et al., 2015). U njoj su ukomponirani svi potrebni elementi korišteni prilikom izrade modela i provođenja eksperimenata. Odabir optimizacijske i propusne funkcije vrši se pozivom njenog imena, jer su sve poznatije funkcije ukomponirane u knjižnicu. Također pruža mogućnost pohrane izgrađenog modela što značajno olakšava eksperimentiranje s brojem epoha treniranja. Jedina prilagodba koja je bila nužna zbog metrike koja se koristila na natjecanju *Semeval* bila je dodavanje funkcije koja računa prosječan iznos odziva po kategorijama u koje je model vršio klasifikacije.

U sklopu knjižnice *Keras* postoji mogućnost automatskog uranjenog završetka treniranja (engl. *early-stopping*) oslanjajući se na željeni parametar kao što je npr. greška predviđanja na skupu podataka za validaciju. Budući da je broj epoha za sve varijacije implementacije manji od 10, automatsko prekidanje manje je pouzdano od ručnog nadgledanja treninga, stoga metoda nije implementirana u treniranje modela.

4.3. Osvrt na rezultate

Implementacija klasičnim metodama strojnog učenja ostvarila je konačnu točnost od 62.3%, a implementacija metodama dubokog učenja 64.24% i time ušla u 10 najbo-

	verzija	negativni	neutralni	pozitivni
SVM-model	početni	24.19%	33.79%	65.7%
SVM-model	konačni	54.26%	62.19%	70.5%
RNN sa LSTM arhitekturom	početni	22.96%	66.48%	64.22%
RNN sa LSTM arhitekturom	konačni	64.23%	59.74%	68.74%

Tablica 4.5: Točnost modela po klasama

ljih rezultata natjecanja *Semeval*. Rezultat je zadovoljavajuć s obzirom na manji opseg eksperimenata koji su provedeni na modelu s LSTM arhitekturom. Nekoliko je postupaka koji bi se mogli implementirati kako bi se model unaprijedio kao što je izrada vlastitih vektora riječi na temelju mikroobjava s *Twittera* umjesto korištenja gotovih vektora riječi iz knjižnice kao što je *Spacy*. Također bi se mogao implementirati mehanizam pozornosti (engl. *Attention mechanism*) koji je zadnjih godina zaprimio značajnu pozornost, a koji je jedan od razloga uspješnosti Googleova modela *BERT*. Zbog ograničenog vremena valjano je isproban samo jedan optimizacijski algoritam (*Adam*), što ostavlja mogućnost daljnjeg povećanja točnosti. Što se SVM-modela tiče, implementirane su poprilično jednostavne značajke, pa je rezultat i viši nego bi se moglo očekivati, ali implementiranje pametnijih značajki svakako bi doprinijelo porastu točnosti. U tablici 4.5 vidljiva je točnost modela, odnosno odziv po svakoj klasi u koju se radila klasifikacija. Oba početna modela najslabije su klasificirali negativne primjere. SVMmodel najbolje je klasificirao pozitivne primjere, a RNN-model sa LSTM arhitekturom podjednako dobro je klasificirao pozitivne i neutralne primjere. U konačnom modelu došlo je do promjene u odzivu pozitivnih i negativnih primjera za manje od 10% u odnosu na početni model, ali točnost odziva za negativne primjere skočila je na gotovo 300% početne točnosti. U SVM modelu značajno je narasla točnost odziva za neutralne i negativne primjere. Za negativne primjere porast je iznosio preko 100%, a za neutralne nešto manje od 100%. Prepoznavanje pozitivnih primjera također je naraslo za nešto manje od 10%.

5. Zaključak

Cilj je ovog rada upoznavanje s analizom sentimenta u tekstu, posebice mikroblogovima društvene mreže *Twitter*. Tema je obrađena uz pomoć zadatka s natjecanja *Semeval* u kojem je osnovni podzadatak određivanje polariteta mikroobjave, odnosno klasifikacija objave u pozitivnu, neutralnu ili negativnu skupinu. Najbolji model za rješavanje navedenog zadatka ostvaruje prosječan odziv u iznosu od 68.1%. U ovom radu rješavanju problema pristupljeno je iz dva pravca. Jedan je koristio metode klasičnog strojnog učenja, konkretno SVM-model, a drugi metode dubokog učenja, konkretno RNN-mrežu sa LSTM arhitekturom. Prvim pristupom ostvaren je prosječan odziv od 62.3%, a drugim pristupom ostvaren je bolji rezultat od 64.24%. S tom točnošću model zauzima $10.$ mjesto na natjecanju *Semeval 2017*. U radu je proučena izrada značajki za oba pristupa i time se pokazalo kako su pristupi izradi značajki vrlo različiti, odnosno da u slučaju primjena klasičnih metoda strojnog učenja zahtijevaju puno više pozornosti. Uspješnost modela u klasičnom pristupu počiva upravo na izradi značajki, a s druge strane, u dubokom učenju izrada značajki malen je korak u izradi modela. Većina posla koji je potrebno odraditi prilikom gradnje i treniranja modela u dubokom učenju odlazi na odabir i prilagodbe značajki modela i hiperparametre. Opseg eksperimenata koji zahtjeva kvalitetno pronalaženje hiperparametara vremenski je i računski zahtjevan posao. Rezultat modela daleko je od najboljeg koji je ostvaren na natjecanju i za njime zaostaje za nešto manje od 4%. Kako bi se ta razlika smanjila potrebno je unaprijediti implementaciju, a nekoliko mogućih koraka koji bi poželjno doprinijeli točnosti modela su korištenje vlastoručno izrađenih vektora koji se temelje na tekstovima mikroobjava platforme *Twitter*, dodavanje mehanizma pozornosti u duboko učenje i konačno bolja obrada ulaznih podataka u pogledu korištenja pametnije tokenizacije. U ovom je radu korišten tokenizator knjižnice *Spacy* koji nije posebno prilagođen za tip podataka kakav se nalazi u skupu podataka. Ugradnjom navedenih poboljšanja, a i opsežnijom provedbom eksperimenata, može se očekivati unaprjeđenje modela i povećanje njegove točnosti. Prostora za poboljšanje ostalo je još puno, s obzirom na to da *state-of-the-art* model postiže točnost od 68.1% koja nije impre-

sivna s obzirom na točnost koja se postiže na ostalim problemima iz područja analize sentimenta (Zimbra et al., 2018). *Twitter* je jedna od najpopularnijih društvenih platforma današnjice, stoga je značajna i vrijednost implementacija koje se bave analizom mikroobjava s *Twittera*, pa se može očekivati daljnji napredak u ovom području.

LITERATURA

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, i Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Anas Al-Masri. What are overfitting and underfitting in machine learning?, Jun 2019. URL `https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-\learning-a96b30864690`.

Christos Baziotis, Nikos Pelekis, i Christos Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. U *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, stranice 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics.

Steven Bird, Ewan Klein, i Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Margaret M Bradley i Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology . . . , 1999.

François Chollet et al. Keras. `https://keras.io`, 2015.

Mathieu Cliche. BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. U *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, stranice 573–580, Vancouver, Canada, Kolovoz 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2094. URL https://www.aclweb.org/anthology/S17-2094.

Doris Zhou. Implementations of various sentiment analysis methods in python, 2020. URL https://github.com/dwzhou/SentimentAnalysis. [Online; accessed 13-April-2020].

Liu Hu. Lexicon, 2006. URL https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon.

Bojana Dalbelo Bašić Jan Šnajder. *Strojno učenje*. 2014.

Diederik P Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

David Lozić, Doria Šarić, Ivan Tokić, Zoran Medić, i Jan Šnajder. TakeLab at SemEval-2017 task 4: Recent deaths and the power of nostalgia in sentiment analysis in twitter. U *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, stranice 784–789, Vancouver, Canada, Kolovoz 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2132. URL https://www.aclweb.org/anthology/S17-2132.

Tomas Mikolov, Kai Chen, Gregory S Corrado, i Jeffrey A Dean. Computing numeric representations of words in a high-dimensional space, Svibanj 19 2015. US Patent 9,037,464.

Andrew Ng. Neural networks and deep learning. URL https://www.coursera.org/lecture/neural-networks-deep-learning/activation-functions-4dDC1.

Razvan Pascanu, Tomas Mikolov, i Yoshua Bengio. On the difficulty of training recurrent neural networks. U *International conference on machine learning*, stranice 1310–1318, 2013.

Nils Reimers i Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

Salman Aslam. Twitter by the numbers: Stats, demographics and fun facts, 2020. URL https://www.omnicoreagency.com/twitter-statistics/. [Online; accessed 13-April-2020].

Sara Rosenthal, Noura Farra, Preslav Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. 2017.

Amy Beth Warriner, Victor Kuperman, i Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207, 2013.

David Zimbra, Ahmed Abbasi, i Daniel Zeng. The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation. *ACM Transactions on Management Information Systems*, xx, No. x, 05 2018. doi: 10.1145/3185045.

Marko Čupić. *Umjetna inteligencija*. 2016. URL java.zemris.fer.hr/nastava/ui/.

POJMOVNIK

CNN konvolucijska neuronska mreža (engl. *Convolutional neural network*). 4

LSTM ćelija s dugoročnom memorijom (engl. *Long short-term memory*). iv, 2, 4, 7, 11–13, 20–24, 26, 31

NLTK knjižnica za obradu prirodnog jezika (engl *Natural Language Toolkit*). 14

RNN povratna neuronska mreža (engl. *Reccurent neural network*). 1, 4, 11, 20, 24, 26, 31

SVM stroj potpornih vektora (engl. *Support-vector machine*). iv, 1, 5, 7–9, 15, 16, 19, 20, 24, 26, 31

Strojno učenje za analizu sentimenta u mikroblogovima

Sažetak

Ovaj se rad bavi analizom sentimenta u mikroblogovima društvene platforme *Twitter*. Proučavanje teme analize sentimenta ostvareno je uz pomoć četvrtoga zadatka s natjecanja *Semeval* koji je bio vrlo popularan zadatak nekoliko godina u nizu u kojima se natjecanje održavalo. Napravljena je usporedba pristupa klasičnim strojnim učenjem, odnosno korištenja SVM-modela i pristupa dubokog učenja, odnosno korištenja povratne neuronske mreže (RNN) s arhitekturom ćelije s dugoročnom memorijom (LSTM). S pristupom klasičnog strojnog učenja ostvarena je točnost od 62.3%, a pristupom dubokog učenja ostvarena je nešto bolja točnost od 64.24%, što implementaciju stavlja na 10. mjesto implementacija koje su bile predane u sklopu natjecanja 2017. godine.

Ključne riječi: strojno učenje, duboko učenje, obrada prirodnog jezika, analiza sentimenta, analiza mikroblogova, Semeval

Machine Learning for Sentiment Analysis in Microblogs

Abstract

The theme of this thesis is sentiment analysis on micro blogs from the social platform *Twitter*. The study of sentiment analysis was done with a help of *Semeval* competition task 4, very popular and attractive task for several years in a row. Comparison of traditional machine learning and deep learning techniques was made by implementing two models, one using SVM, and the other one using RNN with LSTM architecture. The best result with SVM model was 62.3%, and 64.24% with LSTM. Implementation would take 10th place on scoreboard of *Semeval 2017* edition of the competition.

Keywords: Machine learning, Deep learning, Natural language processing, Sentiment analysis, Microblogs analysis, Semeval.