

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 000

# **Strojno učenje za analizu sentimenta u mikroblovima**

Ivan Križanić

Zagreb, lipanj 2020.

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

*TODO*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Povezani radovi</b>	<b>3</b>
2.1. Rad tima <i>BB_twtr</i> - najuspješniji model današnjice . . . . .	3
2.2. Rad tima <i>DataStories</i> . . . . .	4
2.3. Rad tima <i>TakeLab</i> - pristup klasičnim strojnim učenjem . . . . .	5
<b>3. Implementacija</b>	<b>6</b>
3.1. O zadatku 4 natjecanja <i>Semeval 2017</i> i analizi sentimenta u mikroblo- govima . . . . .	6
3.2. Odabir metoda i pristupa . . . . .	7
3.2.1. Klasično strojno učenje - SVM-model . . . . .	7
3.2.2. Duboko učenje . . . . .	10
3.3. Obrada ulaznih podataka . . . . .	13
3.4. Značajke . . . . .	14
3.4.1. Značajke u klasičnom pristupu . . . . .	14
3.4.2. Značajke u dubokom učenju . . . . .	17
<b>4. Provedba eksperimenata</b>	<b>18</b>
4.1. Podatci . . . . .	18
4.2. Eksperimentiranje . . . . .	19
4.2.1. SVM-model . . . . .	19
4.2.2. LSTM-model . . . . .	20
<b>5. Zaključak</b>	<b>22</b>
<b>Literatura</b>	<b>23</b>

# 1. Uvod

Mikroblogovi su danas jedan od najčešće korištenih i najčešće proučavanih oblika komunikacije na internetu. Pronalaza se na iznimno popularnim društvenim mrežama, kao što su Twitter i Facebook, koji broje milijune korisnika diljem svijeta. Ljudi ih objavljuju u stvarnom vremenu, izražavajući svoje osjećaje, stavove i razmišljanja u 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 značenja.

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 podacima 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 sa Twittera i drugih društvenih platformi (*brand-mentions.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 i 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 LSTM inačici modela.

Rad je struktruno podijeljen na sljedeći način. U prvom se dijelu nalazi osvrt na radove koji su utjecali na ovaj rad, odnosno glavni izvori 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. Povezani radovi

Na temu analize sentimenta napisano je mnogo radova, a velik broj bavi se upravo mikroblovima s društvenih mreža i to vrlo često upravo 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%, što odgovara 11. mjestu na ljestvici predanih implementacija natjecanja *Semeval 2017* (Sara Rosenthal, Noura Farra, Preslav Nakov, 2017). Prvo mjesto sa 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 mikroblova. 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 aktuelni *state-of-the-art* model u području analize sentimenta mikroblova. Radi se o radu *BB\_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs* (Cliche, 2017). P problemu 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 sa natjecanja i model izgrađen od 10 CNN i 10 LSTM mreža 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 zadacima ostvarili

najbolji rezultat.

Budući da navode CNN i LSTM modele mreža kao najbolje u području analize sentimenta, u implementaciji ovog rada upotrijebljava se LSTM model mreže 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 traniranje 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 daljnoj su obradi primijenili standardne postupke pročišćavanja teksta koji se koriste u obradi prirodnog jezika.

Osvrnuli su se na CNN i naglasili problematiku gubitka informacije o poretku riječi prilikom uporabe istih. Iz tog su razloga preferirali RNN, konkretnije napredniju izvedenicu koja primjenjuje ćelije s dugoročnom memorijom odnosno 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 preminute 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 svojestvenima njohovoj društvenoj ulozi, a za pojmove kojima je često pridjeljena 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 mikroblovima

Verzija zadatka s kojim se ovaj rad bavi peta je u nizu na najtecanju *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 podzadataka 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 mikrobloge pozitivnog, neutralnog i negativnog sentimenta.

Analiza sentimenta u tekstovima kao što su mikroblovi 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 kolovijalnih 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 postizanja 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 pristuno nizanje istog slova u riječima poput "*cool*" koje možemo pronaći u obliku kao što je "*coool*" 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 ukoliko 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 ukoliko su podatci korišteni u npr. klasifikaciji označeni sa oznakama klase kojoj pripadaju, onda je riječ o nadziranom strojnom učenju, dok se o nenadziranom strojnom učenju radi ukoliko 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 vlastaručnoj 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:

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

a za nelinearnost se može upotrijebiti preslikavanje  $\phi$ . 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. Ukoliko 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:

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

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

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

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

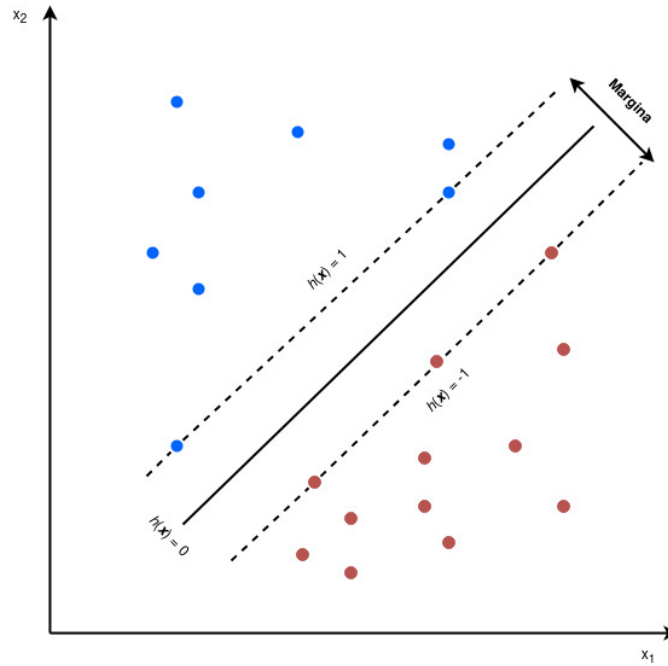
$$y^{(i)} (\mathbf{w}^T \phi(\mathbf{x}^{(i)}) + w_0) \geq 1, \quad n = 1, \dots, N$$

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:

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

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

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



**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:

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

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

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

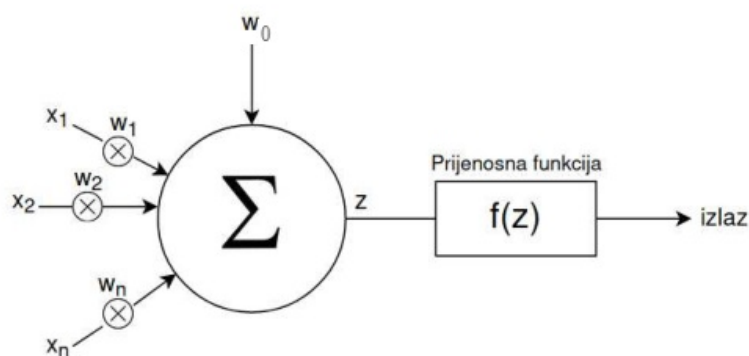
### Problem višeklasne klasifikacije

Standarnom 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

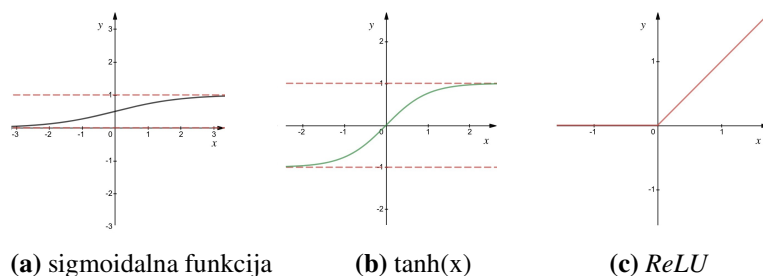
koju se rad osvrće koristi metodu jedan-naspram-ostali (engl. *one-vs-rest*) koja je zadana metoda knjižnice *scikit-learn*. Princip se temelji na svođenju problema na  $K - 1$  problema binarne klasifikacije, gdje je  $K$  broj klasa koje početni problem može klasificirati. Tada svaki od  $K - 1$  binarnih klasifikatora  $h_i$  odjeljuje klasu  $C_i$  od svih preostalih klasa. Problem pristupa javlja se ukoliko 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 na taj način izlaznu vrijednost. Ulazi u neuron množe se sa 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



**Slika 3.3:** Aktivacijske funkcije

## Prijenosne funkcije

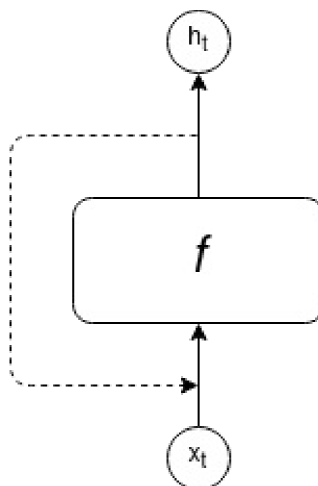
Propusne funkcije mogu biti razne, a najčešće su sljedeće:

- 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  $\max(0, x)$
- tangens 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 tanges hiperbolične funkcije. Sigmoidalna funkcija odgovorna je za odabir vrijednosti koje će sudjelovati u modifikaciji memorije, a tanges hiperbolični odgovoran je za pridjeljivanje odgovarajuće težine ulazu. Vrata za zaboravljanje odgovorna su za prebiranje vrijednosti iz prethodne iteracije i ulaznog



podatka i također funkciniraju 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 jednažbama je sljedeća:

$$f_t = \sigma_g (W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g (W_i x_t + U_i h_{t-1} + b_i)$$

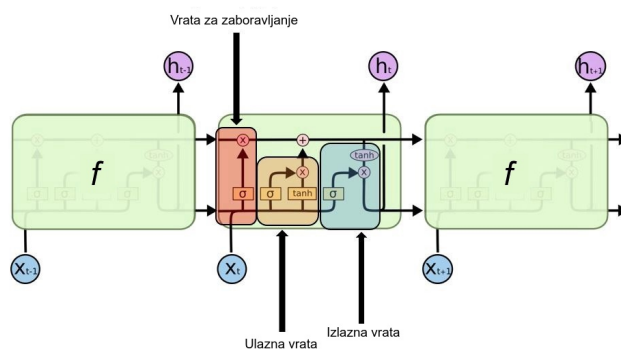
$$o_t = \sigma_g (W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = \sigma_h (W_c x_t + U_c h_{t-1} + b_c)'$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ \sigma_h (c_t)$$

gdje je  $x_t$  vektor ulaznih podataka,  $f_t$  aktivacijski vektor vrata za zaborav,  $o_t$  aktivacijski vektor izlanih 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 zamijena 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. Mikrobloγοvi 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 mikrobloγοva koji su označeni koristeći alat koji je napravio spomenuti tim *DataStories* s natjecanja *Semeval* (Baziotis et al., 2017). U mikrobloگو 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 glavnu 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 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:

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

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:

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

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:

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

U implementaciji rješenja koristim gotovu metodu iz knjižnice *scikit-learn*. Korištenjem nastalih značajki u SVM-modelu dobio sam točnost od 39.5%, 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 *Wikipedie*, 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 na način 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 na način 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). 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 naznačavana zastavicom 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. "*coool*"), 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 pridjeliti 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 riječim (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']
```

( <allcaps> off topic </allcaps> ) - there is only <number> episodes on the first disk of <hashtag> dexter </hashtag> . please hurry , <user> with the 2 nd <hashtag> fit blog </hashtag>.

## 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 dvostruko više pozitivno označenih mikroblogova nego negativnih, dok u skupu za testiranje ima 50% više mikroblogova s negativnom oznakom. Precizni podatci o broju i udjelima mikrovlogova 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 podacima

## 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 iz razloga 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.

### 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 podacima. 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 podacima. Oda-  
bir manje vrijednosti hiperparametra  $c$  utječe na povećanje širine maksimalne mar-  
gine, ali kao posljedicu ostavlja dio podataka pogrešno razdijeljen. Odabirom veće  
vrijednosti smanjuje se širina maksimalne margine hiperravnine, ali zato manje poda-  
taka biva krivo razdijeljeno tijekom treniranja. Vrijednosti hiperparametra  $c$  koje ima  
smisla probati dolaze iz iznimno širokog intervala, stoga se u ovoj implementaciji vrši  
treniranje za vrijednosti hiperparametra od  $10^{-7}$  do  $10^{15}$  povećavajući vrijednost pa-  
rametra 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 mo-  
del nasumičnog pogađanja imao bi točnost od 33.32%, a početni model ima točnost od  
39.85% što ga čini korektnim početnim modelom. S obzirom na to da je razvoj zna-  
čaka 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 4.1. 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.

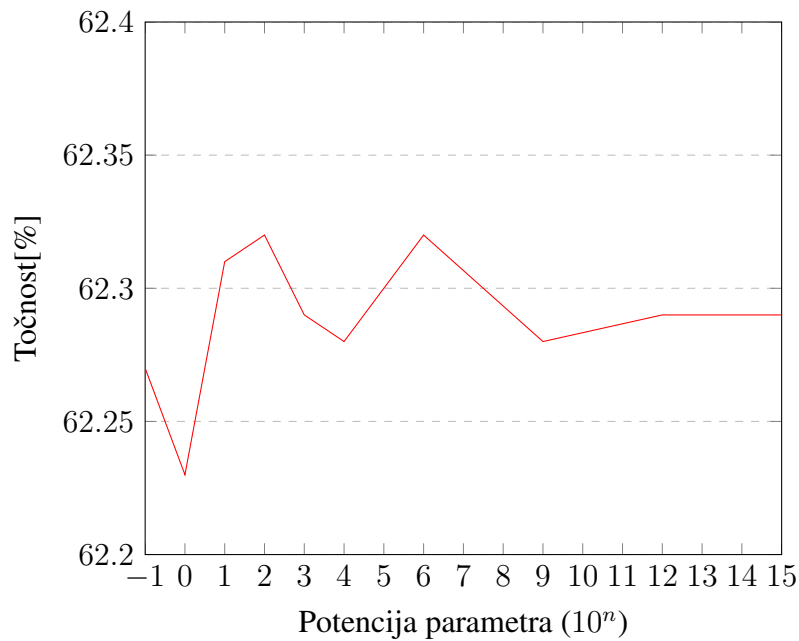
Značajke	Rezultat
<i>Tf-Idf</i> ( <i>Bag-of-Words</i> )	39.85%
<i>Spacy</i> vektori riječi	60.90%
<i>Spacy</i> vektori riječi + <i>ANEW</i> + polaritet riječi	61.91%
<i>Spacy</i> vektori riječi + <i>ANEW</i> + polaritet riječi + značajke prebrojavanja	62.32%

**Tablica 4.2:** Rezultati eksperimenata SVM-modela

#### 4.2.2. LSTM-model

Eksperimentiranje u implementaciji koja koristi metode dubokog učenja znatno je ve-  
ćeg opsega nego kod klasičnog strojnog učenja. LSTM arhitektura RNN-modela ima  
velik broj parametara koji se mogu konfigurirati, pa je stoga skup mogućih postava eks-  
perimenata prevelik da bi se u realnom vremenu isprobao. Hiperparametri koji utječu  
na točnost predviđanja koju utrenirana neuronska mreža ostvaruje su sljedeći: broj ne-  
urona u LSTM sloju, veličina mini serije (engl. *mini-batch*), broj LSTM slojeva, oda-





**Slika 4.1:** Utjecaj hiperparametra  $c$  na točnost SVM-modela

bir 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 provedeni u raznim etapama kako bi se što bolje precizirao skup eksperimenata koji donose najbolje rješenje.

### Optimizacijska funkcija

## **5. Zaključak**

# LITERATURA

Christos Baziotis, Nikos Pelekis, i Christos Doukeridis. 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.

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. 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.

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.
- Marko Čupić. *Umjetna inteligencija*. 2016. URL [java.zemris.fer.hr/nastava/ui/](http://java.zemris.fer.hr/nastava/ui/).

# POJMOVNIK

**CNN** konvolucijska neuronska mreža (engl. *Convolutional neural network*). 3

**LSTM** ćelija s dugoročnom memorijom (engl. *Long short-term memory*). 1, 3, 6

**RNN** povratna neuronska mreža (engl. *Reccurent neural network*). 3

**SVM** stroj potpornih vektora (engl. *Support-vector machine*). iv, 1, 4, 6

## Strojno učenje za analizu sentimenta u mikroblovima

### Sažetak

Ovaj se rad bavi analizom sentimenta u mikroblovima 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 osvarena je nešto bolja točnost od 64%, što implementaciju stavlja na 11. mjesto implementacija koje su bile predane u sklopu natjecanja 2017. godine.

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

The theme of this thesis is sentiment analysis on microblogs from the social platform *Twitter*.

## Machine Learning for Sentiment Analysis in Microblogs

### Abstract

Abstract.

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