TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja
Text Analysis and Knowledge Engineering Lab
Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4755

Primjena strojnog učenja za tematsku analizu sentimenta

Stipan Mikulić

Zagreb, lipanj 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 11. ožujka 2016.

ZAVRŠNI ZADATAK br. 4755

Pristupnik: **Stipan Mikulić (0036472661)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Primjena strojnog učenja za tematsku analizu sentimenta**

Opis zadatka:

Korisnički komentari na internetu vrijedan su izvor informacija za analizu stavova i mišljenja ljudi o događajima i njihovim protagonistima, političkim odlukama i političkim subjektima, ideološkim pitanjima itd. Porastom raspoloživih količina korisnički generiranog sadržaja povećalo se zanimanje za strojnom analizom sentimenta, kojom se utvrđuje je li tekst usmjeren pozitivno, negativno ili neutralno. Tehnike tematske analiza sentimenta kombiniraju otkrivanje tema i analizu sentimenta, kako bi se odredio sentiment usmjeren prema određenoj temi.
U okviru završnoga rada potrebno je proučiti postupke za analizu sentimenta i postupke za otkrivanje tema, s naglaskom na postupke temeljene na strojnom učenju. Razraditi model za tematsku analizu sentimenta u korisničkim komentarima na hrvatskome jeziku temeljen na strojnom učenju. Model primijeniti na podatcima prikupljenima sa stranica političkih stranaka i vlade na društvenoj mreži Facebook. Izgraditi i ručno označiti odgovarajući skup tekstnih podataka na hrvatskome jeziku za razvoj i ispitivanje modela. Provesti eksperimentalno vrednovanje modela, uključivo usporedbu s referentnim modelom i statističku obradu rezultata. Radu priložiti izvorni i izvršni kod razvijenog sustava, označene skupove podataka i potrebnu dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 18. ožujka 2016.
Rok za predaju rada: 17. lipnja 2016.

Mentor:

Doc. dr. sc. Jan Šnajder

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof. dr. sc. Siniša Srbljić

Zahvaljujem mojoj obitelji na neprestanoj podršci tokom studija.
Zahvaljujem mentoru i cijeloj TakeLab ekipi na pomoći i savjetima.
Zahvaljujem Damiru Korenčiću na pristupu njegovih resursima.

SADRŽAJ

1. Uvod

U današnje vrijeme bilježi se nagli porast korisnički generiranog sadržaja. Velike količine nestrukturiranih tekstnih podataka zahtijevaju neki način obrade i analize. Zbog ljudske ograničenosti za obradu tako velikih količina podataka javlja se potreba za rješavanjem tog problema pomoću računala. Ovaj problem pripada području računarske znanosti (engl. *computer science*), umjetne inteligencije (engl. *artificial intelligence*) i strojnog učenja (engl. *machine learning*) koje se naziva obrada prirodnog jezika (engl. *natural language processing, NLP*). Obrada prirodnog jezika jedan je od najvažnijih modernih tehnoloških alata za rješavanje problema kao što su strojno prevođenje (engl. *machine translation*), detekcija tema (engl. *topic detection*), detekcija neželjenih poruka (engl. *spam detection*), analiza sentimenta (engl. *sentiment analysis*), crpljenje i pretraživanje informacija (engl. *information retrieval, information extraction*) itd.

U ovom radu koncentrirat ćemo se na razvoj računalnog modela, temeljenog na strojnom učenju, za analizu aktualnih političkih tema na društvenim mrežama. Zapravo ćemo razviti sustav s dva modela strojnog učenja. Prvi će biti model grupiranja koji ćemo koristiti za detekciju teme, dok će drugi biti model analize sentimenta. S ova dva modela sustav će moći prepoznati koje su teme aktualne na političkim stranicama Facebooka u Hrvatskoj te što ljudi misle o tim temama tj. dobit ćemo širu sliku javnog mnijenja.

U okviru završnog rada razvijen je model nenadziranog učenja[1] (engl. *unsupervised learning*) za detekciju tema. Model je primijenjen na korpus hrvatskih tekstova koji se sastojao od novinskih članaka i poruka s Facebooka. Provedena je i evaluacija modela. Za potrebe analize sentimenta razvijen je model nadziranog učenja[2] (engl. *su-*

[1] Pristup strojnom učenju pri kojem dani podatci nemaju ciljnu vrijednost i potrebno je naći pravilnost u podacima. Dijeli se na grupiranje (engl. *clustering*), otkrivanje novih vrijednosti ili vrijednosti koje odskaču (engl. *novelty/outlier detection*), smanjenje dimenzionalnosti (engl. *dimensionality reduction*) (Šnajder i Dalbelo Bašić, 2012).

[2] Pristup strojnom učenju pri kojem je unaprijed poznata oznaka klase y kojoj pripada primjer x iz skupa za učenje. S obzirom na varijablu y, razlikujemo klasifikaciju – ako je y diskretna ili nebrojčana

pervised learning). Pomoću razvijenog modela želimo utvrditi da li je pojedini komentar pozitivan, negativan ili neutralan. Pošto se koristi nadzirani model strojnog učenja, bilo je potrebno ručno označiti skup tekstnih podataka na hrvatskom jeziku. Provedeno je eksperimentalno vrednovanje modela.

Rad je strukturiran tako da je u drugom poglavlju opisana detekcije tema a u trećem poglavlju analiza sentimenta. U oba poglavlja su opisani sami problemi, slični radovi, model i programska implementacija modela te metode evaluacije. Četvrto poglavlje opisuje dodatne alate za dohvaćanje i spremanje podataka. U petom poglavlju se nalazi zaključak rada.

vrijednost, regresiju – ako je y kontinuirana ili brojčana vrijednost (Šnajder i Dalbelo Bašić, 2012).

2. Detekcija teme

2.1. Opis problema i slični radovi

Uslijed velikog povećanja korisnički generiranog sadržaja na društvenim mrežama detekcija teme se izdvaja kao nužan alat za analizu i otkrivanje aktualnih tema u društvu.

Od sličnih radova želimo izdvojiti (Ahmed Rafea, 2013) i (Mario Cataldi i Schifanella, 2010). U radu (Ahmed Rafea, 2013) je korišteno hijerarhijsko grupiranje za detekciju tema dok se u radu (Mario Cataldi i Schifanella, 2010) detekcija teme temelji na generiranju skupa ključnih riječi uz pomoć teorije grafova. Iako smo iz ovih radova dobili smjernice što se tiče pristupa problemu detekcije teme, odlučili smo koristiti LDA algoritam koji se ne spominje ni u jednom od navedenih radova.

U ovom radu detekcija tema podijeljena je u tri koraka:

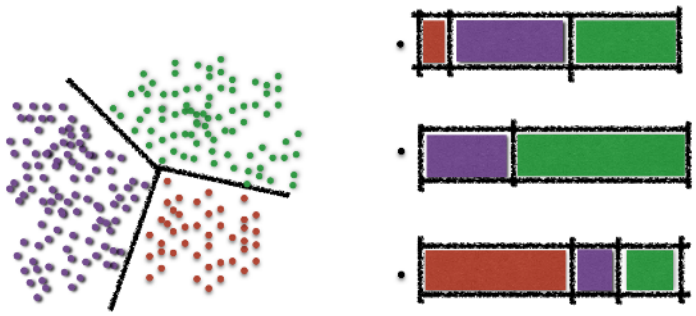
1. Preprocesiranje postova (engl. *posts preprocessing*)
2. Grupiranje (engl. *clustering*)
3. Evaluacija modela i označavanje tema (engl. *model evaluation and clusters labeling*)

Podatci korišteni za detekciju teme su postovi sa stranica Facebooka te novinski članci s hrvatskih portala. Članci su korišteni kao potpora malom broju postova.

Tablica 2.1: Skup podataka za detekciju teme

Postovi	**Članci**
2692	211479

Prikupljene nestrukturirane podatke je prvo trebalo preprocesirati koristeći alat za procesiranje hrvatskog jezika. Iz dobivenih procesiranih podataka složena je vreća



(a) a **(b)** b

Slika 2.1: Čvrsto grupiranje (a) i Meko grupiranje (b)[2]

riječi (engl. *bag-of-words*)[1] struktura za svaki post i članak. Tako strukturirani podatci su sad pogodni za analizu pa se šalju na ulaz algoritma iz kojeg ćemo dobiti grupirane postove i članke. Treći korak je evaluirati dobiveni izlaz algoritma te prepoznati i označiti teme koje odgovaraju različitim grupama.

2.2. Model detekcije teme

Detekcija tema pripada nenadziranom strojnom učenju, točnije grupiranju. Cilj algoritama grupiranja je taj da se skup objekata resporedi po grupama. Grupiranje se obavlja na način da se u jednu grupu rasporede objekti koji su međusobno slični dok su manje slični objektima iz drugih grupa. Konkretno u ovom radu grupiraju se postovi s hrvatskih političkih Facebook stranica.
Grupiranje možemo podijeliti prema "čvrstoći" granica između grupa. Takvu podjelu čine dvije skupine (Šnajder i Dalbelo Bašić, 2012):

1. Čvrsto grupiranje (engl. *hard clustering*): svaki primjer može pripadati isključivo jednoj grupi.

2. Meko grupiranje (engl. *soft clustering*): jedan primjer može pripadati u više grupa, i to eventualno s različitim stupnjem ili različitom vjerojatnošću pripadanja.

[1]Pojednostavljeni model strukturiranja objekata u procesiranju prirodnog jezika. U ovom radu se koristi da bi svaki post ili članak mogli prikazati vektorom jednake veličine tako što će svaki indeks vektora odgovarati broju pojavljivanja pojedine riječi iz posta.

[2]http://chdoig.github.io/pytexas2015-topic-modeling/#/2/5

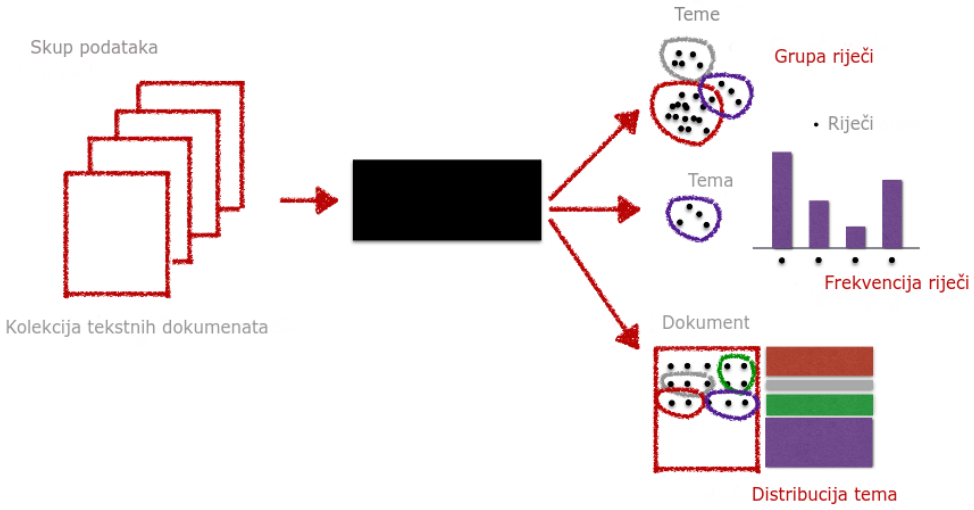
Prije nego krenemo u detaljni opis modela dobro je navesti neke od definicija modela detekcije tema:

- Model detekcije tema je statistički model za otkrivanje apstraktnih tema koje se javljaju u skupini dokumenata.[3]
- Model detekcije tema je paket algoritama koji otkrivaju skrivene tematske strukture u kolekciji dokuemnata. Ti algoritmi nam pomažu otkriti nove načine pretraživanja i sažimanja velikih arhiva teksta[4]
- Modeli detekcije tema omogućavaju nam jednostavnu analizu velikih količina nestrukturiranog teksta. Teme se sastoje od grupe riječi koje se često pojavljuju zajedno.[5]

U ovom radu korišten je algoritm Latentna Dirichletov dodjela (engl. *Latent Dirichlet Allocation – LDA*) algoritam grupiranja iz knjižnice gensim.[6] LDA spada u algoritme mekog grupiranja. Prvo ćemo algoritam promatrati kao crnu kutiju (engl. *black box*). Ulaz algoritma će biti kolekcija tekstnih dokumenata, a kao izlaz ćemo dobiti dvije stvari:

- Grupe riječi s frekvencijama gdje će svaka grupa definirati jednu temu (važno je primjetiti da jedna riječ može biti u više grupa);
- Distribuciju pripadnosti temama za svaki dokument, što nam govori koje su teme zastupljene u pojedinom dokumentu.

[3] https://en.wikipedia.org/wiki/Topic_model
[4] http://www.cs.princeton.edu/ blei/topicmodeling.html
[5] http://mallet.cs.umass.edu/topics.php
[6] https://radimrehurek.com/gensim
[7] http://chdoig.github.io/pytexas2015-topic-modeling/#/2/6



Slika 2.2: LDA kao crna kutija[7]

Opis u nastavku napravljen je prema (Doig, 2015a).

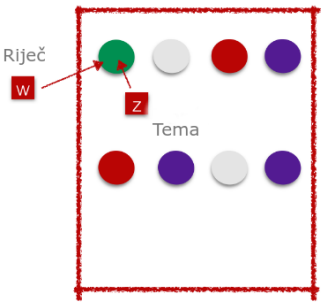
1. **Inicijalizacija parametara**

   Neki od važnijih parametara modela su:

   - num_topics – broj tema;
   - alpha and eta – parametri koji utječu na distribuciju dokument-tema i tema-riječ. Oba su inicijalno postavljena na $1.0/num\_topics$;
   - id2word – struktura koja preslikava id riječi u same riječi;
   - corpus – kolekcija tekstnih dokumenata;
   - passes – broj prolazaka kroz korpus.

2. **Inicijalizacija tema slučajnim odabirom**

[8]http://chdoig.github.io/pytexas2015-topic-modeling/#/3/7



Slika 2.3: Odabir slučajnih tema za riječi[8]

3. **Iteracija**

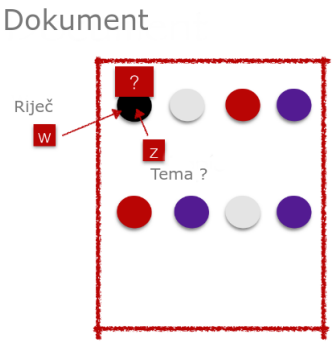
   Za svaku riječ u svakom dokumentu:

   - Odrediti nove teme za riječ u odnosu na sve ostale riječi i njima pridružene teme.

   - Preslagivanje tema se određuje na osnovu iduća dva pitanja:

     (a) Koje se teme pojavljuju u dokumentu?

     (b) Koje teme odgovaraju danoj riječi?



Slika 2.4: Preslagivanje slučajnih tema za riječi[9]

4. **Vrati rezultat**

   Rezultati algoritma su gore opisane strukture.

[9]http://chdoig.github.io/pytexas2015-topic-modeling/#/3/9

2.3. Implementacija modela

Implementacija sustava detekcije teme napisana je u programskom jeziku Python, verzija 2.7.9. Model opisan u prethodnom poglavlju implementiran je u knjižnici `gensim`[10] koji smo i koristili. Jezik je odabran radi visoke razine apstrakcije, jednostavnosti i lakoće pisanja programskog koda te zbog velike dostupnosti vanjskih knjižnica potrebnih za procesiranje teksta.

Cijeli cjevovod (engl. *pipeline*) implementacije modela podijeljen je na tri dijela među kojima vlada lančana ovisnost. Prenošenje potrebnih podataka između dijelova implementacije osigurano je serijalizacijom pomoću knjižnice `json`.[11]
Cjevovod čine sljedeće izvršne datoteke: `crawler.py`, `post_preprocessing.py` i `topic_detection.py`

Prvi dio je prikupljanje podataka za analizu koji je ostvaren u `crawler.py` datoteci. Završetkom izvođenja prvog dijela Facebook postovi su spremljeni u bazu podataka. Također, u bazu su spremljeni i novinski članci.

Drugi dio je procesiranje prikupljenih podataka koje se izvršava pokretanjem datoteke `post_preprocessing.py`. Procesiranje postova se obavlja pomoću alata za procesiranje hrvatskog jezika kojeg je ustupio TakeLab[12]. Rezultat procesiranja postova je serijalizirana `json` datoteka koju čine procesirani postovi. Svaki se post iz običnog teksta pretvori u riječnik koji je zapravo brojač riječi u postu, ali bez zaustavnih riječi.[13] Oblik riječnika je `lemma`[14] `-> broj pojavljivanja u postu`.

Konačno, treći dio čini sam algoritam detekcije tema. U ovom dijelu se iz procesiranih postova grade strukture pogodne za ulaz opisanog modela detekcije tema. Nakon što model završi grupiranje tema generiraju se datoteka u koju se spremaju dobivene grupe te datoteka za vizualizaciju pomoću knjižnice `pyLDAvis`[15]

[10] https://radimrehurek.com/gensim
[11] https://docs.python.org/2/library/json.html
[12] http://takelab.fer.hr/
[13] Najčešće korištene riječi u jeziku.
[14] Osnovi dio riječi, korijen riječi.
[15] https://github.com/bmabey/pyLDAvis

```
from collections import defaultdict
frequency = defaultdict(int)
for post in posts:
    for k, v in post.items():
        frequency[k] += v
texts = [[word for word, freq in post.items() if re.search('[a-zA-Z]
    ', word) and len(word) > 1] for post in posts]

dictionary = gensim.corpora.Dictionary(texts)
mm = [dictionary.doc2bow(text) for text in texts]

lda = gensim.models.LdaModel(corpus=mm, id2word=dictionary,
    num_topics=number_topics, update_every=0, passes=20)
```

Isječak 2.1: Pozivanje LDA algoritma

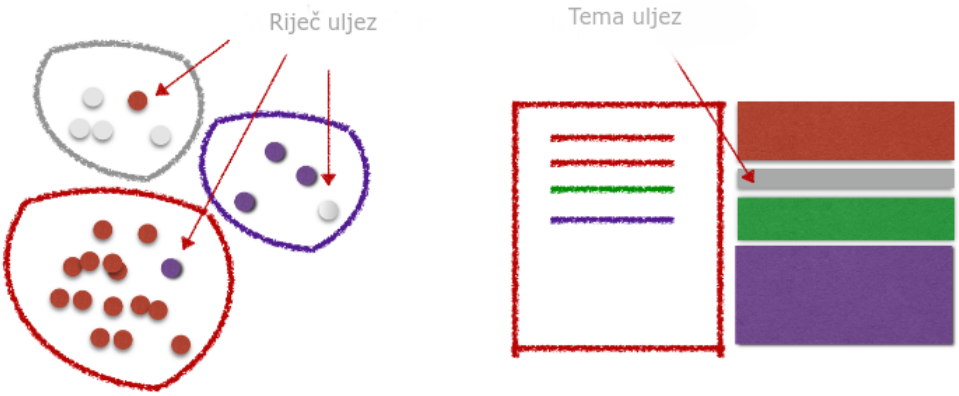
2.4. Evaluacija

Problem s evaluacijom modela nenadziranog učenja je taj što nemamo označene podatke tj. ne znamo kakvu bi distribuciju tema trebali dobiti. Ipak postoje načini evaluiranja modela detekcije tema. Neki od tih su:

- **Čovjek u petlji (engl. *Human-in-the-loop*)**[16](Chang et al., 2009)

    1. Riječ uljez (engl. *word intrusion*)
       Metoda u kojoj slučajnim odabirom mijenjamo riječi među temama. Nakon toga provjeravamo može li čovjek prepoznati koja je riječ uljez među riječima koje pripadaju određenoj temi. Ako osoba prepozna tu riječ znači da su teme dobro podjeljenje, inače teme nemaju dobru podjelu.
    2. Tema uljez (engl. *topic intrusion*)
       Metoda u kojoj se testerima pokaže dio dokumenta. Također su im prikazane četiri teme od kojih tri imaju najveću pripadnost tom dokumentu. Tema uljez je slučajno odabrana među ostalim temama. Ako tester prepozna uljez temu znači da su za dani dokument teme dobro određene.

[16]Definira se kao model za koji je potrebna interakcija s ljudima.



Slika 2.5: Human in the loop[17]

- **Kosinusna sličnost (engl. *cosine similarity*)**(Rehurek, 2011-2016)
  Metoda u kojoj se svaki dokument podijeli na dva dijela te se vrši usporedba prvih i drugih dijelova između istih dokumenata te između različitih. Pri usporedbi istih bolje je imati veći sličnost, dok usporedbama različitih manju.

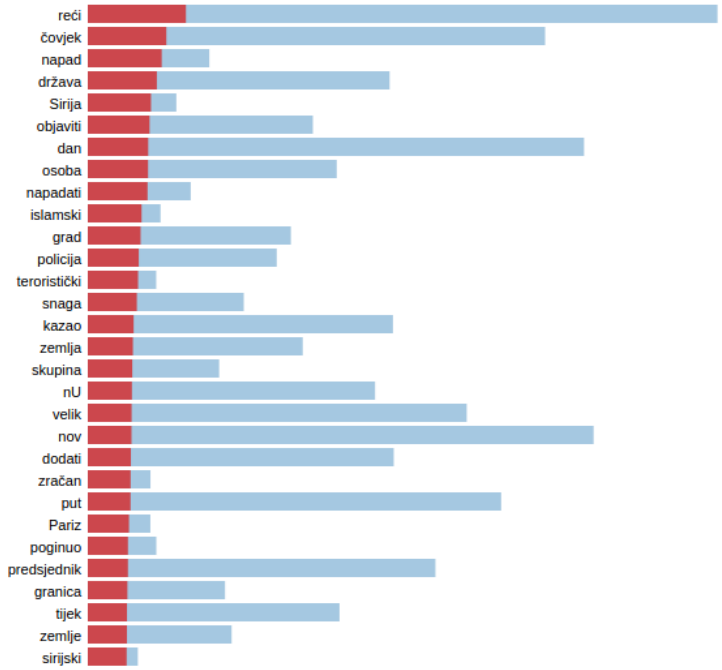
U ovom radu se koristila human-in-the-loop metoda, točnije riječ uljez metoda (engl. *word intrusion*). Imenovanje tema je obavljeno ručno.

[17] http://chdoig.github.io/pytexas2015-topic-modeling/#/3/13

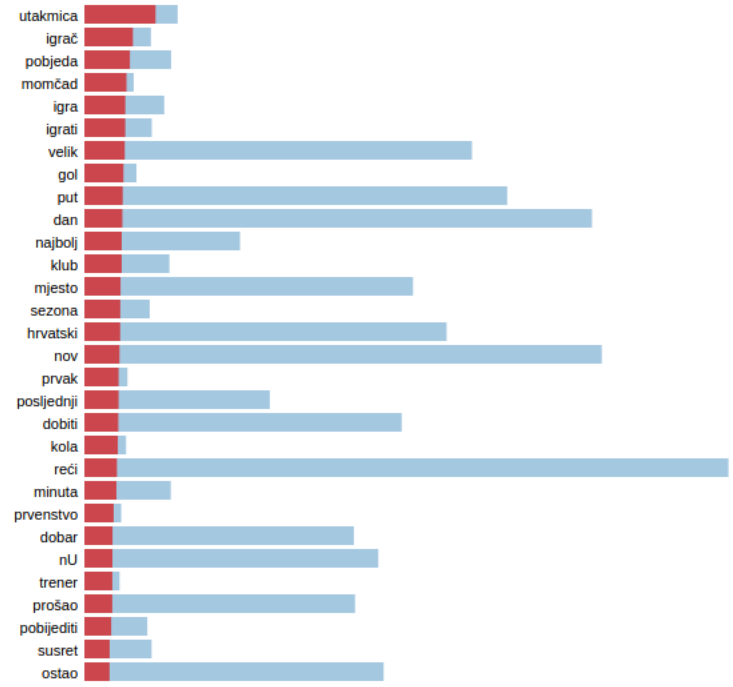
Tablica 2.2: Označene teme

Broj teme	Ime teme
1	Gospodarstvo
2	Hrvatska politika
3	Hrvatska politika
4	Obitelj
5	Terorizam
6	Sport
7	Općenito
8	Općenito
9	Kriminal
10	Svjetska politika
11	Općenito
12	Općenito
13	Hrvatska politika
14	Općenito
15	Općenito

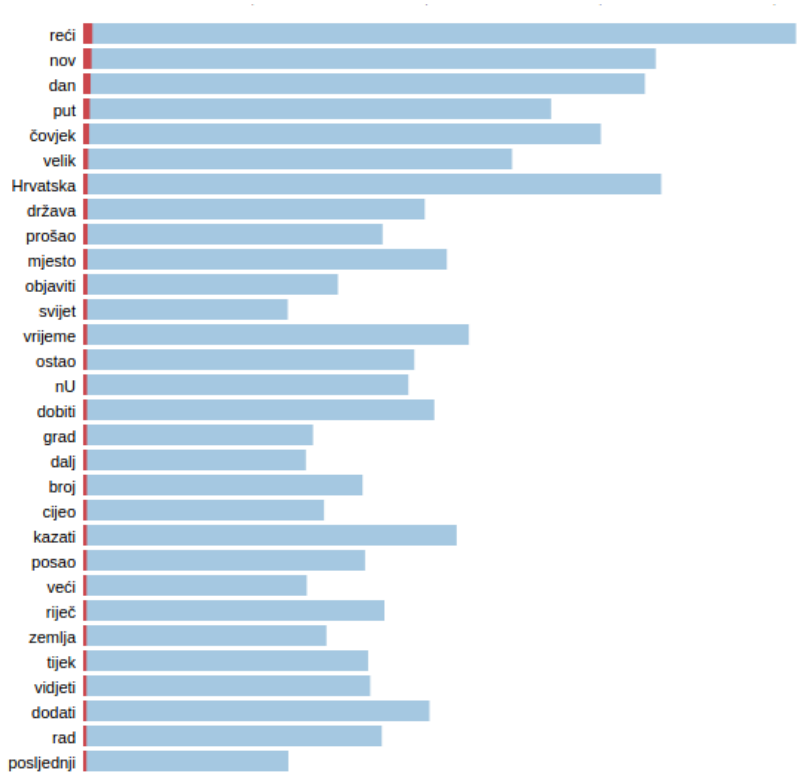
Zamjenom riječi iz tema Terorizam (slika 2.6) i Sport (slika 2.7) tester je uspio pronaći uljeza pa smo zaključili da su dane teme dobro podijeljene. Ipak, dosta tema je su označene kao iste zbog loše podjele i preopćenitih riječi u njima. Kada uzmemo u obzir teme 14. (slika 2.8) i 15. (slika 2.9) koje su označene kao Općenito tester nije mogao zaključiti koja je riječ uljez što znači da nisu dobro podjeljene.



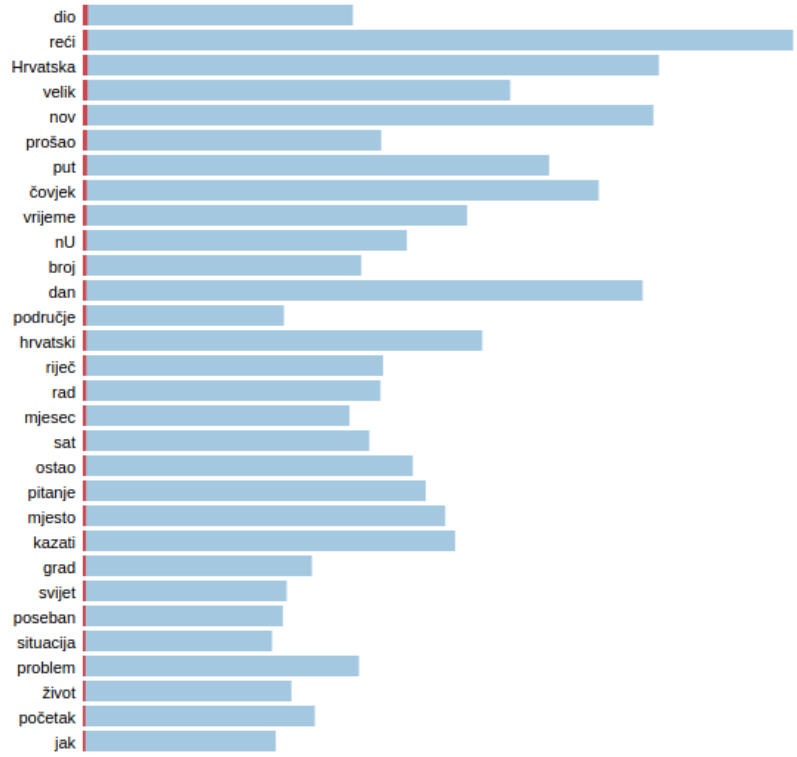
Slika 2.6: Distribucija riječi za temu Terorizam



Slika 2.7: Distribucija riječi za temu Sport



Slika 2.8: Distribucija riječi za temu 14 označenu kao Općenito



Slika 2.9: Distribucija riječi za temu 15 označenu kao Općenito

2.5. Poboljšanja

Modela detekcije teme sasvim sigurno ima mjesta za poboljšanje. Trenutno najbolje grupiranje daje previše generalizirane grupe. Poboljšanja su moguća u filtriranju riječi iz dokumenata. Moglo bi se pokrenuti model bez glagola ili samo filtrirati određenje glagole koji ne mogu dati nikakvu semantiku temama. Također, vrijedni pokušaji poboljšanja bi bili pokrenuti LDA sa raznovrsnijim izborom parametara *alpha* i *eta* te povećati broj iteracija.

3. Analiza sentimenta

3.1. Opis problema i slični radovi

U posljednjem desetljeću analiza sentimenta postala je jedan od aktualnijih problema u procesiranju prirodnog jezika. Ima vrlo veliku primjenu u praktičnim problemima kao što su recenzije proizvoda, analiza javnog mnijenja itd. Također ima veliku primjenu u marketingu i korisničkim službama. Osnovni cilj jednostavne analize sentimenta je klasificirati tekst u jednu od tri klase: pozitivno, negativno ili neutralno. Drugi oblik analize sentimenta je binarna klasifikacija teksta u klase objektivan ili subjektivan. Složeniji problem analize sentimenta jest klasifikacija u više razina (engl. *multi-way scale*). Klasifikacijom u više razina, osim što želimo klasificirati sentiment u jednu od zadanih klasa, potrebno je odrediti i razinu za svaku klasu (npr. ljestvica od 1 do 5 pri recenzijama za svaku klasu), koju možemo tumačiti kao intenzitet sentimenta. Kada govorimo o dosegu analize sentimenta, on se može provoditi nad riječima ili izrazima (engl. *word- or phrase-level*), rečenicama (engl. *sentence-level*) ili nad čitavim dokumentima (engl. *text- or document-level sentiment analysis*).

Analiza sentimenta je popularan problem, stoga imamo jako puno radova koji obrađuju ovu temu. Od sličnih radova smo odlučili izdvojiti (Habernal et al., 2013) i (Pang et al., 2002). U radu (Habernal et al., 2013) razvijeni su dva klasifikatora: Maximalna entropija i SVM. Kao značajke su korišteni N-grami, POS, emotikoni i TF-IDF. Najbolje rezultate je davao model SVM. U radu (Pang et al., 2002) uz navedena dva klasifikatora iz prethodnog rada, razvijen je i Naivni Bayesov klasifikator. I u ovom radu je metoda potpornih vektora dala najbolje rezultate. Poučeni primjerima iz ovih radova odlučili smo koristiti metodu potpornih vektora za analizu sentimenta.

U ovom se radu koristila klasifikacija sentimenta u jednu od klasa: pozitivno, negativno ili neutralno. Doseg analize je cijeli tekstni dokument. Analiza sentimenta u ovom radu dijeli se na tri dijela:

1. Pretprocesiranje komentara (engl. *comments preprocessing*)
2. Klasifikacija zasnova na značajkama (engl. *feature-based classification*)
3. Evaluacija modela (engl. *model evaluation*)

Prije analiziranja sentimenta potrebno je prikupiti komentare sa stanica Facebooka.

Tablica 3.1: Skup podataka za analizu sentimenta

Komentari
193017

3.2. Model analize sentimenta

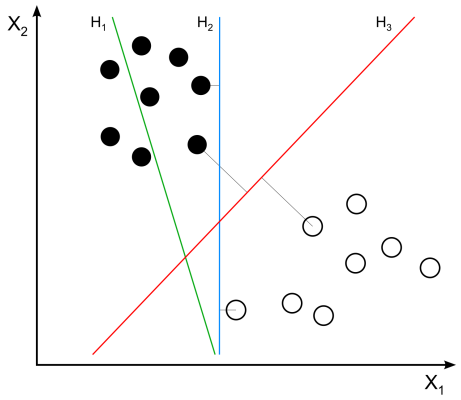
Problem analize sentimenta se rješava modelima nadziranog strojnog učenja. Neke od metoda rješavanja problema analize sentimenta u komentarima su:

1. Naivan Bayesov klasifikator (engl. *Naive Bayes classifier*),
2. Stroj potpornih vektora (engl. *Support Vector Machines, SVM*),
3. Maksimalna entropija (engl. *Maximum Entropy , MaxEnt*).

Poučeni iskustvom iz radova u literaturi problem klasifikacije komentara odlučili smo riješiti metodom potpornih vektora.

Pregled SVM modela napravljen je prema (Meyer, 2015) i `sklearn-svm`.[1] Osnovni model metode potpornih vektora rješava problem binarne klasifikacije. Ideja je ulazne podatke prikazati u prostoru te naći optimalnu hiperravninu koja razdvaja klase. Optimalna hiperravnina mora imati najveću marginu razdvajanja klasa, gdje je margina udaljenost između točaka koje pripadaju suprotnim klasama. Potporni vektori su oni koji se nalaze na rubovima i najbliži su podacima. Na slici 3.1 prikazano je kako SVM odabire najbolju hiperravninu. Ravnina H3 je odabrana jer razdvaja klase s najvećom marginom.

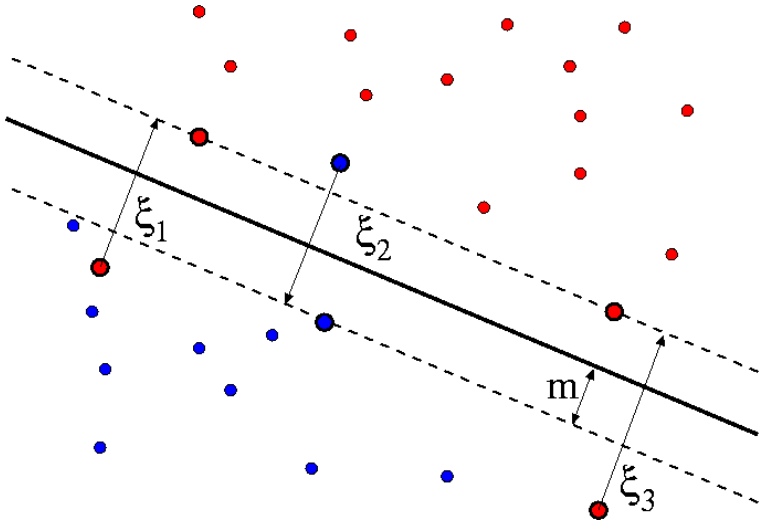
[1] http://scikit-learn.org/stable/modules/svm.html
[2] https://en.wikipedia.org/wiki/Support_vector_machine



Slika 3.1: Prikaz kako SVM odabire najbolju hiperravninu za klasifikaciju.[2]

3.2.1. Metoda meke margine

Metoda meke margine koristi se kada ravninom nije moguće razdvojiti vektore različitih klasa. Taj problem se rješava na način da omogućimo određenu pogrešku prilikom presjecanja prostora ravninom. Stoga uvodimo varijablu ξ_i (engl. *slack variable*) koja predstavlja pogrešku klasificiranja nekog vektora $\vec{x}_i$ prema ravnini podjele. Pogreška raste s udaljenošću vektora od ravnine. Vrijednost ξ_i točno klasificiranih primjera je 0. Uvodimo i parametar C koji kontrolira kaznu za krivu klasifikaciju. Pošto nam je cilj minimizirati broj krivo klasificiranih primjera, izračunatoj margini dodajemo $C \sum_{i=1}^{N} \xi_i$ gdje je N broj primjera za treniranje. Ovaj se postupak naziva metoda meke margine (engl. *soft margin method*).

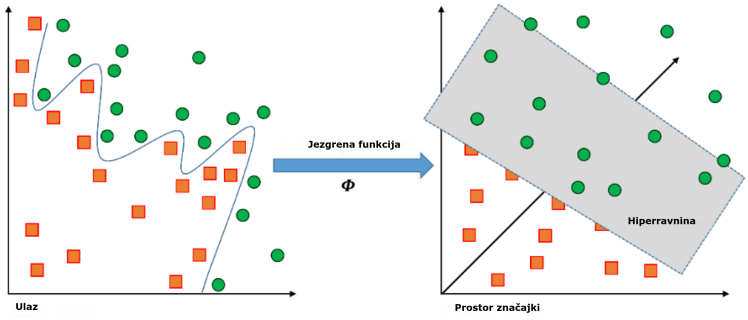


Slika 3.2: Vizualizacija metode meke margine.[3]

[3]http://emilea-stat.stochastik.rwth-aachen.de/cgi-bin/WebObjects/EMILeAstat.woa/wa/module?module=384C2AD2-4504-714A-C17D-90C3A4EFEB82

3.2.2. Nelinearno klasificiranje

U situacijama kada klase ne možemo podijeliti linearno, prelazimo u prostor viših dimenzija u kojem je onda moguće linearno razdijeliti klase. Transformacijska funkcija koja preslikava linearno nerazdvojivog prostora u višedimenionalni linearno razdvojiv prostor naziva se jezgrena funkcija (engl. *kernel function*).



Slika 3.3: Transformacija u višedimenzionalni linearno rezdvojiv prostor.[4]

Zbog posebnih parametara jezgrenih funkcija potrebno je napraviti njihovu optimizaciju. Najčešće rješenje je isprobavanje tipičnih vrijednosti za svaki parametar. Karakteristične vrijednosti za $C \in (2^{-5}, 2^{-3}, ..., 2^{15})$ i $\gamma \in (2^{-15}, 2^{-13}, ..., 2^{3})$ (Chih-Wei Hsu i Lin, 2003).

Tablica 3.2: Najčešće jezgrene funkcije

Ime funkcije	$f(x_1, x_2)$
Linearna	$(x_1^T x_2)$
Polinomijalna	$(\gamma(x_1^T x_2) + r)^d$
RBF	$exp(-\gamma\|x_1 - x_2\|^2)$
Sigmoidalna	$tanh(\gamma(x_1^T x_2) + r)$

3.2.3. Višeklasno klasificiranje

Kada imamo više od dvije klase u koje želimo klasificirati primjere koristimo višeklasni model (engl. *multiclass model*). Problem višeklasnih modela SVM tipično rješava tako da ga reducira na više binarnih klasifikatora.

[4] http://www.mdpi.com/1424-8220/16/5/631/htm

Postoje dva načina implementacije višeklasnih modela:

1. Jedan protiv svih (engl. *one-versus-all*)
   Koristeći ovu metodu gradimo binarnih klasifikatora koliko imamo klasa. Klasifikator s najvećom točnošću provodi klasifikaciju.

2. Jedan protiv jednog (engl. *one-versus-one*)
   Koristeći ovu metodu gradimo $n * (n-1)/2$ binarnih klasifikatora jer gradimo klasifikator za svaki par klasa. Klasifikacija se provodi tako što se vektoru pridodaje klasa koja mu je najviše puta dodijeljena.

3.3. Implementacija modela

Implementacija sustava za analizu sentimenta napisana je u programskom jeziku Python, verzija 3.4.3. SVM model opisan u prethodnom poglavlju implementiran je u knjižnici `scikit-learn`.[5]

Cijeli cjevovod (engl. *pipeline*) implementacije modela podijeljen je na tri dijela među kojima vlada lančana ovisnost. Prenošenje potrebnih podataka između dijelova implementacije osigurano je serijalizacijom pomoću knjižnica `json`[6] i `pickle`.[7]

Cjevovod implementacije je podijeljen na tri dijela: prikupljanje podataka, pretprocesiranje i analiza sentimenta. Prije pokretanja analize sentimenta bilo je potrebno označiti skupove za treniranje i testiranje što je urađeno preko sučelja alata za označavanje.

Datoteka `crawler.py` služi za prikupljanje komentara s stranica Facebooka koji se spremaju u bazu podataka.

Drugi dio cjevovoda je pretprocesiranje označenih komentara koje se izvršava pokretanjem skripte `comments_preprocessing.py`. Označeni komentari podjeljeni su na skupove za traniranje i testiranje u omjeru 70:30.

Tablica 3.3: Statistka označenog skupa podataka

Negativno	Pozitivno	Neutralno	Ukupno
522	98	480	1100

[5] http://scikit-learn.org/stable/
[6] https://docs.python.org/2/library/json.html
[7] https://docs.python.org/2/library/pickle.html

Treći korak je sama analiza sentimenta koju pokrećemo skriptom `svm_model.py`. Skripta na početku čita serijalizirane pretprocesirane označene komentare te iz njih gradi strukturu značajki za model koja je deteljnije opisana u sekciji ispod. Kada su strukture značajki izgrađene slijedi treniranje pa testiranje modela. U ovom radu koristili smo SVM model s višeklasnom jedan protiv jednog (engl. *one-vs-one*) klasifikacijom. Za odabir optimalnog modela klasifikatora koristili smo k-struku unakrsnu validaciju (engl. *k-fold cross validation*), točnije peterostruku unakrsnu validaciju. Općenito, k-struka unakrsna validacija dijeli skup za treniranje na k dijelova te su u k ponavljanja jedan dio koristi za validaciju a ostalih $k-1$ za treniranje.

Optimalni parametri SVM modela dobiveni k-strukom unakrsom validacijom su: $C = 69.123, kernel = RBF, \gamma = 0.00069$.

3.3.1. Značajke modela

Značajke modela s najboljom točnošću sastojale su se od vokabulara dobivenog iz označenih komentara, leksikona sentimenta i dodatnih značajki.

Tablica 3.4: Vektor značajki

Indeks	Opis značajke	Tip vrijednosti
1..7676	frekvenija riječi iz riječnika	0..N
7677	suma težina negativnog leksikona	0..N
7678	suma težina pozitivnog leksikona	0..N
7679	broj uskličnika	0..N
7680	broj velikih slova	0..N
7681	broj uzastopnih ponavljanja slova	0..N

3.4. Evaluacija

3.4.1. Priprema skupa za treniranje

Pošto je cijeli skup za treniranje označila jedna osoba, nije bilo potrebe za analizom suglasnosti označivača.

3.4.2. Evaluacijske mjere

Evaluacijske mjere su standardni način vrednovanja klasifikatora. Osnovne mjere su točnost (engl. *accuracy*), preciznost (engl. *precision*), odziv (engl. *recall*) i F-mjera (engl. *F-score*). Evaluacijske mjere se računaju iz matrice zabune (engl. *confusion matrix*) gdje svaki stupac predstavlja predviđenu klasu a svaki red stvarnu klasu. Kod višeklasne klasifikacije za svaku klasu definiramo:

- TP_i – ispravno pozitivni (engl. *true positive*), i-ti element dijagonale.
- FP_i – lažno pozitivni (engl. *false positive*), zbroj nedijagonalnih elemenata i-tog retka.
- FN_i – lažno negativni (engl. *false negative*), zbroj nedijagonalnih elemenata i-tog stupca.
- TN_i – ispravno negativni (engl. *true negative*), zbroj po elementima izvan retka i i stupca i.

Točnost (engl. *accuracy*) se definira kao udio točno klasificiranih primjera u skupu svih primjera.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

Preciznost (engl. *precision*) se definira kao udio točno klasificiranih primjera u skupu pozitivno klasificiranih primjera.

$$P = \frac{TP}{TP + FP} \tag{3.2}$$

Odziv (engl. *recall*) se definira kao udio točno klasificiranih primjera u skupu svih pozitivnih primjera.

$$R = \frac{TP}{TP + FN} \tag{3.3}$$

Tradicionalna F-mjera se naziva F1-mjera koja se računa kao harmonijska sredina preciznosti i odziva.

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \tag{3.4}$$

Općenito F-mjera se može iskazati kao težinski prosjek preciznosti i odziva gdje parametar β daje veću važnost preciznosti ili odzivu.

$$F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P + R} \tag{3.5}$$

Kod višeklasne klasifikacije računamo evaluacijske mjere za svaku klasu zasebno te uzimamo prosjek svih klasa kao konačnu mjeru sustava.

Tablica 3.5: Testni skup podataka

Negativno	Pozitivno	Neutralno
158	26	143

Matrica zabune za testni skup podataka našeg troklasnog klasifikatora je

$$\begin{bmatrix} 99 & 0 & 59 \\ 5 & 0 & 21 \\ 62 & 0 & 81 \end{bmatrix}$$

Stupci i retci redom označavaju klase negativno, pozitivno i neutralno. Tako prvi stupac označava da je naš klasifikator predvidio 99 negativnih, 5 pozitivnih i 62 neutralna primjera negativno.

U tablici 3.6 možemo vidjeti vrijednosti evaluacijskih mjera za svaku klasu posebno te za cijeli sustav.

Tablica 3.6: Vrijednosti evaluacijskih mjera za testni skup

Mjera	Negativno	Pozitivno	Neutralno	Ukupno
Točnost	62.6%	0.0%	56.6%	55.0%
Preciznost	60.0%	0.0%	50.0%	51.0%
Odziv	63.0%	0.0%	57.0%	55.0%
F1-mjera	61.0%	0.0%	53.0%	53.0%

Dobiveni rezultati evaluacije su vrlo loši. Niti jedan testni primjer nije predviđen kao pozitivan. Razlog leži u tome što je skup za treniranje imao manje od 10% pozitivnih primjera. Također, suprotno očekivanjima, dodavanjem bigrama riječi nije poboljšana točnost sustava.

3.5. Poboljšanja

Poboljšanja u analizi sentimenta mogu se naći u boljem odabiru značajki i označavanjem većeg skupa komentara. Trenutni skup označenih komentara je nedovoljan za veliki broj značajki, stoga bi smanjenje broja značajki i povećanje skupa za treniranje sigurno dovelo do boljih rezultata te onemogućilo modelu da se u ovoj mjeri prilagodi skupu za treniranje. Treba napomenuti da dodatni problem predstavlja loša pismenost autora komentara te prečesti zatipci.

4. Dodatni alati

4.1. Prikupljač podataka

Prikupljač podataka je bot[1] koji sistematično pretražuje internet u svrhu skupljana velike količine podataka. U ovom radu prikupljač podataka je implementiran pomoću `Facebook Graph API-a`[2] radi dohvata podataka s političkih Facebook stranica. Za komunikaciju s Facebook Graph API-jem korištena knjižnica `urllib`.[3]

4.1.1. Implementacija web crawlera

Programski kod vezan za prikupljača podataka može se naći u direktoriju src/crawler. Sama implementacija prikupljača podataka može se pronaći u datoteci crawler.py. Crawler dohvaća podatke sa zadanih Facebook stranica u povijest do zadanog datuma te ih sprema u bazu podataka. Dohvaćani podatci su postovi te komentari tih postova uz metapodatke kao što su datum i autor.

4.1.2. Statistika podataka

Stranice Facebooka s kojih su prikupljeni podatci: *Vlada Republike Hrvatske, MOST nezavisnih lista, HDZ – Hrvatska demokratska zajednica, SDP Hrvatske, Živi zid, Hrvatska narodna stranka – liberalni demokrati (HNS).*

[1] Skraćeni naziv za robota. Računalni program koji se izvodi samostalno.
[2] https://developers.facebook.com/docs/graph-api
[3] https://docs.python.org/3/library/urllib.request.html

Tablica 4.1: Statistika podataka

Stranica	Broj postova	Broj komentara
Vlada Republike Hrvatske	170	22842
MOST nezavisnih lista	189	38740
HDZ - Hrvatska demokratska zajednica	104	2558
SDP Hrvatske	139	10777
Živi zid	1804	117470
Hrvatska narodna stranka – liberalni demokrati (HNS)	286	630
Ukupno	2692	193017

Također, u svrhu detekcije teme korišteno je i 211,479 novinskih članaka kako bi se povećao mali broj postova.

4.2. Alat za označavanje

Alat za označavanje čini izvršna datoteka `annotation.py` koja učita serijalizirane komentare te iz nakon označavanja razvrstava u predviđene datoteke za svaku klasu. Napravljen je jednostavo sučelje za označavanje tako što se za trenutni komentar upisuje sentiment kao što je prikazano na slici 4.1.

```
Comments left:  287
Negatives:  522
Positives:  98
No sentiment:  480
Nadam se da će ljudi iz Mosta početi misliti svojom glavom , a ne glavom velikog vođe.

Upisite sentiment komentara iznad ( 1 - pozitivno, 0 - negativno, n - ne znam, enter -
 preskoci, q - gotovo )  :

############################################################
```

Slika 4.1: Alat za označavanje

4.3. Baza podataka

Baza podataka je korištena za spremanje podataka za analiziranje dohvaćenih pomoću web crawlera. Korištena je `sqlite3`[4] baza podataka. Za komunikaciju s bazom podataka korištena je knjižnica `sqlite3`[5].

Tablica 4.2: Baza podataka

Tablica	Opis
Article	skup hrvatskih novinskih članka
Post	postovi s facebook stranica
Comment	komentari s facebook stranica
Page	facebook stranice
Vocabulary	vokabular procesiranog teksta

4.3.1. Struktura baze podataka

U nastavku su prikazani detaljni opisi tablica u bazi podataka.

Tablica 4.3: Tablica Page

Stupac	Opis
Id	identifikator
Name	ime stranice

[4]https://www.sqlite.org/

[5]https://docs.python.org/2/library/sqlite3.html

Tablica 4.4: Tablica Post

Stupac	Opis
Id	identifikator
Text	tekst post
Date	datum objave posta
Page id	identifikator facebook stranice

Tablica 4.5: Tablica Article

Stupac	Opis
Id	identifikator
Text	tekst članka
Date	datum objave članka

Tablica 4.6: Tablica Vocabulary

Stupac	Opis
Id	identifikator
Text	lemma riječi
POS	POS oznaka riječi
Freq	frekvencija pojavljivanja riječi

Tablica 4.7: Tablica Comment

Stupac	Opis
Id	identifikator
Text	tekst komentara
Date	datum objave komentara
Post id	identifikator posta
Comment id	id roditeljskog komentara
Author	ime autora komentara

5. Zaključak

Porastom raspoloživih količina korisnički generiranog sadržaja povećalo se zanimanje za strojnom analizom sentimenta, kojom se utvrđuje je li tekst usmjeren pozitivno, negativno ili neutralno. Cilj ovog rada je pomoću tehnika tematske analiza sentimenta razviti model temeljen na strojnom učenju (engl. *machine learning*) i procesiranju prirodnog jezika (engl. *natural language processing*), kako bi se odredio sentiment usmjeren prema određenoj temi.

Razvijeni su modeli za detekciju tema i analizu sentimenta. Za potrebe detekcije teme razvijen je sustav koji koristi LDA algoritam za grupiranje. Model je koristio velik skup novinskih članaka i postova s stranica Facebooka. Detekcija teme nije dala najbolje rezultate. Polovica dobivenih grupa su preopćenite i određene općenitim riječima. Jedan od problema leži u kratkoći i malom broju postova. U svrhu analize sentimenta koristili smo stroj potpornih vektora. Analiza sentimenta rješava se metodama nadziranog učenja, pa smo trebali označiti skup podataka za treniranje modela. Označeno je 1100 komentara s stranica Facebooka. Nažalost, model nije dao poželjne rezultate. Iz sličnih radova iz literature kao što su (Pang et al., 2002) i (Habernal et al., 2013) možemo zaključiti da ima mjesta za poboljšanja koja se kriju u pažljivijem odabiru značajki.

Što se tiče detekcije teme, mišljenja smo da se poboljšanje krije u pažljivijem filtriranju riječi koje nisu specifične ni za jednu temu. Kada je riječ o analizi sentimenta, unatoč poražavajućoj točnosti od 55% mišljenja smo da je pristup i odabir modela bio dobar. Moguća poboljšanja vidimo u odabiru boljih značajki te u većem skupu za treniranje.

LITERATURA

Nada A. Mostafa Ahmed Rafea. Topic extraction in social media. 2013.

Ravi Arunachalam i Sandipan Sarkar. The new eye of government: Citizen sentiment analysis in social media. 2013.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, i David M. Blei. Reading tea leaves: How humans interpret topic models. U *Neural Information Processing Systems*, 2009. URL `docs/nips2009-rtl.pdf`.

Chih-Chung Chang Chih-Wei Hsu i Chih-Jen Lin. A practical guide to support vector classification. 2003.

Christine Doig. Introduction to Topic Modeling in Python. `http://chdoig.github.io/pytexas2015-topic-modeling/#/`, 2015a.

Christine Doig. Introduction to Topic Modeling in Python. `https://www.youtube.com/watch?v=BuMu-bdoVrU&nohtml5=False`, 2015b.

Ivan Habernal, Tomaš Ptaček, i Josef Steinberger. Sentiment analysis in czech social media using supervised machine learning. 2013.

Bing Liu. *Sentiment Analysis and Subjectivity*. Department of Computer Science, University of Illinois at Chicago.

Luigi Di Caro Mario Cataldi i Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. 2010.

David Meyer. Support vector machines. 2015.

Saif M. Mohammad i Xiaodan Zhu. Sentiment Analysis of Social Media Texts. `https://www.youtube.com/watch?v=zv16Xyph7Ss`, 2014.

Bo Pang, Lillian Lee, i Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. 2002.

Radim Rehurek. Topic Modeling for Fun and Profit. http://radimrehurek.com/topic_modeling_tutorial/2%20-%20Topic%20Modeling.html, 2011-2016.

Ruchika Sharma i Amit Arora. Improve sentiment analysis accuracy using multiple kernel approach. 2013.

Jan Šnajder i Bojana Dalbelo Bašić. *Strojno učenje*. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2012.

Primjena strojnog učenja za tematsku analizu sentimenta

Sažetak

Uslijed ogromnog povećanja korisnički generiranog sadržaja na društvenih mrežama, detekcija tema i analiza sentimenta nameću se kao nezaobilazni alati za analizu javnog mijenja. U radu su proučeni postupci za detekciju tema u tekstnih dokumenata te analizu sentimenta prema detektiranim temama. Naglasak je stavljen na postupke temeljene na modelima strojnog učenja. Detekcijom tema nastoje se prepoznati teme o kojima se priča na društvenim mrežama te analizom sentimenta se određuje da li je stav o tim temama pozitivan, negativan ili neutralan. Rad se sastoji od cijelog procesa prikupljanja i preprocesiranja podataka. Nakon toga slijedi izgradnja modela detekcije tema i analize sentimenta te evaluacija modela. Dodatno, za analizu sentimenta je bilo potrebno označiti podatke za treniranje modela.

Ključne riječi: obrada prirodnog jezika, analiza sentimenta, strojno učenje, detekcija teme, latentna Dirichletova dodjela, stroj potpornih vektora, računalna lingvistika.

Application of Machine Learning for Topic-Based Sentiment Analysis

Abstract

Due to the huge increase in user-generated content on social networks, topic detection and sentiment analysis imposed as indispensable tools for the analysis of public opinion. The thesis examines the procedures for the topic detection in text documents and an sentiment analysis towards the detected topics. Special focus is placed on a procedures based on machine learning. Topic detection attempts to identify the topics on which the story on social networks and sentiment analysis to determine whether the position on these issues is positive, negative or neutral. The work consists of the entire process of collecting and preprocessing data. This is followed by the construction of topic detection and sentiment analysis models and evaluation of those models. In addition, sentiment analysis required data annotation for training the model.

Keywords: natural language processing, sentiment analysis, machine learning, topic detection, latent Dirichlet allocation support vector machines, Croatian language, computational linguistics.