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. 4274

Kontekstno ovisna analiza sentimenta izraza hrvatskoga jezika

Paula Gombar

Zagreb, srpanj 2015.

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

Zagreb, 13. ožujka 2015.

ZAVRŠNI ZADATAK br. 4274

Pristupnik: **Paula Gombar (0036474619)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Kontekstno ovisna analiza sentimenta izraza hrvatskoga jezika**

Opis zadatka:

Porastom raspoloživih količina korisnički generiranog sadržaja povećalo se zanimanje za strojnom analizom sentimenta, kojom se utvrđuje je li tekst usmjeren pozitivno, negativno ili neutralno. Uobičajeni postupci analize sentimenta temelje se na leksikonima apriornog sentimenta, koji svakoj riječi pridružuju oznaku sentimenta. Međutim, sentiment pojedinačnog izraza u rečeničnome kontekstu općenito ne mora odgovarati apriornom sentimentu riječi od kojih je taj izraz sastavljen. Preciznije modeliranje sentimenta riječi i fraza u kontekstu, odnosno semantička kompozicija sentimenta, važan je zadatak u obradi prirodnoga jezika i preduvjet za preciznu analizu sentimenta.
U okviru završnoga rada potrebno je proučiti postupke za analizu sentimenta s naglaskom na postupke temeljene na semantičkoj kompoziciji sentimenta i postupke temeljene na strojnom učenju. Razraditi model za kontekstno-ovisnu analizu sentimenta izraza hrvatskoga jezika temeljen na modelu nadziranog strojnog učenja, po uzoru na rad Wilsona i dr. (2005). Model treba koristiti niz značajki ekstrahiranih iz teksta, uključivo sintaktičke značajke. Izgraditi i ručno označiti odgovarajući skup tekstnih podataka na hrvatskome jeziku za razvoj i ispitivanje modela. Razviti programsku implementaciju modela te ga primijeniti na podatke na hrvatskome jeziku. Provesti iscrpno eksperimentalno vrednovanje modela, uključivo usporedbu s referentim modelom, statističku obradu rezultata te analizu pogrešaka. Radu priložiti izvorni i izvršni kod razvijenog sustava, označene skupove podataka i potrebnu dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 13. ožujka 2015.
Rok za predaju rada: 12. lipnja 2015.

Mentor:

Doc. dr. sc. Jan Šnajder

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof. dr. sc. Siniša Srbljić

Zahvaljujem Matiji Folnoviću, Antei Hadviger, Davidu Kišu, Lovru Klečini, Ivanu Paljaku, Marseli Polić, Ivanu Sekuliću, Matiji Šantlu, Marinu Tomiću i Borni Vukadinoviću na izdvajanju svog vremena i označavanju podataka. Zahvaljujem mentoru doc. dr. sc. Janu Šnajderu, kolegi mag. ing. Domagoju Alagiću i cijeloj TakeLab ekipi na pomoći, savjetima i otključavanju zavoda. Zahvaljujem svojoj obitelji, psu, prijateljima i boljoj polovici na strpljenju i podršci tokom čitavog studija.

Sadržaj

1. Uvod **1**

2. Analiza sentimenta **3**

2.1. Opis problema 3

2.2. Strojno učenje u obradi prirodnog jezika 4

2.3. Srodni radovi 6

3. Model kontekstno ovisne analize sentimenta **9**

3.1. Značajke klasifikatora 10

3.2. Stroj potpornih vektora 12

3.2.1. Linearna klasifikacija 13

3.2.2. Klasifikacija meke margine 15

3.2.3. Nelinearna klasifikacija 16

3.2.4. Višeklasni model 17

4. Implementacija modela **19**

4.1. Web crawler 19

4.2. Alat za označavanje 19

4.3. Alati za obradu podataka 21

4.4. Alati za vektorizaciju podataka 21

4.5. Klasifikatori 22

4.5.1. Klasifikator temeljen na apriornom sentimentu 22

4.5.2. Ostali klasifikatori 23

5. Evaluacija **26**

5.1. Skup podataka 26

5.2. Suglasnost označivača 29

5.3. Evaluacijske mjere 29

5.4. Klasifikacija subjektivnosti rečenica 30

5.5. Klasifikacija sentimenta rečenica 31
5.6. Klasifikacija sentimenta označenih izraza 32
5.7. Cjevovod klasifikatora 33
5.8. Diskusija rezultata i analiza pogrešaka 34
5.9. Usporedba rezultata s referentnim modelom za engleski jezik . . . 35

6. Zaključak **36**

Literatura **37**

A. Upute za označavanje **39**
A.1. Motivacija 39
A.2. Opis posla 39
A.2.1. Označavanje teksta 40
A.2.2. Pravila označavanja 40
A.3. Alat za označavanje 41
A.3.1. Instalacija i pokretanje programa 41
A.3.2. Početak rada 42
A.3.3. Označavanje 43
A.3.4. Kraj rada 43

1. Uvod

U današnje vrijeme postoji ogromna količina korisnički generiranog sadržaja. Najčešće su ti podaci nestrukturirani, a za bilo kakvu analizu sadržaja potrebno je prijeći iz nestrukturiranog u strukturirani oblik kako bismo omogućili računalima da izvuku značenje iz ljudskog jezika. Tom problemu doskače obrada prirodnog jezika (engl. *natural language processing*), područje računalne znanosti (engl. *computer science*), umjetne inteligencije (engl. *artificial intelligence*) i računalne lingvistike (engl. *computational linguistics*) koje se bavi interakcijom računalnih i ljudskih (prirodnih) jezika.

Da bismo shvatili zadatak obrade prirodnog jezika, potrebno je definirati prirodan jezik. Prirodan jezik je jezik koji koriste ljudi, dok su primjeri umjetnih jezika programski jezici i jezik predikatne logike. Obrada prirodnog jezika uključuje analizu morfoloških, sintaktičkih i semantičkih svojstava jezika, s ciljem da računalo kao stroj uspije razumjeti ljudski jezik. Jedan od prvih, ali i najpoznatijih radova na tu temu jest (Turing, 1950), u kojem je opisan eksperiment gdje računalo pokušava imitirati čovjeka, što se pokazalo izuzetno složenim zadatkom.

Danas se obrada prirodnog jezika pretežito oslanja na statističko strojno učenje (engl. *machine learning*), ali unatoč tome često najbolje rezultate daju modeli statističkog strojnog učenja u kombinaciji s modelima temeljenim na ručno stvorenim pravilima (engl. *rule-based*).

Primjene obrade prirodnog jezika su brojne. Počevši od sažimanja teksta (engl. *text summarization*), strojnog prevođenja (engl. *machine translation*), parsanja (engl. *parsing*), analize sentimenta (engl. *sentiment analysis*) do razrješavanja višeznačnosti riječi (engl. *word sense disambiguation, WSD*), crpljenja informacija (engl. *information retrieval, IR*) i pretraživanja informacija (engl. *information extraction, IE*).

Ovaj se rad fokusira na primjenu obrade prirodnog jezika u analizi sentimenta. Strojnom analizom sentimenta utvrđuje se je li tekst usmjeren pozitivno, negativno ili neutralno. Uobičajeni postupci analize sentimenta temelje se na lek-

sikonima apriornog sentimenta, koji svakoj riječi pridružuju oznaku sentimenta. Međutim, sentiment pojedinačnog izraza u rečeničnome kontekstu općenito ne mora odgovarati apriornom sentimentu riječi od kojih je taj izraz sastavljen.

Kada govorimo o riječima kao o leksemima[1] (grc. *léxis* = riječ), njihovo značenje je dvojako: 1. denotativno (neutralno) značenje izravno govori o imenovanoj stvari iz izvanjezične zbilje na neutralan i obavijestan način, ili 2. konotativno (obilježeno) značenje govori o imenovanoj stvari iz izvanjezične zbilje uz unošenje emocija, afekata ili pristranih doživljaja

Budući da je tema rada kontekstno ovisna analiza sentimenta, fokus je na konotativnom značenju. Analiza sentimenta složen je zadatak obrade prirodnog jezika, a često se još naziva i rudarenjem mišljenja (engl. *opinion mining*). Primjene nalazi u analizi korisničkih postova na sve popularnijim društvenim mrežama (engl. *social networks*), blogovima i forumima, kao i u poslovnoj inteligenciji (engl. *business inteligence*). Srodni problemi su crpljenje informacija, određivanje naklonosti (engl. *bias identification*) i sumarizacija teksta.

U ovom radu, razvijen je model kontekstno ovisne analize sentimenta izraza za hrvatski jezik, a rađen je po uzoru na postojeći model razvijen za engleski jezik, opisan u radu (Wilson et al., 2005). Model se sastoji od tri zadatka: 1. klasifikacija subjektivnosti rečenice (engl. *sentence-level subjectivity classification*); 2. klasifikacija sentimenta rečenice (engl. *sentence-level sentiment classification*) i 3. klasifikacija sentimenta označenih izraza (engl. *phrase-level sentiment classification*). Kao skup podataka (engl. *dataset*) korištene su recenzije igara s hrvatskog *gaming* portala HCL.hr[2]. Ukupno je prikupljeno 4427 rečenica iz 95 tekstova.

Rad je strukturiran tako da je u poglavlju 2 opisan problem za koji se razvija model, zatim je u poglavlju 3 opisan razvijeni model i koje zadatke pokušava riješiti. U poglavlju 4 objašnjena je programska implementacija modela te korišteni alati, dok je u poglavlju 5 dan pregled metoda evaluacije i točnost modela u pojedinim zadacima. Naposljetku, u poglavlju 6 dan je pregled svega postignutog u okviru rada, kao i moguće buduće nadogradnje. U dodatku A prikazane su upute namijenjene označivačima.

[1] Ukupnost svih oblika i značenja koje ima jedna riječ.

[2] `http://www.hcl.hr/recenzije.php`.

2. Analiza sentimenta

2.1. Opis problema

Analiza sentimenta ponekad predstavlja netrivijalan problem ljudima, a pogotovo računalima. Njen cilj jest identifikacija i ekstrakcija subjektivne informacije iz skupa podataka. Nositelji sentimenta (engl. *sentiment carriers*) iskazuju sentiment, a njihova analiza svodi se na identifikaciju stavova, osjećaja i mišljenja.

Osnovni zadatak analize sentimenta je klasifikacija polariteta (engl. *polarity classification*) teksta, što u osnovnom slučaju predstavlja klasifikaciju u jednu od tri skupine: pozitivno, negativno ili neutralno. Složeniji problem analize sentimenta jest klasifikacija u više razina (engl. *multi-way scale*). Primjerice, dok bi osnovna klasifikacija polariteta recenzije nekog proizvoda mogla reći smatra li se on dobrim ili lošim, klasifikacija polariteta u više razina bi mogla reći koliko je on dobar ili loš (na primjer na ljestvici od 1 do 5).

Što se tiče dosega analize sentimenta, on se može provoditi nad riječima ili izrazima (engl. *word- or phrase-level*), rečenicama (engl. *sentiment-level*) ili nad čitavim dokumentima (engl. *text- or document-level sentiment analysis*). Svaka razina uvodi drukčije poteškoće, olakotne okolnosti i izazove.

Budući da računalo inicijalno nema nikakvo prethodno znanje o prirodnom jeziku, često pomaže ako raspolažemo sakupljenim znanjem o promatranoj temi ili leksikonom sentimenta riječi (engl. *sentiment lexicon*). Primjerice, okvirni koraci u algoritmu analize sentimenta mogli bi izgledati ovako:

1. učitaj leksikon apriornog[1] sentimenta,
2. učitaj tekst koji je potrebno klasificirati,
3. iskoristi podatke iz leksikona za klasifikaciju sentimenta teksta.

[1]Koji se uzima kao takav bez prethodnog iskustva ili dokaza, koji sadržava preduvjerenje.

Izazovi ovog zadatka su brojni, počevši od toga da računala ne razumiju ljudski jezik, nego ga moraju reprezentirati u strukturiranijem kontekstu. Nadalje, ljudi izražavaju svoja mišljenja, stavove i osjećaje na složen i jedinstven način. To se očituje u činjenici da se ponekad ni ljudi ne mogu složiti oko značenja nekog teksta, pogotovo ako postoje određene kulturološke ili lingvističke razlike u izražavanju. Također, računalo jako teško može razaznati suptilnost izražavanja, sarkastične komentare ili retorička pitanja koja možemo naći u subjektivnom tekstu.

Još jedna od poteškoća u analizi sentimenta je svakako semantička kompozicionalnost sentimenta, gdje važnu ulogu igraju modifikatori i negacija. Primjerice, logično je da upotrebom negacije sentiment poprima polaritetno značenje, dok modifikatori služe za gradaciju sentimenta. Zadatak je računala da točno identificira sve komponente koje pridodaju sentimentu trenutno promatranog izraza, što često nije jednostavan slučaj.

Ipak, znanost je postigla značajan napredak u ovom području kontinuirano unaprijeđujući razne metode i modele. Iskustvo u ovom području pokazalo je da modeli razvijeni za tekstove iz specifične domene ne moraju nužno dobro raditi kada se primijene na ostale domene. Uzrok toga je specifičnost izražavanja i određeno domensko znanje (engl. *domain knowledge*), čime se sužava leksikon ako govorimo samo o specifičnoj domeni.

2.2. Strojno učenje u obradi prirodnog jezika

Strojno učenje potpodručje je računarske znanosti i grana umjetne inteligencije koja istražuje razvoj algoritama koji mogu učiti iz podataka i predviđati na temelju njih. Algoritmi stvaraju model definiran parametrima, a učenje znači da algoritam optimizira parametre modela temeljem podataka. Strojno učenje svoje korijene nalazi u raspoznavanju uzoraka (engl. *pattern recognition*) i statistici, a začetnik filozofije oko toga mogu li strojevi učiti upravo je Alan Turing poznatim pitanjem *Can machines think?* postavljenim u radu (Turing, 1950).

Tri su osnovna pristupa strojnom učenju:

1. **Nadzirano učenje** (engl. *supervised learning*) – potrebno je naći preslikavanje $\hat{y} = f(x)$, a ulazni podaci su oblika $(ulaz, izlaz) = (x, y)$. S obzirom na varijablu y, razlikujemo:

   (a) klasifikaciju – ako je y diskretna ili nebrojčana vrijednost,

(b) regresiju – ako je y kontinuirana ili brojčana vrijednost.

2. **Nenadzirano učenje** (engl. *unsupervised learning*) – dani podaci nemaju ciljnu vrijednost i potrebno je naći pravilnost u podacima, a dijeli se na:

   (a) grupiranje (engl. *clustering*),

   (b) procjenu gustoće (engl. *density estimation*),

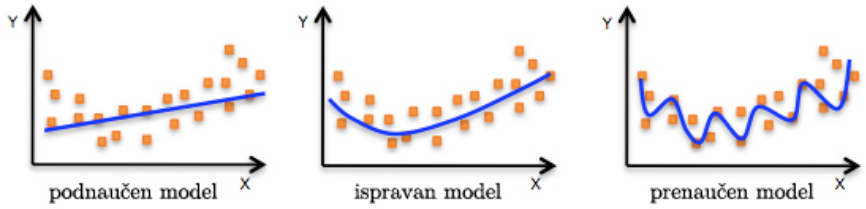
   (c) smanjenje dimenzionalnosti (engl. *dimensionality reduction*).

3. **Podržano ili ojačano učenje** (engl. *reinforcement learning*) – uči se optimalna strategija na temelju pokušaja s odgođenom nagradom.

Najčešći pristup u analizi sentimenta jest klasifikacija (engl. *sentiment classification*), primjerice klasifikacija rečenica kao subjektivnih ili objektivnih, ili pak grupiranje, primjerice grupiranje sličnih dokumenata ili recenzija.

Uobičajeni koraci u primjeni strojnog učenja su: 1. odabir modela, 2. učenje modela i 3. evaluacija.

Učenje modela provodi se nad skupom za učenje (engl. *train set*), a evaluacija nad skupom za ispitivanje (engl. *test set*). Potrebno je naglasiti da model ne smije prije vidjeti primjere iz skupa za ispitivanje. Kada model nema nikakvog znanja o primjerima iz skupa za ispitivanje, dobiva se pesimističnija, pa tako i realnija procjena performansi modela. Međutim, čak i kada se skup za učenje razlikuje od skupa za ispitivanje može se dogoditi da dođe do prenaučenosti ili podnaučenosti modela. Prenaučenost znači da model loše generalizira, a javlja se zbog prevelike složenosti modela, dok do podnaučenosti dolazi zbog toga što je model prejednostavan. Ilustracija ovih primjera prikazana je slikom 2.1.



Slika 2.1: Različiti modeli ovisno o naučenosti

U svijetu obrade prirodnog jezika, najčešće se koriste algoritmi temeljeni na strojnom učenju, a pogotovo na statističkim modelima. Razlog tomu je činjenica da je potrebno automatski naučiti pravila analizirajući velik korpus (engl. *corpus, pl. corpora*). Model se uči na skupu primjera za učenje koji su predstavljeni

vektorom značajki (engl. *features*) ekstrahiranih iz početnog teksta, a upravo je ovaj korak izrade modela ključan za klasifikaciju teksta. U tom se koraku tekst tipično predstavlja vektorom značajki (engl. *feature vector*). Jedna takva značajka bi mogla biti prisustvo ili frekvencija pojavljivanja određene riječi ili n-grama, POS oznaka neke riječi i slično. U radu (Pang et al., 2002) pokazalo se da je značajka prisustva, umjesto frekvencije, pokazala bolje rezultate.

Kada govorimo o analizi sentimenta, najčešće korištena metoda strojnog učenja je klasifikacija. Klasifikacijom možemo odgovoriti na pitanje koliko je nešto pozitivno, pobliže odrediti semantičku relaciju između dva objekta, pridijeliti oznaku čitavom dokumentu i sl. Kao što je prije spomenuto, kod analize sentimenta čest je zadatak odrediti polaritet nečega, tj. je li nešto pozitivno ili negativno i u kojoj mjeri (engl. *sentiment polarity classification*). Većina klasifikatora teksta temelji se na metodi potpornih vektora (engl. *support vector machines*) jer često pružaju najbolje rezultate u odredivanju polariteta teksta.

S druge strane, nedostatak bilo kojeg modela strojnog učenja jest taj što isti klasifikator ima znatno lošije rezultate kada ga primijenimo nad drukčijom domenom. Jedan od primjera spomenut u radu (Pang i Lee, 2008) je uporaba riječi *nepredvidljiv* – dok je ta riječ pozitivna u kontekstu recenzije filma, u kontekstu recenzije auta definitivno je negativna.

Nadalje, još jedna neizbježna stavka kod analize sentimenta jest potreba za korištenjem leksikona sentimenta (engl. *sentiment lexicon*). Do tih leksikona može se doći ručno, kao rezultat označavanja ljudi, ili strojno, kao rezultat metoda temeljenih na rječniku (engl. *dictionary-based*) ili metoda temeljenih na korpusu (engl. *corpus-based*). Kada govorimo o *kontekstno* ovisnoj analizi sentimenta, jasno je da konotativno značenje riječi ne mora odgovarati značenju te riječi iz leksikona sentimenta, no na modelu je da ih nauči razlikovati.

2.3. Srodni radovi

Primarni rad po uzoru na kojeg je razvijen model u ovom radu je (Wilson et al., 2005). U tom radu razvijen je model za kontekstno ovisnu analizu sentimenta izraza engleskog jezika. Model koristi strojno učenje u dvije faze – prvo određuje je li izraz neutralan ili polaran, a zatim pobliže klasificira polaritet polarnih izraza kao pozitivan, negativan, neutralan ili miješan. Prije razvijanja samog modela, autori su na raspolaganju imali veliki korpus prethodno označenih subjektivnih izraza unutar rečenica, što u ovom radu nije slučaj. Ukupno je klasificirano 15991

subjektivnih izraza iz 425 dokumenata (8984 rečenica), a smišljeno je ukupno 28 sintaktičkih i semantičkih značajki. Prvi klasifikator odgovara prvoj fazi, a drugi drugoj fazi. Oba klasifikatora znatno su bolja od primitivnog klasifikatora koji pridjeljuje sve primjere većinskoj klasi iz skupa za učenje. Očekivano, prvi klasifikator ima bolje performanse od drugog. Odabrane značajke i zlatni standard korišteni u radu detaljnije su opisani u radu (Wilson et al., 2009).

U radu (Pang et al., 2002) provodi se analiza sentimenta na razini dokumenta (engl. *document-level*), a ne izraza. Isproban je nešto drukčiji pristup od klasificiranja dokumenata po temi, a to je klasificiranje dokumenata po sveukupnom sentimentu. Bolje rečeno, pokušava se odrediti je li recenzija pozitivna ili negativna. Pritom se koriste tri metode za klasifikaciju sentimenta: naivni Bayes (engl. *Naive Bayes*), klasifikacija najveće entropije (engl. *maximum entropy classification*) i stroj potpornih vektora (engl. *support vector machines*). Iako su sve tri metode rezultatima iznad *baselinea*, u radu se pokazalo da pružaju veću točnost kada se koriste za klasifikaciju teme dokumenta, nego za klasifikaciju sveukupnog sentimenta. To je objašnjeno time da se tema može naslutiti iz čestih ključnih riječi, dok se sentiment izražava puno suptilnije i tako se teže otkriva.

Još jedan rad koji je inspirirao model za hrvatski jezik je rad (Pang i Lee, 2004), koji je uveo novinu u analizi sentimenta, jer koristi tehnike kategorizacije teksta (engl. *text categorization*) nad subjektivnim dijelovima teksta za određivanje polariteta. Umjesto dotadašnjeg istraživanja u klasifikaciji dokumenata koje predlaže korištenje klasifikatora nad cijelim tekstom, u ovom radu predložena je uporaba klasifikatora samo nad subjektivnim rečenicama prvotnog teksta. Autori rada prvo su prikupili stranice s interneta u potrazi za automatski označenim subjektivnim izrazima iz domene filmskih recenzija. Korištene metode su naivni Bayes (engl. *Naive Bayes*) i stroj potpornih vektora (engl. *support vector machines*).

Suprotan pristup dosad navedenima jest metodologija korištena u radu (Taboada et al., 2011). Pristup se temelji na leksikonu sentimenta (engl. *lexicon-based approach*) za analizu sentimenta. Zaključak rada jest da su metode temeljene na leksikonu sentimenta robusne i primjenjive na više domena (engl. *cross-domain*). Model je primarno namijenjen engleskom jeziku, ali se kasnije razvio i za španjolski i čak pokazao bolje rezultate. Razvijen je model analize sentimenta pod nazivom *The Semantic Orientation CALculator (SO-CAL)* koji koristi rječnike koji sadrže podatke o polaritetu i jačini riječi. Bitni faktori u analizi sentimenta u ovom radu bili su pridjevi i pojačivači (engl. *intensifiers*), kao i negacije.

Što se tiče leksikona sentimenta, upravo njegova izgradnja cilj je rada (Glavaš et al., 2012), gdje je korišten hibridni pristup izgradnji leksikona. Rad se temelji na analizi engleskog i hrvatskog jezika, a izgrađen je leksikon od 41359 riječi. Korišteni pristup temelji se na korpusu i kombinira polunadzirane modele temeljene na grafovima i nadzirane modele. U radu je fokus na tri različita zadatka, od kojih svaki pokriva drukčiji aspekt leksikona sentimenta. Prvi zadatak je rangiranje po polaritetu (engl. *polarity ranking task*), odnosno određivanje relativnog ranga riječi po njenoj pozitivnosti ili negativnosti. Drugi zadatak je regresija polariteta (engl. *polarity regression task*) gdje se svakoj riječi pokušava pridijeliti realan broj između 0 i 1 kao ocjena pozitivnosti i negativnosti. Posljednji, treći, zadatak najbitniji je za model razvijen u ovom radu, a to je klasifikacija sentimenta (engl. *sentiment classification task*), gdje je svaka riječ klasificirana u pozitivnu, negativnu ili neutralnu klasu. Upravo se leksikon sentimenta razvijen u trećem zadatku rada koristi u modelu ovog rada.

Još jedan rad koji se bavi analizom sentimenta izraza hrvatskog jezika je (Biđin et al., 2014). U tom radu koristi se model duboke neuronske mreže kako bi se postigla klasifikacija sentimenta izraza koji se sastoji od točno dvije riječi (engl. *bigram*). Rad se referira na model razvijen za engleski jezik i primjenjuje jednake postavke na hrvatski jezik. Model je testiran nad dva različita skupa podataka. Prvi skup podataka sastoji se od 1500 prethodno označenih izraza na hrvatskom jeziku, a drugi skup podataka su recenzije filmova, ali na engleskom jeziku. Model postiže bolje rezultate na skupu podataka na hrvatskom jeziku, a sveukupno pokazuje da je model dubokih neuronskih mreža zadovoljavajuć pristup u analizi sentimenta.

3. Model kontekstno ovisne analize sentimenta

Razvijeni model za kontekstno ovisnu analizu izraza hrvatskog jezika inspiriran je modelom za engleski jezik opisanim u radu (Wilson et al., 2005) i koristi metodu nadziranog strojnog učenja. Model koristi niz značajki ekstrahiranih iz teksta, uključivo sintaktičke značajke. Osnovni zadaci modela definirani su redom:

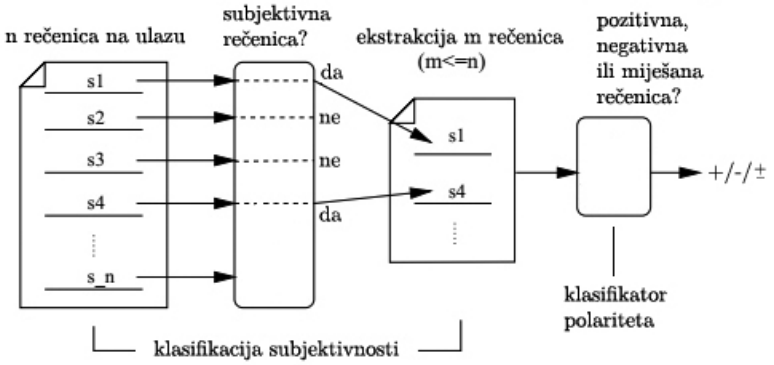
1. klasifikacija subjektivnosti rečenice (engl. *sentence-level subjectivity classification*): oznake subjektivna ili objektivna,

2. klasifikacija sentimenta subjektivnih rečenica (engl. *sentence-level sentiment classification*): oznake pozitivna, negativna ili miješana,

3. klasifikacija polariteta izraza (engl. *phrase-level sentiment classification*): oznake pozitivan ili negativan.

Prvi zadatak može se nazvati detektorom subjektivnosti (engl. *subjectivity detector*). Sve rečenice koje prođu kroz njega prozvane su subjektivnima i zatim ulaze u klasifikator sentimenta subjektivnih rečenica koji zatim odredi je li rečenica pozitivnog, negativnog ili miješanog polariteta. Kao dodatan, četvrti zadatak razvijen je klasifikator čiji tok podataka ide iz izlaza prvog klasifikatora u ulaz drugog klasifkatora i naziva se cjevovodom klasifikatora (engl. *pipeline*). Tok podataka u tom slučaju prikazan je na slici 3.1.

Oprimjereno, model bi u prvom zadatku trebao moći rečenicu *Grafika je vrlo dobra, a kamera odlično prati lika.* klasificirati kao subjektivnu, a u drugom zadatku kao pozitivnu subjektivnu rečenicu. Treći zadatak klasificira segmente rečenice, stoga bi izraz *grafika je vrlo dobra* model trebao klasificirati kao pozitivan izraz. Ako uzmemo objektivnu rečenicu, npr. *Glavni lik je vodoinstalater Mario.*, prvi klasifikator trebao bi je klasificirati kao objektivnu rečenicu, što znači da u cjevovodu ona neće ni doći do drugog klasifikatora. Treba uočiti da



Slika 3.1: Klasifikacija polariteta nakon klasifikacije subjektivnosti rečenice, cjevovod 1. i 2. klasifikatora

bi, pri samostalnom ispitivanju klasifikatora, drugi klasifikator ipak svrstao ovu objektivnu rečenicu u jedne od tri skupine, ovisno o naučenim parametrima.

3.1. Značajke klasifikatora

Značajke su most koji povezuje čisti tekst i ulaz klasifikatora. Iz čistog teksta prvo se osmisle značajke koje opisuju neke karakteristike teksta i zatim se tekst pretvori u vektor značajki. Vektor takvih značajki tako (idealno) reprezentiraju tekst u vektorskom prostoru, koji je pogodan, a i neophodan, za daljnju obradu. Naime, klasifikator prima ulazne podatke u sljedećem obliku vektora značajki:

$$< klasa >< indeks_1 >:< vrijednost_1 >< indeks_2 >:< vrijednost_2 > \ldots$$

gdje $< klasa >$ označava klasu dodijeljenu zlatnim standardom. Zlatni standard je oznaka dobivena konsenzusom označivača. Korištena implementacija klasifikatora u ovom radu ne zahtijeva puni zapis vektora značajki, koji često ima nekoliko stotina tisuća značajki. Umjesto toga, koristi se rijetki format (engl. *sparse format*) zapisa gdje je moguće izostaviti par $< indeks_i >:< vrijednost_i >$ ako je $< vrijednost_i >$ jednak 0. Ovakav način zapisa omogućava brži račun i manje memorijsko zauzeće.

Budući da se model sastoji od tri različita zadatka, potrebno je zasebno promatrati značajke za svaki od njih. Zajedničke značajke svim zadacima su one koje predstavljaju model *vreće riječi* (engl. *bag-of-words model*). Model vreće riječi je tehnika pojednostavljivanja prikaza teksta takva da se prvo izgradi rječnik od svih riječi iz ulaznog skupa podataka. U ovom slučaju to znači iteriranje po svim

Tablica 3.1: Vektor značajki korišten u prvom i drugom zadatku

Indeks	Opis značajke	Tip vrijednosti
0 . . . 1341	prisutnost riječi iz rječnika	binaran
1342	broj riječi u instanci	0 . . . N
1343	sadrži *?*	binaran
1344	sadrži *!*	binaran
1345	postotak pridjeva	0 . . . 1
1346	postotak priloga	0 . . . 1
1347	postotak čestica	0 . . . 1
1348	sadrži suprotni veznik	binaran
1349	sadrži isključni veznik	binaran
1350	sadrži riječ iz top 100 pozitivnih	binaran
1351	sadrži riječ iz top 100 negativnih	binaran
1352	sadrži uzorak POS SUPROTNI_VEZNIK NEG	binaran
1353	sadrži uzorak NEG SUPROTNI_VEZNIK POS	binaran
1354	sadrži uzorak NEG_MODIFIKATOR . . . POS	binaran

lematiziranim oblicima riječi koje se pojavljuju u 95 tekstova recenzija i spremanje u rječnik. Tim postupkom stvoren je rječnik od 13642 jedinstvena zapisa. Potrebno je naglasiti da rječnik nije pročišćen od zapisa koji nisu riječi same po sebi, npr. razni oblici interpunkcije. To je zato što će i takvi zapisi poslužiti za analizu sentimenta u daljnjim koracima.

Model vreće riječi zatim nalazi frekvencije pojavljivanja svake riječi. Međutim, u ovom radu korištena je samo zastavica prisutnosti: i-ti zapis u vektoru značajki je 1 ako se riječ pojavljuje, 0 ako se riječ ne pojavljuje u trenutnoj instanci (izraz ili rečenica, ovisno o zadatku). U radovima (Pang et al., 2002) i (Pang i Lee, 2004) pokazalo se da je to bolji odabir od zapisa frekvencije riječi.

Dakle, prvih 13641 zapisa u vektoru značajki isto je u sva tri zadatka i predstavlja model vreće riječi. Prvi i drugi zadatak, klasifikacija subjektivnosti rečenice i klasifikacija sentimenta subjektivnih rečenica, dalje imaju identične značajke. U oba klasifikatora instancu predstavlja rečenica, a razlika je u broju klasa – klasifikator definiran u prvom zadatku na izlazu ima dvije moguće klase, a onaj drugom tri. Značajke su prikazane tablicom 3.1. Značajke na indeksu 1350 i 1351 ovise o leksikonu sentimenta koji je razvijen u sklopu rada (Glavaš et al., 2012). Suprotni veznici su *a*, *ali*, *dok*, *god*, *nego*, *no*, *već*, *pa*, *pak* i *ipak*.

Tablica 3.2: Vektor značajki korišten u trećem zadatku

Indeks	Opis značajke	Tip vrijednosti
0 . . . 1341	prisutnost riječi iz rječnika	binaran
1342	broj riječi u instanci	0 . . . N
1343	sadrži *?*	binaran
1344	sadrži *!*	binaran
1345	sadrži suprotni veznik	binaran
1346	sadrži isključni veznik	binaran
1347	sadrži riječ iz top 100 pozitivnih	binaran
1348	sadrži riječ iz top 100 negativnih	binaran
1349	sadrži uzorak POS SUPROTNI_VEZNIK NEG	binaran
1350	sadrži uzorak NEG SUPROTNI_VEZNIK POS	binaran
1351	sadrži uzorak NEG_MODIFIKATOR . . . POS	binaran
1352	omjer pozitivnih i negativnih riječi	0 . . . 1

Isključni veznici su *samo*, *tek*, *jedino* i *osim*. Značajka na indeksu 1354 ispituje NEG_MODIFIKATOR, varijablu koja je predstavljena riječima *ne*, *neće*, *nema*, *nisu*, *nikako*, *nije*, *nipošto*, *nikad*, *ni*, *niti*.

Treći zadatak, klasifikacija polariteta izraza, nešto je drukčiji od prethodna dva, stoga su i značajke za taj zadatak drugačije. Kao što se vidi u tablici 3.2, izbačene su značajke koje promatraju udio broja pridjeva, priloga i čestica u instanci, zato što je to više pokazatelj subjektivnosti ili objektivnosti, a u trećem zadatku zanima nas polaritet, tj. je li nešto pozitivno ili negativno. Stoga je dodana značajka koja promatra omjer pozitivnih i negativnih riječi.

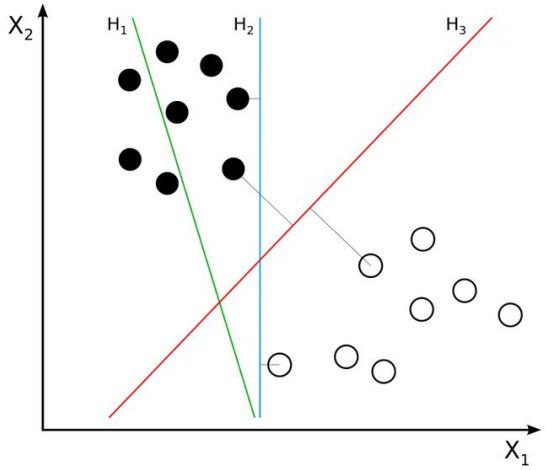
3.2. Stroj potpornih vektora

Stroj potpornih vektora (engl. *support vector machines, SVM*)[1] nadzirana je metoda strojnog učenja i koristi se za klasifikaciju. Metoda je nadzirana zato što radi tako da dobije skup za učenje s označenim pripadajućim klasama i na temelju toga stvori model koji je sposoban pridijeliti oznake tih klasa novim, neviđenim podacima. U ovom radu, SVM se koristi kao binarni klasifikator izraza i rečenica i višeklasni klasifikator rečenica.

Ideja ove metode je prikazati ulazne podatke kao točke u prostoru, mapirane

[1] `http://scikit-learn.org/stable/modules/svm.html`.

na način da su točke različitih klasa odvojene. Ova metoda nalazi hiperravninu koja ima najveću marginu razdvajanja klasa, gdje je margina udaljenost između točaka koje pripadaju suprotnim klasama. SVM maksimizira marginu oko hiperravnine razdvajanja. Potporni vektori su oni koji se nalaze na rubovima te praznine i najbliži su podacima. Primjer linearne klasifikacije ilustriran je na slici 3.2.



Slika 3.2: Primjer kako SVM odabire najbolju hiperravninu za klasifikaciju. H_1 ne razdvaja klase, dok H_2 i H_3 da. Međutim, H_3 to čini s najvećom marginom.

3.2.1. Linearna klasifikacija

Formalno, kažemo da imamo L primjera za učenje, gdje svaki primjer $\mathbf{x_i}$ ima D značajki i pripada jednoj od dvije klase $y_i = -1$ ili $+1$. To znači da je skup za učenje oblika:

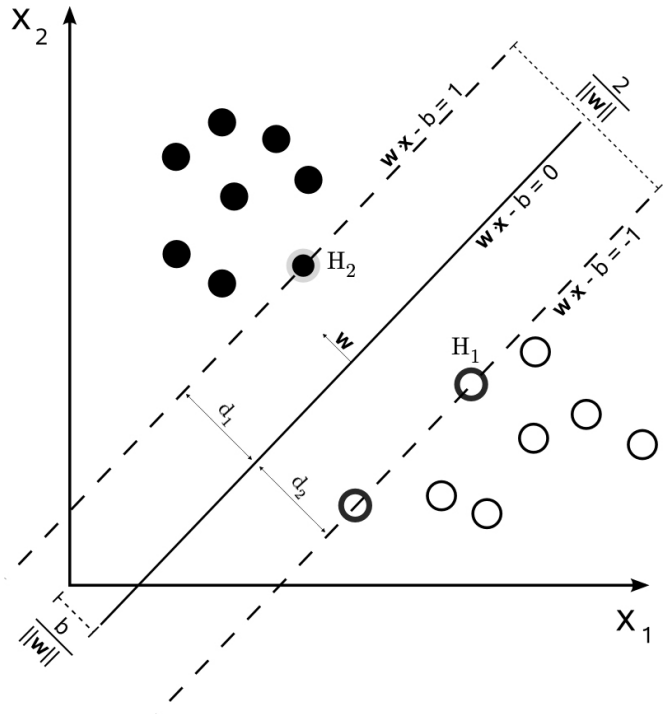
$$\{(\mathbf{x_i}, y_i)\}_i \quad \text{gdje je} \quad i = 1 \ldots L, y_i \in \{-1, 1\}, \mathbf{x} \in \Re^D$$

Ako pretpostavimo da su podaci linearno razdvojivi, to znači da možemo povući pravac na grafu koji razdvaja dvije klase $\mathbf{x_1}$ i $\mathbf{x_2}$ kada je $D = 2$ i hiperravninu na grafovima $\mathbf{x_1}, \mathbf{x_2} \ldots \mathbf{x_D}$ kada je $D > 2$.

Hiperravnina se može opisati izrazom $\mathbf{w} \cdot \mathbf{x} + b = 0$, gdje je $\mathbf{w}$ normala na hiperravninu, a $\frac{b}{\|\mathbf{w}\|}$ okomita udaljenost hiperravnine od ishodišta.

Ako gledamo sliku 3.3, implementacija SVM-a svodi se na odabir parametara $\mathbf{w}$ i b tako da se podaci za učenje mogu prikazati kao:

$$\mathbf{x_i} \cdot \mathbf{w} - b \geq +1 \quad \text{za} \quad y_i = +1 \tag{3.1}$$



Slika 3.3: Hiperravnina kroz dvije linearno razdvojive klase. Potporni vektori nalaze se na margini ravnine.

$$\mathbf{x_i} \cdot \mathbf{w} - b \leq -1 \quad \text{za} \quad y_i = -1 \tag{3.2}$$

Jednadžbe 3.1 i 3.2 mogu se prikazati jednom jednadžbom:

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i \tag{3.3}$$

Ako gledamo samo točke koje se nalaze najbliže hiperravnini (na slici 3.3 označene kružićima, drugim riječima potporni vektori) koja dijeli klase, onda se ravnine označene s H_1 i H_2 mogu napisati kao:

$$\mathbf{x_i} \cdot \mathbf{w} + b = +1 \quad \text{za} \quad H_1 \tag{3.4}$$

$$\mathbf{x_i} \cdot \mathbf{w} + b = -1 \quad \text{za} \quad H_2 \tag{3.5}$$

Ravnina H se još naziva i prostorom značajki (engl. *feature space*). Na slici 3.3 također su označene udaljenosti d_1 i d_2 koje označavaju udaljenost ravnine H_1 od hiperravnine i ravnine H_2 od hiperravnine, respektivno. Ti parametri bitni su za definiciju margine SVM-a. Naime, ekvidistanca hiperravnine od H_1 i H_2 povlači $d_1 = d_2$. Margina se definira kao udaljenost između najbliže tocke hiperravnine i hiperravnine. Zadatak SVM-a je maksimizirati ovaj parametar.

Ispostavlja se da je margina jednaka $\frac{1}{\|\mathbf{w}\|}$ i maksimizacija toga ekvivalentna je pronalasku:

$$\min\|\mathbf{w}\| \quad \text{tako da je} \quad y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i \tag{3.6}$$

SVM ovaj problem rješava upotrebom Lagrangeovih multiplikatora[2] i pronalaženjem optimalnih parametara $\mathbf{w}$ i b pomoću kojih se definira optimalna orijentacija hiperravnine. Nakon što imamo ove parametre, klasifikacija se svodi na:

$$y' = sgn(\mathbf{w} \cdot \mathbf{x}' + b)$$

3.2.2. Klasifikacija meke margine

Kako bismo mogli provoditi binarnu klasifikaciju podataka koji nisu u potpunosti linearno razdvojivi, potrebno je popustiti ograničenja uvedena u jednadžbama 3.1 i 3.2. Stoga uvodimo nove varijable ξ_i, $i = 1, \ldots L$ (engl. *slack variables*) koje dozvoljavaju „krivu" klasifikaciju i time dobivamo SVM meke margine (engl. *soft margin SVM*). Nove jednadžbe su oblika:

$$\mathbf{x_i} \cdot \mathbf{w} + b \geq +1 - \xi_i \quad \text{za} \quad y_i = +1 \tag{3.7}$$

$$\mathbf{x_i} \cdot \mathbf{w} + b \leq -1 + \xi_i \quad \text{za} \quad y_i = -1 \tag{3.8}$$

$$\xi_i \geq 0 \quad \forall i \tag{3.9}$$

Kombinacijom jednadžbi 3.7 i 3.8 dobivamo:

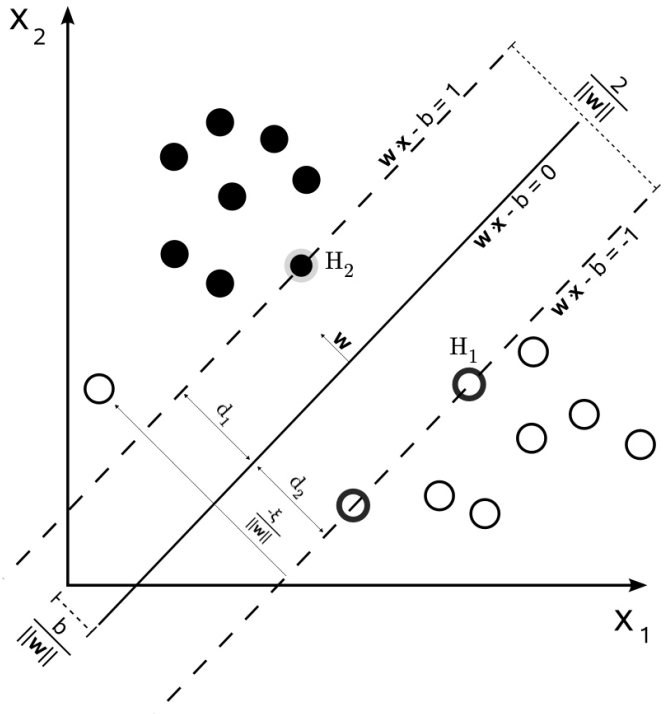
$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0 \quad \text{gdje je} \quad \xi_i \geq 0 \quad \forall i \tag{3.10}$$

Na slici 3.4 može se vidjeti da točke smještene na krivoj strani hiperravnine s obzirom na svoju klasu imaju kaznu (engl. *penalty*) koja raste s udaljenosti od potpornog vektora. Budući da želimo minimizirati broj krivo klasificiranih točaka, zadatak je modificirati jednadžbu 3.6 i pronaći sljedeće:

$$\min\|\mathbf{w}\| + C\sum_{i=1}^{L}\xi_i \quad \text{tako da je} \quad y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0 \quad \forall i \tag{3.11}$$

Ovdje parametar C kontrolira *trade-off* između kazne za krivu klasifikaciju i širine margine. Ponovno se uvode Lagrangeovi multiplikatori i pronalaze optimalni parametri za orijentaciju hiperravnine.

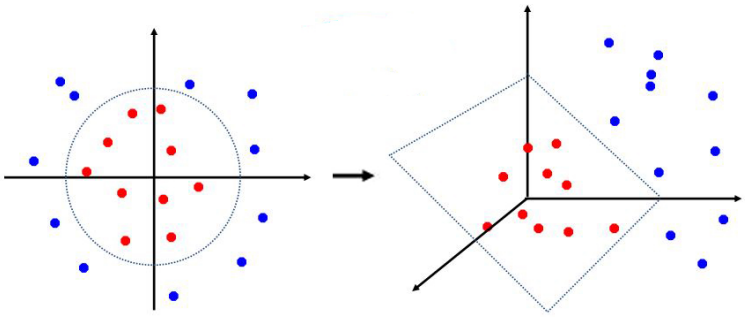
[2]Za više detalja, vidi `http://www.tristanfletcher.co.uk/SVM\%20Explained.pdf`.



Slika 3.4: Hiperravnina kroz dvije klase koje se ne mogu u potpunosti linearno razdvojiti.

3.2.3. Nelinearna klasifikacija

Kada točke ne možemo podijeliti linearno, možemo prijeći u prostor više dimenzije, kako bismo u tom više-dimenzionalnom prostoru opet mogli pribjeći linearnoj klasifikaciji. Primjer pretvorbe nelinearne u linearnu klasifikaciju mijenjanjem broja dimenzija prostora prikazan je na slici 3.5.



Slika 3.5: Primjer povećanja dimenzionalnosti prostora kao rješenje nelinearne klasifikacije.

Za prelazak u više-dimenzionalan prostor koristi nam „jezgreni trik" koji koristi funkcije jezgre (engl. *kernel function*) $k(\mathbf{x_i}, \mathbf{x_j})$. Uobičajene funkcije jezgre prikazane su tablicom 3.3.

Zaključujemo, konačna optimizacija modela svodi se na odabir prikladne funkcije jezgre, parametara funkcije jezgre i parametra meke margine C. Budući da je Gaussova funkcija jezgre čest odabir, najbolja kombinacija parametara C i γ nalazi se uporabom pretrage po rešetci (engl. *grid search*). Pretraga po rešetci je drugo ime za optimizaciju hiperparametara, a radi tako da isprobava sve moguće vrijednosti predanih parametara i vrednuje trenutni model najčešće koristeći krosvalidaciju. Tipične vrijednosti parametara koji se šalju u pretragu po rešetci za stroj potpornih vektora su $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}\}$ i $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{1}, 2^{3}\}$. Za svaku kombinaciju parametara, ispituju se performanse modela koristeći unakrsnu validaciju (engl. *cross-validation*). Unakrsna validacija prvo podijeli skup podataka za testiranje na k dijelova, nad $k-1$ dijelova trenira model, a jedan dio koristi za testiranje. Navedeni postupak ponavlja se k puta i odabiru parametri koji su postigli najbolje rezultate.

Tablica 3.3: Prikaz uobičajenih funkcija jezgri

Ime funkcije	$k(\mathbf{x_i}, \mathbf{x_j})$
Linearna	$(\mathbf{x_i^T x_j})$
Polinomijalna	$(1 + \mathbf{x_i^T x_j})^d$
Gaussov RBF[3]	$\exp(-\gamma\|\mathbf{x_i} - \mathbf{x_j}\|^2), \gamma > 0$
Sigmoid	$\tanh(\gamma \mathbf{x_i^T x_j} + r)$

3.2.4. Višeklasni model

U slučaju kada imamo više od dvije klase u koje želimo klasificirati primjere, koristimo višeklasni model (engl. *multiclass model*). Budući da SVM sam po sebi ne podržava višeklasnu klasifikaciju, taj se problem rješava reduciranjem višeklasnog problema na više binarnih klasifikatora. Novoizgrađeni binarni klasifikatori implementiraju jednu od dvije metode:

1. jedan protiv svih (engl. *one-versus-all*),
2. jedan protiv jednog (engl. *one-versus-one*).

Obje metode detaljno su opisane u radu (Pal, 2008). U modelu razvijenom u ovom radu, koristi se metoda jedan protiv jednog.

[3]Radial Basis Function.

U metodi jedan protiv svih, svaki klasifikator odmjerava se s ostalima, a klasifikacija se određuje temeljem *winner-takes-all* strategije. Drugim riječima, pobjednički klasifikator je onaj s najvećom točnošću i on provodi klasifikaciju. S druge strane, ako primjenjujemo metodu jedan protiv jednog i imamo n klasa, izgradit će se $\frac{n*(n-1)}{2}$ klasifikatora. Ovdje se klasifikacija provodi temeljem *max-wins voting* strategije. To znači da nekoj instanci svaki klasifikator dodijeli klasu i time poveća broj glasova (engl. *vote*) za tu klasu. Naposljetku je instanca smještena u klasu koja je dobila najveći broj glasova.

4. Implementacija modela

Model razvijen u ovom radu napisan je u programskom jeziku Python, verzija 2.7.9. Priručnik za upotrebu može se naći u radu (Rossum, 1995). Uz Python, za sitne manipulacije datotekama korišten je i skriptni jezik Bash.[1] Svi izvorni kodovi nalaze se u direktoriju `src`, dok se svi ulazni podaci, korišteni leksikoni i resursi nalaze u direktoriju `data`. U direktoriju `upute` mogu se naći upute za označavanje, koje se također mogu naći u dodatku A priloženom na kraju rada. Također, u direktoriju `annotators` nalaze se pripremljeni podaci i alati za označivače.

4.1. Web crawler

Prvi korak u razvoju modela bio je pripremiti ulazne podatke (engl. *dataset*). To je napravljeno *crawlanjem* hrvatskog *gaming* portala HCL.hr. Izvorni kod web crawlera nalazi se u datoteci `crawler.py`. Za dohvaćanje HTML-a putem URL-a korištena je knjižnica `urllib`[2], a za parsanje HTML-a u tekst korištena je knjižnica `Beautiful Soup`[3].

4.2. Alat za označavanje

Korišten je jednostavan alat za označavanje imena MAE (Multi-purpose Annotation Environment)[4]. Alat je razvijen i opisan u radu (Stubbs, 2011). Pisan je u programskom jeziku Java, a dozvoljava korisnicima da definiraju vlastite sheme označavanja. Izlazni format je *stand-off* XML (engl. *Extensible Markup Language*) formata. *Stand-off* označava da se anotacije spremaju odvojeno od

[1] `http://www.gnu.org/software/bash/`.

[2] `https://docs.python.org/2/library/urllib.html`.

[3] `http://www.crummy.com/software/BeautifulSoup/`.

[4] `https://code.google.com/p/mae-annotation/`.

samih podataka koji su označavani. Suprotnost bi bilo *inline* označavanje, gdje su oznake i početni tekst spremljeni na istom mjestu.

Shema označavanja regulirana je `scheme.dtd` datotekom koja se mora učitati prije početka označavanja. DTD je skraćenica od *Document Type Definition*, i opisnik je XML formata. U ovom modelu, u `scheme.dtd` datoteci zapisane su moguće klase koje označivači mogu pridijeliti segmentu teksta. Sadržaj te datoteke prikazan je isječkom 4.1.

Isječak 4.1: Sadržaj datoteke `scheme.dtd`

```
<!ENTITY name "SentimentAnnotation">
<!ELEMENT POSITIVE (#PCDATA)>
<!ELEMENT NEGATIVE (#PCDATA)>
```

Deklarirane su dvije moguće opcije označavanja – `POSITIVE` i `NEGATIVE`. Oznaka `PCDATA` označava *parsed character data* i taj će dio parser obraditi i po potrebi pretvoriti u stablastu strukturu ako sadrži unutarnje oznake. Suprotnost je oznaka `<![CDATA[]]>` i tekst unutar te oznake tretirat će se kao običan tekst, zato što `CDATA` znaci *Unparsed Character Data*. Upravo je unutar tih oznaka pohranjen tekst recenzije. Isječkom 4.2 prikazan je primjer *stand-off* XML-a nastao označavanjem recenzije *Lego Batman 3: Beyond Gotham*.

Isječak 4.2: Primjer *stand-off* XML datoteke

```
<?xml version="1.0" encoding="UTF-8" ?>
<SentimentAnnotation>
<TEXT><![CDATA[
// ovdje se nalazi tekst recenzije}
]]></TEXT>
<TAGS>
<POSITIVE id="P3" start="2198" end="2288" text="Ukupno 150 igrivih
    likova cini ove Legace pravim malim rajem za sve ljubitelje DC
    stripova" />
<POSITIVE id="P11" start="5327" end="5404" text="nece moci
   odoljeti ovoj sarenoj drogici i adiktivnom zvuku skupljanja
   kockica" />
<NEGATIVE id="N1" start="1139" end="1205" text="Nazalost i dalje
   ne postoji mogucnost kooperativnog igranja online" />
<NEGATIVE id="N9" start="4953" end="5039" text="oduzeli su nam i
```

```
    otvoreni svijet koji je savrseno funkcionirao u prethodnim
    nastavcima" />
</TAGS>
</SentimentAnnotation>
```

Oznake `POSITIVE` i `NEGATIVE` označavaju polaritet segmenta teksta, a oznaka `id` predstavlja identitet oznake. Identitet oznake počinje s `P` ako je segment pozitivan, a s `N` ako je segment negativan i slijedi redni broj označenog segmenta u dokumentu. Oznake `start` i `end` označavaju indekse početnog i završnog znaka u dokumentu, a oznaka `text` sadrži segment označenog teksta.

Nakon što su označivači označili svoj dio, podaci su prikupljeni i smješteni u direktorij `data/oznaceno_sve`.

4.3. Alati za obradu podataka

Što se tiče gotovih alata za obradu podataka, u direktoriju `output-pre` nalazi se rezultat alata za segmentaciju rečenica. Ovisnosni parser koristi te datoteke kako bi reprezentirao svaku rečenicu u ConLL formatu i spremio ih u direktorij `output-dp`.

Prije pripreme podataka za vektorizaciju, bilo je potrebno poravnavati *stand-off* XML datoteke nastale označavanjem s izlazom ovisnosnog parsera koji se koristio kasnije u modelu. Modul koji to obavlja zove se `align.py`, a čita podatke iz direktorija `data/oznaceno_sve` i ispisuje ih u direktorij `data/aligned`.

Ostali alati za obradu podataka nalaze se u modulima pod nazivom `tool1.py` do `tool5.py` i oni pripremaju tekst za vektorizaciju.

4.4. Alati za vektorizaciju podataka

Vektorizacija podataka podrazumijeva implementaciju modela *bag of words*. Stoga modul `dict.py` čita leme svih rijeci iz direktorija `data/output-dp` i gradi rječnik jedinstvenih zapisa koji sprema u datoteku `data/dict.txt`.

Budući da se model sastoji od tri zadatka, razvijena su tri modula za vektorizaciju teksta. Imena modula su `features1.py`, `features2.py` i `features3.py`.

4.5. Klasifikatori

Prije samih klasifikatora, napisana je Bash skripta `generate_train_test.sh` koja automatski dijeli čitav ulazni skup podataka na skup za učenje i skup za testiranje u omjeru 70 : 30.

Modul koji vrši klasifikaciju je `classify.py` i prima jedan argument komandne linije – broj zadatka koji želimo izvršiti – 1, 2 ili 3. Za implementaciju klasifikatora korištena je knjižnica `scikit-learn`[5], opisana u radu (Pedregosa et al., 2011). Za baratanje vektorima značajki korištena je knjižnica `numpy`[6], opisana u radu (Van Der Walt et al., 2011). Za serijalizaciju objekata korištena je knjižnica `cPickle`[7].

Za sva tri zadatka modela, isprobano je više klasifikatora kako bi se njihovi rezultati mogli uspoređivati s onim ciljnim, SVM klasifikatorom. Isprobani su sljedeći klasifikatori:

1. klasifikator temeljen na apriornom sentimentu
2. naivni Bayes, (engl. *naive Bayes*)
3. klasifikator većinske klase (engl. *majority class classifier*)
4. metoda potpornih vektora (engl. *support vector machines, SVM*)

4.5.1. Klasifikator temeljen na apriornom sentimentu

Klasifikator temeljen na apriornom sentimentu specifičan je za svaki od triju zadataka i on označava podatke onako kako bi to čovjek to radio, bez ikakvog strojnog učenja, samo na temelju apriornog sentimenta. Za prvi zadatak, klasifikaciju subjektivnosti rečenica, ovaj klasifikator gleda udio pozitivnih i negativnih riječi u receniči. Ako je ukupan udio pozitivnih i negativnih riječi u rečenici veći od proizvoljnog praga, rečenicu proglašava subjektivnom. U suprotnom, rečenica je objektivna.

U drugom zadatku, klasifikaciji sentimenta subjektivnih rečenica, klasifikator mora odrediti je li rečenica pozitivna, negativna ili miješana. To određuje tako da opet gleda udio pozitivnih i negativnih riječi u rečenici. Ako je udio pozitivnih riječi jednak udjelu negativnih riječi te ako je zbrojeni udio tih riječi veći od

[5] `http://scikit-learn.org/stable/`.

[6] `http://www.numpy.org/`.

[7] `https://docs.python.org/2/library/pickle.html`.

proizvoljnog praga, rečenicu proglašava miješanom. U suprotnom, jednostavno gleda kojih riječi ima više, pozitivnih ili negativnih, te klasificira rečenicu kao pozitivnu ili negativnu.

U trećem zadatku, klasifikaciji polariteta izraza, klasifikator jednostavno uspoređuje udjele pozitivnih i negativnih riječi u izrazu. Ako je udio pozitivnih rijeci veći, izraz je pozitivan. U suprotnom je negativan.

4.5.2. Ostali klasifikatori

Ostali klasifikatori su naivni Bayes, većinski klasifikator i SVM klasifikator. Oni se pozivaju jednako, neovisno o zadatku.

Naivni Bayes radi s pretpostavkom da sve značajke jednako doprinose klasifikaciji, što znači da potpuno zanemaruje bilo kakve korelacije izmedu značajki. Ovakav jednostavan pristup pokazuje visoke rezultate upravo kada se primjenjuje kategorizaciju teksta. Najprikladnija vrsta Bayesovog klasifikatora za ovaj slučaj je Multinomial Naive Bayes, korišten u isječku 4.3.

Isječak 4.3: Pozivanje naivnog Bayesa

```
import numpy
from sklearn.datasets import load_svmlight_file
from sklearn.naive_bayes import MultinomialNB

X_train, y_train = load_svmlight_file(trainfile)
X_test, y_test = load_svmlight_file(testfile)

print "=== Naive Bayes ==="
clf = MultinomialNB()
clf.fit(X_train.toarray(), y_train)
print "Baseline: " + str(clf.score(X_test.toarray(), y_test))
```

Klasifikator većinske klase (engl. *majority class classifier*) u skupu podataka za treniranje nađe najzastupljeniju klasu i u skupu za testiranje jednostavno sve proglasi tom klasom. Primjer je prikazan isječkom 4.4.

Isječak 4.4: Pozivanje većinskog klasifikatora

```
import numpy
from sklearn.datasets import load_svmlight_file
from sklearn.dummy import DummyClassifier
```

```
X_train, y_train = load_svmlight_file(trainfile)
X_test, y_test = load_svmlight_file(testfile)

print "=== Majority Class Classifier ==="
clf = DummyClassifier(strategy='most_frequent')
clf.fit(X_train.toarray(), y_train)
print "Baseline: " + str(clf.score(X_test.toarray(), y_test))
```

Naposljetku, klasifikator u koji se polaze najviše nade je metoda potpornih vektora. Klasifikator je treniran nad skupom za učenje i isprobavao je različite vrijednosti parametara C i γ koristeći pretragu po rešetci i unakrsnu validaciju. Detaljnije o važnosti tih parametara opisano je u poglavlju 3.2. Način pozivanja i izbor mogućih kombinacija parametara prikazani su isječkom 4.5.

Isječak 4.5: Pozivanje SVM klasifikatora

```
import numpy
from sklearn import svm, metrics, grid_search
from sklearn.datasets import load_svmlight_file
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

X_train, y_train = load_svmlight_file(trainfile)
X_test, y_test = load_svmlight_file(testfile)

print "=== Using Grid Search ==="
Cs = [2 ** val for val in np.linspace(-5, 15, num = 10)]
gs = [2 ** val for val in np.linspace(-15, 3, num = 9)]
parameters = {'svc__kernel':['linear', 'rbf'], 'svc__C':Cs, '
    svc__gamma':gs}

estimators = [('normalize', StandardScaler(with_mean = False)), ('
    svc', svm.SVC())]
pip = Pipeline(estimators)
clf = grid_search.GridSearchCV(pip, parameters, verbose=1, cv=5,
    n_jobs=-1)
clf.fit(X_train, y_train)
```

```
clf.best_estimator_.fit(X_train, y_train)
print "Score: " + str(clf.best_estimator_.score(X_test, y_test))
```

Četvrti i posljednji zadatak modela je cjevovod (engl. *pipeline*) prvog i drugog klasifikatora. To je izvedeno modulom `classify.py`. Naime, oba klasifikatora nakon učenja zapisuju optimalne parametre u serijaliziranom obliku u datoteke `data/svm/params1.txt` i `data/svm/params2.txt`. Modul `classify.py` čita te parametre i pokreće prvi klasifikator, zatim izlaz prvog klasifikatora dovodi kao ulaz drugom klasifikatoru, te pokreće drugi klasifikator.

Na samom kraju, čitav model sveo se na jednostavnu Bash skriptu `zr.sh` opisanu isječkom 4.6.

Isječak 4.6: Skripta koja pokreće model

```
python features1.py
echo "Done with features1!"
sh ./generate_train_test.sh 1 3099
python classify.py 1 > results1.txt
echo "Done classifying task 1!"

python features2.py
echo "Done with features2!"
sh ./generate_train_test.sh 2 1096
python classify.py 2 > results2.txt
echo "Done classifying task 2!"

python tool6.py
python features3.py
echo "Done with features3!"
sh ./generate_train_test.sh 3 883
python classify.py 3 > results3.txt
echo "Done classifying task 3!"

sh ./generate_train_test.sh 4 3099
python pipeline.py > resultsp.txt
echo "Done classifying pipeline!"
```

5. Evaluacija

5.1. Skup podataka

Kao skup podataka (engl. *dataset*) korištene su recenzije igara s hrvatskog *gaming* portala HCL.hr[1]. *Crawlani* su tekstovi iz sljedećih kategorija *PC igre*, *Trash igre*, *Igre za konzole*, *Igre za mobitele* i *Retro igre*. Ukupno je *crawlano* 4427 rečenica iz 95 tekstova.

Dani skup tekstnih podataka ručno je označen kako bi se mogao koristiti za iscrpno ispitivanje modela. Nakon predstavljanja detaljnih uputa namijenjenih označivačima i razjašnjenih upita, postignut je standard označavanja kojeg su se pridržavali svi označivači. Upute za označavanje mogu se naći u dodatku A. Ručnim označavanjem izgrađen je korpus pogodan za bilo koji od definiranih zadataka klasifikacije.

Tablica 5.1: Ukupni skup podataka

Broj tekstova	Broj rečenica	Broj odsječaka	Prosječna duljina odsječka
95	4427	1566	9 riječi

Označivači su odabirom teksta definirali odsječke koji im se čine subjektivnima i označili mu polaritet (pozitivno ili negativno). Svaku recenziju označila su tri proizvoljno odabrana označivača. Zlatni standard (engl. *gold standard*) definiran je uzevši dvotrećinsku većinu označenih riječi u tekstu. Drugim riječima, ako su 2 od 3 označivača proglasila riječ dijelom odsječka i ako se označeni polariteti odsječka podudaraju, riječ je dio rezultirajućeg odsječka. Jedan primjer prikazan je na slici 5.1.

Prema zlatnom standardu, rezultirajući odsječci sami po sebi predstavljaju ulaz za klasifikator polariteta izraza, dok su rečenice subjektivne ako se unutar

[1] `http://www.hcl.hr/recenzije.php`.

O_1	Pa... simpatična i crtana, no ništa spektakularno.
O_2	Pa... simpatična i crtana, no ništa spektakularno.
O_3	Pa... simpatična i crtana, no ništa spektakularno.
R	Pa... simpatična i crtana, no ništa spektakularno.

Slika 5.1: Rezultirajući odsječak dobiven iz tri različito označena odsječka istog polariteta

njih nalazi barem jedan rezultirajući odsječak (polaritet nebitan), a objektivne su ako ne sadrže niti jedan označeni odsječak.

Nakon definiranja zlatnog standarda, potrebno je reprezentirati korpus u pogodnijem obliku za klasifikaciju i ekstrahiranje značajki.

Pogodna reprezentacija podataka za ovaj zadatak je CoNLL format[2]. Detaljno objašnjenje CoNLL formata prikazano je tablicom 5.2.

Tablica 5.2: Objašnjenje vrijednosti pojedinih stupaca CoNLL formata

#	Oznaka	Opis vrijednosti polja
1	ID	Broj tokena u rečenici, počevši od 1.
2	FORM	Oblik riječi ili interpunkcija.
3	LEMMA	Lema riječi, ako postoji. Ako ne, donja crta.
4	CPOSTAG	*Coarse-grained* vrsta riječi, ovisna o jeziku.
5	POSTAG	*Fine-grained* vrsta riječi, ovisna o jeziku. Identična prethodnoj vrijednosti ako ne postoji.
6	FEATS	Lista sintaktičkih i/li morfoloških značajki odvojenih znakom | ili donja crta ako ne postoji.
7	HEAD	Korijen trenutnog tokena, ili vrijednost polja ID ili 0.
8	DEPREL	Relacija ovisnosti o polju HEAD.
9	PHEAD	Projektivni korijen trenutnog tokena, ili vrijednost polja ID ili 0 ili donja crta ako nije primjenjivo.
10	PDEPREL	Relacija ovisnosti o polju PHEAD.

Budući da su CPOSTAG i POSTAG (4. i 5. stupac) ovisne o jeziku, u tablici 5.3 prikazana je veza između vrste riječi i njezine oznake.

[2] `http://ilk.uvt.nl/conll/#dataformat`.

Tablica 5.3: Veza između vrste riječi i CoNLL oznake

Vrsta riječi	CoNLL oznaka	Vrsta riječi	CoNLL oznaka
imenica	N	veznik	C
glagol	V	broj	M
pridjev	A	čestica	Q
zamjenica	P	usklik	I
prilog	R	kratica	Y
prijedlog	S	ostatak	X

Prikaz jedne rečenice iz skupa podataka u ConLL formatu dan je na slici 5.2. Može se vidjeti da je CoNLL format blago modificiran i sadrži 11. stupac. Vrijednost tog stupca može biti BPOS, BNEG, EPOS, ENEG ili ništa. Ako je vrijednost BPOS ili BNEG, to znači da označeni odsječak nekog polariteta počinje na toj riječi, a EPOS ili ENEG označava da je ta riječ kraj označenog odsječka. U primjeru na slici 5.2 označeni odsječak obuhvaća čitavu rečenicu.

1	Jedina	jedini	A	A	_	3	Atr	_ _	BPOS
2	simpatična	simpatičan	A	A	_	3	Atr	_ _	
3	stvar	stvar	N	N	_	6	Sb	_ _	
4	na	na	S	S	_	3	Prep	_ _	
5	igri	igra	N	N	_	4	Adv	_ _	
6	je	biti	V	V	_	0	Pred	_ _	
7	muzika	muzik	N	N	_	6	Pnom	_ _	
8	u	u	S	S	_	6	Prep	_ _	
9	glavnom	glavni	A	A	_	10	Atr	_ _	
10	izborniku	izbornik	N	N	_	8	Atr	_ _	EPOS

Slika 5.2: Prikaz rečenice u CoNLL formatu

Kako bismo od obične rečenice početnog teksta došli do CoNLL formata, potrebno je prvo segmentirati rečenice i tokenizirati riječi. Za to je korišten alat za pretprocesiranje koji je razvijen u sklopu TakeLaba[3]. Nadalje, potrebno je provesti označavanje vrste riječi (engl. *part-of-speech, POS*) i lematizaciju, a za to je korišten model iz rada (Agić et al., 2013) koji interno koristi CST[4] za lematizaciju i HunPos[5] za označavanje vrste riječi. Krajnji je korak obaviti *dependency parsing*, za što se pobrinuo alat opisan u radu (Agić i Merkler, 2013) temeljen na parseru MST[6].

[3]Text Analysis and Knowledge Engineering Lab, `http://takelab.fer.hr/`.

[4]`http://cst.dk/online/lemmatiser/uk/`.

[5]`https://code.google.com/p/hunpos/`.

[6]`http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html`.

5.2. Suglasnost označivača

Za početak, promotrimo tablicu 5.1 koja pruža informacije o ukupnim podacima.

Tablica 5.4 prikazuje koliko je ukupno riječi označeno (unija sve troje označivača po dokumentu) i koliko je ukupno riječi označeno zlatnim standardom (uzimajući dvotrećinsku većinu).

Tablica 5.4: Prikaz ukupno označenih podataka

Zlatni standard	Ukupno označeno
14205 riječi	29081 riječi

Najviše označenih riječi po zlatnom standardu je 531 u recenziji *The Order 1886* i to čini 28.2% označenih riječi u tekstu. S druge strane, recenzija *Shadowgun [iOS]* nema niti jedne označene riječi po zlatnom standardu.

5.3. Evaluacijske mjere

Osnovne mjere u evaluaciji klasifikatora su točnost (engl. *accuracy*), preciznost (engl. *precision*) i odziv (engl. *recall*). Uz to, koristi se F-mjera[7] koja predstavlja harmonijsku sredinu preciznosti i odziva. Također, razlikujemo mikro- i makro-F mjeru. Razlika je u tome što makro-F_1 sve klase tretira kao da su jednako zastupljene. Zbog toga primjeri iz malih klasa imaju veći utjecaj na mjeru nego što bi imali kod mikro-F1, a razlika je vidljiva kod neuravnoteženih skupova.

Ove mjere dobivene su s obzirom na broj TP (engl. *true positive*), TN (engl. *true negative*), FP (engl. *false positive*) i FN (engl. *false negative*) primjera. TP predstavlja primjer kojeg je klasifikator ispravno svrstao u klasu. TN predstavlja primjer za kojeg je klasifikator ispravno rekao da podatak ne pripada nekoj klasi. FP predstavlja primjer kojeg je klasifikator smjestio u krivu klasu, dok FN predstavlja primjer za koji je klasifikator krivo rekao da ne pripada nekoj klasi.

Ukupan broj primjera definira se kao TP + TN + FP + FN. Broj točno klasificiranih primjera je TP + TN, dok TP + FN označava broj pozitivnih primjera, a TN + FP broj negativnih primjera.

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

[7] `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`.

skupu svih primjera.

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

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

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

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

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

F1-mjera je harmonijska sredina preciznosti i odziva.

$$F_1 = \frac{2}{(1/P) + (1/R)} = \frac{2PR}{P + R}$$

Kada govorimo o višeklasnoj klasifikaciji, uvodimo pojam makro-uprosječene ili jednostavno makro-F1 mjere. Ova mjera tretira sve klase kao da su jednako zastupljene.

$$F^{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} F_i$$

S druge strane, mikro-uprosječena F1 mjera uzima u obzir zastupljenost klasa i dobiva se tako da se posumiraju TP, FP i FN po svim klasama. Konačno, dobivamo formulu:

$$F^{\text{micro}} = \frac{2TP}{2TP + FP + FN}$$

Navedene mjere izračunavaju se na slučajnom uzorku. U ovom radu korištena je metoda izdvajanja (engl. *holdout method*) gdje se ulazni podaci dijele na skup za učenje i skup za testiranje, a evaluacija se provodi isključivo nad skupom za testiranje.

5.4. Klasifikacija subjektivnosti rečenica

Ulazni skup podataka prvog klasifikatora prikazan je tablicom 5.5, a točnosti pojedinih klasifikatora prikazane su tablicom 5.6.

U ovom zadatku, naivni Bayes postigao je prilično visok rezultat, iako ne viši od SVM-a. Moguće da je to zato što naivni Bayes pretpostavlja da nema nikakve korelacije izmedu značajki i da je svaka jednako vrijedna.

Tablica 5.5: Skup podataka prvog klasifikatora

Ukupno rečenica	Objektivnih rečenica	Subjektivnih rečenica
4427	3165 (71.5%)	1262 (28.5%)

Tablica 5.6: Točnosti klasifikatora u prvom zadatku

Klasifikator apriornog sentimenta	Naivni Bayes	Većinska klasa	SVM
70.5%	73.3%	71.5%	74.4%

Tablica 5.7: Mjere i optimalni parametri SVM klasifikatora u prvom zadatku

Točnost	Preciznost	Odziv	F1-mjera	Jezgra	C	γ
74.4%	58.8%	33.6%	42.8%	rbf	27.8	5.078e-05

Detaljnije mjere SVM klasifikatora prikazane su tablicom 5.7, kao i optimalni parametri.

Možemo vidjeti da je odziv dosta malen naspram točnosti, što bi se moglo objasniti činjenicom da je početni skup podataka neuravnotežen. Upravo zbog tog razloga i većinski klasifikator postiže visok rezultat. Kao pozitivna klasa izabrana je klasa subjektivnih rečenica. Gledajući relativno loš odziv, možemo zaključiti da klasifikator bolje klasificira objektivne rečenice nego subjektivne.

5.5. Klasifikacija sentimenta rečenica

Zadatak drugog klasifikatora jest odrediti u koju od tri klase spada subjektivna rečenica – pozitivnu, negativnu ili miješanu. Ulazni skup podataka drugog klasifikatora prikazan je tablicom 5.8. Vidljivo je da je udio negativnih rečenica gotovo pa jednak udjelu pozitivnih rečenica, tako da su te klase ujednačene, dok je miješanih rečenica samo 68.

Tablica 5.8: Skup podataka drugog klasifikatora

Ukupno rečenica	Pozitivnih rečenica	Negativnih rečenica	Miješanih rečenica
1262	667 (52.9%)	527 (41.8%)	68 (5.3%)

Točnosti pojedinih klasifikatora prikazane su tablicom 5.9, a detaljnije mjere i optimalni parametri SVM klasifikatora tablicom 5.10.

Prema matrici zabune prikazanoj tablicom 5.11 možemo vidjeti da klasifikator

Tablica 5.9: Točnosti klasifikatora u drugom zadatku

Klasifikator apriornog sentimenta	Naivni Bayes	Većinska klasa	SVM
47.1%	72.3%	61.7%	64.1%

Tablica 5.10: Mjere SVM klasifikatora u drugom zadatku

Točnost	Preciznost	Odziv	Makro-F1	Mikro-F1	Jezgra	C	γ
63.1%	44.8%	35.9%	35.2%	64.1%	rbf	15.8	5.078e-05

dobro klasificira negativne i pozitivne rečenice, iako ih ponekad miješa. Klasifikator ima problema s miješanim rečenicama. Moguće da je to tako zbog toga što miješanih rečenica ima vrlo malo općenito, pa model nije naučen na njih.

Tablica 5.11: Matrica zabune klasifikatora sentimenta rečenica

		Zlatni standard			
		Negativna	Pozitivna	Miješana	Ukupno
Predviđeno	Negativna	98	31	0	129
	Pozitivna	89	144	1	234
	Miješana	6	9	1	16
	Ukupno	205	164	0	363

5.6. Klasifikacija sentimenta označenih izraza

Ulazni skup podataka trećeg klasifikatora prikazan je tablicom 5.12. Možemo vidjeti da su klase uravnotežene.

Tablica 5.12: Skup podataka trećeg klasifikatora

Ukupno izraza	Pozitivnih izraza	Negativnih izraza
1566	822 (52.5%)	744 (47.5%)

Točnosti pojedinih klasifikatora prikazane su tablicom 5.13. Klasifikator apriornog sentimenta pokazuje najgore rezultate, dok je SVM klasifikator znatno bolji od svih ostalih.

Detaljnije mjere i optimalni parametri SVM klasifikatora prikazani su tablicom 5.14. Kao pozitivna klasa odabrana je klasa pozitivnih izraza.

Tablica 5.13: Točnosti klasifikatora u trećem zadatku

Klasifikator apriornog sentimenta	Naivni Bayes	Većinska klasa	SVM
45.3%	63%	52.1%	69.8%

Tablica 5.14: Mjere i optimalni parametri SVM klasifikatora u trećem zadatku

Točnost	Preciznost	Odziv	F1-mjera	Jezgra	C	γ
69.8%	73.7%	65.3%	69.3%	rbf	8.96	3.051e-05

5.7. Cjevovod klasifikatora

U ovom zadatku prvo dajemo prvom klasifikatoru da odredi koje rečenice su subjektivne, zatim te subjektivne rečenice šaljemo na ulaz drugog klasifikatora čija je zadaća odrediti je li rečenica pozitivna, negativna ili miješana. Objektivne rečenice ne idu dalje u proces.

Ulaz u prvi klasifikator opisan je tablicom 5.5. Izlaz prvog klasifikatora prikazan je tablicom 5.15.

Tablica 5.15: Izlaz iz prvog klasifikatora

Objektivnih rečenica	Subjektivnih rečenica
3343 (75.5%)	1084 (24.5%)

Drugi klasifikator zatim mora klasificirati detektiranih 1084 subjektivnih rečenica u tri različite klase koristeći SVM klasifikator. Uspješnost tog zadatka prikazan je tablicom 5.16.

Tablica 5.16: Mjere SVM klasifikatora u *pipelineu*

Makro točnost	Makro preciznost	Makro odziv	Makro-F1	Mikro-F1
64.2%	25.2%	24.6%	24.3%	64.2%

U matrici zabune (engl. *confusion matrix*) prikazanoj tablicom 5.17 možemo vidjeti da model zna dobro klasificirati objektivne rečenice, ali ima problema s pozitivnim i negativnim rečenicama, što je zadatak drugog klasifikatora. Miješanih rečenica ima jako malo u odnosu na čitav ulazni skup podataka, tako da klasifikator ni njih ne zna smjestiti u odgovarajuću klasu. Budući da je prvi klasifikator u svom zadatku postigao bolje rezultate od drugog klasifikatora, valjana

je pretpostavka da je drugi klasifikator više pridonio lošem ukupnom rezultatu cjevovoda klasifikatora.

Tablica 5.17: Matrica zabune cjevovoda klasifikatora

		Zlatni standard				
		Objektivna	Pozitivna	Negativna	Miješana	Ukupno
Predviđeno	Objektivna	824	70	75	9	978
	Pozitivna	170	25	21	1	217
	Negativna	105	10	3	0	118
	Miješana	13	0	2	0	15
	Ukupno	1112	105	101	10	1328

5.8. Diskusija rezultata i analiza pogrešaka

Možemo vidjeti da je model dobro naučio klasifikaciju subjektivnosti rečenica i klasifikaciju sentimenta označenih izraza, dok postiže lošije rezultate u klasifikaciji sentimenta čitavih rečenica. Pokazalo se da naivni Bayes, kao naizgled jednostavan klasifikator, vrlo dobro konkurira mnogo složenijim modelima kada je u pitanju odgovarajući skup podataka. Unatoč tome, metoda potpornih vektora i dalje je najbolji odabir za ovakav problem. Uparena s pretragom po rešetci (engl. *grid searchom*) i unakrsnom validacijom (engl. *cross-validation*), model može vrlo dobro učiti i iz dosta malog skupa podataka.

Budući da cjevovod koristi prvi, zatim drugi klasifikator, potrebno je poboljšati klasifikator subjektivnih rečenica kako bismo poboljšali i čitav cjevovod. Cjevovod klasifikatora može biti primjenjiv na čitave dokumente, tako da na ulaz dobije tekst dokumenta, a na izlazu se pojave pozitivne i negativne rečenice. Ova primjena posebno je korisna za automatsku analizu recenzija bilo kakve vrste. Upravo zbog toga odabrane su recenzije igara kao početni skup podataka.

Nadalje, moguća nadogradnja modela bila bi automatsko detektiranje početka i kraja odsječka koji predajemo klasifikatoru sentimenta izraza. Zasada je to napravljeno ručno, tako da su označivači odabirali odsječak subjektivnog teksta koji predstavlja neki izraz. Na taj način, zadatak bi bio zaokružen tako da bi izlaz cjevovoda bio ulaz u model automatskog pronalaska odsječaka, zatim bi to bilo predano klasifikatoru sentimenta izraza i na kraju bismo dobili pozitivne i negativne izraze koji čine polazni dokument.

Što se tiče pogrešaka, do propusta je moglo doći u bilo kojem koraku analize sentimenta. Za početak, lako je moguće da označivači nisu imali potpuno iste kriterije označavanja i time se već gubi informacija bitna za daljnje korake. Zatim, korišteni ovisnosni parser i alat za lematizaciju nisu savršeni i došlo je do krivih podataka, a greška se samo propagirala u budućim koracima. Na kraju, moguće je da značajke klasifikatora nisu dovoljno pomno odabrane i tu uvijek ima mjesta za nadogradnju. Primarna nadogradnja u ovom koraku bila bi uvođenje značajki povezanih sa sintaksom jezika.

5.9. Usporedba rezultata s referentnim modelom za engleski jezik

Referentni model za engleski jezik (Wilson et al., 2005) radi nad znatno većim skupom podataka. Taj model razvijen je nad leksikonom od 15991 izraza iz 425 dokumenata. Od ukupno 8984 rečenica, 28% je objektivno, a 72% je subjektivno, što je značajna razlika u odnosu na model razvijen u ovom radu, gdje je većina rečenica objektivna.

Uspoređujući klasifikatore subjektivnosti rečenica, model za engleski jezik koristi 28 značajki i postiže točnost od 75.9%, što je vrlo slično modelu za hrvatski jezik (74.4%). Kada gledamo klasifikator sentimenta rečenica, model za engleski jezik koristi 10 značajki i postiže točnost od 65.7%, dok model za hrvatski jezik postiže točnost od 64.1%.

Možemo vidjeti da su rezultati modela za hrvatski jezik zadovoljavajući u odnosu na model za engleski jezik, iako model za hrvatski jezik radi nad dvostruko manje rečenica i deset puta manje izraza, a ni nema značajke koje uzimaju u obzir sintaksna svojstva jezika, dok model za engleski jezik to ima.

6. Zaključak

Analiza sentimenta netrivijalan je zadatak i čest je problem u obradi prirodnog jezika. Analiza sentimenta sastoji se od mnogo koraka koji većinom uključuju spretno baratanje podacima i reprezentaciju teksta u što pogodnijem obliku za analizu i učenje modela, kao i odabir prikladnog modela.

U ovom radu razvijen je model za kontekstno ovisnu analizu sentimenta izraza hrvatskog jezika, što se pokazalo kao izazovan posao. Potrebno je označiti početni korpus, pripaziti na semantičke i sintaktičke osobitosti jezika i sukladno tome osmisliti značajke. Konačan ishod je mogućnost klasificiranja rečenice u subjektivne i objektivne, a subjektivne detaljnije kao pozitivne, negativne ili miješane, kao i klasifikacija samih odsječaka teksta u pozitivne i negativne.

Budući da je u svakom zadatku naučeni SVM klasifikator postigao veću točnost od naivnih rješenja, rezultati su zadovoljavajući. Također, uspoređujući s referentnim modelom za engleski jezik opisanim u radu (Wilson et al., 2005), dobivaju se slični rezultati, iako je model za engleski jezik učen na više od dvostruko većem skupu podataka.

Buduća nadogradnja bila bi svakako iskorištavanje sintaktičkih značajki hrvatskog jezika, jer su sadašnje značajke jedva zagrebale površinu. Također, ne može škoditi i povećanje označenog korpusa, kao i veći broj korištenih resursa poput leksikona sentimenta. Možda najbitnija nadogradnja bila bi automatsko prepoznavanje početka i kraja odsječka izraza kojem se klasificira sentiment.

Na kraju, pokazalo se da je razvijeni model za hrvatski jezik zadovoljavajuć i može se primjenjivati u analizi sentimenta teksta već sada, iako ima mjesta za nadogradnju.

Literatura

Agić i Merkler. Three syntactic formalisms for data-driven dependency parsing of croatian. U *Text, Speech and Dialogue. Lecture Notes in Computer Science*, stranice 560—567. Association for Computational Linguistics, 2013.

Agić, Ljubešić, i Merkler. Lemmatization and morphosyntactic tagging of croatian and serbian. U *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, stranice 48–57. Association for Computational Linguistics, 2013.

Siniša Biđin, Goran Glavaš, i Jan Šnajder. Predicting croatian phrase sentiment using a deep matrix-vector model. U *Proceedings of the 9th Language Technologies Conference 2014*. Information Society, 2014.

Goran Glavaš, Jan Šnajder, i Bojana Dalbelo Bašić. Experiments on hybrid corpus-based sentiment lexicon acquisition. U *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, stranice 1–9. Association for Computational Linguistics, 2012.

Mahesh Pal. Multiclass approaches for support vector machine based land cover classification. *arXiv preprint arXiv:0802.2411*, 2008.

Bo Pang i Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. U *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, stranica 271. Association for Computational Linguistics, 2004.

Bo Pang i Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

Bo Pang, Lillian Lee, i Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. U *Proceedings of the ACL-*

02 conference on Empirical methods in natural language processing-Volume 10, stranice 79–86. Association for Computational Linguistics, 2002.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, i E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

Amber Stubbs. Mae and mai: lightweight annotation and adjudication tools. U *Proceedings of the 5th Linguistic Annotation Workshop*, stranice 129–133. Association for Computational Linguistics, 2011.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, i Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37 (2):267–307, 2011.

Alan M Turing. Computing machinery and intelligence. *Mind*, stranice 433–460, 1950.

Stefan Van Der Walt, S Chris Colbert, i Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Theresa Wilson, Janyce Wiebe, i Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. U *Proceedings of the conference on human language technology and empirical methods in natural language processing*, stranice 347–354. Association for Computational Linguistics, 2005.

Theresa Wilson, Janyce Wiebe, i Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433, 2009.

Dodatak A
Upute za označavanje

A.1. Motivacija

Porastom raspoloživih količina korisnički generiranog sadržaja povećalo se zanimanje za strojnom analizom sentimenta, kojom se utvrđuje je li tekst usmjeren pozitivno ili negativno. Uobičajeni postupci analize sentimenta temelje se na leksikonima apriornog sentimenta, koji svakoj riječi pridružuju oznaku sentimenta. Međutim, sentiment pojedinačnog izraza u rečeničnome kontekstu općenito ne mora odgovarati apriornom sentimentu riječi od kojih je taj izraz sastavljen. Preciznije modeliranje sentimenta riječi i fraza u kontekstu, odnosno semantička kompozicija sentimenta, važan je zadatak u obradi prirodnoga jezika i preduvjet za preciznu analizu sentimenta.

U okviru Završnog rada potrebno je razviti model za kontekstno-ovisnu analizu sentimenta izraza hrvatskoga jezika. Prvi korak u tome jest izgradnja i ručno označavanje odgovarajućeg skupa tekstnih podataka na hrvatskome jeziku. U tom plemenitom zadatku potrebna je tvoja pomoć, dragi Označivaču. Ovaj dokument namijenjen je označivačima kao vodilja u poslu ručnog označavanja sentimenta dijelova teksta.

A.2. Opis posla

Ukupan dataset sastoji se od 100 tekstova. Tekstovi su recenzije raznih igara s *Hrvatskog gaming portala* (`www.hcl.hr`). Svaki označivač označit će 30-ak tekstova, od kojih nijedan ne prelazi dvije kartice teksta. Procijenjeno trajanje posla je 5 sati.

A.2.1. Označavanje teksta

U tekstu je potrebno označiti sentiment dijelova teksta. Pod oznakom se podrazumijeva *pozitivno* ili *negativno*. Koji je to točno dio teksta koji nosi sentiment, ostavlja se označivaču na odluku, što znači da sam određuje koje su granice označenog izraza. U ovome dokumentu, zelenom bojom označen je pozitivan sentiment, a crvenom negativan. Slijedi primjer:

> Grafika? Pa.... simpatična i crtana, no ništa spektakularno.

A.2.2. Pravila označavanja

Potrebno je uvesti pravila označavanja kako bi označavanje polučilo željeni rezultat, a i kako bi se uskladio način označavanja svih označivača.

1. **Označeni izraz mora biti neprekinuti segment teksta**

> Super Mario je, uz mnogobrojne druge igre, oduševio publiku.

U ovom primjeru jasno je da je *uz mnogobrojne druge igre* umetnuto u glavnu rečenicu i da je pozitivan sentiment zapravo čitav izraz *Super Mario je (...) oduševio publiku*, no nemojte razlamati segmente i radije označite samo *oduševio publiku*.

2. **Označavajte pozitivne i negativne stvari vezane uz igru**

Primjerice, ako je izrečena objektivna činjenica, ali joj se može odrediti sentiment, označite je. Pritom koristite kontekstualno i opće znanje.

> Super Mario srušio je sve rekode prodavanosti.

3. **Ograničite označeni sentiment na maksimalno 1 rečenicu i uvijek težite označavanju minimuma koji ima smisla**

Primjerice, umjesto cijele rečenice, označite segmente:

> Ovo su jednostavno igre koje moramo poštovati, igre bez kojih gaming ne bi bio ono što je danas.

> Svejedno, dosadit će vam brzo. Zaobići, osim ako morate potrošiti 20$ bezveze i dokazati se kao ultimativni smetlar.

Napomena: izbjegavate označavanje samo jedne riječi (npr. *loš, dobar*). Takvo označavanje moguće je u pojedinim slučajevima, npr. kada je subjekt neizrečen.

4. **Označavajte sentiment uzimajući u obzir cijeli kontekst članka, ne samo tu rečenicu**

   Primjerice, sljedeći primjer izraza je pozitivan jer se u ostatku članka jasno vidi da je recenzija igre pozitivna.

> Na scenu je stupio Mario i učinio nešto što niti jedna igra nikad nije i neće.

5. **Ako niste sigurni trebate li označiti neki izraz i kakvog je on sentimenta, radije nemojte**

A.3. Alat za označavanje

Alat koji ćete koristiti za označavanje je MAE (Multi-purpose Annotation Environment), jednostavan program napisan u Javi. Uz same upute, dobit ćete ZIP arhivu u kojoj se nalazi JAR[1] datoteka pomoću koje pokrećete program.

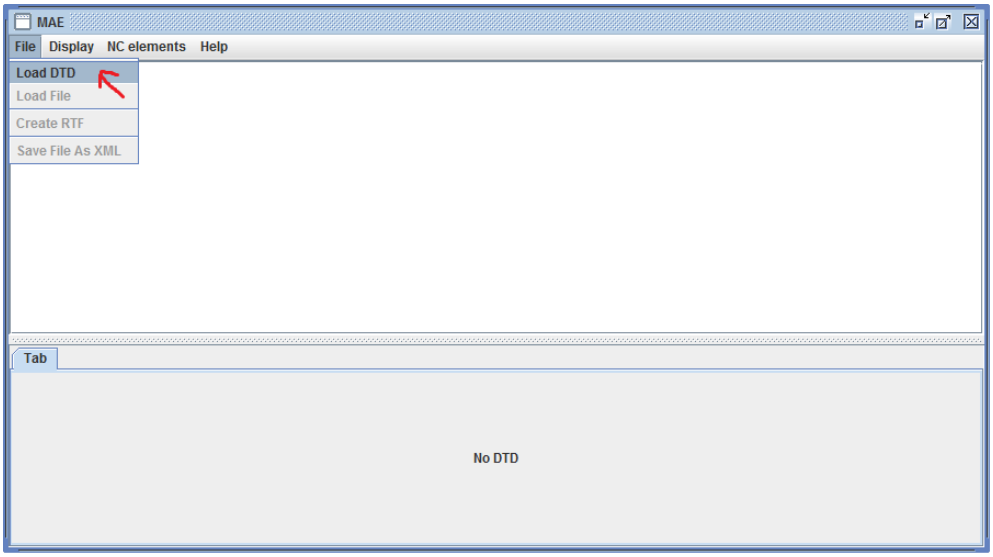
A.3.1. Instalacija i pokretanje programa

Kako biste pokrenuli JAR, na računalu je potrebno imati instaliran Java Runtime Environment. Ako ga nemate, možete ga skinuti s linka `http://java.com/en/download/`. Program možete pokrenuti duplim klikom na datoteku `mae_v0.9.6.jar` ili iz terminala naredbom `java -jar mae_v0.9.6.jar`.

[1] Java Archive ili izvršna datoteka projekta napisanog u Javi.

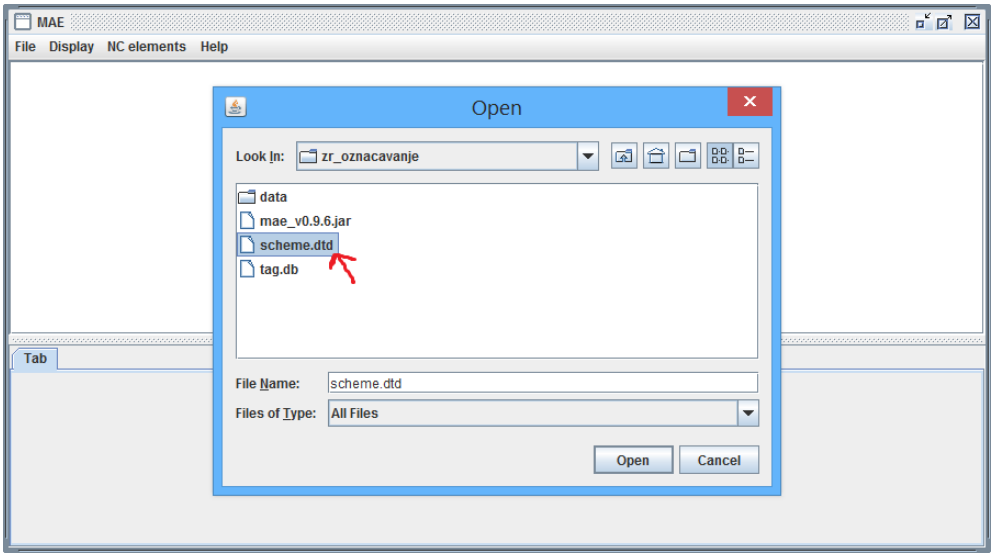
A.3.2. Početak rada

Nakon pokretanja programa, dočekat će vas sljedeće sučelje:



Slika A.1: Početno sučelje programa

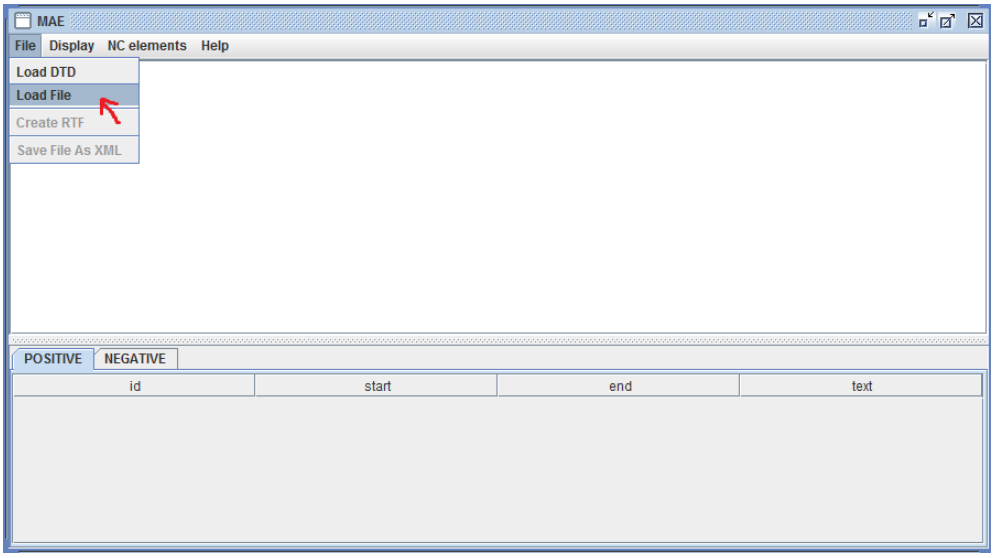
Prostor s bijelom pozadinom rezerviran je za tekst dokumenta, dok je sivi prostor ispod njega gdje piše *No DTD* rezerviran za označene segmente. DTD znači *Document Type Definition*, što je u ovom slučaju datoteka koja definira oznake. Prvi korak je odabrati opciju *Load DTD* u izborniku *File* i odabrati datoteku `scheme.dtd`.



Slika A.2: Otvaranje datoteke koja definira shemu označavanja

Drugi korak je odabrati tekstualnu datoteku koju označavamo. To radimo

opcijom *Load File* u izborniku *File* i odaberemo dokument iz foldera `data`. Neoznačeni tekstovi su u TXT formatu, a program sprema označene datoteke kao XML datoteke. Ako tek počinjete s označavanjem nekog teksta, otvarate TXT datoteku, a ako ste prekinuli i spremili označavanje nekog teksta, možete nastaviti označavanje tako da loadate XML datoteku.



Slika A.3: Otvaranje tekstualne datoteke koju označavamo

A.3.3. Označavanje

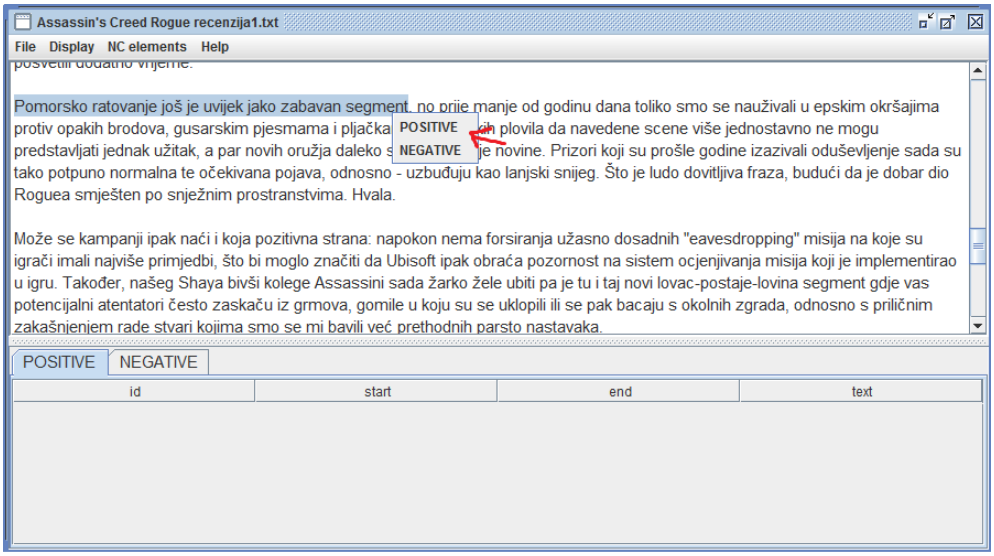
Jednom kada ste otvorili DTD shemu za označavanje i tekstualni dokument (to ćete morati raditi prilikom svakog pokretanja programa), počinje označavanje. Ako želite označiti neki segment teksta, povucite mišem preko željenog segmenta i stisnite desni gumb. Trebao bi se prikazati izbornik kao na slici i klikom na jednu od opcija odabirete oznaku segmenta.

Ako kojim slučajem želite izbrisati nešto što ste već označili, u tablici oznaka pod stupcem *text* stisnite desni klik i prikazat će vam se opcija za brisanje.

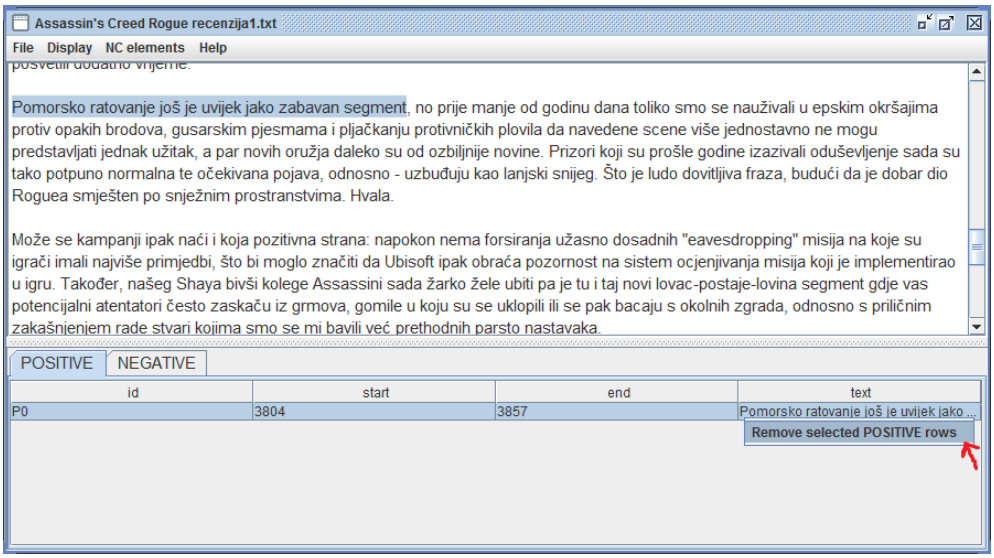
Ako želite brzo doći do označenog segmenta, dupli klik na polje pod stupcem *id* odvest će vas direktno na željeni segment.

A.3.4. Kraj rada

Kada ste završili s dnevnom dozom označavanja, spremite datoteku koju ste označavali. **Oprez – program ne pita jeste li sigurni da želite izaći bez spremanja ako slučajno ugasite program bez prethodnog spremanja!**

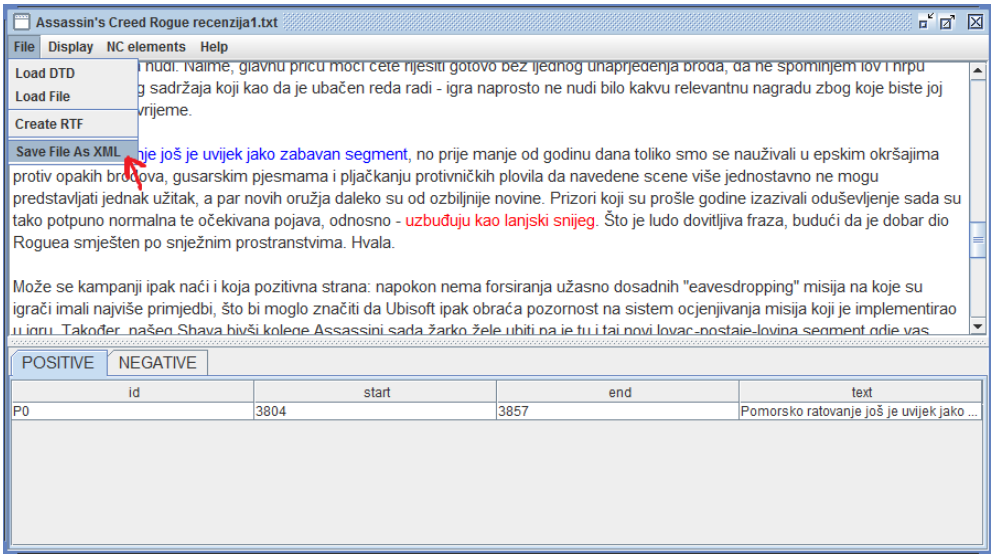


Slika A.4: Označavanje segmenta desnim klikom



Slika A.5: Brisanje označenog segmenta

Program sprema datoteke kao XML datoteke. Datoteku spremite na isto mjesto otkud ste je loadali i ne mijenjajte joj ime. Ako se želite vratiti na već označenu datoteku ili jednostavno nastaviti prekinuto označavanje, možete loadati isti XML dokument i nastaviti gdje ste stali.



Slika A.6: Spremanje označene datoteke

Kontekstno ovisna analiza sentimenta izraza hrvatskoga jezika

Sažetak

Porastom raspoloživih količina korisnički generiranog sadržaja povećalo se zanimanje za strojnom analizom sentimenta, kojom se utvrđuje je li tekst usmjeren pozitivno, negativno ili neutralno. U radu su proučeni postupci za analizu sentimenta s naglaskom na postupak temeljen na strojnom učenju. Razrađen je model za kontekstno ovisnu analizu sentimenta izraza hrvatskoga jezika temeljen na modelu nadziranog strojnog učenja, po uzoru na rad Wilsona i dr. (2005). Izgrađen je i ručno označen odgovarajući skup tekstnih podataka na hrvatskome jeziku za razvoj i ispitivanje modela. Provedeno je iscrpno eksperimentalno vrednovanje modela, statistička obrada rezultata i analiza pogrešaka.

Ključne riječi: obrada prirodnog jezika, analiza sentimenta, strojno učenje, stroj potpornih vektora, hrvatski jezik, računalna lingvistika.

Contextual Sentiment Analysis of Croatian Expressions

Abstract

Given the increase in the amount of available user-generated content, there has been rising interest in sentiment analysis based on machine learning, which determines whether the text attitude is positive, negative or neutral. This paper examines methods for sentiment analysis with a special focus on methods based on machine learning. A model for contextual sentiment analysis of Croatian expressions has been devised, based on the supervised machine learning model described in the work of Wilson et al. (2005). An appropriate dataset consisting of texts in Croatian was manually built and annotated. It was used for model development and testing. An exhaustive experimental evaluation of the model was conducted, along with statistical result and error analysis.

Keywords: natural language processing, sentiment analysis, machine learning, support vector machines, Croatian language, computational linguistics.