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. 3790

Strojno učenje pravila za klasifikaciju dokumenata

Stjepan Glavina

Zagreb, lipanj 2014.

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

Zagreb, 13. ožujka 2014.

ZAVRŠNI ZADATAK br. 3790

Pristupnik: **Stjepan Glavina**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Strojno učenje pravila za klasifikaciju dokumenata**

Opis zadatka:

Sadržajna klasifikacija teksta jedan je od osnovnih zadatka dubinske analize teksta. Uobičajeni postupci temelje se na vektorskoj reprezentaciji značenja dokumenata u sprezi s modelima statističkog strojnog učenja. Prema učinkoviti, takvi modeli nisu lako tumačivi, odnosno nude objašnjenje za klasifikaciju pojedinačnih dokumenata. Alternativu predstavljaju modeli temeljeni na strojno učenim pravilima. Takvi su pravila tumačiva i korisnik ih može po potrebi prilagođavati.
U okviru završnoga rada potrebno je proučiti pristupe za klasifikaciju dokumenata te pristupe za strojno učenje pravila, uključivo algoritam RIPPER i njegovu hijerarhijsku inačicu. Razraditi postupak za hijerarhijsku klasifikaciju dokumenata na hrvatskome jeziku temeljen na strojno učenim pravilima. Razviti programsku implementaciju sustava za klasifikaciju temeljenog na pravilima koji omogućava učenje novih pravila, uređivanje postojećih pravila te klasifikaciju i objašnjenje klasifikacije pojedinačnih dokumenata. Razmotriti proširenje sustava modeliranjem pouzdanosti pravila. Eksperimentalno ispitati rad sustava na ručno označenim zbirkama dokumenata na hrvatskome i engleskome jeziku. Radu priložiti izvorni i izvršni kod razvijenog sustava, skupove podataka i programsku dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 14. ožujka 2014.
Rok za predaju rada: 13. lipnja 2014.

Mentor:

Doc. dr.sc. Jan Šnajder

Djelovođa:

Doc. dr.sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof. dr.sc. Siniša Srbljić

SADRŽAJ

1. Uvod

Napredak računalne tehnologije je omogućio stvaranje, spremanje i prikupljanje enormnih količina podataka. Ta činjenica je svima očita, a jedan dobar primjer koji ju demonstrira je pojava i razvoj interneta. Digitalni dokumenti kojim se raspolaže su raznovrsni; to može biti tekst, slika, video, glazba itd. Međutim, najjednostavniji i navažniji oblik medija je ipak tekst pisan prirodnim jezikom, a samo tim oblikom se ovaj rad i bavi. Kako broj digitalnih tekstnih dokumenata sve brže raste, javila se potreba da se sav taj sadržaj dovede u red i na neki način organizira. Internetske tražilice su primjer vrlo uspješnog organiziranja: omogućuju brzo pretraživanje relevantnih dokumenata na korisnički upit. Da bi se pravi dokumenti uopće mogli efikasno dohvatiti, bitno je da repozitorij dokumenata u podatkovnim centrima tražilice bude dobro strukturiran. Klasifikacija teksta je jedan od osnovnih problema koji se time bave. Klasifikacijom automatski određujemo kojoj od unaprijed zadanih kategorija pojedini dokument pripada.

U ovom radu je opisana implementacija klasifikacije dokumenata i testirana na šest različitih skupova dokumenata (*korpusa*), predstavljenim u poglavlju 6.1. Jedno od najčešćih i najboljih rješenja je klasifikacija pomoću stroja s potpornim vektorima (engl. *support vector machine – SVM*). Glavni nedostatak tog pristupa je što trenirani SVM ne nudi čovjeku lako razumljivi uvid u način na koji klasificira pojedine dokumente. Nažalost, SVM se praktički ponaša kao crna kutija. Alternativna metoda klasifikacije koja je implementirana se bazira na strojno učenim pravilima. Takva pravila se mogu s lakoćom čitati, uređivati, brisati i dodavati jer su konceptualno jednostavna, čitljiva i razumljiva. Naravno, taj pristup dolazi s kompromisom: klasifikacija je manje precizna nego u slučaju SVM-a. No iako do neke mjere kaska za SVM-om, svejedno je vrlo upotrebljiv i vrijedan pažnje.

U nastavku 2. poglavlje detaljnije opisuje problem klasifikacije teksta, 3. i 4. poglavlje dvije glavne metode klasifikacije kojima se rad bavi, 5. poglavlje opisuje detalje implementiranog programa u sklopu rada, dok 6. poglavlje analizira i uspoređuje rezultate vrednovanja različitih metoda (i njihovih varijanti) klasifikacije.

2. Problem klasifikacije teksta

Klasifikacijom teksta određujemo kojoj od zadanih kategorija dokument pripada. Npr. tipične kategorije na internetskim portalima koji objavljuju vijesti su *Hrvatska*, *Svijet*, *Crna kronika*, *Kultura*, *Znanost* i slično. Klasifikacijom se članci (vijesti) mogu automatski razvrstati u kategorije, a upravo takav klasifikator je jedno od rješenja koja su implementirana u sklopu ovog rada.
Postoje i druge zanimljive primjene. Sebastiani (2002) navodi sljedeće primjere upotrebe klasifikacije teksta:

- klasifikacija patenata (organizacija u kategorije radi lakšeg otkrivanja postojećih sličnih patenata)
- klasifikacija internetskih članaka (grupiranje članaka u tematske kategorije)
- filtriranje neželjene pošte (grupiranje poruka u dvije kategorije: željena i neželjena pošta)
- identifikacija autora teksta (na temelju unaprijed zadanog skupa autora)
- identifikacija roda autora (radi li se o autoru ili autorici; slično prethodnom primjeru)
- klasifikacija teksta prema žanru (identifikacija vrste dokumenta, npr. radi li se o recenziji proizvoda ili reklami za proizvod)

Vrijedi spomenuti da se svi ovi navedeni primjeri odnose samo na upotrebu nadziranog strojnog učenja. To znači da je klasifikator učen već unaprijed pripremljenim ručno klasificiranim dokumentima u predefinirane kategorije.
Kod nenadziranog učenja nema ručno klasificiranih dokumenata niti poznatih kategorija, stoga je u tom slučaju zadatak klasifikatora da sam otkrije moguće kategorije i u njih klasificira dokumente. Međutim, klasifikacija nenadziranim strojnim učenjem je izvan opsega ovog rada.

Klasifikacija teksta je inherentno subjektivan zadatak. Više osoba se neće složiti u apsolutno svim slučajevima oko pripadnosti pojedinog dokumenta pravim kategorijama. Kod svrstavanja u kategoriju je bitno pravilno interpretirati značenje rečenica

i sam ton teksta, tj. ono što piše *između redaka*. Dodatno, možda je potrebno i upotrijebiti opće znanje ili poznavanje tematike te uočiti sličnosti s ostalim dokumentima slične vrste, što samo otežava stvar. Takva razina razumijevanja teksta je delikatan i izuzetno težak posao za računalnu implementaciju. Računalna klasifikacija se najčešće svodi samo na analizu skupa riječi koje dokument sadrži, što implicira da se struktura rečenica i njihovo značenje u potpunosti odbacuju. Klasifikacija bazirana samo na temelju skupa riječi u dokumentu je u većini slučajeva učinkovita. Nažalost, kod klasificiranja dokumenata u srodne kategorije ta površna metoda analize teksta osobito zakazuje, što se vidi na primjeru u odjeljku 6.1.2.

2.1. Ulaz u klasifikator

Da bi ga bilo jednostavnije i praktičnije reprezentirati u klasifikacijskim algoritmima, originalni tekstni dokument je najprije potrebno pretprocesirati. Iz dokumenta se prvo ekstrahiraju sve riječi koje sadrži, zatim lematiziraju i na kraju uklanjaju suvišne riječi.

Lematizacijom se riječi svode na korijenski oblik; npr. riječi "psa" i "psima" se pretvaraju u jednostavno "pas" i na taj način smatraju ekvivalentnima. Višestruke pojave iste riječi se zadržavaju. Razlog tome je očevidan: ako se riječ "Hrvatska" u nekom novinskom članku pojavljuje deset puta, a riječ "Rusija" samo jednom, mnogo je vjerojatnije da je Hrvatska glavna tema članka, dok Rusija gotovo sigurno nije središnji subjekt članka. Zadržavanjem višestrukih pojavljivanja iste riječi se ova informacija čuva.

Postoje neinformativne riječi (engl. *stopwords*) koje služe samo kod gradnje rečeničnih konstrukcija, a ne pomažu u identifikaciji klase dokumenta. To su veznici, čestice, prijedlozi, prilozi i slične vrste riječi. One predstavljaju samo smetnju kod klasifikacije pa ih je stoga potrebno ukloniti. Njihovo uklanjanje se radi uz pomoć unaprijed izrađenog popisa takvih funkcijskih riječi.

Rezultat obrade ulaznog teksta je vreća riječi (engl. *bag of words*) koja karakterizira originalni dokument.

2.1.1. Primjer

Slijedi primjer pretprocesiranja teksta jedne kratke vijesti preuzete s interneta.
Izvor: Hrvatska izvještajna novinska agencija (2014).
Ovo je originalni dokument:

```
Međunarodni monetarni fond (MMF) razmatra mogućnost da
```

```
odobri isplatu 180 milijuna eura Bosni i Hercegovini
kako bi toj zemlji pomogao u suočavanju s posljedicama
nedavnih poplava, potvrđeno je u petak iz ureda te
financijske institucije u Sarajevu.
```

Sljedeći korak je uklanjanje rečenične strukture. Nestaju interpunkcijski znakovi i brojevi, velika slova se pretvaraju u mala, a riječi se lematiziraju. Poredak riječi nije važan, pa su riječi radi preglednosti sortirane po abecedi. Rezultat:

```
bi bosna dati euro financijski fond hercegovina
institucija isplata iz je kako međunarodan milijun
mmf mogućnost monetaran nedavan odobriti petak pomoći
poplava posljedica potvrđen razmatrati sarajev
suočavanje te toj ured zemlja
```

Još je samo potrebno ukloniti funkcijske riječi poput "bi", "iz" i "je", a zadržati one koje su zbilja nosioci sadržaja. Nakon uklanjanja se dobiva konačna vreća riječi, karakteristična za početni tekst:

```
bosna dati euro financijski fond hercegovina institucija
isplata međunarodan mmf mogućnost monetaran nedavan
odobriti petak pomoći poplava posljedica potvrđen
razmatrati sarajev suočavanje ured zemlja
```

Iz ove konačne krnje forme vijesti više nije moguće rekonstruirati niti razumjeti originalni tekst. No unatoč tome, moguće je bez poznavanja teksta iz same vreće riječi zaključiti o čemu se otprilike radi. Primjerice, budući da vreća sadrži riječi "bosna", "hercegovina" i "sarajev", vijest spominje državu BiH ili izvještava o događaju koji se u njoj dogodio. Pojava riječi "poplava" i "suočavanje" indicira da bi se moglo raditi o elementarnoj nepogodi iz crne kronike, dok "euro" i "financijski" ukazuju na gospodarstvo i novac.

Već na ovom primjeru je vidljivo da sama vreća riječi sadrži dovoljno informacija za barem grubu klasifikaciju dokumenata, ali i da se značenje riječi može krivo interpretirati. Moguće je da je riječ "poplava" iskorištena u metaforičkom smislu, pa se zapravo uopće ne radi o elementarnoj nepogodi nego nekakvoj *novčanoj poplavi*. Ovaj problem nije lako rješiv; implementacija klasifikatora u sklopu ovog rada ga naprosto mora pretrpjeti.

2.2. Izlaz iz klasifikatora

Klasifikator za zadani dokument i svaku od kategorija određuje pripada li dokument u nju ili ne. Stoga u suštini postoji onoliko klasifikatora koliko i kategorija, a svaki od njih daje binarni odgovor (*pripada* ili *ne pripada*). Dokument iz primjera 2.1.1 bi se klasificirao u kategorije poput *Crna kronika*, *Hrvatska i regija* ili pak *Gospodarstvo*, dok sigurno ne bi u *Sport* ili *Kultura*.

U nekim skupovima tekstova su kategorije organizirane u hijerarhijsku strukturu. Npr. kategorija *Zagreb i Županija* može biti podkategorija *Hrvatske*, a *Film* podkategorija *Kulture*. Na taj način kategorije čine hijerarhijsko stablo, a za svaki dokument vrijedi sljedeća invarijanta: ako dokument pripada u neku kategoriju, onda pripada i u njenu roditeljsku kategoriju. Kod hijerarhijske inačice algoritam klasifikacije počinje od korijenskih kategorija i za svaku od njih odredi pripada li joj ulazni dokument. Ako pripada nekoj kategoriji, postupak se rekurzivno ponavlja za njene direktne podkategorije. U slučaju da dokument ne pripada kategoriji, onda definitivno ne može pripadati niti bilo kojoj njenoj podkategoriji.

2.3. Metode klasifikacije

Problem klasifikacije dokumenata se može riješiti ručno konstruiranim algoritmima za klasifikaciju na temelju ugrađenih pravila. To spada u inženjering znanja (engl. *knowledge engineering*). Detaljnom analizom određenog skupa dokumenata se mogu uočiti i razviti pravila koja dokumente dobro klasificiraju unutar te problemske domene. Iako mogu biti vrlo učinkoviti, ručno izgrađene algoritme je vremenski skupo graditi od temelja za svaku novu primjenu klasifikacije. Cilj ovog rada je analizirati algoritme koji samo uz pomoć strojnog učenja klasificiraju dokumente, neovisno o vrsti tekstnih dokumenata unutar skupa. Strojno učeni klasifikatori su generalniji i zbog toga mnogo šire primjenjivi. Dovoljno je implementirati samo jedan algoritam za klasifikaciju i jednostavno ga naučiti da sam klasificira dokumente iz bilo kakvog skupa dokumenata. Osim toga, današnji klasifikatori bazirani na principu strojnog učenja su prema učinkovitosti čak i bolji od onih baziranih na inženjeringu znanja te na razini ljudske ručne klasifikacije dokumenata (Sebastiani, 2002).

Da bi se klasifikatora strojno učilo, najprije je potreban dovoljno velik skup već označenih dokumenata. Dokumente mora čovjek ručno označiti; to je praktički najveći zadatak u koji je potrebno uložiti ljudski napor. Označavanje dokumenata uglavnom nije velik problem jer je u mnogim slučajevima skup označenih dokumenata odmah

lako dostupan. Ako je knjižnici potrebna automatizirana metoda klasifikacije knjiga, dovoljno je strojno naučiti klasifikator već postojećim rasporedom knjiga na policama. Čak i u nesretnim slučajevima kad nema gotovog označavanja dobra vijest je činjenica da je za označavanje potrebno manje vještine nego za inžinjering znanja.

Skup označenih dokumenata se prije učenja (treniranja) prvo dijeli na dva skupa dokumenata: skup za treniranje i skup za testiranje (engl. *training set* i *test set*). Dijeljenje u dva skupa je obično u omjeru $1:1$ ili $1:2$. Klasifikator je strojno učen skupom za treniranje, a zatim se njegova efikasnost evaluira pomoću skupa za testiranje.

U okviru rada su implementirane dvije metode klasifikacije strojnim učenjem:

- uz pomoć stroja potpornih vektora
- uz pomoć strojno učenih pravila

Obje metode imaju svoje prednosti i mane, a u poglavljima koja slijede su objašnjeni njihovi principi rada, prednosti i mane te usporedba na rezultatima evaluacije. Klasifikacija uz pomoć stroja potpornih vektora primarno služi samo kao *baseline*, tj. za konkretnu procjenu relativne učinkovitosti metode bazirane na strojno učenih pravila, koja i jest glavna tema.

3. Klasifikacija teksta uz pomoć SVM-a

Stroj s potpornim vektorima (SVM) je metoda nadziranog strojnog učenja koja višedimenzionalne vektore za treniranje optimalnom hiperravninom razdvaja u dvije kategorije. Svi vektori koji se nalaze s jedne strane hiperravnine pripadaju u jednu kategoriju, a s druge strane u drugu kategoriju. Pronađena hiperravnina se zatim koristi za klasifikaciju vektora (iz skupa vektora za testiranje): za vektor je potrebno samo odrediti s koje strane hiperravnine se nalazi. SVM radi brzo i efikasno čak i kada se radi o vrlo mnogo dimenzija, pa se zato SVM odmah nameće kao vrlo praktično rješenje problema klasifikacije dokumenata. Svaki pretprocesirani dokument, tj. njegova vreća riječi, može se reprezentirati vektorom značajki. Dimenzija vektora je jednaka broju različitih riječi koje se spominju u kompletnom skupu dokumenata za treniranje, što u praksi odgovara desecima tisuća riječi. Za svaku riječ koja se pojavljuje u vreći riječi se na odgovorajuće mjesto u vektoru upiše njen broj pojavljivanja unutar dokumenta. Doduše, umjesto točno tog broja obično se koriste njegove varijacije koje daju bolje rezultate. Budući da jedan dokument ne sadrži jako puno različitih riječi, vektori značajki su vrlo rijetki, tj. većina njihovih elemenata je jednaka nuli.

Konkretna reprezentacija vektora je napravljena prema modelu kojeg koriste Lewis et al. (2004). Vrijednost $w_d(t)$ za riječ t (engl. *term*) u vektoru značajki dokumenta d se računa prema formuli:

$$w_d(t) = (1 + \ln n(t,d)) \cdot ln(|D|/n(t)) \tag{3.1}$$

Značenja oznaka u formuli (u ovom kontekstu se *skup* odnosi samo na skup dokumenata za treniranje):

- $|D|$ je broj ukupan dokumenata u skupu
- $n(t)$ je broj dokumenata u skupu koji sadrže riječ t barem jednom
- $n(t,d)$ je broj pojavljivanja riječi t u dokumentu d

Za riječi t koje se ne nalaze u dokumentu d vrijedi $w_d(t) = 0$.

Kao posljednji korak u pripremi vektora značajki preostalo je skaliranje. Vektori se euklidski normiraju, tj. skaliraju tako da im euklidska duljina bude jednaka 1. Formula za normiranje je:

$$w'_d(t) = \frac{w_d(t)}{\sqrt{\sum_u w_d(u) \cdot w_d(u)}} \tag{3.2}$$

Glavni nedostatak klasifikacije uz pomoć SVM-a je što rezultat treniranja (separacijska hiperravnina) nije čovjeku razumljiva reprezentacija modela za klasifikaciju dokumenata. Iz same hiperravnine je teško razlučiti kako ona radi i na koji način pojedine dokumente klasificira. Iako SVM daje dobre rezultate (kao što se vidi u poglavlju 6.2), bilo bi zgodno da se njime trenirani klasifikator može lako razumjeti i po potrebi ručno uređivati te nadograditi ljudskim iskustvenim znanjem. Takve mogućnosti imaju značajnu praktičnu vrijednost. Klasifikacija uz pomoć pravila je druga metoda klasifikacije koja ne pati od spomenutog problema.

4. Klasifikacija teksta uz pomoć pravila

Kod primjera 2.1.1 je već sugerirano da se na temelju pojave riječi *poplava* i *suočavanje* može zaključiti da originalni dokument pripada kategoriji *Crna kronika*. Ovaj način razmišljanja je motivacija za klasifikaciju teksta uz pomoć pravila. Može se konstruirati niz pravila oblika: *Ako dokument sadrži skup riječi* $\{t_1, t_2, t_3, ...\}$*, onda pripada kategoriji* K.

Glavna ideja ove metode je strojnim učenjem izgraditi skup pravila koja dobro klasificiraju dokumente. Izgrađena pravila su jednostavna i pregledna. Na bilo kojem dokumentu je jasno koja pravila su *upalila*, a koja nisu. U slučaju da korisnik primijeti da klasifikator pogrešno klasificira određenu vrstu dokumenata ili ponavlja iste greške, jednostavno je korigirati loša pravila ili pak dodati nova. Cilj je da se uz povremenu ljudsku intervenciju dotjerivanja pravila uz malo napora naprave klasifikatori koji su istovremeno i razumljivi i vrlo precizni.

U ovom radu je implementacija klasifikatora dizajnirana tako da se za svaku kategoriju izgradi popis pravila u sljedećem obliku. Svako pravilo je napisano u svom retku, a sastoji se samo od popisa riječi koje ono očekuje u dokumentu. Naprimjer, pravila za *Crnu kroniku* bi mogla ovako izgledati:

```
policija očevid poginuo
policija provalnik
sat očevid smrtan
voziti nesreća prevrnuti
ozljeda preminuo
nesreća očevid cesta
```

Interpretacija skupa pravila: ako dokument sadrži riječi *policija*, *očevid* i *poginuo*; ili ako sadrži riječi *policija* i *provalnik*; ili ako sadrži riječi *sat*, *očevid* i *smrtan*; ili ...; onda pripada u kategoriju *Crna kronika*.

U ovom primjeru pokušaj klasificiranja vijesti o tragičnim poplavama neće rezultirati uspjehom, budući da se poplave nigdje ne spominju među pravilima. Ako korisnik koji koristi klasifikator za razvrstavanje vijesti u kategorije primijeti ovaj propust, može jednostavno dodati i novo pravilo (ili nekoliko njih) koje će ispravno klasificirati većinu vijesti o poplavama.

Jedna bitna razlika u odnosu na klasifikaciju pomoću SVM-a je irelevantnost ponavljanja riječi u dokumentu. Potpuno je svejedno pojavljuje li se neka riječ samo jednom ili više puta. Ovo implicira da klasifikacija uz pomoć pravila raspolaže s manje informacija o dokumentu, pa je već u početku do neke mjere u inferiornoj poziciji u odnosu na SVM. Međutim, tom problemu može doskočiti na razne načine, iako u okvirima ovog rada oni nisu bili razmatrani.

Postoje različiti algoritmi (i njihove varijante) koji automatski izgrađuju pravila za klasifikaciju dokumenata, a ovdje je implementiran algoritam RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*), i to po uzoru na RIPPER koji opisuju Menzies (2006) i Lanzi (2007).

4.1. RIPPER

Algoritam za jednu kategoriju postupno izrađuje pravila koja pokrivaju sve pozitivne primjerke dokumenata (one koji joj pripadaju) uz nastojanje da se pokrije čim manje negativnih dokumenata (onih koji joj ne pripadaju). Počinje se od kompletnog skupa dokumenata za trening, zatim se konstruira jedno pravilo pa obrišu svi dokumenti koje ih pokriva. To novo pravilo se dodaje u skup pravila, a algoritam se nastavlja na isti način sve dok skup dokumenata ne postane prazan. Procedura je opisana algoritmom 1.

Kod konstrukcije jednog pravila se skup dokumenata prvo dijeli na dva dijela: skup za gradnju i skup za rezanje. Obično je skup za gradnju dvostruko veći od skupa za rezanje. U prvoj fazi pravilo prvo naraste pomoću skupa za gradnju, a u drugoj fazi se skraćuje pomoću skupa za rezanje. Procedura je opisana algoritmom 2.

Postupak izgradnje se radi tako dugo ima i pozitivnih i negativnih dokumenata (algoritam 3). U svakom koraku se traži riječ koja maksimizira prirast informacije FOIL (TNM003: Introduction to Data Mining). Prirast informacije za neku riječ w se definira na sljedeći način:

$$prirast(w) = p_1 \cdot (log_2 \frac{|P'|}{|P'| + |N'|} - log_2 \frac{|P|}{|P| + |N|}) \tag{4.1}$$

Algorithm 1 RIPPER

Ulaz: P, N – skupovi pozitivnih i negativnih dokumenata.

Izlaz: R – popis pravila.

$R := \emptyset$

while $P \neq \emptyset$ **do**

$r := KonstruirajPravilo(P, N)$

ukloni dokumente iz skupa P koje r pokriva

ukloni dokumente iz skupa N koje r pokriva

$R := R \cup \{r\}$

end while

return R

Algorithm 2 KonstruirajPravilo

Ulaz: P, N – skupovi pozitivnih i negativnih dokumenata.

Izlaz: W – pravilo kao niz riječi.

podijeli $P \cup N$ na skupove $P_{izgradnja}, N_{izgradnja}, P_{rezanje}, N_{rezanje}$

$W := GradiPravilo(P_{izgradnja}, N_{izgradnja})$

$W := ReziPravilo(W, P_{rezanje}, N_{rezanje})$

return W

U jednadžbi 4.1 je P' podskup od P, a sadrži sve njegove dokumente koji nemaju u sebi riječ w. Analogno se definira i skup N'.

Algorithm 3 GradiPravilo

Ulaz: P, N – skupovi pozitivnih i negativnih dokumenata.
Izlaz: W – izgrađeno pravilo kao niz riječi.
$W := []$
while $P \neq \emptyset$ **and** $N \neq \emptyset$ **do**
  pronađi riječ w s maksimalnim prirastom informacije
  ukloni dokumente iz skupa P koji sadrže w
  ukloni dokumente iz skupa N koji sadrže w
  $W := W + [w]$
end while
return W

Ova izgradnja pravila će rezultirati prespecifičnim pravilom; ono će sadržavati više riječi nego što je potrebno. Rezanje pravila je pokušaj generaliziranja pravila tako da pokriva malo širi skup dokumenata, ali ipak ne preširok. U algoritmu 4 se s kraja niza riječi koje definiraju pravilo uklanja sufiks, tj. zadržava samo neki prefiks. Traži se prefiks pravila koji je optimalan prema kriteriju opisanom u pseudokodu.

Posljednja procedura kompletira opis rada RIPPER-a implementiranog u ovom radu. Postoji još mnogo različitih varijanti algoritma. Neke od jednostavnih se dobivaju drugačijim odabirom funkcije za ocjenjivanje sufiksa kod rezanja pravila, ili funkcije za izračunavanje prirasta informacije. Spomenute funkcije su odabrane jer su se eksperimentalno ponašale najbolje na korištenim korpusima, iako se nije dublje zalazilo u ovo područje istraživanja.

Algorithm 4 ReziPravilo

Ulaz: W, P, N – pravilo koje treba skratiti te skupovi pozitivnih i negativnih dokumenata.

Izlaz: W – skraćeno pravilo kao niz riječi.

$W_{opt} := W$

$x_{opt} := -\infty$

for svaki prefiks W_i niza W **do**

  $p :=$ broj dokumenata u skupu P koje W_i pokriva

  $n :=$ broj dokumenata u skupu N koje W_i pokriva

  $x_i := (p - n)/(p + n)$

  if $x_i > x_{opt}$ **then**

    $x_{opt} := x_i$

    $W_{opt} := W_i$

  end if

end for

return W_{opt}

5. Programska implementacija

Implementiran je program koji čita korpus dokumenata, strojno ih uči klasificirati pomoću SVM-a i RIPPER-a, vrednuje klasifikatore te prikazuje rezultate vrednovanja u tablici. Algoritmi su memorijski i vremenski vrlo zahtjevni, pa je iz tog razloga C++ odabran kao primarni programski jezik.

Da se ne bi kod svakog pokretanja programa svi dokumenti unutar korpusa iznova pripremali, to mora biti unaprijed riješeno: dokumenti su podijeljeni na skup za treniranje i skup za testiranje. Osim toga, svaki dokument još treba biti zasebno pripremljen: prvo lematiziran, a zatim irelevantne riječi uklonjenje (odjeljak 2.1.1). Priprema teksta na engleskom se vrši posebnom skriptom napisanom u Pythonu koja koristi biblioteku NLTK[1]. Za hrvatski jezik je korišten interni lematizator razvijen u FER-ovom laboratoriju TakeLab (Šnajder et al., 2008).

Klasifikacija uz pomoć stroja s potpornim vektorima je implementirana koristeći biblioteku *liblinear*[2] (Fan et al., 2008). Ona pruža gotove SVM algoritme s linearnim kernelima. U početku je korišten *liblinear* kao običan program, tj. programima *liblinear-train* i *liblinear-test* su bile zadavane datoteke s vektorima za trening i testiranje. Međutim, budući da su ulazne datoteke ogromne, njihovo pisanje i čitanje s diska je vrlo sporo što otežava rad. Kao rješenje tog problema je biblioteka liblinear ugrađena u program, te se poziva direktno iz koda.

Klasifikacija uz pomoć pravila je u cijelosti implementirana prema algoritmu 1, opisanom u odjeljku 4.1.

Hijerarhijska inačica klasifikatora baziranih na SVM-u i RIPPER-u je također implementirana. Ona koristi hijerarhijsku strukturu kategorija u korpusu. Način rada hijerarhijske klasifikacije je opisan algoritmom 5.

Osim što je algoritam klasifikacije malo drugačiji, u hijerarhijskoj inačici postoje i razlike kod treniranja. Treniranje za kategoriju se vrši samo na dokumentima koji pripadaju u roditeljsku kategoriju. U teoriji to predstavlja olakšanje u postupku treni-

[1] `http://www.nltk.org/`

[2] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

Algorithm 5 HijerarhijskiKlasificiraj

Ulaz: K, kat, D – klasifikator, kategorija i dokument
Izlaz: **true** ili **false** – pripada li dokument u kategoriju ili ne
$kat_r :=$ Roditelj(kat)
if $kat_r \neq nil$ **and** HijerarhijskiKlasificiraj(K, kat_r, D) **then**
  if Klasificiraj(K, kat_r, D) **then**
    return $true$
  end if
end if
return $false$

ranja za klasifikator, budući da mora razlikovati znatno manji skup kategorija (samo one koje su srodne unutar jedne razine hijerarhijskog stabla).

5.1. Hibridni klasifikator

Kao dodatna vrsta klasifikacije je još implementiran i hibridni klasifikator kao kombinacija druga dva klasifikatora. Princip rada je vrlo jednostavan: akko barem jedan od klasifikatora svrstava dokument u neku kategoriju, onda ga svrstava i hibrid. Kako je hijerarhijski SVM eksperimentalno točniji od običnog (poglavlje 6), a običan RIPPER točniji od hijerarhijskog, za hibridnu varijantu su odabrani hijerarhijski SVM i običan RIPPER.

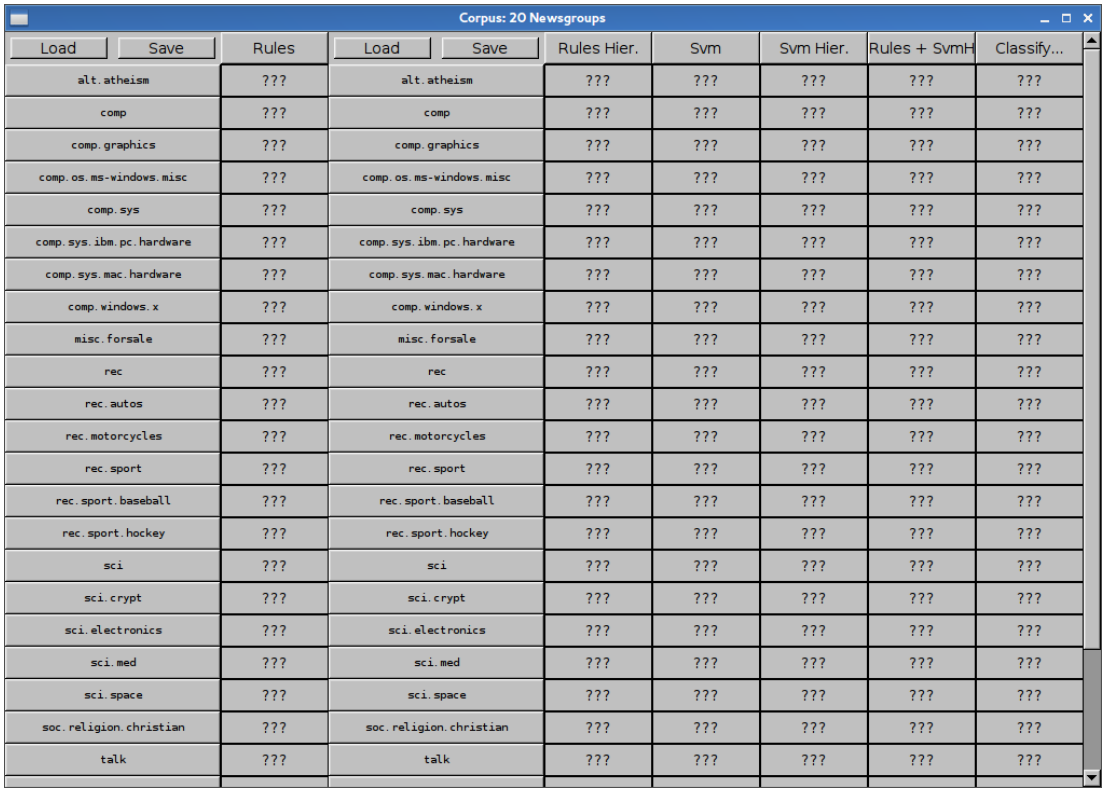
5.2. Grafičko sučelje

Grafičko sučelje programa je izvedeno pomoću biblioteke FLTK[3] (*Fast Light Toolkit*). Najistaknutija vrlina FLTK-a je lakoća i jednostavnost izrade grafičkih sučelja, a to je bio i prvi kriterij pri izboru *GUI toolkita*. Glavni prozor s učitanim korpusom *20 Newsgroups* je prikazan na slici 5.1.

Slika 5.2 prikazuje rezultat vrednovanja svih klasifikatora. Svaka ćelija tablice prikazuje F_1-mjeru za odgovarajući klasifikator i kategoriju, a približavanjem kursora ćeliji se otkrivaju vrijednosti preciznosti i odziva.

Klasifikatori bazirani na pravilima nude mogućnost prikaza i uređivanja samih pravila. Prozor za uređivanje pravila se otvara klikom na ime kategorije, a primjer jednog

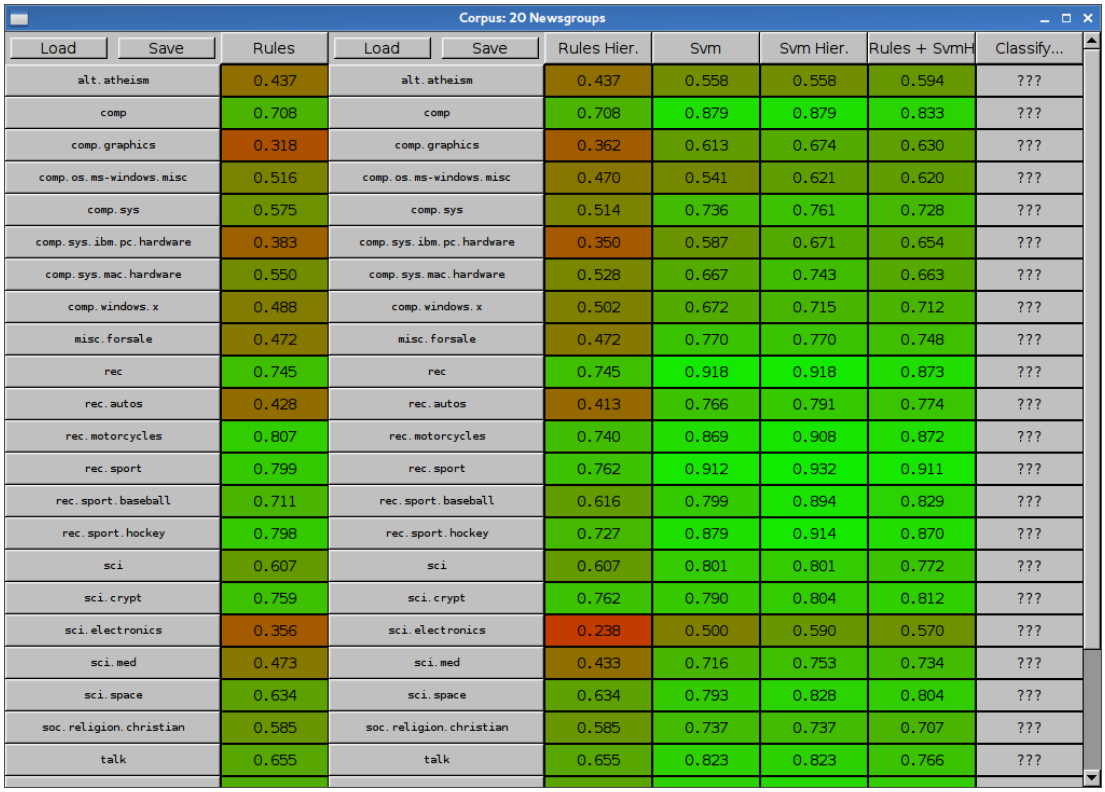
[3] `http://www.fltk.org/`



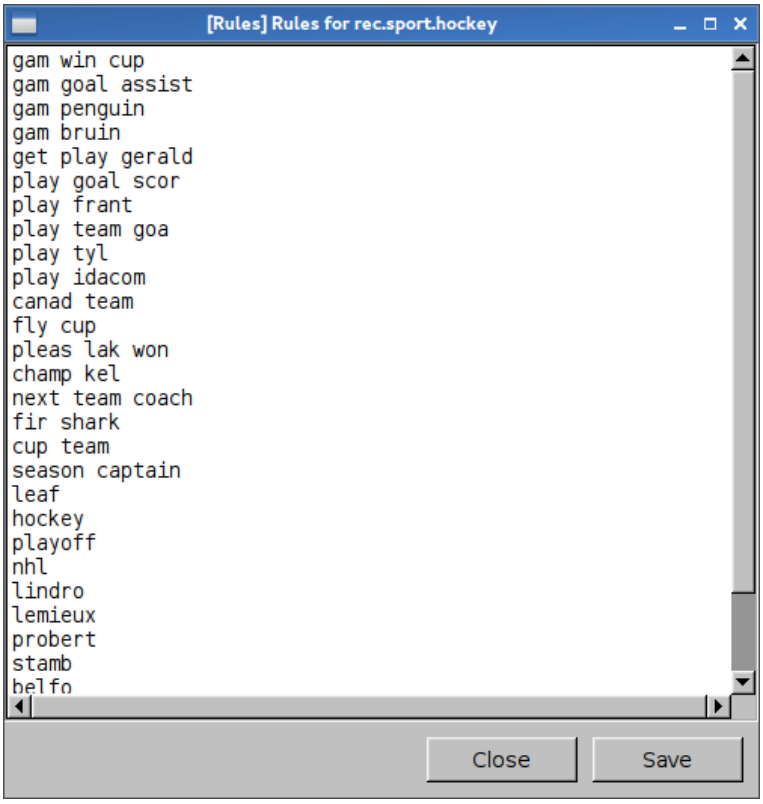
Slika 5.1: Glavni prozor nakon učitavanja korpusa

takvog prozora je na slici 5.3.

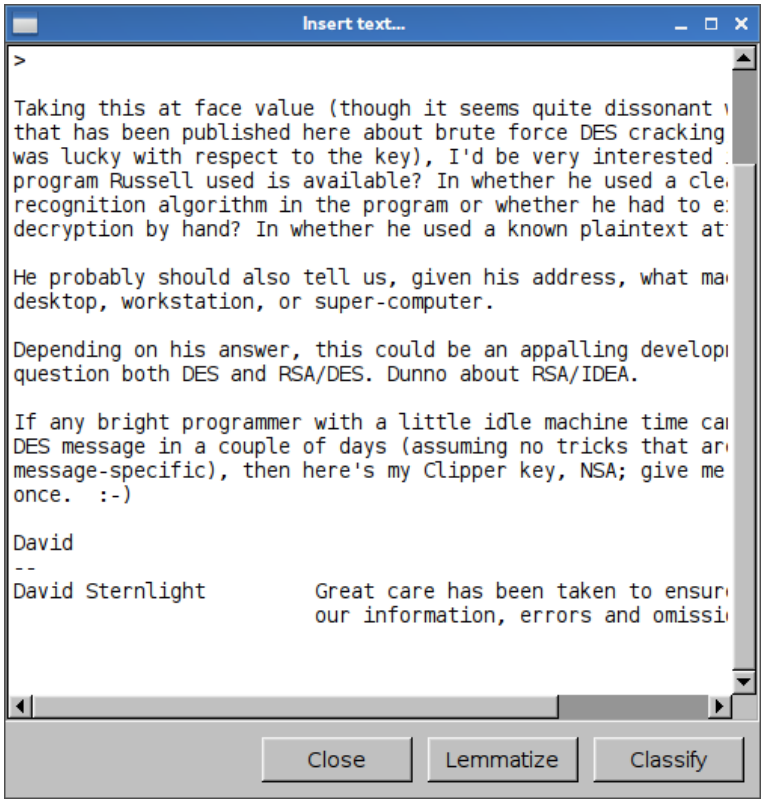
Posljednji stupac tablice omogućuje klasifikaciju pojedinačnih dokumenata. Korisnik u prozor (slika 5.4) upisuje sadržaj teksta, a klikom na gumb ga može lematizirati (obavezan korak prije klasifikacije) i klasificirati. Još se i odabire vrsta klasifikatora, a na kraju posljednji stupac tablice prikazuje u koje kategorije dokument pripada.



Slika 5.2: Prikaz vrednovanja klasifikatora



Slika 5.3: Uređivanje pravila



Slika 5.4: Prozor za klasifikaciju jednog dokumenta

6. Vrednovanje

Da bi se efikasnost klasificiranja dokumenata za neku kategoriju vrednovala, potrebno je prebrojati koliko puta se kod klasifikacije testnih dokumenata dogodio svaki od ova četiri slučaja:

- *true positive (tp)*: pozitivan dokument je točno klasificiran
- *true negative (tn)*: negativan dokument je točno klasificiran
- *false positive (fp)*: negativan dokument je pogrešno klasificiran kao pozitivan
- *false negative (fn)*: pozitivan dokument je pogrešno klasificiran kao negativan

Preciznost (engl. *precision*) je omjer broja dokumenata koji su ispravno klasificirani kao pozitivni i ukupnog broja pozitivno klasificiranih dokumenata.

$$precision = \frac{tp}{tp + fp} \tag{6.1}$$

Odziv (engl. *recall*) je omjer broja dokumenata koji su ispravno klasificirani kao pozitivni i ukupnog broja pozitivnih dokumenata.

$$recall = \frac{tp}{tp + fn} \tag{6.2}$$

Cilj klasifikatora je da ove dvije mjere budu po vrijednosti čim bliže broju 1.

U statističkoj analizi se za točnost klasifikatora koristi još jedna mjera, a kombinira preciznost i odziv. Naziva se F_1-mjera, a računa na sljedeći način:

$$F_1 = \frac{2 \cdot recall \cdot precision}{recall + precision} \tag{6.3}$$

To je zapravo harmonijska sredina preciznosti i odziva. Kod nje je također cilj da bude čim bliže broju 1. Iako se F_1-mjera može računati za svaku kategoriju zasebno, u nastavku se koristi kao mjera točnosti klasifikatora na svim kategorijama zajedno. Dva su načina računanja ukupne F_1-mjere:

1. Mikro F_1 - računa se odjednom pomoću svih zajedno zbrojenih vrijednosti tp, tn, fp i fn (kroz sve kategorije); zbrojene vrijednosti se uvrštavaju u formule 6.1 i 6.2, a izračunati rezultati u formulu 6.3

2. Makro F_1 - računa se pomoću prosječne preciznosti i prosječnog odziva unutar svake kategorije zasebno; prosječna preciznost i odziv se uvrštavaju u formulu 6.3

Pet različitih klasifikatora je vrednovano na skupovima dokumenata:

1. RIPPER - klasifikacija uz pomoć pravila; bez uporabe hijerarhije kategorija
2. RIPPER (hijer.) - klasifikacija uz pomoć pravila; s uporabom hijerarhije kategorija
3. SVM - klasifikacija uz pomoć SVM-a; bez uporabe hijerarhije kategorija
4. SVM (hijer.) - klasifikacija uz pomoć SVM-a; s uporabom hijerarhije kategorija
5. RIPPER + SVM (hijer.) - hibrid 1. i 4. klasifikatora

6.1. Skupovi podataka

Četiri skupa podataka (korpusa) su korištena za trening i evaluaciju klasifikatora. Dva skupa sadrže dokumente na engleskom jeziku, a druga dva na hrvatskom.

6.1.1. Reuters

Reutersov korpus sadrži preko 800,000 objavljenih novinskih članaka na engleskom jeziku, koji su ručno označeni i pripremljeni u svrhu znanstvenog istraživanja (Lewis et al., 2004). Sastoji se od tri skupa kategorija:

- *Industries* (354 kategorija): hijerarhijska kategorizacija članaka prema vrsti posla (npr. *Metals and minerals* i *Agriculture and horticulture*)
- *Regions* (366 kategorija): kategorizacija članaka prema svjetskim regijama (uglavnom države); nije hijerarhijska
- *Topics* (103 kategorije): hijerarhijska kategorizacija članaka prema temi (npr. *Crime/Law Enforcement* i *Fashion*)

Budući da je korpus jako velik, radi praktičnosti je ograničen samo na prvu četvrtinu od dostupnih dokumenata za testiranje. Skup dokumenata za treniranje je nepromijenjen, tj. korišten je takav kakav je i već ponuđen.

6.1.2. 20 Newsgroups

20 Newsgroups je skup od oko 20,000 dokumenata na engleskom jeziku iz 20 različitih *newsgroupa* (Rennie, 2014). Neki primjeri tih 20 kategorija su *comp.graphics*, *comp.sys.ibm.pc.hardware* i *alt.atheism*. Razlikujemo još 7 dodatnih (poput *comp*, *comp.sys* i *sci*), koje su nazvane prema ponavljajućim prefiksima početnih 20 kategorija. Tako je na jednostavan način izgrađena hijerarhija kategorija. Korpus je već unaprijed podijeljen na skup za trening i skup za testiranje, i to kronološki prema datumu objave: starija polovica dokumenata je u skupu za trening.

6.1.3. Vjesnik

Vjesnikov korpus se sastoji od oko 260,000 novinskih članaka prikupljenih kroz period od 10 godina. Za treniranje je iskorišten skup od oko 25,000 članaka objavljenih 2003. godine, dok su za testiranje korišteni članci iz 2004. godine, kojih ima oko 24,000. Postoji 12 kategorija, a primjeri su *Crna kronika*, *Gospodarstvo*, *Sa svih strana* i *Teme dana*. Kategorije u ovom korpusu nemaju hijerarhijsku strukturu, a svi članci su na hrvatskom jeziku.

6.1.4. Narodne Novine

Oko 13,000 zakonodavnih dokumenata na hrvatskom jeziku je u korpusu Narodnih Novina. Kategorije su strukturirane u veliku hijerarhiju od oko 40,000 kategorija. Budući da treniranje klasifikatora za toliko kategorija iziskuje odviše vremena, promatrane su samo kategorije u prve dvije razine hijerarhijskog stabla, a ima ih 172. Neki primjeri kategorija su *Parlament*, zatim *Prava i slobode* te *Kultura i religija*.

6.2. Eksperimenti i rezultati

6.2.1. Reuters

Kategorizacija *Industries* sadrži mnogobrojne i vrlo srodne kategorije. Postupak klasifikacije je u ovom slučaju često subjektivan i netrivijalan za obavljanje, što se vidi na rezultatima u tablici 6.1. Budući da je točna klasifikacija vrlo rijetka, hibridni klasifikator daje najbolje rezultate: RIPPER i SVM u kombinaciji ostvaruju mnogo veću preciznost nego samostalno.

Tablica 6.1: Reuters - Industries: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.289	0.173	0.099	0.715
RIPPER (hijer.)	0.288	0.166	0.096	0.642
SVM	0.444	0.223	0.129	**0.844**
SVM (hijer.)	0.466	0.256	0.152	0.816
RIPPER i SVM (hijer.)	**0.483**	**0.288**	**0.181**	0.702

Regije je mnogo lakše odrediti zato što novinski članci gotovo uvijek u sebi sadrže imena samih regija o kojima se radi (tablica 6.2). Iako postoji mnogo različitih kategorija, klasifikatori znatno bolje prepoznavaju relevantne regije nego tematske karakteristike članaka.

Tablica 6.2: Reuters - Regions: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.752	0.606	0.459	0.892
RIPPER (hijer.)	0.752	0.606	0.459	0.892
SVM	**0.840**	0.623	0.461	**0.961**
SVM (hijer.)	**0.840**	0.623	0.461	**0.961**
RIPPER i SVM (hijer.)	0.831	**0.660**	**0.525**	0.889

Topics sadrži manji broj kategorija nego prethodne dvije kategorizacije, a posljedica toga su manje mogućnosti za pogrešnu klasifikaciju. Rezultati su dobri (tablica 6.3), a SVM je i u ovom slučaju mnogo uspješniji od RIPPER-a.

Tablica 6.3: Reuters - Topics: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.671	0.398	0.298	0.597
RIPPER (hijer.)	0.667	0.387	0.299	0.549
SVM	0.805	0.551	0.421	**0.798**
SVM (hijer.)	**0.808**	**0.571**	0.446	0.795
RIPPER i SVM (hijer.)	0.774	0.561	**0.493**	0.651

6.2.2. 20 Newsgroups

Glavni problem s kojim se klasifikatori suočavaju su slične kategorije, a u ovom slučaju je i čovjeku teško klasificirati dokumente. Naprimjer, ako se u vreći riječi spominju neke svjetske religije, spada li dokument u *alt.atheism*, *soc.religion.christian* ili *talk.religion.misc*? Na to pitanje je teško odgovoriti bez poznavanja originalnog dokumenta.

Tablica 6.4: 20 Newsgroups: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.627	0.579	0.491	0.704
RIPPER (hijer.)	0.601	0.551	0.497	0.618
SVM	0.781	0.723	0.655	0.807
SVM (hijer.)	**0.805**	**0.763**	0.716	**0.816**
RIPPER i SVM (hijer.)	0.769	0.735	**0.764**	0.707

6.2.3. Vjesnik

Neke kategorije Vjesnikovog korpusa su jednostavne za klasifikaciju (*Hrvatska*, *Sport*, *Kultura*), ali postoje i vrlo zahtjevne, poput *Komentari*, *Sa svih strana* i *Teme dana*. Takve kategorije se zapravo određuju prema načinu pisanja članka, a vrlo teško vrećom riječi, budući da mogu biti o bilo kojoj temi. Rezultati po kategorijama su miješani (ima i dobrih i loših), pa je ukupna ocjena na kraju osrednja (tablica 6.5).

Tablica 6.5: Vjesnik: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.577	0.538	0.457	0.654
RIPPER (hijer.)	0.577	0.538	0.457	0.654
SVM	**0.768**	**0.721**	0.639	**0.828**
SVM (hijer.)	**0.768**	**0.721**	0.639	**0.828**
RIPPER i SVM (hijer.)	0.738	0.707	0.**707**	0.708

6.2.4. Narodne Novine

Iako postoji mnogo kategorija zakonodavnih dokumenata, SVM ih dosta uspješno klasificira (tablica 6.6). U usporedbi s njime, RIPPER nema ni približno dobre rezultate.

Razlog tome se nalazi u činjenici da za neke kategorije postoji relativno malo primjeraka dokumenata, a posljedica su pravila koja obuhvaćaju razne specifičnosti dokumenata umjesto specifičnosti samih kategorija. Tada obično bude izgrađeno malo kratkih pravila koja su nisu dovoljno generalna za kategoriju da bi uspješno klasificirala dokumente iz testnog skupa.

Tablica 6.6: Narodne Novine: vrednovanje klasifikatora

	Mikro F_1	Makro F_1	Makro preciznost	Makro odziv
RIPPER	0.542	0.423	0.300	0.716
RIPPER (hijer.)	0.530	0.412	0.324	0.565
SVM	0.759	0.632	0.492	**0.884**
SVM (hijer.)	**0.767**	**0.653**	0.527	0.855
RIPPER i SVM (hijer.)	0.735	0.637	**0.551**	0.755

6.3. Diskusija

Iz tablice rezultata je vidljivo da je SVM po pitanju točnosti na svakom korpusu superiorniji RIPPER-u. Da će se eventualno doći do tog zaključka je bilo očekivano i od samog početka, prije implementacije i testiranja; rezultati nisu iznenađenje. Iako se kod vrednovanja optimizira F_1-mjera, općenito se parametri algoritama mogu postaviti tako da se preciznost poveća, a odziv smanji; ili obrnuto. Svi korpusi su dizajnirani tako da dokument ili pripada ili ne pripada u kategoriju, ne postoji ništa između. Za mnoge dokumente se može reći da *donekle pripadaju* u neku kategoriju ili pak da se tiču šireg spektra kategorija. Ako klasifikator zaključi da bi dokument mogao pripadati u više kategorija odjednom, za svaku od njih će izjaviti da tamo i pripada. Iako je klasifikacija subjektivna, *dobar pokušaj* klasificiranja se jednako kažnjava kao i potpuno pogrešno klasificiranje.

Primjer: novinski članak piše o ljetovanju poznatog glumca na hrvatskoj obali. Točna novinska kategorija je *Život*. Klasifikator stavlja članak u kategorije *Život*, *Kultura* (spominje se glumac) i *Hrvatska* (spominje se hrvatska obala). To rezultira jednom točnom i dvjema pogrešnim klasifikacijama, ali se boduje jednako kao i klasifikacija: *Život*, *Crna kronika*, *Sport*. Očito je da je prvi primjer klasifikacije mnogo bolji od drugog, no to je primjer neadekvatnosti u sustavu ocjenjivanja kojeg treba uzeti u obzir.

Ručnom analizom generiranih pravila se vidi da su zbilja razumljiva i imaju smisla,

a uočavaju se i neki njihovi nedostaci, osobito prevelika specifičnost u nekim slučajevima. Primjer koji to ilustrira je kategorija *Zagreb i Županija* iz Vjesnikovog korpusa. Većina generiranih pravila za nju je logična, međutim pojavljuje se i nekolicina neobičnih. Neka od tih neobičnih su sljedeća:

```
centar invalid nalaziti
gradski sat ured zdravstvo
gradski gradonačelnica zamjenik
ulica komunalan
srijeda plućan
zelen skijati
biometeorološki
dioksid plućan
kreativan eko
mališani
```

Navedena pravila se uopće ne tiču grada Zagreba niti Zagrebačke županije, nego igrom slučaja ispada da se te riječi nalaze u dokumentima za trening koji su označeni tom kategorijom. Osim toga, nema dokumenata koji sadrže te riječi i istovremeno pripadaju nekoj drugoj kategoriji da bi se ova besmislena pravila eliminirala. Takva pravila su osobito loša jer kad god na skupu dokumenata za testiranje upale, u većini slučajeva se ne radi o pozitivnom nego negativnom dokumentu.

Hijerarhijska metoda klasifikacije uz SVM povećava preciznost, što ima smisla budući da se klasifikator fokusira na klasifikaciju u uže skupove kategorija. Međutim, hijerarhijska metoda uz RIPPER smanjuje preciznost, suprotno početnim očekivanjima. U takvoj inačici algoritma generiran skup pravila je manji nego inače i sadrži manje specifičnosti kategorija. Ovo polje ostavlja mnogo prostora za daljnju improvizaciju, ali u radu je korištena samo jednostavna i naivna vrsta hijerarhijske klasifikacije.

Hibridni klasifikator kao kombinacija RIPPER-a i hijerarhijskog SVM-a je dao malo slabije ukupne rezultate nego sam hijerarhijski SVM. Hibrid eksperimentalno daje vrlo visoku razinu preciznosti, iako je s druge strane vrijednost odziva manja. Radi tog svojstva se u praksi taj klasifikator nameće kao zgodno rješenje u primjenama gdje je potrebno samo sugerirati klasifikaciju dokumenta umjesto da ga se konačno svrsta ili ne svrsta u kategoriju. Može primjerice služiti kao pomoćni alat u ručnoj klasifikaciji radi ubrzanja posla ili kao međuprocedura u nekom većem postupku analize teksta, kao što su algoritmi za dohvat informacija.

Jedna od ideja za poboljšanje RIPPER-a, razmatrana još i prije implementacije, je

za svako pravilo (ili riječ unutar pravila) odrediti mjeru pouzdanosti. Motiv za takvu modifikaciju algoritma je sugestija da ne pridonose sva pravila jednako točnosti algoritma. Primjerice, više slabijih pravila bi moglo imati jednaku snagu kao manje jačih pravila. Zbrajanjem pouzdanosti pravila (ili drugim načinom kombiniranja) se izračuna mjera pripadnosti u kategoriju, pa ako je ona veća od specifične granice, onda se može dokument klasificirati u kategoriju.
Bilo je pokušaja implementacije ove ideje na razne načine, ali bezuspješno: točnost se ili nije promijenila, ili još gore: smanjila se. Dva su razloga za to kriva:

1. Teško je precizno odrediti pouzdanost pravila. Da bi se tako nešto izračunalo, potrebne su velike količine dokumenata za treniranje ili možda i još neki drugi izvor podataka na temelju kojeg bi se izračunale pouzdanosti. Iako kategorije uvijek imaju određene sebi svojstvene ključne riječi, one obično odmah impliciraju kategoriju, što znači da im je teško staviti pouzdanost na nekakvu skalu. U svakom slučaju, ovo je netrivijalan problem.

2. Kod klasifikacije dokumenata obično upali samo nekolicina pravila; u većini slučajeva samo jedno ili dva. Ako pravila imaju specificiranu pouzdanost, njihova kombinacija ne daje mnogo informacija. U gotovo svakom slučaju samo jedno od tih pravila koja su upalila će mjerom pouzdanosti dominirati nad ostalima. Stoga se u suštini niti ne isplati kombinirati pouzdanosti pravila jer je svejedno razmatraju li se zasebno ili ne.

Klasifikacija uz isprobane algoritme računanja pouzdanosti pravila se odmah pokazala kao promašaj. To ne znači da je ideja sama po sebi pogrešna, nego se mogućnost da se RIPPER unaprijedi na neki sličan način ostavlja otvorenom.

7. Zaključak

Ovaj rad se bavi klasifikacijom dokumenata uz pomoć pravila izgrađenih strojnim učenjem. Implementiran je program koji za zadani korpus izgrađuje pravila za klasifikaciju njegovih dokumenata. Učinkovitost klasifikatora baziranog na tim pravilima je uspoređena sa klasifikatorom baziranim na stroju s potpornim vektorima (engl. *support vector machine – SVM*).

Klasifikacija dokumenata uz pomoć pravila i klasifikacija uz pomoć SVM-a nemaju iste primjene; pravila je moguće ručnom intervencijom uređivati i poboljšavati. Kako se klasifikacija uz pomoć pravila ponaša u praksi nakon nekog broja ručnih adaptacija promašenih klasifikacija ostaje otvoreno pitanje, ovaj rad ga nije razriješio. Možda je moguće uz razumno malo truda ručnim dotjerivanjem strojno naučenih pravila postići točnost na razini SVM-a; ili čak i višu. No to nije lako eksperimentalno ispitati unutar okvira ovog rada.

Hijerarhijske inačice algoritama su ovisno o klasifikatoru pomogle ili odmogle u cilju povećanja točnosti. Iskorištavanje hijerarhijske strukture je bio dobar pokušaj, no čini se da treba uložiti više truda da bi taj pristup bio uspješan kod strojno učenih pravila pomoću RIPPER-a. U principu nema razloga da hijerarhijska inačica odmaže jer teoretski samo klasifikatoru olakšava problem; ovo je potrebno bolje istražiti.

Još jedan prijedlog za poboljšanje klasifikacije pomoću pravila je bolje pretprocesiranje dokumenata. Npr. mogao bi se svakoj riječi dodijeliti faktor važnosti na temelju odnosa broja pojavljivanja unutar dokumenta i ukupnog broja pojavljivanja u svim dokumentima. U svakom slučaju, pravila se mogu koristiti i praktična su kao baza (početno rješenje) u problemu inženjeringa znanja, što zapravo i jest glavna intencija klasifikacije pomoću pravila. Algoritam RIPPER je dao zadovoljavajuće rezultate, no njime generirana pravila sama po sebi uglavnom nisu dovoljno točna za praktičnu uporabu. Uz to treba napomenuti da je ovdje implementiran algoritam veoma jednostavan. Bolje varijante su dobro istražene, a kad bi bile implementirane, nudile bi i bolje rezultate. Koliko bolje teško je reći, ali sama klasifikacija pomoću pravila gotovo sigurno neće nadmašiti klasifikaciju pomoću SVM-a.

LITERATURA

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, i Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Hrvatska izvještajna novinska agencija, 2014. URL `http://www.hina.hr/Free/POOL`.

Pier Luca Lanzi. Machine Learning and Data Mining: 12 Classification Rules, 2007. URL `http://www.slideshare.net/pierluca.lanzi/machine-learning-and-data-mining-12-classification-rules`.

D. D. Lewis, Y. Yang, T. Rose, i F. Li. Rcv1: A new benchmark collection for text categorization research. 2004.

Tim Menzies. Data mining: Rules, 2006. URL `http://csee.wvu.edu/~timm/cs591o/old/Rules.html`.

Jason Rennie. 20 Newsgroups, 2014. URL `http://qwone.com/~jason/20Newsgroups/`.

Fabrizio Sebastiani. Text categorization. 2002.

Jan Šnajder, B Dalbelo Bašić, i Marko Tadić. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing & Management*, 44 (5):1720–1731, 2008.

TNM003: Introduction to Data Mining. TNM003: Introduction to Data Mining. 2006. URL `http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec4.pdf`.

Strojno učenje pravila za klasifikaciju dokumenata

Sažetak

Klasifikacija dokumenata je jedan od osnovnih i najvažnijih problema analize tekstnih dokumenata. Najčešće metode se baziraju na vektorskoj reprezentaciji vreće riječi karakteristične dokumentu i vrlo su učinkovite. Međutim, način rada takvih strojno učenih modela je obično izrazito težak za tumačenje i ručno uređivanje. Zbog toga se predlaže drugi pristup klasifikaciji, a to je uz pomoć pravila. Prednost pravila jer što ih korisnik može lako interpretirati i uređivati. Rad se bavi klasifikacijom uz pomoć pravila u kombinaciji sa strojnim učenjem te uspoređuje učinkovitost sa klasifikatorom baziranim na SVM-u.

Ključne riječi: strojno učenje, klasifikacija dokumenata, RIPPER, SVM, pravila, lematizacija

Machine Learning of Document Classification Rules

Abstract

Document classification is one of basic and most important problems of textual document analysis. Most common methods are based on vector representation of words (bag of words) and are very effective. However, trained models developed by machine learning are difficult to interpret and edit by hand. Therefore, a different approach is suggested, namely, rule based classification. The advantage of rules is their simplicity; they are easy to interpret and edit. This paper discusses rule based classification in combination with machine learning and compares its effectiveness with an SVM based classifier.

Keywords: machine learning, document classification, RIPPER, SVM, rules, lemmatization