Laboratorij za analizu teksta i inženjerstvo znanja – TakeLab

Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikrolektroniku, računalne i inteligentne sustave
Unska 3, 10000 Zagreb, Hrvatska

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 457

Prepoznavanje i klasifikacija imenovanih entiteta u tekstovima na hrvatskome jeziku

Tomislav Lombarović

Zagreb, lipanj 2012.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 5. ožujka 2012.

DIPLOMSKI ZADATAK br. 457

Pristupnik: **Tomislav Lombarović**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Prepoznavanje i klasifikacija imenovanih entiteta u tekstovima na hrvatskome jeziku**

Opis zadatka:

Prepoznavanje i klasifikacija imenovanih entiteta jedan je od glavnih zadataka ekstrakcije informacija. Postupci temeljeni na pravilima iziskuju ručno oblikovanje većeg broja pravila, dok postupci temeljeni na strojnome učenju iziskuju ručno označavanje velikog skupa dokumenata. Hibridni postupci, koji kombiniraju pravila i metode strojnog učenja, u načelu postižu najbolje rezultate. Problem ručnog označavanja može se ublažiti primjenom polunadziranih metoda strojnog učenja.
U okviru diplomskog rada potrebno je proučiti postupke za prepoznavanje i klasifikaciju imenovanih entiteta. Posebno proučiti metode polunadziranog strojnog učenja i njihovu primjenu u zadatcima obrade prirodnog jezika. Razraditi i implementirati hibridni postupak za prepoznavanje i klasifikaciju imenovanih entiteta u tekstovima na hrvatskome jeziku. Postupak treba kombinirati metode temeljene na popisima i ručno oblikovanim pravilima s polunadziranim strojnim učenjem temeljenima na klasifikatoru maksimalne entropije (MaxEnt), stroju s potpornim vektorima (SVM) i skrivenim Markovljevim modelom (HMM). Provesti označavanje odgovarajućeg skupa za učenje prema normi MUC-7 i odabrati najprikladnije značajke uzevši u obzir ograničenost jezičnotehnoloških alata za hrvatski jezik. Provesti eksperimentalno vrednovanje točnosti ekstrakcije uporabom različitih metoda strojnog učenja, analizu značajki, analizu pogrešaka, analizu različitih načina hibridizacije i analizu utjecaja veličine početnog skupa za učenje. Radu priložiti izvorni programski kod, programsku dokumentaciju i označene skupove podataka.

Zadatak uručen pristupniku: 9. ožujka 2012.
Rok za predaju rada: 21. lipnja 2012.

Mentor:

Doc.dr. sc. Jan Šnajder

Djelovođa:

Prof.dr.sc. Domagoj Jakobović

Predsjednik odbora za diplomski rad profila:

Prof.dr.sc. Siniša Srbljić

Zahvaljujem mentoru
doc. dr. sc. Janu Šnajderu, na podršci, izvrsnim savjetima i pruženoj pomoći prilikom izrade ovoga rada, kao i tijekom cijelog diplomskog studija.

SADRŽAJ

1. Uvod

Pretraživanje i strojno razumijevanje informacija danas je, s obzirom na količinu i sve veći porast količine dostupnih informacija, vrlo važno područje računarstva. Web je primjer takvog repozitorija koji sadrži velike količine dokumenata iz najrazličitijih područja ljudske djelatnosti. Da bi se te informacije mogle učinkovito koristiti potrebno je razviti tehnologije za pretraživanje i dohvaćanje ciljanih informacija. Danas su na webu najpopularnije tražilice koje se zasnivaju ne pretraživanju informacija prema ključnim riječima. Uz pretraživanje po ključnim riječima mogu se koristiti i dostupni meta-podaci, kao i informacije o međusobnoj povezanosti stranica (engl. *hyperlinks*). Takve tražilice pretražuju nestrukturirane podatke i kao rezultate pretrage vraćaju rangirane popise dokumenata.

Međutim, za inteligentno pretraživanje i ekstrakciju informacija potrebno je dublje razumjeti sadržaj dokumenta. Bitan korak u razumijevanju sadržaja dokumenta jest i prepoznavanje imenovanih entiteta (engl. *named entities*) o kojima se u tekstu govori. Ti entiteti mogu biti imena osoba, organizacija, mjesta, vremenski izrazi i slične informacije koje imaju jasno definiranu semantiku. Informacije o takvim entitetima mogu se koristiti za naprednije pretraživanje dokumenata i dohvat ciljanih informacija. Također, moguće je donositi različite zaključke o entitetima i njihovim međusobnim vezama.

Problem prepoznavanja takvih entiteta naziva se prepoznavanje i klasifikacija imenovanih entiteta (engl. *named entity recognition and classification - NERC*). Prepoznavanje se odnosi na određivanje položaja entiteta u tekstu – određivanje početka i kraja pojedinog entiteta. U procesu klasifikacije određuje se tip, odnosno klasa, kojoj entitet pripada (npr. ime osobe, organizacije, vremenski izraz i sl.).

Skupovi entiteta koji se prepoznaju mogu biti različiti, ovisno o domeni primjene sustava koji se razvija. U medicini to primjerice mogu biti nazivi bolesti, nazivi proteina u bioinformatičkim sustavima, ili nazivi raznih proizvoda ili kompanija u sustavima koji bi se koristili za istraživanje tržišta. Često korišten skup općenitije primjene sadrži sljedećih 7 entiteta: osoba, lokacija organizacija, datum, vrijeme, postotak i valutni iz-

raz. Ovaj skup biti će korišten i u ovom radu.

Za rješavanja problema mogu se koristiti različiti pristupi. To mogu biti sustavi temeljeni na pravilima, popisima entiteta, različiti oblici strojnog učenja ili hibridni sustavi koji kombiniraju više pristupa. Loše svojstvo sustava temeljenih na pravilima jest potreba za njihovom izmjenom tijekom vremena, ili za primjene u drugim domenama, dok se sustavi temeljeni na strojnom učenju mogu lakše prilagođavati promjenama.

U okviru ovog rada bit će proučeni različiti pristupi rješavanju problema pronalaska i označavanja entiteta, biti će izgrađen skup podataka za učenje i vrednovanje sustava te izgrađen hibridni sustava za prepoznavanje i klasifikaciju entiteta u tekstovima na hrvatskom jeziku. Središnji dio sustav bit će zasnovan na metodama strojnog učenja, a za učenje će se koristi polunadzirano učenje.

U slijedećem poglavlju dan je pregled područja i sličnih radova koji se bave problemom prepoznavanja i klasifikacije imenovanih entiteta. U trećem poglavlju bit će ukratko opisan način označavanja i skupovi za učenje sustava. Nakon toga, u četvrtom poglavlju, slijedi opis metoda korištenih u izradi sustava. U petom poglavlju bit će opisana implementacija sustava. Zatim slijede detaljni rezultati vrednovanja sustava. U posljednjem, sedmom poglavlju, nalazi se zaključak rada.

2. Pregled područja i slični radovi

Prepoznavanje i klasifikacija imenovanih entiteta već je duže vrijeme popularan problem u području obrade prirodnog jezika. Naime, izdvajanje takvih informacija iz nestrukturiranih tekstova važan je korak u razvoju naprednijih sustava za ciljani dohvat i razumijevanje informacija. Veće zanimanje znanstvene zajednice za temu započelo je 1995. godine kada je u sklopu 6. po redu *Message Understanding Conference MUC-6* (Grishman i Sundheim, 1995) jedan od zadatak bio upravo prepoznavanje i klasifikacija imenovanih entiteta. U sklopu te konferencije definiran je i pojam *named entity* (Grishman i Sundheim, 1996) koji se danas često koristi u području obrade prirodnog jezika.

Za engleski jezik do sada su razvijeni brojni sustavi za označavanje i klasifikaciju imenovanih entiteta u tekstovima različitih domena, kao i oni za označavanje općenitih (open domain) tekstova (Florian et al., 2003), (Isozaki i Kazawa, 2002). Uz prije spomenuti MUC-6, nekoliko je konferencija bilo posvećeno rješavanju ovog problema prvenstveno za engleski jezik, ali i za druge jezike: MUC-7, ACE Evaluation i CoNLL-2003 (uz engleski, evaluacija i za njemački jezik). Na CoNLL-2003 provedeno je vrednovanje za španjolski i nizozemski jezik.

2.1. Klase entiteta

Prilikom izrade sustava NERC potrebno je najprije odlučiti koje klase entiteta će biti označene u tekstovima. U prvim je sustavima zadatak bio definiran pronalaženjem *vlastitih imena* u tekstovima, a imena su najčešće bila podijeljena u tri skupine: imena osoba, lokacija i organizacija. Ova je skupina poznata pod nazivom *enamex*, a taj naziv počeo se koristiti na konferenciji MUC-6. Skupine se dodatno mogu dijeliti u finije klase stvarajući taksonomije klasa. Primjerice, osobe se mogu podijeliti u imena političara, glumaca, pjevača i sl. (Fleischman i Hovy, 2002), a lokacije u nazive gradova, država, rijeka, planina, ulica i sl. (Lee i Lee, 2005).

Druge dvije važne skupine entiteta su *numex* i *timex*. Skupina *numex* sastoji se

od postotnih i valutnih izraza. Postotni izraz je svaki izraz koji označava postotnu vrijednost ili raspon vrijednosti, a valutni izraz označava određenu količinu ili raspon količine novca u nekoj valuti. Skupina timex također se sastoji od dviju klasa: datum i vrijeme. Klasa datum označava datume bez određenog vremena u danu, dok skupina vrijeme označava vrijeme u danu.

Osim opisanih općenitih klasa entiteta, sustavi s posebnim primjenama mogu koristiti domenski specifične klase entiteta. U bioinformacijskim sustavima to na primjer mogu biti klase *DNA*, *PROTEIN* i *TIP STANICE* koji su korišteni u (Shen et al., 2003). Za filmsku domenu klase mogu biti naziv filma, ime glumca i sl. (Etzioni et al., 2005).

2.2. Metode prepoznavanja i klasifikacije entiteta

Metode za rješavanje problema mogu se podijeliti u dvije skupine: metode temeljene na pravilima i metode temeljene na strojnom učenju. Metode temeljene na pravilima češće su korištene u početku razvoja područja, dok se u novije vrijeme sve više koriste metode strojnog učenja (Nadeau i Sekine, 2007). Osnovna prednost metoda temeljenih na pravilima jest u činjenici da nisu potrebni označeni podaci za učenje, no takve sustave je potrebno ručno prilagođavati za nove domene ili taksonomije klasa. S druge strane, sustavi temeljeni na strojnom učenju mogu se jednostavno trenirati na novim podacima i s novim klasama, ali je potrebno imati označene podatke.

Sustavi temeljeni na strojnom učenju mogu koristiti različite tehnike učenja. Metode nadziranog učenja koriste samo označene podatke. Dosta često korištena metoda jest stroj s potpornim vektorima (Kazama et al., 2002), (Mayfield et al., 2003), (Isozaki i Kazawa, 2002). Problem kod ovog pristupa jest relativno sporo učenje SVM-a, a (Isozaki i Kazawa, 2002) se bavi optimizacijom SVM-a za ovaj problem.

Na konferenciji CoNNL 2003 (Sang i Meulder, 2003) sudjelovalo je 16 sustava od kojih je velika većina bila temeljena na strojnom učenju. Najuspješnijima su se pokazali sustavi temeljeni na klasifikatoru maksimalne entropije (engl. *maximum entropy classifier*), a čak 5 sustava se temeljilo na toj metodi. Svi najbolji sustavi koristili su i nekakav dodatan izvor informacija uz podatke za učenje, većina popis pojmova, a neki sustavi i izlaze drugih NERC označivača (Sang i Meulder, 2003). Najbolje rezultate dao je sustav temeljen na kombiniranju nekoliko metoda: MaxEnt, *transformation-based learningn*, HMM i *robust risk minimization*, kod kojih je F1 mjera iznosila 88.76% (*exact-match*). Drugi rezultat za engleski jezik postigao je sustav (Chieu i Ng, 2003) u kojem je korišten klasifikator maksimalne entropije, a F1 mjera iznosila je 86.84% (*exact-match*). Najbolji rezultat potignut na konferenciji MUC-7 je (Mikheev

et al., 1998), a F1 mjera iznosi 93.39%.

Za učenje je moguće kombinirati označene i neoznačene dokumente, a metode koje se temelje na toj ideji pripadaju skupini metoda polunadziranog strojnog učenja (engl. *semi-supervised learning*). Jedna od metoda polunadziranog učenje jest samoučenje (engl. *self learning*). Kod primjene ove metode klasifikator se najprije uči na manjem, označenom skupu, a zatim se, koristeći taj klasifikator, označavaju neoznačeni podaci i među njima se izabiru novi primjeri za učenje. U radu (Ji i Grishman, 2006) autori se bave proučavanjem metoda za izbor novih primjera za učenje. Korištenjem neoznačenih podataka F1 mjera sustava poboljšana je za 1.7.

Za temu ovog rada posebno su važni sustavi za tekstove na hrvatskome jeziku. Do sada je razvijen jedan sustav za označavanje imenovanih entiteta u tekstovima na hrvatskom jeziku (Bekavac i Tadić, 2007). Sustav je razvijen na Filozofskom fakultetu u Zagrebu a zasniva se na skupu ručno izrađenih pravila. Pravila su dobivena iz podskupa "Hrvatskog Nacionalnog Korpusa"(Tadić, 2002), a F1 mjera performansi sustava na novinskim tekstovima iznosi 90%, što je vrlo dobar rezultat.

2.3. Vrednovanje sustava

Za vrednovanje NERC sustava koristi se nekoliko mjera. Najjednostavniji način vrednovanja pod nazivom *exact-match* pozitivno boduje izlaz klasifikatora samo u slučaju kada su točno određene i granice i klasa nekog entiteta. Ova metoda korištena je na evaluacijama IREX i CONLL. Na evaluacijama MUC korištena je druga metoda koja najprije zasebno boduje točno označavanje granice i klase entiteta (Grishman i Sundheim, 1995). Točnim određivanjem klase smatra se označavanje u kojem pravi i označeni entitet imaju preklapanje u barem jednom tokenu, a označena je ispravna klasa entiteta. Točnim određivanjem granica smatra se označeni entitet čiji se tokeni u potpunosti podudaraju s tokenima originalnog entiteta, bez obzira je li entitet označen ispravnom klasom. Konačan rezultat vrednovanja jest mikro-uprosječeni F1 za sve entitete.

3. Podaci za učenje i vrednovanje sustava

Podaci za učenje i vrednovanje sustava prikupljeni su s portala *Poslovni.hr*. Za učenje su korišteni označeni i neoznačeni članci. Ukupno je označeno 210 članaka, od kojih se 140 koristiti za učenje, 40 za vrednovanje sustava te 30 za provjeru tijekom učenja modela. Podjela skupova izvršena nasumičnim odabirom. Članci su uglavnom poslovne tematike.

Označeni skup sadrži ukupno 73 295 tokena, što je prosječno 354 tokena po dokumentu. U tekstovima je označeno 4 717 imenovanih entiteta, odnosno 22.78 entiteta po dokumentu. Detalje raspodjele entiteta po klasama prikazuje Tablica 3.1.

Tablica 3.1: Raspodjela entiteta

	Osoba	Lokacija	Org.	Val. izraz	Postotak	Datum	Vrijeme
#	706	834	1979	460	463	266	9
Udio [%]	14.96	17.68	41.95	9.76	9.81	5.64	0.19

Budući da se sustav temelji na polunadziranom strojnom učenju, za učenje su korišteni i neoznačeni podaci. Korišteno je ukupno 1000 neoznačenih članaka koji sadrže 349 481 tokena, što prosječno iznosi 349.48 tokena po dokumentu.

3.1. Označavanje skupova

Članci su označeni prema preporukama MUC-7 (Chinchor, 1998). Ukupno je definirano sedam različitih entiteta i pripadnih oznaka podijeljenih u tri skupine. U nastavku će biti detaljnije opisani entiteti i pravila za njihovo označavanje.

3.1.1. ENAMEX skupina

Skupina definira tri tipa entiteta: osoba, organizacija i lokacija.

Osoba - označava ime osobe u tekstu. Ime ne mora nužno biti puno ime i prezime, označava se svako pojavljivanje bilo imena ili prezimena, ili punog imena osobe. Puno ime koje uključuje ime i prezime označava se kao jedan entitet. Titule poput *dr.* ili *mr.* ne smatraju se dijelom entiteta. Oznaka za ime osobe izgleda ovako:

```
<enamex type="Person">Ivo Ivić</enamex>
```

Imena osoba koja u tekstu ne predstavljaju osobe se ne označavaju, npr. *Alzheimer* u izrazu *Alzheimerova bolest* se ne označava.

Lokacija - označava pojavljivanje bilo kakve lokacije u tekstu. Lokacija može biti nazivi države ili grada, ulica, trg, kao i ostali geografski pojmovi poput naziva kontinenta, rijeka i planina. Oznaka je opisana ovako:

```
<enamex type="Location">Hrvatska</enamex>
```

Organizacija - označava dobro definirani naziva bilo kakve organizacije, npr.: nazivi tvrtki, obrazovnih ili državnih institucija i sl. Oznaka za entitet je:

```
<enamex type="Organization">Hrvatska narodna banka</enamex>
```

Kako je spomenuto, radi se on najsloženijoj grupi entiteta i ovdje će biti opisane neke konvencije korištene prije označavanju.

Entitet mora biti dobro definiran, odnosno mora biti jasno o kojem se točno entitetu radi. U tekstu “Vlada je danas potpisala ugovor...” riječ “Vlada” ne smatra se entitetom jer nije dobro definirano od kojoj se vladi radi. U rečenici “Hrvatska Vlada je danas potpisala ugovor...” označava se entitet “Hrvatska Vlada”:

“<enamex type="Organization">Hrvatska Vlada</enamex> je danas potpisala ugovor...”

Ako se dio jednog entiteta može sam za sebe smatrati entitetom, on se svejedno ne označava posebno. Tako u prethodnom primjeru riječ “Hrvatska” ne označavamo kao zaseban entiteta tipa *Location*. Entiteti koji se pojavljuju kao dio metonima također trebaju biti označeni, npr. u rečenici “Službeni Zagreb odlučio je povisiti....” označava

se entitet *Zagreb* oznakom *Lokacija*.

3.1.2. NUMEX skupina

Skupina definira dva tipa entiteta: postotak i valutni izraz.

Postotak - označava postotni izraz. Oznaka se zapisuju ovako:

```
<numex type="Percent">50 %</numex>
```

Postotni izrazom se smatra na bilo koji način zapisani postotni izraz. Dio koji označava vrijednost može biti zapisan kao broj ili riječ, a oznaka postotka korištenjem oznake %, ili riječi *posto*, *postotni*, *postotaka* i sl. Također, može se raditi i o rasponu vrijednosti, a cijeli raspon se označava kao jedan entitet. Npr.

```
<numex type="Percent">pet do 67 posto</numex>
<numex type="Percent">4 ili 5%</numex>
```

Valutni izraz - označava količinu novca, dakle radi se o kombinaciji brojevnog izraza i identifikatora valute. Zapisuje se ovako:

```
<numex type="Money">105 kuna</numex>
```

Vrijede slična pravila kao i kod postotnih izraza, rasponi se označavaju kao jedna entitet:

```
<numex type="Money">dva do tri milijuna kuna</numex>
<numex type="Money">15 ili 50 $</numex>
```

3.1.3. TIMEX skupina

Skupina definira dva tipa entiteta: datum i vrijeme.

Datum - označava datum, bez vremena u danu. Datum ne mora biti potpuno definiran u smislu da sadrži dan, mjesec i godinu. Primjer oznake:

```
<timex type="Date">5.  1.  2011.  godine</timex>
```

Dopušteni su proizvoljni formati zapisa, npr. zapis mjeseca u obliku riječi kao i rasponima datuma:

```
<timex type="Date">srpnju 2010.</timex>
<timex type="Date">srpnja do prosinca 2013.  godine</timex>
```

Vrijeme - označava izraz za vrijeme u danu. Oznaka je:

```
<timex type="Time">17:30 sati</timex>
```

Opet slično kao i kod datuma, dopušteni su proizvoljni zapisi vremena i rasponi koji se smatraju jednim entitetom:

```
<timex type="Time">17:00 do 19:30</timex>
<timex type="Time">19 - 22:30</timex>
```

Vremenskim izrazima smatraju se i izrazi poput “treći kvartal 2010. godine”, “21. stoljeće”. S druge strane, vremenske oznake koje ne definiraju točno vrijeme se ne označavaju: “prije nekoliko sati”, “jučer ujutro” i sl.

4. Sustava za označavanje i klasifikaciju

4.1. Uvod

Pristup rješenju problema u ovome radu zasniva se na kombinaciji nekoliko metoda koje se često koriste za prepoznavanje i klasifikaciju imenovanih entiteta. Kao važan izvor informacija koristiti će se popisi pojmova za tri osnovna tipa entiteta: osobe, lokacije i organizacije. Međutim, pokazalo se da označavanje koje se zasniva samo na popisima ne može ostvariti dovoljno dobre performanse.

Središnji dio sustava biti će utemeljen na metodama strojnoga učenja. Prednost korištenja metoda strojnog učenja jest u laganoj prilagodljivosti, za razliku od metoda temeljenih na pravilima. Naime, ručno pisana pravila moraju se ručno prilagođavati, npr. za druge domene, dok se sustavi temeljeni na učenju mogu ponovno učiti nad novim podacima, bez potrebe za mijenjanjem samog sustava. U sklopu rada biti će proučena tri klasifikatora: klasifikator maksimalne entropije, stroj s potpornim vektorima i klasifikator temeljen na skrivenim Markovljevim modelima. Za učenje klasifikatora biti će primijenjena metoda samoučenja (engl. *self training*). To je metoda polunadziranog strojnog učenja, pa će se na taj način moći iskoristiti dostupnost velike količine neoznačenih podataka.

Treći dio sustava čine pravila. To su dodatna pravila kojima se mogu razriješiti određene nejednoznačnosti te regularni izrazi. Regularni izrazi biti će korišteni za prepoznavanje entiteta iz skupine NUMEX i TIMEX, dakle entiteta koji imaju dobro definiranu strukturu pa su regularni izrazi pogodna metodama za prepoznavanje takvih entiteta.

4.2. Popisi

Popisi su važan izvor informacija o entitetima te će u radu biti korišteni kao jedna od značajki na ulazu klasifikatora. Iako će sustav raspoznavati sedam vrsta entiteta, bit će korištena tri popisa: osobe, lokacije i organizacije, dok za preostale tipove popisi nemaju smisla (postotak, valutni izrazi, datum i vrijeme).

Popis lokacija izgrađen je većim dijelom iz popisa hrvatskih naselja, a uz to je dodan popis svih država, glavnih gradova i ostalih gradova s više od 100 000 stanovnika. Nakon obrade popis sadrži oko 8 850 lokacija.

Popis organizacija izgrađen je uglavnom iz popisa hrvatskih tvrtki, a uz njih su dodane i veće svjetske tvrtke. Popisi većih tvrtki izvan Hrvatske dobiveni su iz popisa tvrtki prema zaradama koji su dostupni na webu. Npr. popis 500 najvećih tvrtki u SAD-u prema prihodu *FORTUNE 500* [1]. Nakon obrade popis sadrži 252 680 pojmova uključujući akronime organizacija. Akronimi su generirani iz naziva organizacije uzimanjem prvog slova svake riječi u nazivu.

Popis osoba izgrađen je iz popisa imena osoba koje žive u Hrvatskoj. U popisu nisu sadržana puna imena i prezimena nego su puna imena podijeljena na imena i prezimena te tako dodani u popis. Ovaj popis sadrži oko 22 330 imena i prezimena. U njega nisu dodana ona imena i prezimena koja se pojavljuju previše rijetko. Naime, takva imena zbog vrlo rijetkog pojavljivanja ne mogu značajno doprinijeti performansama sustava, a mogu doprinijeti lažnom prepoznavanju riječi koje su slične takvom imenu ili prezimenu. Prag za prezimena jest 10 pojavljivanja, a za imena 30 pojavljivanja.

Moguće je odmah primijetiti jedan nedostatak korištenja popisa. Naime, dok su popisi lokacija i osoba, odnosno njihovih imena, relativno nepromjenjivi u vremenu, popis organizacija bitno se mijenja svakodnevnim nastajanjem i nestajanjem organizacija. Sustav koji bi se temeljio samo na popisima morao bi redovito biti nadograđivan nazivima novonastalih organizacija kako bi ih mogao prepoznati u tekstu.

4.2.1. Označavanje korištenjem popisa

Označavanje korištenjem popisa kao predgledne tablice (engl. *Lookup table*) daje vrlo loše rezultate i nije prikladno za označavanje pojmova u prirodnom tekstu. U tekstu se pojmovi često pojavljuju u različitima morfološkim oblicima, pa samo pogled u tablicu pojmova ne daje dobre rezultate. Npr. i za vrlo jednostavne primjere poput:

[1] `http://money.cnn.com/magazines/fortune/fortune500/2011/full_list/index.html`

"Managment *Agrokora* uputio je zahtjev za kupnjom 50% dionica *Mercatora*."
pogled u popis ne bi prepoznao niti jedan od dva entiteta koji se spominju u tekstu. Još složeniji primjeri su oni koji se sastoje od nekoliko riječi, gdje svaka od riječi može biti različita od riječi u pravom nazivu:

"Danas smo posjetili *Željka Rohatinskog*, guvernera *Hrvatske narodne banke*. "
u ovom primjeru također su dva entiteta, od koji je svaki složen, a svaka od riječi koja čini entitet je promijenjena u odnosu na izvorni oblik pojma.

4.2.2. Pretraživanje popisa

Zbog opisanih problema pretraživanje popisa mora biti ostvareno tako da može dohvatiti pojmove uzevši u obzira da će se oni u tekstu vrlo često pojaviti u drugačijem obliku.

U prvom kraku pojam se pokušava pronaći u izvornom obliku, a ako to ne uspije prelazi se na složenije postupke kako bi se pokušao dohvatiti izvorni pojam. U prvom koraku se pojam dijeli u riječi te se dohvaćaju svi pojmovi koji počinju s istim troslovnim prefiksom kao i prva riječ iz pojma koji se traži. Dohvaćanje kandidata korištenjem prefiksa pogodno je za pojmove na hrvatskom jeziku, jer se morfološke promjene na riječima vrlo rijetko događaju u prefiksu riječi. Nakon dohvaćanja pojmova kandidata, uspoređuju se riječ po riječ u pojmu iz rječnika i pojmu za označavanje. Najprije se izvorna riječ pokušava dobiti iz promatrane odbijanjem čestih nastavaka ili poništavanjem čestih promjena koje nastaju na kraju riječi. Taj se popis sastoji od tridesetak nastavaka koji su ručno, a izrađen je ručno. Ako ni tada riječ nije prepoznata, provjerava se imaju li riječi jednak korijen i dovoljno malu *Levenshteinovu* udaljenost i u slučaju da uvjeti nisu zadovoljeni zaključuje se da se riječi ne podudaraju. Ako se pojam sastojao od više riječi kandidat se odbacuje čim mu se ne podudaraju dvije riječi na istim pozicijama.

Ovakav način pretraživanja omogućiti će dohvat nekih pojmova koji nisu mogli biti dohvaćeni izravno, ali također postoje i problemi. Ako se dopuste prevelike promijene na riječima, pojavit će se problem čestog lažnog prepoznavanja pojmova. S druge strane, ako se dopuste samo malene promjene na riječima češće će biti situacije da se ne može prepoznati pojam koji se zaista pojavio u tekstu.

Još jedan problem su imena tvrtki, koja u izvornom obliku sadrže i kraticu za oblik tvrtke: *d.d.* za dioničko društvo i *d.o.o.* za društvo s ograničenom odgovornošću. U tekstovima se imena tvrtki pojavljuje sa tim nastavkom ili bez njega, a vrlo se često koristi i akronim umjesto naziva. Ovi problemi relativno se lako rješavaju tako da se

prilikom pretraživanja dopusti nedostatak nastavaka *d.o.o* i *d.d.* u nazivu. Pojavljivanje akronima je riješeno dodavanjem mogućih akronima u popis. Akronimi su generirani iz naziva tvrtki uzimajući početna slova riječi u nazivu organizacije.

4.2.3. Početno slovo

Budući da većina promatranih entiteta mora započeti velikim početnim slovom, korisna je informacija o početnom slovu pojma koji se želi označiti. Međutim, problem koji se javlja su riječi koje se nalaze na početku rečenice, pa samim time počinju velikim slovom, iako nisu pojam iz rječnika.

Da bi se smanjio utjecaj ovog problema iz svih dokumenta koji su korišteni u razvoju sustava, izgrađen je rječnik riječi koje se ne nalaze na početku rečenice. Pomoću toga rječnika moguće je odrediti da li se neka riječ uobičajeno piše malim ili velikim početnim slovom. Smatra se da se riječ uobičajeno piše velikim slovom ako se u tekstu u više od 50% slučajeva pojavljivala pisana velikim početnim slovom.

4.2.4. Višestruko pojavljivanje u popisima

Čest problem jest da se jedan pojam prepoznaje u dva ili čak sva tri popisa. Budući da se u označavanju pomoću popisa ne promatra kontekst nekog pojma u takvim je slučajevima moguće jedino odbiti klasifikaciju promatranog pojma. Problem dodatno pojačava opisani način pretraživanja pojmova. Budući da su dopuštena odstupanja teksta koji se prepoznaje od točnog naziva pojma, moguće je da se pojam prepozna u više popisa čak i ako se on zaista nalazi u samo jednom popisu.

4.3. Značajke

Za rješavanje problema korištenjem metoda strojnog učenja ključan je izbor značajki kojima će primjeri za klasifikaciju biti predstavljeni klasifikatorima.

4.3.1. Okolina promatranog tokena

Ove se značajke sastoje od lematizirane verzije tekstne reprezentacije tokena koji okružuju token za koji se generiraju značajke. Optimalna veličine okoline biti će određena postupkom unakrsne provjere prilikom vrednovanja sustava.

4.3.2. Oznake prethodnih tokena

Tokeni se označavaju slijedno, slijeva nadesno, pa su dostupne oznake za tokene koji prethode promatranom tokenu. Kao i za prethodno opisanu značajku, optimalna veličina okoline biti će određena postupkom unakrsne promjene prilikom vrednovanja sustava.

4.3.3. Velika i mala slova

Koriste se dvije značajke koje opisuju kakvim je slovima napisana promatrana riječ. Prva značajka poprima sljedeće vrijednosti:

1. Capitalized – riječ počinje velikim slovom, ostala slova mala;
2. AllCapitals – sva slova su velika;
3. Mixed – sadrži i velika i mala slova;
4. AllSmallCaps – sadrži samo mala slova;
5. NotApplicable – ne sadrži slova;

Druga značajka je binarna vrijednost koja označava pojavljuje li se riječ u tekstu najčešće pisana malim ili velikim početnim slovom. Ova informacija procjenjuje se kako je već ranije opisano iz velike količine teksta.

4.3.4. Sufiks riječi

Značajka se dobiva jednostavnim uzimanjem sufiksa promatranog tokena.

4.3.5. Riječ i korijen riječi

Jedna značajka je tekstna reprezentacija tokena, a druga je korijen riječi. Korijen se dobiva odbacivanjem nastavka riječi iza zadnjeg samoglasnika.

4.3.6. Oznake dobivene označavanjem pomoću popisa

Ova značajka predstavlja oznaku koju je promatranom tokenu odredio označivač koji koristi samo popise pojmova.

4.4. Klasifikator maksimalne entropije

Klasifikator maksimalne entropije (engl. *maximum entropy classifier, MaxEnt*) (Berger et al., 1996) često je korišten za probleme klasifikacije u slučaju kada su značajke nominalne vrijednosti, odnosno kada ne postoji prirodan uređaj značajki. Takav slučaj je i u problemu pronalaženja i klasifikacije imenovanih entiteta gdje su značajke riječi, oznake prethodnih tokena, prefiksi, sufiksi i sl. Prednost klasifikatora maksimalne entropije nad Bayesovim klasifikatorom jest u činjenici da MaxEnt klasifikator ne pretpostavlja uvjetnu nezavisnost među značajkama. Zbog toga je učenje složenije, klasifikator mora učiti iterativno, dok se Bayesov klasifikator uči jednostavnim brojanjem pojavljivanja neke značajke i izlaza, odnosno promatrane klase. Klasifikator MaxEnt iterativno uči koeficijente pridružene pojedinim značajkama.

4.4.1. Model MaxEnt

Model MaxEnt definira vjerojatnost da uzorak $\mathbf{x}$ pripada klasi y na sljedeći način:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}, y)\right)$$

gdje je $\mathbf{x}$ vektor značajki, n broj značajki, λ_i je koeficijent uz značajku i, $Z(\mathbf{x})$ normalizacijska konstanta koja osigurava da p zaista bude vjerojatnost, a funkcija $f_i(\mathbf{x}, y)$ je definirana na sljedeći način:

$$f_i(\mathbf{x}, y) = \begin{cases} 1, \ ako\ je\ \mathbf{x_i} > 0\ i\ \mathbf{x}\ pripada\ klasi\ y \\ 0, \ \textit{inače} \end{cases} \tag{4.1}$$

$f_i(\mathbf{x}, y)$ u ovom se kontekstu naziva značajkom.

Parametar modela je vektor λ koji određuje koeficijente vezane uz pojedinu značajki.

Načelo najveće entropije kaže da model p^* mora biti izabran tako da ima najveću moguću entropiju $H(p)$ među svim modelima koji zadovoljavaju ograničenja zadatka C:

$$p^* = \underset{p \in C}{\operatorname{argmax}} H(p)$$

U modelu MaxEnt ograničenja definiraju da očekivana vrijednost svojstva $f_i(\mathbf{x}, y)$ mora biti jednaka nekoj konstanti K_i:

$$\mathbb{E}(f_i(\mathbf{x}, y)) = \sum_{\mathbf{x},y} p(\mathbf{x}, y) f_i(\mathbf{x}, y) = K_i$$

Budući da je ovaj izraz nemoguće izračunati zbog sumacije po svim mogućim vektorima $\mathbf{x}$, koristi se slijedeća aproksimacija:

$$\sum_{\mathbf{x},y} p(\mathbf{x}, y) f_i(\mathbf{x}, y) \approx \sum_{\mathbf{x},y} \tilde{p}(\mathbf{x}) p(y|\mathbf{x}) f_i(\mathbf{x}, y)$$

gdje je $\tilde{p}(\mathbf{x})$ relativan frekvencija pojavljivanja uzorka $\mathbf{x}$ u podacima za učenje, odnosno, procjena najveće izglednosti.

Prirodan izbor za konstante K_i je empirijsko očekivanje značajke f_i:

$$E_{\tilde{p}f_i} = \sum_{\mathbf{x},y} \tilde{p}(\mathbf{x}, y) f_i(\mathbf{x}, y)$$

iz čega slijede ograničenja:

$$\sum_{\mathbf{x},y} \tilde{p}(\mathbf{x}) p(y|\mathbf{x}) f_i(\mathbf{x}, y) = E_{\tilde{p}f_i}$$

Sada je definiran optimizacijski problem pronalaska takvog modela p koji zadovoljava zadana ograničenja i ima maksimalnu entropiju.

Za određivanje optimalnog modela koji zadovoljava ovo načelo mogu se koristiti različite optimizacijske metode, npr. *iteratively reweighted least squares*, *Quasi-Newton method* ili *generalized iterative scaling - GIS*. Implementacija klasifikatora SharpEntropy[2] koja će biti korištena u izradi rada koristi GIS metodu.

GIS algoritam

GIS je jednostavan algoritam za optimizaciju parametara MaxEnt modela (Curran i Clark, 2003). Koraci algoritma su slijedeći:

– Postavi $\lambda_i^{(0)}$ na bilo koju vrijednost, primjerice:

$$\lambda_i^{(0)} = 0$$

– Ponavljaj dok model ne konvergira:

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \frac{1}{C} \log \frac{E_{\tilde{p}f_i}}{E_{p^{(t)}f_i}}$$

[2]http://www.codeproject.com/Articles/11090/ Maximum-Entropy-Modeling-Using-SharpEntropy

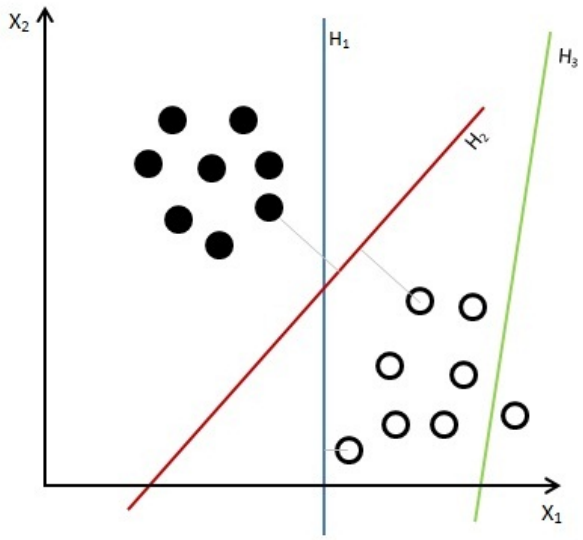
gdje je $E_{\tilde{p}f_i}$ empirijski očekivana vrijednost za f_i, $E_{p^{(t)}f_i}$ je očekivana vrijednost s obzirom na model p, (t) indeks trenutne iteracije, a konstanta C je definirana ovako:

$$C = \max_{\mathbf{x},y} \sum_{i=1}^{n} f_i(\mathbf{x}, y)$$

4.5. Stroj s potpornim vektorima

Stroj s potpornim vektorima (engl. *support vector machine*) naziv je za skupinu srodnih metoda strojnog učenja iz skupine algoritama nadziranog učenja (engl. *supervised learning*). Klasifikator SVM često pokazuje vrlo dobre performanse u rješavanju problema iz obrade prirodnog jezika. Osim za klasifikaciju, SVM se može koristiti i kod regresijskih problema.

SVM je u osnovnoj verziji binaran klasifikator. Ideja SVM-a jest primjere iz skupa za treniranje prikazati kao točke u višedimenzionalnom prostoru (dimenzije prostora određene su brojem značajki u vektoru značajki promatranih primjera za treniranje). SVM ulazni prostor nastoji podijeliti na dva djela, od kojih svaki sadrži primjere iz samo jedne klase. Hiperravnina koja dijeli prostor konstruira se tako da su podprostori pri tome što bolje razdvojeni, odnosno da je što veći razmak najbližih primjera za učenje od granice podprostora. Na slici 4.1 prikazan je binarni klasifikacijski problem te mogući vektori koji razdvajaju primjere u dvije klase.



Slika 4.1: Primjer podjele prostora

Slika ilustrira na koje načine SVM može pokušati razdvojiti primjere za klasifi-

kaciju. Dakle, vektori prikazani crvenom i plavom bojom uspješno razdvajaju skup primjera u dvije klase, dok vektor prikazan zelenom bojom nije dobro podijelio primjere. Osim prikazanih vektora, moguće je konstruirati još beskonačno mnogo drugih vektora, no potrebno je izabrati samo jedan. U ovom slučaju SVM bi odabrao vektor prikazan crvenom bojom jer on najbolje razdvaja primjere, odnosno postavlja najveću marginu između dvije promatrane klase.

Formalizacija problema

Podaci za treniranje klasifikatora prikazani su kao skup točaka D u višedimenzionalnom prostoru (Cortes i Vapnik):

$$D = \{(\mathbf{x}_i, c_i | x_i \in \mathbb{R}^p), c_i \in \{-1, 1\}\}_{i=1}^{n}$$

gdje je c_i jednako -1 ili 1, što predstavlja klasu objekta x_i, a svaki $\mathbf{x}_i$ je p dimenzionalni vektor značajki.

Zadaća SVM-a jest pronaći ravninu u višedimenzionalnom prostoru koja uz najveću marginu razdvaja vektore x_i kojima je vrijednost $c = -1$ od onih kojima je $c = 1$.

Bilo koju ravninu u višedimenzionalnom prostoru možemo zapisati ovako:

$$\mathbf{w} \cdot \mathbf{x} - b = 0$$

gdje je $\mathbf{w}$ vektor normale na ravninu, a $\frac{b}{||\mathbf{w}||}$ je udaljenost ravnine od ishodišta.

Potrebno je pronaći $\mathbf{w}$ i b tako da ravnina najbolje razdvaja primjere za učenje u dvije klase, odnosno potrebno je pronaći dvije najudaljenije paralelne ravnine, sa svojstvom da razdvajaju podatke u dvije klase.

Te ravnine možemo zapisati na sljedeći način:

$$\mathbf{w} \cdot \mathbf{x} - b = -1$$

$$\mathbf{w} \cdot \mathbf{x} - b = 1$$

ili kraće:

$$y_i(\mathbf{w} \cdot \mathbf{x} - b) = 1$$

Kako bi se postigla što veća margina želimo da te ravnine budu što više udaljene. Udaljenost ovih dviju ravnina možemo izraziti kao $\frac{2}{\|\mathbf{w}\|}$, iz čega slijedi da treba minimizirati $\|w\|$.

Kako bismo osigurali da se niti jedan uzorak ne nalazi unutar margine potrebno je dodati sljedeća ograničenja:

$$\mathbf{w} \cdot \mathbf{x_i} - b \geq 1$$

za x_i iz prve klase, i:

$$\mathbf{w} \cdot \mathbf{x_i} - b \leq -1$$

za x_i iz druge klase.

Sada je definiran sljedeći minimizacijski problem:

minimizirati $\|\mathbf{w}\|$, uz ograničenja:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1$$

Ovo je problem kvadratnog optimiranja za čega postoje učinkovite metode rješavanja, a jedna od njih je metoda Lagrangeovih multiplikatora.

4.5.1. Klasifikacija u više klasa

Budući da je SVM primarno binarni klasifikator, potrebna je metoda kojom možemo iskoristiti ovakav klasifikator i za višeklasne klasifikacijske probleme. Takav sustav može se ostvariti na dva načina, konstruirajući binarne klasifikatore za svaki par klasa (engl. *one against one*), ili konstruirajući klasifikatore koji razdvajaju jednu klasu od svih ostalih (engl. *one against all*).

One against one

Ako se radi o problemu klasifikacije u N klasa, konstruira se ukupno $\frac{N(N-1)}{2}$ klasifikatora. Novi ulazni primjer klasificira se svakim od konstruiranih klasifikatora, a rezultati je klasa koja je dobila najviše glasova. U idealnom slučaju to bi značilo $N-1$ glasova - ako je binarni klasifikatora koji odlučuje između prave klase i neke od preostalih $N-1$ klasa svaki put ispravno odlučio.

One against all

U ovom pristupu stvara se ukupno N klasifikatora, od kojih svaki odgovara na pitanje je li novi uzorak u klasi C_i ili u nekoj drugoj klasi. Da bi ispravno mogli klasificirati uzorak samo jedan klasifikator smije pozitivno odgovoriti. Ako je više klasifikatora utvrdilo da primjer pripada klasi C_i, nemoguće je odrediti koja je od tih klasa prava.

Iz gore opisanog vidi se da pristup *one against one* omogućuje veću robusnost, jer je moguće utvrditi klasifikaciju imenovanih u slučaju da neki od klasifikatora ne donese ispravnu odluku.

4.6. Skriveni Markovljev model

Treći iskušani pristup su skriveni Markovljevi modeli (engl. *hidden Markov model - HMM*). Ovaj pristup također se često koristi za označavanje imenovanih entiteta te za označavanje vrste riječi (engl. *POS tagging*), a prednost pred klasičnim klasifikatorima je u činjenici da HMM označivać označava cijeli niz, odnosno rečenicu, a ne zasebne tokene u rečenici. Na taj se način optimizira cijeli niz oznaka, dok drugi klasifikatori donose lokalno optimalnu odluku označavajući samo jedan token.

Skriveni Markovljev model predstavljen je skupom stanja koja nisu vidljiva promatraču. Model može s određenim vjerojatnostima prijeći iz jednog skrivenog stanja u drugo, i u svakom stanju može s određenom vjerojatnošću emitirati izlaz. Izlaz je vidljiv promatraču i na temelju izlaza može se zaključivati o skrivenim stanjima, vjerojatnostima prijelaza i vjerojatnostima emitiranja izlaza u svakom stanju.
Formalno, HMM čini:

- S – skup stanja veličine N,
- A – matrica vjerojatnosti prijelaza između stanja, veličine $N \times N$,
- V – skup mogućih izlaza veličine M,
- B – matrica vjerojatnosti emitiranja simbola u stanju, veličine $N \times M$,
- π – vektor veličine N koji definira početne vjerojatnosti za stanja.

HMM se u početnom trenutku t_0 nalazi u nekom stanju $s \in S$ i u svakom diskretnom trenutku $t_{0..\infty}$ prema matrici vjerojatnosti A prelazi u novo stanje (moguće i u isto) te emitira simbol $O \in V$ prema matrici vjerojatnosti B. Model HMM parametriziran je trojkom parametara $\lambda = (A, B, \pi)$

Prema (Rabiner, 1989), postoje tri osnovna zadatka u radu s HMM-om:

- Problem 1: Uz zadani niz izlaza $O = O_1O_2...O_T$ i model $\lambda(A, B, \pi)$, potrebno je učinkovito izračunati $P(O|\lambda)$ - vjerojatnost da je model generirao zadani niz izlaza;
- Problem 2: Uz zadani niz izlaza $O = O_1O_2...O_T$ i model $\lambda(A, B, \pi)$, treba pronaći niz skrivenih stanja $Q = q_1q_2...q_T$ koji je na neki način optimalan.

– Problem 3: Kako možemo podesiti parametre modela $\lambda = (A, B, \pi)$ tako da se maksimizira $P(O|\lambda)$, gdje su O nizovi izlaza iz skupa za učenje.

Označavanje imenovanih entiteta u tekstu moguće je modelirati skrivenim Markovljevim modelom. U tom modelu skup stanja čine moguće oznake entiteta, dok su izlazi svi mogući tokeni. Iz označenoga skupa za učenje tada je potrebno izračunati nepoznate parametre λ. Ako ovako modeliramo problem, označavanje novog teksta svodi se na rješavanje *Problema 2*.

No, za rješavanje *Problema 2* potrebno je najprije odrediti parametre modela (*Problem 3*). Matricu prijelaza za skrivena stanja lako je izračunati iz skupa za učenje. U skupu za učenje prebrajaju se svi prijelazi iz nekog stanja s_i u sva ostala stanja $s \in S$, pa je vjerojatnost prijelaza u neko stanje s_j, $p(s_i, s_j)$ jednaka relativnoj frekvenciji prijelaza iz s_i u s_j, odnosno, koristimo metodu najveće izglednosti. Slično se određuje i matrica početnih vjerojatnosti π – ta vjerojatnost jednaka je relativnoj frekvenciji pojavljivanja stanja s_i na početku rečenice, odnosno, ponovno radimo procjenu najveće izglednosti.

Preostaje odrediti matricu B koja određuje vjerojatnost emitiranja simbola (riječi) u nekom skrivenom stanju. Ako bi matricu odredili računajući vjerojatnosti kao relativnu frekvenciju emitiranja neke riječi u određenom stanju, takav model ne bi imao mogućnost generalizacije za nove riječi, kao ni za viđene riječi koje u nekim stanjima nisu nikad emitirane. Za tu svrhu se generiraju dodatni modeli iz kojih se računa tražena vjerojatnost za dosad neviđene riječi. U radu se iskušani pristupi računanja tih vjerojatnosti na temelju informacija o tipu riječi dobivenom iz popisa pojmova, prethodne riječi u kontekstu i informacije obliku riječi (mala/velika slova).

4.7. Učenje

Kako bi se iskoristila velika količina dostupnih neoznačenih podataka za učenje klasifikatora, u ovome radu koristit će se polunadzirano strojno učenje (engl. *semi-supervised learning*). Polunadzirano strojno učenje koristi dostupne označene podatke, a uz njih pokušava iskoristiti i neoznačene podatke koji su često lako dostupni.

Samoučenje

Samoučenje (engl. *self-training*) jedna je varijanta polunadziranog strojnog učenja. Ideja samoučenja je jednostavna: klasifikator se najprije uči na označenim podacima, zatim se tako naučeni klasifikator koristi za označavanje na neoznačenom skupu i iz

novooznačenih podatak tada ponovno uči.

Za samoučenje je bitno da klasifikator daje mjeru pouzdanosti, što se koristi pri izboru novih primjera za učenje. Naime, za učenje se ne uzimaju svi novodobiveni podaci, nego samo oni za koje klasifikatora daje pouzdanosti veću od zadanog praga. Taj prag je parametar algoritma, a optimalna vrijednosti se može tražiti postupkom unakrsne provjere.

Samoučenje je iterativan algoritam, odnosno, postupak označavanja neoznačenih podataka i ponovno učenje na njima može se ponavljati više puta dok se ne dosegnu maksimalne performanse klasifikatora.

Pseudokod algoritma prikazan je u nastavku (Abney, 2007):

Algorithm 1 Pseudokod algoritma samoučenja

function SELFTRAIN(L_0,U)
  L_0 su označeni podaci, U su neoznačeni podaci
  $c \leftarrow treniraj(L_0)$
  repeat
    $L \leftarrow L_0 + odaberiPrimjere(label(U, c))$
    $c \leftarrow treniraj(L)$
  until kriterij zaustavljanja nije ispunjen
  return c
end function

Kako je vidljivo iz pseudokoda, algoritam ne propisuje točan kriterij zaustavljanja. Kao kriterij zaustavljanja može se koristiti fiksan broj iteracija, vrijeme izvođenja ili broj iteracija bez dovoljnog napretka u rezultatu.

U primjeni algoritama samoučenja na učenje klasifikatora za označavanje imenovanih entiteta pojavljuje se još jedan problem koji je potrebno razmotriti. Naime, kao značajke se koriste i oznake prethodnih tokena. Kada se naučenim klasifikatorom označe novi podaci, za svaki od tokena dobivamo oznaku i pouzdanost te oznake. Prilikom izbora novog primjera za učenje uzimaju se oznake iznad određene razine pouzdanosti, a kao značajke koriste se oznake prethodnih tokena koje su također dobivene označavanjem klasifikatora iz prethodne iteracije i nisu potpuno pouzdane. Pitanje je treba li nepouzdanost tih oznaka uzimati u obzir, i treba li i za njih koristiti jednaki prag za izbor. Ovaj fenomen također će biti ispitan tijekom učenja.

4.8. Dodatna obrada izlaza klasifikatora

Nakon što klasifikatori označe tekst moguće je provesti dodatno razrješavanje označenog teksta kako bi se uklonile česte pogreške i popravili konačni rezultati označavanja.

Jedna od čestih grešaka jest da se riječi na početku rečenice označe pogrešno zbog velikog početnog slova. Ovaj problem je riješen tako da se za svaki entitet koji čini početak rečenice provjerava kako se početna riječ entiteta najčešće piše kada nije na početku rečenice. Ako se riječ najčešće ne piše velikim početnim slovom, taj se entitet proglašava pogrešno prepoznatim i dodjeljuje mu se klasa koja označava običan tekst.

Druga metoda razrješavanja zasniva se na popisima entiteta. Sav se tekst označava korištenjem rječnika tako da se u obzir uzimaju samo potpuno jednoznačni entiteti. To znači da se tekst entiteta smije pojaviti samo u jednom od popisa entiteta, a tekst iz popisa mora se točno poklapati s tekstom koji se pojavio u dokumentu. Na ovaj način nije moguće pronaći veliki broj entiteta, ali pronađeni entiteti imaju veliku pouzdanost.

4.8.1. Označavanje numeričkih entiteta korištenjem regularnih izraza

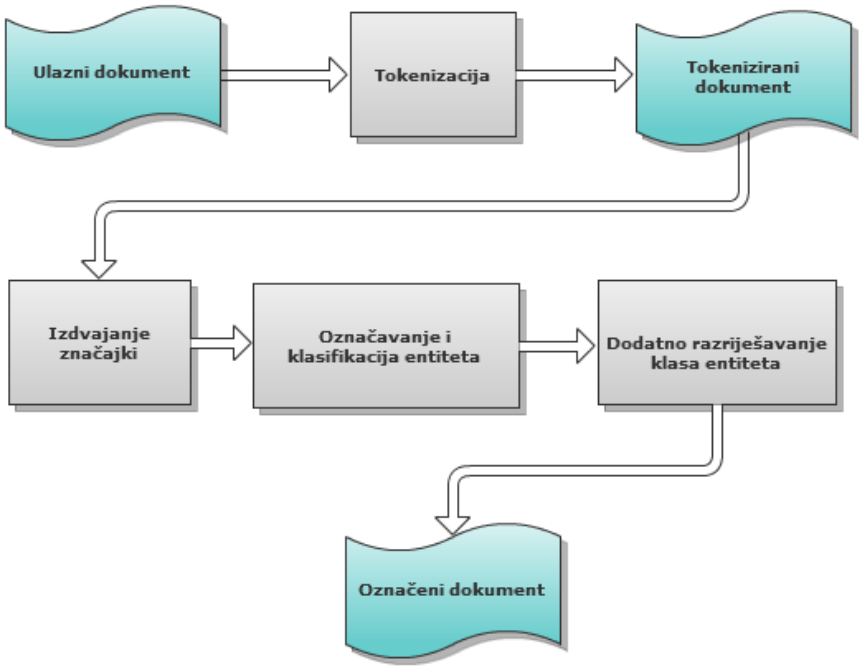
Za označavanje entiteta iz skupina TIMEX (datum i vrijeme) i NUMEX (postotak i valutni izraz) koriste se regularni izrazi. Entiteti iz ove skupine imaju čvrstu strukturu i moguće ih je vrlo učinkovito prepoznati korištenjem regularnih izraza. Prosječna mjera F1 za ove entitete kada se prepoznaju regularnim izrazima kreće se oko 0.95. Iskušan je i pristup korištenjem klasifikatora, no rezultati su lošiji. Osim lošijih rezultata, dodavanje dodatne četiri klase znatno usporava treniranje klasifikatora.

Za svaki od četiri entiteta iz ove skupine napisani su složeni regularni izrazi koji dopuštaju različite načine na koji entiteti mogu biti zapisani. Tako se zapis broja dopušta korištenjem različitih decimalnih oznaka (’,’, ’.’) i različitih simbola za razdvajanje grupa znamenaka(’ ’, ’,’, ’.’). Brojeve je također moguće zapisati i riječima, ali i kombinacijom brojeva i riječi npr. “5,5 tisuća”, “100 milijuna”.

Za valutne izraze dopušteni su različiti zapisi koji uključuju puni naziv valute, standardne skraćenice ili valutne simbole, npr. za *Eure*: euro, EUR, . €

5. Implementacija

U ovom poglavlju bit će ukratko opisana implementacija razvijenog sustava NERC. Slika 5.1 prikazuje grubu shemu sustava.



Slika 5.1: Shema sustava NERC

Ulaz sustava je XML-dokument čija je struktura opisna u *Dodatku A*. Dokument se učitava iz datoteke, tokenizira i iz njega se stvar objekt tipa `Document`. Takav dokument mogu prihvatiti *NERC*-podsustavi, koji tada za svaki token provode generiranje vektora značajki i klasifikaciju. Na klasificiranom dokumentu se zatim provodi dodatno razrješavanje tipova entiteta i označavanje regularnim izrazima za *NUMEX* i *TIMEX* tipove. Tako označeni dokument moguće je ponovno spremiti u XML-datoteku čiji je format također opisan u *Dodatku A*

Sustav je implementiran u programskom jeziku C#, uz korištenje frameworka Microsoft .NET 3.5. Sva funkcionalnost sustava NERC razvijena je u obliku *class li-*

brarya, odnosno može se koristiti kao komponenta *.NET .dll* u drugim aplikacijama. Uz *class library* razvijena je i jednostavna *Windows Forms* aplikacija koja omogućava učitavanja, označavanje, pregled označenih dokumenata te njihovo spremanje u XML format.

U nastavku će ukratko biti opisana struktura koda i najvažniji razredi. Korijenski *namespace* je `NERC` i on sadrži sljedeće prostore imena (engl. *namespace*):

- `NERCSystem` – implementira potporne funkcionalnosti za ostatak sustava,
- `Gazeteer` – implementira funkcionalnosti vezane u popise pojmova,
- `MaxEntTagger` – implementira funkcionalnosti vezane uz sustav za označavanje temelljen na klasifikatoru MaxEnt,
- `SVMTagger` – implementira funkcionalnosti vezane uz sustav za označavanje temeljene na klasifikatoru SVM,
- `HMMTagger` – implementira funkcionalnosti vezane uz sustav za označavanje temeljen na klasifikatoru HMM,
- `Tokenizer` – implementira Tokenizer i potporne klase

5.1. Razred `NERC.Tokenizer.Document`

Ovo je osnovni razred za predstavljanje dokumenta u sustavu. Dokument sadrži naslov koji sadrži jednu ili više rečenica te jedan ili više odlomaka koji se opet sastoje od jedne ili više rečenica.

`Document` se može kreirati iz datoteke *.xml* korištenjem konstruktora:
`public` `Document(String path)` - parametar je put do *.xml* dokumenta.

5.2. Razred `NERC.Tokenizer.Token`

Razred apstrahira jedan token sa sljedećim svojstvima:

- **`public`** `string Value` - tekst tokena;
- **`public`** `type TypeOfWord` - tip tokena (word, space, number...);
- **`public`** `EEntityType EntityType` - tip entiteta;
- **`public double`** `EntityConfidence` - pouzdanost dodijeljenog tipa entiteta;
- **`public`** `string Value` - tekst tokena.

5.3. Razred `NERC.Tokenizer.Tokenizer`

Razred sadrži metode za učitavanje i tokenizaciju dokumenata *.xml*, kao i nekoliko pomoćnih koje će ukratko biti opisane.

public `Document TokenizeXML(String path)` - metoda kao argument prima put do datoteke s dokumentom *.xml*, učitava ga, tokenizira i vraća potpuno izgrađeni `Document` objekt.

public void `SaveDocumentToXML(Document doc, String path)` - metoda prima `Document` objekt i put za spremanje datoteke. Primljeni dokument se serijalizira u niz znakova u obliku datoteke *.xml* i sprema na zadanu lokaciju.

public `List<Token> GetTokens(String str)` - metoda prima proizvoljan niz znakova i vraća tokene sadržane u nizu znakova.

public static `Token.WordCase WordCase(String word)` - metoda prima niz znakova i određuje kakvim je slovima riječ napisana. Povratna enumeracija poprima sljedeće vrijednosti: `Capitalized`, `AllCapitals`, `Mixed`, `AllSmallCaps`, `NotApplicable`.

public static `String StemmWord(String word)` - metoda prima niz znakova i vraća jednostavan korijen dobiven odbijanjem nastavka od zadnjeg samoglasnika u riječi. Riječ se vraća u originalnom obliku ako je kraća od tri slova, ili ako je dio koji se odbija dulji od novonastale riječi.

5.4. Razred `public class WordCapitalization`

Glavna namjena razreda jest određivanje piše li se neka riječ u tekstu uobičajeno velikim ili malim početnim slovom. Ova se informacija koristi kod generiranja značajki za učenje klasifikatora i kod razrješavanja tipova entiteta. Razred se kreira sljedećim konstruktorom:

public `WordCapitalization(String fileName)` - jedini argument je put do datoteke sa serijaliziranim objektom.

public bool `IsNaturalCapizalized(String word)` - metoda prima riječ, a povratna vrijednost označava piše li se riječ uobičajeno velikim (*true*) ili malim (*false*) početnim slovom.

5.5. Razred `Gazetteer`

Ovaj razred implementira popise pojmova. Za kreiranje razreda poziva se glavni konstruktor:

public `Gazetteer(String fileName)` - parametar je put do datoteke koja sadrži serijalizirani popis.

Jedina javna metoda je:

public `Token.EEntityType GetItemType(String text)` - parametar je tekst koji se želi klasificirati, a metoda vraća pronađeni tip. Ako se klasa ne može odrediti, vraća se tip *Token.EEntityType.Other*.

5.6. Razred `NERC.Gazeteer.GazetteerTagger`

Razred implementira označivač korištenjem popisa entiteta. Za kreiranje je potrebno pozvati konstruktor:

public `GazetterNERC(Gazetteer gazetteer)` - parametar je instanca razreda `Gazetteer`

Za označavanje dokumenta potrebno je pozvati metodu

public `Document TagDocument(Document doc)` - parametar je dokument koji je potrebno označiti, a metoda vraća označeni dokument.

5.7. Razred `NERC.SVMTagger.SVMTagger`

Ovaj razred implementira označivač korištenjem klasifikatora SVM. Za kreiranje se koristi sljedeći konstruktor:

public `SVMTagger(String classifierPath)` - jedini parametar je put do datoteke sa serijaliziranim klasifikatorom.

Za označavanje dokumenta poziva se metoda:

public `Document TagDocument(Document doc)` - parametar je dokument koji je potrebno označiti, a povratna vrijednost je označeni dokument.

5.8. Razred `NERC.MaxEntTagger.MaxEntTagger`

Ovaj razred implementira označivač korištenjem klasifikatora SVM. Za kreiranje se koristi sljedeći konstruktor:

public `MaxEntTagger(String classifierPath)` - jedini parametar je put do datoteke sa serijaliziranim klasifikatorom.

Za označavanje dokumenta poziva se metoda:

public `Document TagDocument(Document doc)` - parametar je dokument koji je potrebno označiti, a povratna vrijednost je označeni dokument.

5.9. Razred `NERC.HMMTagger.HMMTagger`

Ovaj razred implementira označivač korištenjem skrivenih Markovljevih lanaca. Za kreiranje koristi se slijedeći konstruktor:

public `HMMTagger(String classifierPath)` - jedini parametar je put do datoteke sa serijaliziranim klasifikatorom.

Za označavanje dokumenta poziva se metoda:

public `Document TagDocument(Document doc)` - parametar je dokument koji je potrebno označiti, a povratna vrijednost je označeni dokument.

5.10. Razred `NERC.NERCSystem.AmbiguityReslover`

Razred implementira metode za obradu dokumenata nakon što ih označe klasifikatori.

`Document ResolveDocument(Document doc)` - kao argument prima označeni dokument, obrađuje ga, te vraća ispravljeni dokument. U obradi se vrši promjena tipa entiteta na temelju početnog slova i označavanje entiteta iz skupine *NUMEX* i *TIMEX*.

5.11. Razred `NERC.NERCSystem.EvaluationResult`

Razred apstrahira rezultat vrednovanja NERC sustava. Istovremeno je podržano vrednovanje prema pravilima MUC te *exact-match* vrednovanje. Svojstva preko kojih se dohvaćaju pojedine mjere su slijedeće:

- **double** `MUCPrecission` - preciznost prema normi MUC,
- **double** `MUCRecall` - odziv prema normi MUC,
- **double** `MUCF1` - mikro-prosječni F1 prema normi MUC,
- **double** `ExactPrecission` - *exact-match* preciznost,
- **double** `ExactRecall` - *exact-match* odziv,
- **double** `ExactF1` - *exact-match* F1.

5.12. Razred `NERC.NERCSystem.Evaluator`

Ovaj razred omogućava vrednovanje sustava NERC. Kreira se pozivanjem konstruktora:

public `Evaluator()` - metoda nema parametara

Evaluacija se može vršiti na dva načina:

public `EvaluationResult Evaluate(String folderPath, INERCTagger NT)` - metoda prima put do mape s označenim dokumentima te klasifikator preko sučelja `INERCTagger`. Vraća ukupni rezultat evaluacije svih dokumenata u danoj mapi.

public `EvaluationResult Evaluate(Document cor, Document outp)` - metoda prima dva dokumenta od kojih je prvi originalni označeni dokument a drugi dokument kojeg je označio klasifikator. Vraća objekt tipa `EvaluationResult` koji sadrži rezultate evaluacije.

5.13. Razred `NERC.NERCSystem.NERCSystem`

Razred predstavlja sustav NERC koji nudi više NERC označivača. Korisnik *.DLL*-a može instancirati samo ovaj razred kako bi mogao koristiti funkcionalnost označavanja dokumenata. Za kreiranje se koristi sljedeći konstruktor:

public `NERCSystem(String dataPath)` - jedini parametar je put do mape koja sadrži serijalizacije svih objekata potrebnih za izgradnju sustava, a to su:

- `capitalization` - datoteka s podacima o malim/velikim početnim slovima za riječi
- `molex-vj-nn-1.31-lm-locase.fsa` - datoteka za izgradnju lematizatora
- `gazetteer` - datoteka za izgradnju Gazetteera kojeg koriste klasifikatori
- `svmTagger` i `svmTagger.converter` - datoteke za izgradnju SVM taggera (opcionalno)
- `maxEntTagger` i `maxEntTagger.features` - datoteke za izgradnju maxEnt taggera (opcionalno);
- `hmmTagger` - datoteka za izgradnju HMM taggera (opcionalno).

Datoteke kod kojih je navedeno da su opcionalne ne moraju obavezno postojati, ali tada odgovarajući označivači neće biti dostupni za korištenje.

Pojedini NERC označivači mogu se dohvatiti preko svojstava razreda:

– **public** `SVMTagger.SVMTagger SvmTagger` - SVM temeljen NERC-sustav,
– **public** `MaxEntTagger.MaxEntTagger MaxEntTagger` - MaxEnt temeljen NERC-sustav,
– **public** `HMMTagger.HMMTagger HMMTagger` - HMM temeljen NERC-sustav.

5.14. Razred `NERC.HelperRepository`

Ovaj razred nudi nekoliko statičkih svojstava za dohvaćanje često korištenih objekata, a sva su svojstva dostupna odmah po kreiranju instance razreda `NERC.NERCSystem.NERCSystem`. Dostupni objekti su:

– **public static** `AmbiguityResolver ambiguityResolver`,
– **public static** `Tokenizer.Tokenizer tokenizer`,
– **public static** `WordCapitalization wordCapitalization`,
– **public static** `GazetterTagger gazetterTagger`,
– **public static** `Gazetteer gazetter`,
– **public static** `FSALemmatization lematizer`.

6. Vrednovanje sustava

6.1. Metode vrednovanja NERC sustava

Vrednovanje sustava za označavanje i klasifikaciju imenovanih entiteta drugačije je od vrednovanja običnih klasifikacijskih sustava te će u ovom poglavlju biti ukratko opisano. Sustav za označavanje i klasifikaciju mora odrediti granice entiteta u tekstu, kao i klasifikaciju svakog tako označenog entiteta. Dakle, točna klasifikacija nekog entiteta sastoji se od nekoliko koraka te u tom postupku označivač može načinit sljedećih pet grešaka:

- označiti entitet tamo gdje on ne postoji u originalnom tekstu,
- ne označiti entitet koji postoji,
- točno označiti granice entiteta, ali pogrešno odrediti njegovu oznaku,
- točno odrediti oznaku, no netočno odrediti granice entiteta,
- netočno odrediti i granice i oznaku entiteta.

S obzirom na navedene greške sustav je moguće vrednovati na različite načine. U literaturi se najčešće koriste dvije metode vrednovanja: MUC-vrednovanje i (engl. *exact-match*) vrednovanje (Ratinov i Roth, 2009).

6.1.1. MUC-vrednovanje

Ovaj način vrednovanja korišten je za vrednovanje označavanja i klasifikacije imenovanih entiteta na *Message Understanding Conference - MUC*. Ovaj pristup zasebno vrednuje točan pronalazak granica entiteta i točnu klasifikaciju entiteta.

Kao točno prepoznavanje tipa entiteta (*TYPE*) računa se pokušaj sustava u kojem je ostvareno bilo kakvo preklapanje s pravim entitetom i točno određivanje klase entiteta. Točnim prepoznavanjem teksta (*TEXT*) smatra se pokušaj u kojem su točno određene granice entiteta u tekstu (preklapanje svih tokena), bez obzira je li klasifikacija prepoznatog entiteta točna.

Kada sustav označi entitete računaju se sljedeće mjere: COR_{TEXT} - broj točnih prepoznavanja tipa *TEXT*, COR_{TYPE} - broj točnih prepoznavanja tipa *TYPE*, *ACT* - broj pokušaja sustava (koliko je sustav pronašao entiteta), i *POS* - stvarni broj entiteta u dokumentu. Ukupan broj točnih prepoznavanja sada je: $COR = COR_{TEXT} + COR_{TYPE}$. Preciznost se tada računa prema formuli:

$$P = \frac{COR}{ACT * 2}$$

, a odziv prema formuli:

$$R = \frac{COR}{POS * 2}$$

Konačna mjera uspješnosti sustava je mikro-prosječni F1 koji se računa formulom:

$$F1 = \frac{2PR}{P + R}$$

6.1.2. *Exact-match* vrednovanje

Kod *exact-match* vrednovanja sustav mora točno odrediti granice entiteta i njegov tip. Bilo kakvo odstupanje od toga računa se kao promašaj. Ovaj način vrednovanja korišten je na konferencijama *IREX* i CONNL (Sang i Meulder, 2003).

U ovom načinu vrednovanja koriste se jednake formule kao i kod MUC vrednovanja, a razlika je u računaju varijable *COR*, koja je u ovom slučaju broj entiteta kojima su točno određene granice i klasa (također u nazivniku nema faktora 2, koji se kod MUC vrednovanja pojavljuje jer sustav za svaki entitet određuje 2 značajke). Konačna mjera također je mikro-prosječni F1.

6.2. Optimizacija parametara

U sljedećim poglavljima bit će opisan postupak optimizacije parametara za klasifikatore MaxEnt i SVM. Na klasifikator HMM nisu primjenjivi jednaki parametri, kao ni metoda samoučenja zbog nedostatka informacije o pouzdanosti oznaka, pa o njemu neće biti riječi u ovom poglavlju.

6.2.1. Izbor značajki

Dvije bitne grupe značajki koje se koriste za klasifikaciju su veličina konteksta (tokeni koji okružuju token koji se klasificira) i oznake klasa prethodnih tokena (oznake tokena koji slijede nisu dostupne). Za obje je grupe potrebno odrediti optimalne vrijednosti (broj tokena u kontekstu i broj oznaka prethodnih tokena). Postupak optimizacije parametara proveden je postupkom jednostruke unakrsne provjere na skupu za provjeru. Za oba parametra korišten je raspon vrijednosti $1..2$ i vrednovanje je provedeno za sve 4 kombinacije tih parametara.

Tablica 6.1 prikazuje utjecaj ovih parametara na performanse sustava temeljenog na klasifikatoru MaxEnt. Oznake $x-y$ u stupcima znače da je korištena tekstna reprezentacija x tokena koji se nalaze prije i nakon promatranog tokena i oznake y prethodnih tokena. Reci tablice prikazuju rezultate po iteracijama algoritma samoučenja. Iteracija 0 prikazuje rezultat bez korištenja neoznačenog skupa, dok preostali reci prikazuju rezultate nakon n-te iteracije učenja na neoznačenim podacima. U svim tablicama koje prikazuju rezultate po iteracijama u zadnjem retku označenom s δ prikazana je razlika najboljih postignutih performansi i performansi u iteraciji 0 (bez primjene samoučenja).

Rezultati prikazani u tablici pokazuju da je optimalno koristiti kontekst veličine dva te oznaku samo jednog prethodnog tokena.

Tablica 6.2 prikazuje utjecaj parametara na performanse sustava temeljenog na SVM klasifikatoru.

Iz tablice je vidljivo da je najbolji rezultat postignut u prvoj iteraciji učenja za slučaj korištenja konteksta veličine dva i oznake jednog prethodnog tokena, kao i kod MaxEnt klasifikatora. Za daljnje testove korišteni su pronađeni optimalni parametri, odnosno, kontekst veličine dva i oznaka jednog prethodnog tokena.

Tablica 6.1: Utjecaj veličine konteksta – klasifikator MaxEnt

	Veličina konteksta							
	1-1		2-1		1-2		2-2	
	MUC	Exact	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	72.8	67.5	72.3	66.3	72.2	66.5	71.8	65.8
Iteracija 1	75.5	70.5	75.4	69.6	75.4	69.8	74.9	68.8
Iteracija 2	76.6	71.1	76.6	70.6	76.2	70.4	77.3	71.4
Iteracija 3	77.0	71.9	78.7	73.0	77.0	71.3	78.6	72.6
Iteracija 4	78.1	72.9	79.4	74.0	**77.6**	**72. 2**	78.1	72.7
Iteracija 5	78.3	73.5	**80.7***	**75.2***	77.5	72.2	78.8	73.1
Iteracija 6	78.3	73.5	80.5	74.7	77.3	71.7	79.5	73.9
Iteracija 7	78.6	73.6	79.7	74.0	76.7	71.0	79.6	73.9
Iteracija 8	**78.9**	**73.7**	79.2	73.3	76.7	71.1	**79.9**	**74.0**
δ	6.1	6.2	8.4	8.9	5.4	5.7	8.1	8.2

Tablica 6.2: Utjecaj veličine konteksta – SVM klasifikator

	Veličina konteksta							
	1-1		2-1		1-2		2-2	
	MUC	Exact	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	81.1	75.0	81.0	75.2	81.3	75.7	80.8	75.6
Iteracija 1	**81.5**	**75.4**	**81.9***	**76.1***	**81.5**	**76.0**	**81.2**	**75.6**
Iteracija 2	80.2	74.3	81.0	75.5	80.8	75.3	80.7	75.0
Iteracija 3	79.4	73.3	80.6	74.4	80.9	75.2	80.9	75.0
Iteracija 4	79.2	73.1	80.3	73.8	79.5	73.4	81.0	74.9
Iteracija 5	79.1	73.0	80.2	73.7	79.3	73.1	80.6	74.0
δ	0.4	0.4	0.9	1.1	0.2	0.7	0.4	0.0

Kod korištenja algoritma samoučenja parametar algoritma je prag pouzdanosti za određivanje onih novooznačeni primjeri koji će biti korišteni u sljedećoj iteraciji učenja. U Tablici 6.3 prikazan je utjecaj tog parametra na performanse označivača MaxEnt.

Tablica 6.3: Utjecaj praga pouzdanosti – klasifikator MaxEnt

	Prag pouzdanosti					
	0.75		0.85		0.95	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	72.3	66.3	72.3	66.3	72.3	66.3
Iteracija 1	76.0	70.0	75.4	69.6	73.4	67.8
Iteracija 2	79.1	73.5	76.6	70.6	73.4	66.7
Iteracija 3	**79.9**	**74.0**	78.7	73.0	73.9	67.9
Iteracija 4	78.3	72.2	79.4	74.0	74.4	68.0
Iteracija 5	78.3	72.2	**80.7***	**75.2***	74.9	69.0
Iteracija 6	78.1	71.8	80.5	74.7	75.3	69.1
Iteracija 7	77.5	71.1	79.7	74.0	75.3	69.0
Iteracija 8	77.4	70.6	79.2	73.3	**75.5**	**69.2**
δ	7.6	7.7	8.4	8.9	3.2	2.9

Najbolji rezultati postignuti su uz prag pouzdanosti 0.85.

Tablica 6.4 prikazuje ovisnost rezultat SVM označivača o pragu pouzdanosti za izbor novih primjera za učenje:

I u ovom slučaju se optimalnim izborom pokazala vrijednost 0.85.

6.2.2. Analiza utjecaja veličine skupova za učenje

U ovom poglavlju bit će ispitan utjecaj veličine skupova za učenje na performanse sustava. Takvom analizom moguće je procijeniti bi li povećanje količine podataka za učenje bitno doprinijelo konačnim rezultatima. Budući da sustav za učenje koristi označene i neoznačene skupove podataka testovi će se provesti zasebno za oba skupa.

Utjecaj veličine označenog skupa

Skup za učenje podijeljen je na tri jednaka djela koji sadrže po 45 članaka te je sustav treniran na skupovima načinjenim od jednog, dva i sva tri takva podskupa.

Tablica 6.4: Utjecaj praga pouzdanosti – klasifikator SVM

	Prag pouzdanosti					
	0.75		0.85		0.95	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	**81.3**	**75.5**	81.0	75.2	81.1	75.4
Iteracija 1	80.9	75.1	**81.9***	**76.1***	81.3	75.6
Iteracija 2	81.1	75.0	81.0	75.5	81.5	75.8
Iteracija 3	80.5	74.4	80.6	74.4	**81.7**	75.9
Iteracija 4	80.1	73.9	80.3	73.8	81.6	76.0
Iteracija 5	80.2	73.8	80.2	73.7	81.6	**76.1**
δ	0.0	0.0	0.9	1.1	0.6	0.7

U Tablici 6.5 prikazani su rezultati vrednovanja sustava za različite veličine skupa za učenje.

Rezultati pokazuju da veličina označenog skupa za učenje značajno utječe na rezultate, a porast performansi za jednako povećanje skupa za učenje je gotov linearan. Ovakvo ponašanje signalizira da bi dodatne količine označenih podataka dodatno mogle poboljšati točnost označavanja entiteta.

U Tablici 6.6 prikazan je utjecaj veličine označenog skupa na performanse klasifikatora temeljenog na stroju s potpornim vektorima.

I ovdje je vidljivo da više podataka za učenje poboljšava konačne rezultate, ali apsolutni iznos poboljšanja manji je nego kod klasifikatora MaxEnt.

Tablica 6.5: Utjecaj veličine skupa za učenje – klasifikator MaxEnt

	Broj označenih članaka					
	45		90		135	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	66.1	59.2	69.8	64.0	73.5	67.4
Iteracija 1	72.4	66.0	74.4	68.2	76.4	70.4
Iteracija 2	74.3	67.4	77.1	70.7	78.1	71.6
Iteracija 3	**75.1**	**68.0**	77.1	70.5	79.3	73.2
Iteracija 4	74.5	66.3	77.7	71.1	80.0	74.0
Iteracija 5	73.5	64.1	77.4	70.1	**80.3***	**74.5***
Iteracija 6	71.4	61.2	77.4	70.6	80.4	74.3
Iteracija 7	70.2	59.1	**78.0**	**71.3**	80.0	73.9
Iteracija 8	69.1	57.8	77.4	70.6	79.9	73.8
δ	9.0	8.8	8.2	7.3	6.8	7.1

Tablica 6.6: Utjecaj veličine skupa za učenje – klasifikator SVM

	Broj označenih članaka					
	45		90		135	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	76.8	69.9	79.3	73.2	**82.2***	**76.1***
Iteracija 1	**77.7**	**71.2**	79.4	73.2	81.5	75.5
Iteracija 2	76.7	69.6	**79.7**	**73.8**	81.3	75.5
Iteracija 3	75.3	68.5	78.6	72.5	81.3	75.0
Iteracija 4	75.1	67.6	79.4	72.3	80.6	74.2
Iteracija 5	74.7	67.0	78.3	72.0	80.2	73.8
δ	0.9	0.3	0.4	0.6	0.0	0.0

Utjecaj veličine neoznačenog skupa

Cilj ovoga testa jest utvrditi koliko količina neoznačenih podataka utječe na konačne performanse sustava. Slično kao u prethodnom testu, neoznačeni skup dijeli se u 3 dijela, a označeni skup je ovaj puta fiksne veličine. Ostali parametri su fiksirani na optimalne vrijednosti pronađene u prethodnim mjerenjima.

U Tablici 6.7 prikazana je ovisnost performansi sustava MaxEnt o veličini neoznačenog skupa za učenje.

Tablica 6.7: Utjecaj veličine neoznačenog skupa za učenje – klasifikator MaxEnt

	Broj neoznačenih članaka					
	330		660		1000	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	72.3	66.3	72.3	66.3	72.3	66.3
Iteracija 1	73.8	68.0	74.1	67.8	75.4	69.6
Iteracija 2	**74.5**	**68.7**	75.3	69.1	76.6	70.6
Iteracija 3	74.3	68.6	75.7	69.5	78.7	73.0
Iteracija 4	74.0	68.2	75.9	69.7	79.4	74.0
Iteracija 5	74.4	68.6	76.4	70.2	**80.7***	**75.2***
Iteracija 6	74.3	68.4	76.4	70.2	80.5	74.7
Iteracija 7	74.4	68.6	**76.9**	**71.1**	79.7	74.0
Iteracija 8	74.3	68.5	76.6	70.6	79.2	73.3
δ	2.2	2.4	4.6	4.8	8.4	8.9

Iz tablice se vidi da količina neoznačenih podataka bitno utječe ne performanse sustava. Mjera F1 kada se koriste svi podaci povećana je za oko 6% u odnosu na rezultat postignut korištenjem samo trećine skupa. Ovakav rezultat sugerira da bi sustav korištenjem još veće količine neoznačenih podataka postigao još bolje rezultate. Međutim, mjerenja s većom količinom neoznačenih podataka nisu provedena zbog ograničenog vremena za izvođenje svih testova.

U Tablici 6.8 prikazan je utjecaj veličine neoznačenog skupa za učenje za sustav temeljen na klasifikatoru SVM.

I ovdje se može primijetiti poboljšanje rezultata korištenjem veće količine neoznačenih podataka, no taj je utjecaj daleko manji nego kod MaxEnt klasifikatora.

Iz svih provedenih testova vidljivo je da samoučenje značajno više doprinosi per-

Tablica 6.8: Utjecaj veličine neoznačenog skupa za učenje – klasifikator SVM

	Broj neoznačenih članaka					
	330		660		1000	
	MUC	Exact	MUC	Exact	MUC	Exact
Iteracija 0	81.1	75.2	81.2	75.3	81.0	76.1
Iteracija 1	81.4	76.0	81.6	76.1	**81.9***	**76.2***
Iteracija 2	81.2	75.3	81.1	75.3	81.0	75.5
Iteracija 3	81.0	74.8	**81.3**	**75.6**	80.6	74.4
Iteracija 4	81.3	75.3	81.2	75.4	80.3	73.8
Iteracija 5	**81.6**	**75.7**	80.5	74.2	80.2	73.7
δ	0.5	0.5	0.1	0.3	0.9	0.1

formansama klasifikatora MaxEnt u odnosu na klasifikator SVM. Također, kod klasifikatora SVM doprinos samoučenja prestaje već nakon prve iteracije, dok je to uglavnom slučaj nakon pet iteracija učenja klasifikatora MaxEnt. Razlog bi mogao biti u činjenici da klasifikator SVM postiže vrlo visoke performanse u odnosu na klasifikator MaxEnt već nakon učenja na označenim podacima, pa klasifikator MaxEnt ima mnogo više mjesta za napredak samo da bi dostigao početne performanse SVM-a, dok je klasifikator SVM većinu dostupnih informacija pokupio već u provom koraku učenja.

6.2.3. Kombiniranje klasifikatora

Budući da su implementirana dva klasifikatora s mjerom pouzdanosti, zanimljivo je pogledati može li se kombinacijom rezultata klasifikatora postići bolji rezultat. U Tablici 6.9 prikazani su rezultati označavanja tekstova koristeći kombinaciju klasifikatora. Klasifikatori su kombinirani tako da se za svaki token točnom smatra ona oznaka za koju se postiže maksimalna težinska suma izlaza dvaju klasifikatora.

Tablica 6.9: Kombinacija SVM i MaxEnt klasifikatora

	Utjecaj klasifkatora [MaxEnt/SVM]					
	0.0/1.0	0.2/0.8	0.4/0.6	0.6/0.4	0.8/0.2	1.0/0
Exact match F1	74.6	74.7	75.6	74.8	73.3	72.7
MUC F1	80.5	80.4	80.9	80.0	79.0	78.1

Iz tablice je vidljivo da ovakav pristup nije utjecao na poboljšanje ukupnih rezultata. Ovakav rezultat je i očekivan s obzirom da oba klasifikatora u većini slučajeva daju jednake rezultate, odnosno, griješe na istim primjerima.

Drugi, nešto složeniji način kombiniranja, jest izgradnja trećeg klasifikatora kojem će ulazne značajke biti izlazi dvaju početnih klasifikatora. Isproban je i ovaj način, tako da su mjere pouzdanosti za sve 4 klase (*Osoba*, *Lokacija*, *Organizacija*, *Drugo*) SVM i MaxEnt složene u vektor značajki duljine osam, a izlazna klasa je točna klasa iz označenog skupa. Skup za učenje generiran je označavanjem svih tokena iz označenog skupa za učenje. Rezultati ovog eksperimenta nalaze se u Tablici 6.10.

Tablica 6.10: Kombinacija SVM i MaxEnt klasifikatora 2

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
MaxEnt & SVM	79.4	84.8	82.0	74.5	79.6	77.0

Iz ove tablice vidimo da ovaj način kombiniranja klasifikatora daje nešto bolje rezultate nego što je bio slučaj kod jednostavne težinske sume rezultata, no rezultat opet nije značajno bolji od rezultata najboljeg pojedinačnog klasifikatora.

6.3. Konačni rezultati

U ovom će poglavlju biti prikazani konačni rezultati triju sustava temeljenih na klasifikatoru maksimalne entropije, stroju s potpornim vektorima i skrivenim Markovljevim modelima. Za sustave MaxEnt i SVM optimalnim se pokazalo koristiti dva okolna tokena koji čine kontekst tokena koji se klasificira, a optimalan prag pouzdanosti za izbor novih primjera za učenje kod samoučenja je 0.85. Prema očekivanjima, najbolji rezultati su postignuti korištenjem svih dostupnih označenih i neoznačenih podataka. Za klasifikator maksimalne entropije optimalnim se pokazalo učenje u pet iteracija samoučenja, dok je za SVM klasifikator to samo jedna iteracija nakon inicijalnog učenja na označenim podacima. Za konačno vrednovanje za učenje koristit će se i podaci koji su prethodno korišteni za validaciju, a vrednovanje će biti vršeno na skupu za testiranje koji se sastoji od 40 dokumenata koji nisu korišteni u fazi podešavanja parametara sustava.

U Tablici 6.11 prikazani su konačne performanse svih triju sustava.

Rezultati pokazuju da najbolje performanse postiže sustav temeljen na stroju s potpornim vektorima, slijedi ga sustav temeljen na MaxEnt klasifikatoru s nešto nižim

Tablica 6.11: Konačni rezultati za sve tipove entiteta

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
SVM	84.9	86.5	85.7	79.5	81.0	80.2
MaxEnt	83.8	79.9	81.8	78.0	74.4	76.1
HMM	82.0	52.4	64.0	75.0	47.9	58.4

rezultatom, a najlošijim se pokazao sustav temeljen na skrivenim Markovljevim modelima. Zanimljivo je primjetiti da SVM i MaxEnt imaju sličnu preciznost, ali je odziv SVM-a značajno veći (za oko 6%) što u konačnici daje 4% veći F1. Rezultati prema exact-match vrednovanju su kod svih klasifkatora manji za oko 5% u odnosu na MUC vrednovanje.

U nastavku će biti prikazane performanse svakog sustava za pojedine klase entiteta. Vrednovanje je provedena na način da se svi entiteti koji nisu onog tipa za koji se provodi vrednovanje tretiraju kao običan tekst. Na taj se način uklanja utjecaj performansi sustava na ostalim klasama, i dobiva se onakav rezultat kakav bi davao sustav koji zaista na izlazu daje binarnu vrijednost klasifikacije. U retku *ENAMEX* prikazan je ukupan rezultat za sve 3 klase iz skupine *ENAMEX*. U retku *Is ENAMEX* nalazi se rezultat za binarnu klasifikaciju – određuje se samo pripada li entitet nekoj klasi iz skupine *ENAMEX* ili se radi o običnom tekstu. Tablica 6.12 prikazuje performanse sustava temeljenog na klasifkatoru SVM.

Tablica 6.12: Konačni rezultati SVM-označivača po tipovima

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
Osoba	78.7	81.1	79.9	75.6	78.0	76.8
Lokacija	91.3	85.7	88.4	88	82.7	85.3
Organizacija	75.7	81.0	78.3	69.5	74.4	71.9
ENAMEX	81.4	83.7	82.6	75.1	77.2	76.2
Is ENAMEX	88.2	89.3	88.7	83.7	84.7	84.2

Iz tablice su vidljive relativno velike razlike u performansama za različite klase. Najbolji rezultati postižu se za klasu *Lokacija*, a najslabiji za klasu *Organizacija*. Iz Tablice 3.1 može se vidjeti da se u tekstovima najviše pojavljuju baš entiteti tipa organizacija, iz čega bi se moglo očekivati da će klasifikator iz puno primjera bolje naučiti takvu klasu. Međutim, promatranjem entiteta tipa *Organizacija* može se vidjeti da su

takvi entiteti najsloženiji, često se sastoje od nekoliko riječi ili pojavljuju samo kao akronimi što objašnjava ovakvo ponašanje klasifikatora.

Tablica 6.13 prikazuje performanse sustava temeljenog na klasifikatoru maksimalne entropije.

Tablica 6.13: Konačni rezultati MaxEnt-označivača po tipovima

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
Osoba	75.2	63.8	69.0	65.9	56.0	60.5
Lokacija	90.5	75.3	82.2	89	74.0	80.8
Organizacija	74.6	77.3	75.9	68.8	71.3	70.0
ENAMEX	79.8	75.4	77.5	72.8	68.8	70.7
Is ENAMEX	86.3	82.4	84.3	81.2	77.6	79.3

I u ovom slučaju sustav pokazuje najbolje performanse za entitete tipa *Lokacija*, ali su najlošiji rezultati ostvareni kod klase *Osoba*. Iz tablice se može vidjeti da je za klasu *Osoba* odziva daleko manji nego za ostale klase, i u usporedbi s odzivom za istu klasu kod klasifikatora SVM.

Tablica 6.14 prikazuje performanse sustava temeljenog na skrivenim Markovljevim modelima.

Tablica 6.14: Konačni rezultati HMM označivača po tipovima

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
Person	80	12.6	21.7	68	10.7	18.5
Location	57.8	58.4	58.1	56.1	56.6	56.3
Organization	82.7	42.8	56.4	74.8	38.6	51
ENAMEX	75.2	42.7	54.5	65.9	37.5	47.8
Is ENAMEX	84.9	56.3	67.7	78.6	52.2	62.7

Ovaj sustav pokazao je daleko najslabije rezultate. Kao i u prethodna dva sustava, najbolji rezultati postignuti su za klasu *Lokacija*. Najlošiji rezultat postignut je za klasu *Osoba* kao i u slučaju sustava temeljenog na klasifikatoru MaxEnt. Problem odziva za tu klasu ovdje je još puno izraženiji, iznosi tek oko 12%, dok je preciznost čak i veća nego kod ostalih sustava.

U Tablici 6.15 su prikazani rezultati za skupine *NUMEX* i *TIMEX*. Ovi rezultati nisu prikazivani posebno za svaki od tri implementirana sustava, jer se u sva tri slučaja

za prepoznavanje ovih tipova koristi jednak podsustav temeljen na regularnim izrazima.

Tablica 6.15: Konačni rezultati za numeričke tipove

	MUC P	MUC R	MUC F1	Exact P	Exact R	Exact F1
Postotak	98.8	98.8	98.8	97.6	97.6	97.6
Novac	99.2	97.7	98.4	98.4	96.9	97.7
NUMEX	99	98.3	98.6	98	97.3	97.6 1
Datum	97.2	95.4	96.3	94.3	92.6	93.5
Vrijeme	83.3	62.5	71.4	66.7	50	57.1
TIMEX	96.4	93.1	94.7	92.9	89.7	91.2

Iz tablice je vidljivo da su performanse na ovim tipovima relativno velike u odnosu na preostala tri tipa entiteta. Takvi rezultati su mogući zbog čvrste strukture entiteta u ovoj skupini, koje je moguće dobro opisati regularnim izrazima. Također se može primijetiti da rezultati za klasu *Vrijeme* bitno odstupaju od ostalih rezultata, no kako se može vidjeti u Tablici 3.1 takvih je entiteta tek 7 (odnosno 0.19%) u označenim skupovima, pa se ovaj rezultat niti ne može smatrati pouzdanim.

6.4. Analiza pogrešaka

U ovom poglavlju navedene su česte pogreške razvijenih NERC sustava.

6.4.1. Označavanje naziva događaja, proizvoda i sl.

Klasifikatori često nazive proizvoda ili događaja označuju kao entitet, najčešće oznakom *Organizacija* ili *Osoba*:

- *"...trgovaca ne radi samo Kplus za Konzum..."* – SVM i MaxEnt označuju *"Kplus"*, kao entitet tipa *Organizacija* iako se ništa ne treba označiti.
- *"... testirala utjecaj odredbi Solvency II na poslovanje ...."* – ponovno oba klasifikatora označuju *"Solvency II"*, iako to nije entitet.

Razlog za ovakve pogreške mogao bi biti u činjenici da se nazivi najčešće pišu velikim početnim slovom, a to je klasifikatorima vjerojatno važan signal da se radi o entitetu. Također, to zapravo i jesu entiteti (ali tipa koje sustav ne bi trebao prepoznavati) i zbog toga imaju slične kontekste kao drugi entiteti.

6.4.2. Označavanje okolnih tokena

Jedna od tipičnih grešaka je i označavanje dodatnih tokena ispred ili iza granice entiteta. Ovaj problem češće se pojavljuje kod SVM-označivača:

- *"...prognozama EBRD za cijelu regiju..."* – SVM ovdje kao organizaciju prepoznaje *"EBRD za"* iako je entitet samo *"EBRD"*.
- *"...tvrtke Ragusa čakovečki poduzetnik..."* – SVM označuje *"Ragusa čakovečki"*, a pravi entitet je *"Ragusa"*.

Iz ovih primjera se vidi da su oba primjera označena oznakom *Organizacija*, a kako se nazivi organizacija često sastoje od nekoliko riječi, moguće je da značajka koja govori da je prošla riječ bila tipa *Organizacija* mnogo utječe na odluku za sljedeću riječ, i uz *pogodan* kontekst nekad dovodi do krivog zaključka da se entitet proteže i kroz sljedeći token.

6.4.3. Neoznačavanje cijelog entiteta

Iako rjeđe nego prethodni tip pogreške, označavanje samo dijela entiteta može se istaknuti kao tipična greška:

- *"...uspjela prodati RTB Bor iako ..."* – označuje se samo *"Bor"* a puni entitet je *"RTB Bor"*.
- *"... okviru svoje tvrtke mStart telekomunikacije te ..."* – SVM označuje *"mStart"*, a pravi entitet je *"mStart telekomunikacije"*.

U ovim primjerima klasifikatori ne uspijevaju prepoznati sve tokene koji čine entitet. Uzrok bi mogao biti u tome što se neprepoznate riječi često koriste i ne pripadaju entitetima *telekomunikacije* pa klasifikatori nisu uspjeli odrediti točan tip, bez obzira na oznaku prethodnog tokena i konteksta u kojem se riječ pojavljuje.

7. Zaključak

Za razvoj naprednih metoda pretraživanja i ekstrakcije informacije bitne su semantičke značajke teksta. Imenovani entiteti skupina su takvih značajki koje dijelovima teksta pridružuju oznake različitih klasa, npr. osoba, lokacija, organizacija i dr.

U ovom diplomskom radu zadatak je bio proučiti metode za označavanje i klasifikaciju imenovanih entiteta i implementirati takav sustav za tekstove na hrvatskom jeziku. Rad je prvenstveno usmjeren na rješavanje ovog problema korištenjem sljedećih metoda strojnog učenja: klasifikatora maksimalne entropije, stroja s potpornim vektorima i skrivenih Markovljevih modela. Uz metode strojnog učenja u radu se kombiniraju i popisi pojmova te pravila.

Razvijena su tri NERC sustava temeljan na navedenim klasifikatorima, a kao metoda učenja korišten je algoritam samoučenja koji iskorištava i dostupne neoznačene podatke. Popisi pojmova korišteni su kao izvor značajki za klasifikatore. Za određene tipove entiteta dobro definirane strukture pogodno je korištenje pravila – u radu je implementiran podsustav za takve tipove korištenjem regularnih izraza. Podaci za učenje također su prikupljeni i označeni u sklopu ovog rada.

Razvijeni sustavi su vrednovani, a osim konačnih performansa sustava ispitan je i utjecaj količine dostupnih podataka za učenje na uspješnost sustava. Rezultati su, očekivano, pokazali da veća količina dostupnih podataka pozitivno utječe na konačne performanse sustava, a za klasifikator maksimalne entropije količina neoznačenih podataka pokazala se znatno važnijom nego što je to slučaj kod klasifikatora SVM.

Rezultati vrednovanja pokazuju da NERC-sustav temeljen na klasifikatoru SVM daje najbolje rezultate, slijedi ga sustav temeljen na klasifikatoru MaxEnt, a sustav temeljen na HMM-u pokazao je znatno lošije rezultate. Mjera F1 najboljeg sustava iznosi 85.7 prema MUC-vrednovanju, odnosno 80.2 prema *exact-match* vrednovanju.

Daljnji rad na sustavu uključivao bi izradu većeg skupa za učenje, što bi prema rezultatima analize utjecaja veličine skupova za učenje moglo dodatno poboljšati performanse sustava. Osim toga treba dodatno razraditi označivač temeljen na popisima kako bi davao kvalitetnije značajke za učenje klasifikatora.

LITERATURA

Steven Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st izdanju, 2007. ISBN 1584885599, 9781584885597.

Božo Bekavac i Marko Tadić. Implementation of Croatian NERC system. U *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, stranice 11–18, 2007.

Adam L. Berger, Vincent J. Della Pietra, i Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, Ožujak 1996. ISSN 0891-2017.

Hai Leong Chieu i Hwee Tou Ng. Named entity recognition with a maximum entropy approach. U *In Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, stranice 160–163, 2003.

Nancy A. Chinchor. Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition. U *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, stranica 21 pages, Fairfax, VA, April 1998. version 3.5, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Corinna Cortes i Vladimir Vapnik. *Machine Learning*, (3):273–297.

James R. Curran i Stephen Clark. Investigating GIS and smoothing for maximum entropy taggers. U *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, stranice 91–98, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, i Alexander Yates. Unsupervised named-entity extraction from the Web: An experimental study. *Artif. Intell.*, 165(1):91–134, Lipanj 2005. ISSN 0004-3702.

Michael Fleischman i Eduard Hovy. Fine grained classification of named entities. U *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, stranice 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Radu Florian, Abe Ittycheriah, Hongyan Jing, i Tong Zhang. Named entity recognition through classifier combination. U *In Proceedings of CoNLL-2003*, stranice 168–171, 2003.

Ralph Grishman i Beth Sundheim. Design of the MUC-6 evaluation. U *Proceedings of the 6th conference on Message understanding*, MUC6 '95, stranice 1–11, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics. ISBN 1-55860-402-2.

Ralph Grishman i Beth Sundheim. Message Understanding Conference-6: a brief history. U *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, stranice 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

Hideki Isozaki i Hideto Kazawa. Efficient support vector classifiers for named entity recognition. U *Proceedings of the 19th international conference on Computational linguistics*, stranice 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

Heng Ji i Ralph Grishman. Data selection in semisupervised learning for name tagging. U *In ACL 2006 Workshop on Information Extraction Beyond the Document:48-55*, 2006.

Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, i Jun'ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. U *In Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, stranice 1–8, 2002.

Seungwoo Lee i Gary Geunbae Lee. Heuristic methods for reducing errors of geographic named entities learned by bootstrapping. U *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, stranice 658–669, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-29172-5, 978-3-540-29172-5.

James Mayfield, , James Mayfield, Paul Mcnamee, i Christine Piatko. Named entity recognition using hundreds of thousands of features. U *In Proceedings of CoNLL-2003*, stranice 184–187, 2003.

Andrei Mikheev, Claire Grover, i Marc Moens. Description of the LTG system used for MUC-7. U *In Proceedings of 7th Message Understanding Conference (MUC-7*, 1998.

David Nadeau i Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. U *Proceedings of the IEEE*, stranice 257–286, 1989.

Lev Ratinov i Dan Roth. Design challenges and misconceptions in named entity recognition. U *Proceedings of the thirteenth conference on computational natural language learning (CONLL)*, stranice 147–155, 2009.

Erik F. Tjong Kim Sang i Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. U *Proceedings of CoNLL-2003*, stranice 142–147, 2003.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, i Chew-Lim Tan. Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. U *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, BioMed '03, stranice 49–56, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

Marko Tadić. Building the Croatian National Corpus. U *LREC 2002 zbornik*, stranice 441–446, 2002.

Dodatak A
Format ulaznih i izlaznih dokumenata

Razvijeni sustav zahtjeva da ulazni dokumenti poštuju sljedeći XML format:

```
<?xml version="1.0" encoding="utf-8"?>
<doc textOnly="no">
  <title>
    <s>Prva rečenica.</s>
    <s>Druga rečenica.</s>
    .
    <s>k-ta rečenica.</s>
  </title>
  <body>
    <section>
      <s>Prvi odlomak, prva rečenica.</s>
      <s>Prvi odlomak, druga rečenica.</s>
      .
      <s>Prvi odlomak, n-ta rečenica.</s>
    </section>
    .
    .
    .
    <section>
      <s>m-ti odlomak, prva rečenica.</s>
      <s>m-ti odlomak, druga rečenica.</s>
      .
      <s>m-ti odlomak, n-ta rečenica.</s>
    </section>
  </body>
</doc>
```

Dakle, dokument se sastoji od naslova i tijela dokumenta. Naslov može sadržati 0 ili više rečenica. Tijelo dokumenta sastoji se od 0 ili više odlomaka, a svaki od njih ponovno sadrži 0 ili više rečenica.

Označeni dokument na izlazu sustava također će poštovati gore opisani format, a entiteti će biti označeni sljedećim XML oznakama:

- Osoba – `<enamex type="Person">Pero Perić</enamex>`
- Lokacija – `<enamex type="Location">Zagreb</enamex>`
- Organizacija – `<enamex type="Organization">Hrvatska narodna banka</enamex>`
- Postotni izraz – `<numex type="Percent">5 do 7%</numex>`
- Valutni izraz – `<numex type="Money">100 kuna</numex>`
- Datum – `<timex type="Date">17. siječnja 2012.</timex>`
- Vrijeme – `<timex type="Time">17:00</timex>`

U nastavku je prikazan primjer označenog dokumenta:

```
<?xml version="1.0" encoding="utf-8"?>
<doc textOnly="no">
  <title>
    <s>Izvoz u <enamex type="Location">Hrvatskoj</enamex> pao
       za <numex type="Percent">6%</numex></s>
  </title>
  <body>
    <section>
      <s>Guverner <enamex type="Organization">Hrvatske
         narodne banke</enamex> <enamex type="Person">Željko
         Rohatinski</enamex> izjavio je da je hrvatski izvoz
         u <timex type="Date">listopadu 2010.</timex> pao za
         <numex type="Percent">5 do 6 posto</numex>.</s>
    </section>
    <section>
      <s>Robna razmjena s članicama <enamex
         type="Organization">EU</enamex> iznosila je skromnih
         <numey type="Money">150 milijuna eura</numex>.</s>
    </section>
  </body>
</doc>
```

Dodatak B
Primjeri označenih dokumenata

Oznake:
Organizacija Osoba Lokacija
Postotni izraz Valutni izraz Datum Vrijeme

Agrokor zaštićuje neretvansku mandarinu

Nakon paške soli u koncernu Agrokor intenzivno rade na tome da oznakom o zemljopisnom podrijetlu zaštite neretvansku mandarinu. Kompliciran i zahtjevan postupak ulazi već u drugu godinu i odvija se u suradnji sa zagrebačkim Prehrambeno - tehnološkim fakultetom, s kojim koncern u vlasništvu Ivice Todorića ima višegodišnju suradnju na različitim projektima.

"Suradnja je posebno intenzivirana od listopada 2009. godine kada smo potpisali ugovor za projekt neretvanske mandarine. Istraživanje je nakon prve godine i predstavljenih obećavajućih rezultata produljeno i na drugu godinu. Pri kraju je i poslužit će kao podloga za pokretanje postupka zaštite mandarine oznakom zemljopisnog podrijetla iz doline Neretve. Postupak je kompleksan i zahtjevan, potrebno je proći propisanu proceduru i prema našim dosadašnjim iskustvima za završetak treba proći i do dvije godine", pojašnjavaju u Agrokoru.

Ta je kompanija još 2008. godine oznakom izvornosti zaštitila pašku sol (Agrokor je vlasnik Solane Pag koja ima tržišni udio od 40 posto). Ove je godine, međutim, pokrenut postupak preregistracije na oznaku zemljopisnog podrijetla radi izvjesnijeg dobivanja oznake na europskoj razini. Pojednostavljeno govoreći, oznaka izvornosti je jača i teža u odnosu na oznaku o zemljopisnom podrijetlu, s tim da je i za ovu posljednju procedura zahtjevna i dugotrajna. Zahtjev za registriranje oznake podnose

udruge proizvođača, a o njemu odlučuje posebno povjerenstvo pri Ministarstvu poljoprivrede. Obavijest o zahtjevu se mora objaviti u Narodnim novinama od kada teče tromjesečni rok za podnošenje prigovora...

Zaštita autohtonih poljoprivredno-prehrambenih proizvoda je važna zato što se takvi proizvodi odmah stavljaju u višu cjenovnu kategoriju i pridonose komercijalnoj vrijednosti proizvoda. U slučaju neretvanske mandarine za Agrokor bi to bila izuzetno bitna stvar s obzirom na to da ta kompanija godišnje otkupi oko 35.000 tona mandarina, ovisno o godini i urodu. Od toga se više od 75 posto izveze, dok se oko 15.000 tona distribuira na domaćem tržištu.

Oko šest tisuća tona mandarina distribuira se kroz Agrokorov trgovački lanac Konzum i njegovu veleprodaju Velpro. Iako je Agrokor najveći domaći poljoprivredni proizvođač s godišnjom proizvodnjom u vrijednosti od oko tri milijarde kuna (za što godišnje dobije oko 150 milijuna kuna poticaja), kada su u pitanju mandarine koncern se pojavljuje samo kao otkupljivač. Surađuje s velikim brojem kooperanata upravo u dolini Neretve gdje su investirali u hladnjaču i kalibratore. U trenutku kad dobiju oznaku o zemljopisnom podrijetlu za neretvansku mandarinu, u Agrokoru će već imati dogovoreno oko 300.000 eura sredstava od EBRD-a za njezin marketing.

Najvažnija izvozna tržišta za neretvansku mandarinu su Engleska i Rusija, te Slovenija, Srbija, BiH, Austrija, Slovačka i Nizozemska. Posebno im u Agrokoru laska što se mandarina u Velikoj Britaniji nalazi na policama najjačih trgovačkih igrača poput Tescoa, Asde i Sainsburyja. Inače, neretvanski kraj ima potencijal za proizvodnju čak 150.000 tona mandarina godišnje (za sada se koristi tek njegova trećina).

Gradska plinara ZG potpisala ugovor s poznatim spasiteljem iz vremena plinske krize

Gradska plinara Zagreb-Opskrba (GPZ-O) potpisala je ugovor s njemačkom tvrtkom E. ON Ruhrgas o isporukama plina u 2012 godini, a to će biti tek manji dio potrebne količine plina na godišnjoj razini koje GPZ-O isporučuje svojim klijentima u Zagrebu i okolici, izvijestili su danas iz Gradske plinare Zagreb iz koje ističu i da je to prva isporuka koja dolazi od strane novog dobavljača.

"Isporuka počinje u siječnju 2012. godine te će biti tek manji dio potrebne količine plina na godišnjoj razini koju će GPZ-O isporučiti svojim klijentima u Zagrebu i okolici", navodi se u priopćenju. Neslužbeno se doznaje da su količine plina koje će Gradskoj plinari Zagreb isporučiti njemački E. ON Ruhrgas na razini 15-ak posto

godišnjih količina koje svojim klijentima isporuči Gradska plinara Zagreb.

Kako se navodi u priopćenju, Gradska plinara Zagreb-Opskrba će iskoristiti nove zalihe kako bi optimizirala vlastiti portfelj zaliha – osobito vezano za diverzifikaciju i konkurentnost nabavne cijene. "S obzirom na to da je ova isporuka prva koja dolazi od strane novog dobavljača, to također ukazuje na korak naprijed ka otvaranju hrvatskog tržišta plina", ističu iz Gradske plinare Zagreb.

Direktor tvrtke GPZ-O Mladen Pejnović ističe kako se partnerstvo s E. ON-om koje je otvoreno prema tržištu, zajedno s infrastrukturnim snagama E. ON-a u regiji, uklapa u potrebe Gradske plinare Zagreb-Opskrba. "To će također pomoći i ubrzati otvaranje tržišta u Hrvatskoj" objašnjava Pejnović. "Sigurni smo da će naše isporuke pridonijeti sigurnosti nabave i konkurentnosti.E. ON Ruhrgas ima široki portfelj koji obuhvaća raznolike sigurne zalihe plina i skladišta. Na temelju ovoga veselimo se daljnjem doprinosu otvaranju hrvatskoga tržišta", izjava je višeg potpredsjednika prodaje za Europu iz tvrtke E. ON Ruhrgas Haralda Drauba koja se prenosi u priopćenju.

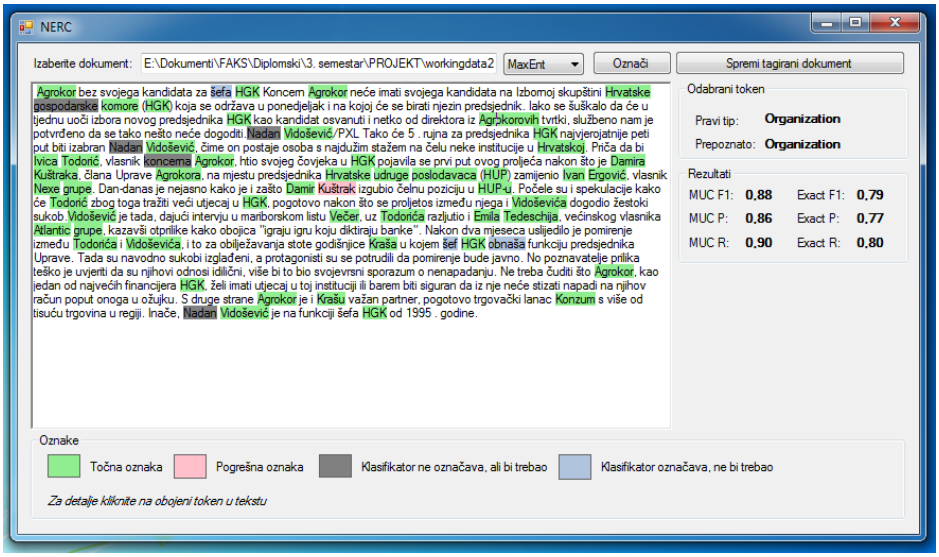
Tvrtka E. ON Ruhrgas, iz Essena, jedna je od velikih europskih tvrtki za plin, klijenti su joj lokalni i regionalni dobavljači energije, industrijski klijenti i elektrane.E-ON Ruhrgas uspio je osigurati hitne isporuke plina u kratkom roku tijekom prekida u nabavi plina u siječnju 2009. godine, podsjeća se u priopćenju.

GPZ-O je najveća i vodeća tvrtka za distribuciju plina u Hrvatskoj i dio je Zagrebačkog holdinga, a opskrbljuje plinom kućanstava, poslovne i industrijske klijenate u Zagrebu i okolici. (Hina / pd)

Dodatak C
Upute za korištenje

Na slici C.1 prikazano je sučelje aplikacije.

Slika C.1: Izgled sučelja aplikacije



Da bi se označilo XML dokument u formatu opisanom u *Dodatku A* najprije treba kliknuti u prazno polje (lijevo od dugmeta "Označi") da bi se otvorio prozor za odabir datotetke. Nakon što je datoteka odabrana, pritiskom na dugme "Označi" dokument će biti označen i prikazan.

Za označavanje se može koristiti NERC sustav temeljen na klasifikatoru MaxEnt, SVM ili HMM. Odabir se vrši u izborniku koji se nalazi desno od imena datoteke.

Desno od prikaza datoteke prikazuje se točnost označavanja ako je dokument već sadržavao oznake, a klikom na bilo koju obojanu riječ s desne strane pokazuje se očekivani i prepoznati tip tokena.

Označena datoteka može se spremiti klikom na dugme "Spremi označeni dokument".

Prepoznavanje i klasifikacija imenovanih entiteta u tekstovima na hrvatskome jeziku

Sažetak

Označavanje i klasifikacija imenovanih entiteta (NERC) bitno je za napredne metode pretraživanja i ekstrakcije informacija, kao i za druge primjene u obradi prirodnog jezika. U ovom radu proučene su različite metode za prepoznavanje i klasifikaciju imenovanih entiteta. Razvijena su tri NERC sustava temeljena na različitim metodama strojnog učenja: MaxEnt, SVM i HMM. Za učenje klasifikatora MaxEnt i SVM korišteno je polunadzirano učenje, a kao izvor značajki korišteni su i popisi pojmova. Za prepoznavanje nekih grupa entiteta koriste se regularni izrazi. Sustav je učen i vrednovan na skupu podataka koji je također pripremljen kao dio ovog rada. Dobiveni rezultati su ohrabrujući, a sustav postiže mikro-prosječni F1 od preko 0.85 prema MUC-ovom načinu vrednovanja.

Ključne riječi: ekstrakcija informacija, imenovani entiteti, klasifikacija, SVM, MaxEnt, HMM, polunadzirano učenje, samoučenje, hrvatski jezik

Named entity recognition and classification for text in Croatian language

Abstract

Named entity recognition and classification (NERC) is very important for advanced information retrieval techniques, information extraction as well as for other applications in natural language processing. In this work we have studied different methods for named entity recognition and classification. We have developed three NERC systems based on different machine learning methods: MaxEnt, SVM and HMM. To train SVM and MaxEnt based systems we use semi-supervised learning, and gazetteer are used as a feature source. For some entity groups regular expression are used. System is trained and evaluated on a set of documents which is also prepared as part of this work. Evaluation results are encouraging; developed NERC system achieves micro-average F1 over 0.85 (MUC evaluation).

Keywords: information extraction, named entities, NERC, classification, SVM, MaxEnt, HMM, semi-supervised learning, self-learning, Croatian language