Laboratorij za tehnologije znanja (KTLab)
Zavod za elektroniku, mikrolektroniku, računalne i inteligentne sustave
Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu

Interni dokument

© 2009 KTLab

Niti jedan dio ovog dokumenta ne smije se fotokopirati,
umnožavati niti prevoditi na drugi jezik
bez prethodnog pismenog odobrenja.

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

INTERNI DOKUMENT

DIPLOMSKI RAD br. 1802

METODA MAKSIMALNE ENTROPIJE I NJENA PRIMJENA ZA OZNAČAVANJE SLIJEDNOG NIZA TEKSTNIH PODATAKA

Grga Ćurković

Zagreb, rujan 2009.

Zahvaljujem prof. dr. sc. Dalbelo Bašić na vodstvu kroz meni, do nedavno, nepoznata područja crpljenja informacija i strojnog učenja.

Zahvaljujem mr. sc. Janu Šnajderu na besprijekornom lektoriranju i redaktiranju.

Zahvaljujem Juri Mijići i Frani Šariću na mnogim korisnim savjetima.

Na kraju zahvaljujem svojim roditeljima i Mirni...

INTERNI DOKUMENT

Popis tablica

INTERNI DOKUMENT

Popis slika

INTERNI DOKUMENT

Sadržaj

INTERNI DOKUMENT

INTERNI DOKUMENT

1. Uvod

Mogućnost da velike količine znanja i informacija pohranimo u pisanom obliku omogućila je razvoj naše civilizacije. Nakon tisuća godina tokom kojih je način pohrane informacija ostao nepromijenjen, tokom zadnjih nekoliko desetljeća svjedoci smo revolucionarnih promjena na tom polju. Računala su nam omogućila da gotovo sve informacije imamo dostupne uvijek i svugdje.

Upravo zbog toga danas smo suočeni s drugim problemom, kako iz goleme količine dostupnih informacija dobiti baš onu informaciju koju trebamo. Iako računala imaju nevjerojatnu mogućnost pohranjivanja podataka, prilikom crpljenja obavijesti suočena su s velikim problemom jer ne razumiju prirodni jezik koji se koristi za pohranu znanja i informacija. Stoga je obrada prirodnog jezika (engl. *Natural Language Processing, NLP*), polje računarne znanosti posvećeno izgradnji sustava koji koriste (prirodne) ljudske jezike kao ulaz i izlaz [1], iznimno bitno područje istraživanja.

Ovaj rad proučava prepoznavanje i klasifikaciju naziva (engl. *Named Entity Recognition and Classification, NERC*), prvi od pet zadataka unutar područja crpljenja informacija. U prvom dijelu rada razmatra se teorijska podloga te mogući pristupi rješavanju tog zadatka. U drugom dijelu rada opisuje se implementacija, testiranje i rezultati odabranog pristupa rješavanju problema.

INTERNI DOKUMENT

2. Teorijska podloga

Prepoznavanje naziva (engl. *Named Entity Recognition, NER*) je postupak kojim se identificiraju i označavaju nazivi u tekstu. Nakon prepoznavanja naziva najčešće se obavlja klasifikacija naziva (engl. *Named Entitiy Classification, NEC*) u unaprijed definirane kategorije. Konkretne implementacije vrlo često sjedinjuju obavljanje ovih radnju u jedinstveni postupak koji se tada naziva prepoznavanje i klasifikacija naziva, u nastavku rada PKN.

Termin *naziv* u ovom radu imati će ono značenje koje u engleskom jeziku ima termin *Named Entity*. Prema konferenciji MUC-7 u nazive pripadaju imena, određeni vremenski i brojčani izrazi i izrazi koji se referenciraju na entitete koji su u ovom slučaju osobe, organizacije i lokacije. Općenito govoreći, termin naziv ne mora nužno obuhvaćati samo one kategorije definirane na *Message Understanding Conferences* 7 (*MUC*-7) već se može proizvoljno proširiti tako da obuhvaća podatke kao što su adrese, imena kemijskih spojeva itd.

Ovaj rad pripada području računarne obradbe prirodnog jezika, znanosti koja se bavi proučavanjem računalnog sustava koji obrađuje prirodni jezik. Cilj ove discipline jest što učinkovitije obraditi jezične podatke, a pritom utrošiti što manje računalnih resursa i vremena. S gledišta znanosti o jeziku ovaj rad se može svrstati u područje računalne lingvistike, znanosti koja se bavi proučavanjem prirodnog jezika pri čemu se koristi računalom kao pomoćnim sredstvom. Cilj ove discipline jest što kvalitetnije opisati jezične činjenice, neovisno o potrošnji računalnih resursa. Ova podjela nastala je u vrijeme kada su računalni resursi bili skupi te je bilo potrebno odlučiti je li važnija brzina ili točnost obrade. Kako je u novije vrijeme došlo do značajnog povećanja performansi računala danas se ove grane sve više približavaju te više nisu potrebna kompromisna rješenja [11].

PKN se na prvi pogled čini kao jednostavan zadatak. Trivijalna rješenja poput prepoznavanja riječi koje su pisane velikim početnim slovom isprva se čine kao adekvatna, no takva rješenja nailaze na nepremostive probleme zbog činjenice da se i prva riječ u rečenici također piše velikim početnim slovom. Korištenje popisa imena, lokacija ili organizacija iziskivalo bi enormnu količinu vremena, a bili bi nepotpuni i zastarjeli već sljedeći dan nakon svog izdavanja zbog, npr., osnivanja novih tvrtki. Priroda hrvatskog jezika unosi dodatne komplikacije prilikom korištenja ovakvog pristupa zbog različitih oblika istih naziva. Čak i da uspijemo izraditi kompletne popise imena, organizacija i

INTERNI DOKUMENT

lokacija, suočili bi se s velikim problemom zbog preklapanja tih popisa. Na primjer, riječ *Hrvatska* bi se osim u popisu lokacija mogla naći i u popisu organizacija (npr. *Hrvatska osiguravateljska kuća*) i u popisu imena (npr. *Ivan Hrvatska*).

Za poznatije jezike postoje gotovi sustavi za PKN, međutim pri razmatranju ideje o preuzimanju takvog sustava treba biti vrlo oprezan. Mogući su znatno lošiji rezultati zbog specifičnosti jezika za koji je izgrađen sustav. Tako primjerice sustav izgrađen za engleski jezik vjerojatno ne bi dao ni približno zadovoljavajuće rezultate za hrvatski jezik zbog različitosti kao što su poredak riječi u rečenici, padeži itd. Postoje sustavi za PKN temeljeni na strojnom učenju koji su jezično neovisni, te bi adaptacija takvog sustava bila moguća, no njihovi rezultati nisu na razini sustava koji su izrađeni za određeni jezik.

2.1. Crpljenje obavijesti

Crpljenje obavijesti definirano je u [2] kao identifikacija, klasifikacija i strukturiranje specifičnih informacija pronađenih u nestrukturiranim izvorima podataka u semantičke klase, čime se informaciju čini primjerenijom za daljnju obradu. Crpljenje obavijesti nerijetko se brka s pronalaženjem obavijesti. Crpljenje obavijesti je pronalaženje važnih podataka unutar dokumenata dok se pronalaženje obavijesti (engl. *information retrieval*) bavi pronalaženjem cijelih dokumenata [3].

PKN je jedan od pet zadataka koji čine crpljenje obavijesti, ovdje ćemo ih izložiti redom kojim se obavljaju:

1. Prepoznavanje i klasifikacija naziva (engl. *Named Entity Recognition and Classification)*
   - pronalazi i klasificira određene nazive u tekstu.
2. Razrješavanje koreferencija (engl. *Coreference Resolution*)
   - identifikacija veza među entitetima u tekstu – osobina je teksta da se na iste izvan jezične entitete najčešće ne referira istim nazivima, već se zamjenjuju skraćenim oblicima ili zamjenicama.
3. Izrada obrasca elementa (engl. *Template Element Production*)
   - na osnovi obavijesti iz prvih dvaju koraka združuju se prikupljeni podatci i dodaju deskriptivne informacije nazivima
4. Konstruiranje odnosa među obrascima (engl. *Template Relation Construction*
   - pronalazi odnose među obrascima izrađenim u prethodnom koraku

INTERNI DOKUMENT

5. Izrada obrasca scenarija (*Scenarion Template Production*)
   - usklađuje rezultate prethodna dva koraka u određeni scenarij događaja

Primjer

Razmotrimo slijedeći primjer kako bismo dobili jasniju predodžbu zadataka crpljenja informacija – (primjer preuzet iz [4]).

Sjajna crvena raketa lansirana je u utorak. Ona je izum dr. Ludoga Znanstvenika. Dr. Znanstvenik je glavni znanstvenik u Mi Gradimo Rakete Inc.

Rezultat svakog od navedenih zadataka crpljenja obavijesti je slijedeći:

- NE obilježava (podcrtano u primjeru) prisutne nazive
- CO otkriva da se Ona odnosi na raketu te da se dr. Ludoga Znanstvenika i Dr. Znanstvenik odnose na istu osobu
- TE opisuje da je raketa Sjajna crvena i da je Znanstvenikov izum
- TR otkriva da Dr. Znanstvenik radi za Mi Gradimo Rakete Inc.
- ST otkriva da se dogodilo lansiranje rakete s raznim sudionicima

2.2. Konferencije o razumijevanju poruka

Već spomenuta konferencija MUC-7 zadnja je u nizu konferencija o razumijevanju poruka koje su devedesetih godina prošlog stoljeća znatno pridonijele razvoju crpljenja obavijesti i postavile brojne standarde. Organizator konferencija je američka agencija *The Defense Advances Research Projects Agency* (*DARPA*). PKN je bila u posebnom žarištu konferencija MUC-6 i MUC-7. Konferencija MUC-7 održana 1998. godine bila je natjecateljskog tipa s ukupno 12 natjecatelja odnosno sustava od koji je većina bila temeljena na ručno kodiranim pravilima.

Prema MUC specifikaciji svaku riječ moguće je svrstati u jednu od osam kategorija: imena osoba, imena organizacija, imena lokacija, oznake datuma, oznake vremena, brojčani postoci, monetarni iznosi i ostalo. MUC specifikacija obuhvaća izraze koji daju odgovore na osnovna obavijesna pitanja (*tko? kada? gdje? što? koliko?*) te su time nosioci velike količine obavijesti u tekstu.

Nazivi su prema konferenciji MUC kategorizirani na sljedeći način:

INTERNI DOKUMENT

- Imena osoba, organizacija i lokacija,
- Vremenski i datumski izrazi,
- Brojčani izrazi postotaka i monetarnih valuta.

Ovim kategorijama pripadaju odgovarajuće oznake: ENAMEX (imena), TIMEX (vremenski izrazi), NUMEX (brojčani izrazi). Ovi entiteti preuzeti su iz smjernica *Text Encoding Initiative* (TEI) za kodiranje i razmjenu elektroničkih tekstova. Dogovoreni jezik za obilježavanje MUC tekstova je *Standard Generalized Markup Language* (SGML), gdje pronađeni entiteti u tekstu trebaju biti obilježeni kao elementi s pripadajućim atributima po sljedećem uzorku:

<*oznaka_entiteta* TYPE="*tip_entiteta*">*entitet*</*oznaka_entiteta*>

Potencijalni elementi i atributi definirani na koferenciji MUC-7 su [5]:

1. Za ENAMEX entitete
    a. ORGANIZATION – tvrtke, državne institucije i druge organizacije
    b. PERSON – vlastita imena i prezimena
    c. LOCATION – imena politički i geografski definiranih lokacija (gradovi, pokrajine, države, rijeke, planine itd.)
2. Za TIMEX entitete
    a. DATE – potpuni ili nepotpuni izrazi za datume
    b. TIME – potpuni ili nepotpuni izrazi za vremena unutar dana
3. Za NUMEX entitete
    a. MONEY – novčani izrazi
    b. PERCENT – postotci

Prema MUC specifikaciji nazivi nisu:

- Opće imenice koje se referiraju na nazive,
- Naslovi,
- Imena skupna i stvari nazvane prema osobama,
- Pridjevi izvedeni od naziva,
- Brojevi koji nisu vremenska razdoblja, datumi, postotci ili novčani izrazi.

Konferencije MUC više se ne održavaju, ali postoji nekoliko konferencija koje se smatraju konferencijama nasljednicama: *Conference on computational Natural Language Learning* (CoNLL) i *Automated Content Extraction* (ACE). CoNLL se održava svake godine s određenom temom. Godine 2002. i 2003. tema je bila „Jezično neovisno

INTERNI DOKUMENT

prepoznavanje i klasifikacija naziva“. Jedan od glavnih ciljeva programa ACE koji se održava pod pokroviteljstvom *National Institute of Standards and Technology* (NIST) jest „detekcija i praćenje naziva“ (engl. *Entity Detection nad Tracking*).

2.3. Metrika

Sve mjere uspješnosti koje će biti obrađene u ovom poglavlju ne mjere apsolutnu uspješnost već ocjenjuju koliko se dva označivača podudaraju. Da bismo dobili apsolutnu mjeru uspješnosti potrebno je utvrditi podudarnost sustava koji ocjenjujemo s idealnim rezultatom. Taj idealni rezultat naziva se zlatni standard (engl. *gold standard*). Za potrebe mjerenja uspješnosti PKN zlatnim standardom smatra se korpus teksta koji su označili lingvisti. Pritom treba imati na umu da niti lingvisti nisu apsolutno precizni.

Također treba voditi računa o tipu i količini teksta koji se koristi kao zlatni standard. Način pisanja i vokabular znatno se razlikuju u različitim tipovima teksta te je sasvim uobičajeno da sustavi izrađeni za pojedine tipove tekstova imaju znatno lošije performanse kada se koriste na drugi tipovima tekstova. U PKN uobičajeno je da se kao tip teksta odabiru novinski članci zbog toga što novinarski stil pisanja sadrži veću količinu obavijesti nego npr. beletristika. Budući da je PKN za hrvatski jezik tek u povojima ne postoji jasno definirani zlatni standard.

2.3.1. Kappa-mjera

Kappa-mjera je statistička mjera podudaranja koja uspoređuje dva označivača. Njezina glavna osobina jest što izbacuje faktor slučajnog podudaranja zbog čega se smatra vrlo konzervativnom mjerom. Ova mjera predstavljena je u [6] te se često naziva Cohenovom kappa-mjerom.

Općenita jednadžba kojom se računa kappa mjera je:

$$\kappa = \frac{Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

Gdje je *Pr(a)* relativno slaganje između dva označivača, a *Pr(e)* je vjerojatnost da su se dva označivača slučajno složila. Kappa-mjera se može primijeniti i na više označivača [7].

Razmotrimo sljedeći primjer s dva označivača (primjer preuzet iz [8]).

INTERNI DOKUMENT

		Označivač A: Da	Označivač A: Ne
Označivač B	Da	a	b
	Ne	c	d

Kappa-mjera računa se po sljedećoj formuli:

$$\kappa = \frac{2(a * d - b * c)}{(a + c)(c + d) + (b + d)(a + b)}$$

U crpljenju obavijesti d je gotovo uvijek vrlo velik i nepoznat broj, a poklapanje dva označivača u tom parametru nije bitno. Zbog toga kappa mjera najčešće nije primjenjiva u mjerenju uspješnosti zadataka iz područja crpljenja obavijesti.

2.3.2. Preciznost, odziv i F-mjera

Preciznost, odziv i F-mjera su mjere uspješnosti koje se često koriste u crpljenju obavijesti. Preciznost (engl. *precision*) se izražava kao omjer svih sustavom pronađenih točnih naziva i svih sustavom pronađenih naziva. Njome se izražava točnost sustava, odnosno izražava koliko je netočnih odgovora sustav ponudio.

Odziv (engl. *recall*) se izražava kao omjer svim sustavom pronađenih točnih naziva i svih naziva u tekstu. Njime se mjeri koliko je sustav sveobuhvatan, odnosno potpun u pronalaženju relevantnih informacija.

Budući da je u usporedbi dvaju sustava poželjno imati jedinstvenu mjeru učinka definirana je F-mjera (engl. *F-measure*) koja kombinira preciznost i odziv u harmonično opterećen pokazatelj u svrhu postizanja njihove optimalne ravnoteže.

Da bi izrazili ove mjere formulama koristimo sljedeće vrijednosti:

C – naziv koji je sustav označio (engl. *correct*)
M – naziv koji sustav nije označio (engl. *missing*)
S – riječ koja nije naziv, označena kao naziv (engl. *spurious*)
I – riječ koja je označena kao naziv, ali svrstana u krivu kategoriju (engl. *incorrect*)

INTERNI DOKUMENT

Preciznost se računa kao:

$$P = \frac{C}{C + S + I}$$

Odziv se računa kao:

$$R = \frac{C}{C + M + I}$$

F- mjera se računa kao:

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

Parametar β odražava relativnu važnost preciznosti i odziva. Ukoliko preciznost i odziv imaju jednaku težinsku vrijednost, tada je β = 1, a takva mjera se naziva F_1 mjera:

$$F_1 = \frac{2 * P * R}{P + R}$$

Načelno govoreći, ovako definirane mjere moguće je primijeniti za mjerenje uspješnosti klasifikacije, ali ako ih pokušamo primijeniti na PKN nailazimo na problem. Mjerenje uspješnosti PKN je mjerenje podudarnosti sa zlatnim standardom u sedam kategorija koje obuhvaćaju nazivi. Budući da je moguće da se naziv sastoji od više riječi konkretne implementacije problem PKN rješavaju tako da svaku od sedam kategorija podijele u četiri potkategorije (početak naziva, sredina naziva, kraj naziva i samostalni naziv), problem PKN može se promatrati kao problem klasifikacije u jednu od 29 kategorija (7 kategorija sa 4 potkategorije svaka i ostalo). Podudarnost tih kategorija sa zlatnim standardnom moguće mjeriti ovako definiranim mjerama, no takva usporedba ne daje adekvatne rezultate. Razmotrimo sljedeći primjer:

INTERNI DOKUMENT

Tablica 1 - Primjer izračuna preciznosti, odziva i F_1 mjere

Zlatni standard		Sustavom označeni korpus	
Riječ	kategorija	Riječ	kategorija
Josip	PERSON-BEGIN	Josip	PERSON-UNIQUE
Lončar	PERSON-END	Lončar	OTHER
je	OTHER	je	OTHER
poznati	OTHER	poznati	OTHER
znanstvenik.	OTHER	znanstvenik.	OTHER

$$P = 0, \quad R = 0, \quad F_1 = 0$$

Iz primjera jasno se vidi da ovakav način mjerenja uspješnosti nije adekvatan. Potrebno je proširiti definiciju tako da se i djelomična poklapanja vrednuju. Da bi se to postiglo nije dovoljno promatrati označeni tekst na razini jedne riječi već je potrebno mjeriti uspješnost na razini naziva.

Za proširenje modela uvodimo slijedeću vrijednost:

PAR – naziv je jednim dijelom označen točno (engl. *Partially Correct*)

Te proširujemo definicije:

$$P = \frac{C + (0.5 * PAR)}{C + S + I + PAR} \qquad REC = \frac{C + (0.5 * PAR)}{C + M + I + PAR}$$

Dijelom točno označeni nazivi se prilikom računanja preciznosti i odziva uzimaju u obzir kao pola ispravno prepoznatog naziva.

2.3.3. Sličnost kappa-mjere i mjere F_1

Razmotrimo ponovno primjer na kojem smo demonstrirali računanje kappa mjere, no ovaj puta zamijenimo varijable *a, b c* i *d* vrijednostima koje smo koristili za izračunavanje mjere F_1:

Označivač A

Označivač B	Da	Ne
Da	a	b
Ne	c	d

Zlatni standard

Označivač	naziv	ostalo
naziv	C	S
ostalo	M	

INTERNI DOKUMENT

U ovom primjeru postoji samo jedna kategorija pa je vrijednost *I* uvijek jednaka 0 zbog čega će biti izostavljena iz izračuna. Preciznost i odziv izražavamo kao

$$P = \frac{C}{C+S} \qquad R = \frac{C}{C+M}$$

Uvrštavanjem u izraz za F_1 mjeru dobivamo sljedeću formulu:

$$F_1 = \frac{2 * C}{2 * C + M + S}$$

Kapa mjera računa se prema formuli

$$\kappa = \frac{2(a * d - b * c)}{(a+c)(c+d) + (b+d)(a+b)}$$

Ako razmatramo PKN, možemo pretpostaviti da je *d* znatno veći od *a*, *b* i *c*. Vodeći se tom pretpostavkom, možemo pojednostaviti formulu za kappa-mjeru na sljedeći način:

$$\kappa = \frac{2 * a}{2 * a + b + c}$$

Dobivena formula identična je izvedenoj formuli za F_1 mjeru. Ovime smo pokazali da F_1 mjera i kappa mjera teže prema istoj vrijednosti [8].

2.3.4. Mjerenje uspješnosti na konferenciji MUC-7

Na sedmoj konferenciji MUC jedna od glavnih tema bilo je rješavanje problema PKN. U sklopu konferencije održano je i natjecanje. Za potrebe konferencije razvijen je program koji je automatski ocjenjivao uspješnost pojedinog sustava [9]. Za ocjenjivanje uspješnosti korištene su preciznost, odziv i F_1 mjera koje su računate prema ranije prikazanim formulama.

INTERNI DOKUMENT

Primjer tipičnog izvještaja s MUC-7 [9] je slijedeći:

	POS	ACT	COR	PAR	INC	MIS	SPU	NON	REC	PRE	UND	OVG	SUB	ERR
SUBTASK SCORES														
enamex														
organizatio	443	444	405	0	18	20	21	18	91	91	5	5	4	13
person	373	371	364	0	2	7	5	0	98	98	2	1	1	4
location	110	122	109	0	0	1	13	3	99	89	1	11	0	11
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
timex														
date	111	112	107	0	0	4	5	6	96	96	4	4	0	8
time	0	0	0	0	0	0	0	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
numex														
money	76	76	73	0	0	3	3	0	96	96	4	4	0	8
percent	17	25	17	0	0	0	8	0	100	68	0	32	0	32
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SECT SCORES														
Header	244	256	233	0	9	2	14	8	95	91	1	5	4	10
Body	2016	2044	1906	0	42	68	96	95	95	93	3	5	2	10
OBJ SCORES														
enamex	926	937	898	0	0	28	39	21	97	96	3	4	0	7
timex	111	112	107	0	0	4	5	6	96	96	4	4	0	8
numex	93	101	90	0	0	3	11	0	97	89	3	11	0	13
SLOT SCORES														
enamex														
type	926	937	878	0	20	28	39	21	95	94	3	4	2	9
text	926	937	876	0	22	28	39	21	95	93	3	4	2	9
status	0	0	0	0	0	0	0	38	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
timex														
type	111	112	107	0	0	4	5	6	96	96	4	4	0	8
text	111	112	98	0	9	4	5	11	88	88	4	4	8	16
status	0	0	0	0	0	0	0	6	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
numex														
type	93	101	90	0	0	3	11	0	97	89	3	11	0	13
text	93	101	90	0	0	3	11	0	97	89	3	11	0	13
status	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALL SLOTS	2260	2300	2139	0	51	70	110	103	95	93	3	5	2	10
								P&R		2P&R		P&2R		
F-MEASURES								93.82		93.32		94.31		

INTERNI DOKUMENT

2.4. Primjena

Sustav za PKN može se primijeniti pri:

- Generiranju meta-podataka namijenjenih objavljivanju na Internetu,
- Poboljšavanju tražilica koje pretražuju Internet,
- Sažimanju dokumenata s obzirom na neku temu,
- Automatskom generiranju dokumentacijskih i knjiških indeksa,
- Crpljenju obavijesti,
- Prepoznavanju naziva u domenama molekularne biologije i bioinformatike gdje, traženi nazivi predstavljaju imena gena i genskih produkata [10].

INTERNI DOKUMENT

3. Sustavi za prepoznavanje i klasifikaciju naziva

Sustavi za PKN mogu se podijeliti u dvije osnovne kategorije: sustavi temeljeni na pravilima i sustavi temeljeni na metodama strojnog učenja. Sustavi temeljeni na pravilima koriste se jezičnim činjenicama, dok se sustavi temeljeni na strojnom učenju oslanjaju na neku od metoda strojnog učenja i unaprijed označeni korpus za učenje. Postoji i treći, hibridni pristup koji objedinjuje sustave temeljene na pravilima i strojnom učenju.

3.1. Sustavi temeljeni na pravilima

Ovi sustavi izrađeni su ručnim kodiranjem pravila specifičnim za jezik za koji se izrađuju. Takvi sustavi najčešće se modeliraju regularnim gramatikama, a njihov uspjeh uvelike ovisi o količini uloženog vremena i intuiciji dizajnera.

Regularne gramatike su gramatike koje opisuju regularne jezike, a regularni jezici su jezici koji se mogu prikazati konačnim automatom (engl. *Finite State Automaton, FSA*). Deterministički konačni automat (DKA) definiran je kao uređena petorka:

$$(Q, \Sigma, \delta, q_0, F)$$

gdje su:

$$Q \equiv \text{konačni skup stanja}$$
$$\Sigma \equiv \text{konačni skup znakova}$$
$$\delta \equiv \text{prijelazna funkcija } \delta\ Q \times \Sigma \rightarrow Q$$
$$q_0 \equiv \text{početno stanje}$$
$$F \equiv \text{skup završnih stanja}$$

Automati su posebno pogodna klasa generatora jezika za računarnu primjenu, dovoljno su izražajni za modeliranje pravila sustava. Beskonteksne gramatike su znatno snažniji, ali ujedno i znatno složeniji formalizam za obradu jezika tako da ne postoji opravdanost za njihovo korištenje pri modeliranju PKN temeljenog na pravilima.

Uporaba regularnih gramatika za obradu jezika u slučaju prepoznavanja i klasifikacije naziva primjerenija je kao rješenje iz sljedećih razloga [11]:

1. Automati su iznimno jednostavni mehanizmi s čitljivim i preglednim zapisom pravila;

INTERNI DOKUMENT

2. Brzina obrade regularnim gramatikama je mnogo veća u odnosu na beskonteksne gramatike. Eksperimentalno je dokazano da je sustav temeljen na konačnom automatu iznimno robustan;
3. Postoji velik broj gotovih alata za obradu prirodnog jezika koji koriste regularne izraze.

Osobine sustava zasnovanog na pravilima su [11][12]:

- ne zahtijevaju korpus za učenje, samo tekstove na temelju koji će se izraditi pravila,
- preglednost sustava je veća što čini otklanjanje grešaka lakšim,
- iznimno lako prepoznaju neke klase koje je teško prepoznati metodama strojnog učenja,
- mogu dati iznimno dobre rezultate ako se u izradu uloži dovoljno vremena,
- autori sustava moraju biti računarni lingvisti,
- performanse su jako ovisne o stručnosti autora,
- potrebno je duže vrijeme za razvoj sustava,
- adaptacija na tekstove iz druge domene može biti zahtjevna zbog drugačijih osobina tekstova,
- adaptacija na prepoznavanje tekstova na drugim jezicima je gotovo nemoguća.

Iako mogu dati vrlo dobre rezultate, regularni izrazi ne mogu u potpunosti riješiti problem PKN zbog velikog broja iznimaka koje se pojavljuju kod gotovo svakog pravila [12]. Općenito govoreći, nemoguće je kodirati svaku uočenu iznimku s obzirom na vremensko ograničenje pri izradi sustava, pogotovo one koje postanu vidljive tek u fazi testiranja sustava. Treba imati na umu da različiti tipovi dokumenata imaju svoje posebnosti koje je također potrebno ručno kodirati.

3.2. Sustavi temeljeni na metodama strojnog učenja

Prema općoj definiciji, strojno učenje *je proučavanje i izgradnja računarskih sustava koji automatski poboljšavaju svoje performanse kroz iskustvo* [13]. Takvi sustavi stječu potrebno znanje na temelju uzoraka iz korpusa za učenje. Nakon faze učenja takvi sustavi se mogu primijeniti za označavanje prethodno neviđenih podataka.

Na osnovi korpusa za uvježbavanje, za svaki se dio teksta w_i izračunava vjerojatnost pridruživanja jedne od mogućih klasa naziva c_i. Jednostavno računanje

INTERNI DOKUMENT

vjerojatnosti $p(c_i|w_i)$ neovisno o drugima gotovo uvijek daje loš rezultat jer se na taj način ne uzima u obzir kontekst. Zbog toga se izračunava vrijednost koja ovisi o kontekstu duljine n oko w_i, $p(ci|w_{i+n}, \ldots, w_i, \ldots w_{i-n})$. Osim konteksta većina sustava koristi i druge jezične i nejezične osobine teksta – značajke.

Strojno učenje može se prema osnovnim principima rada podijeliti na nadgledane i nenadgledane metode učenja. Nadgledane metode učenja temelje se na prepoznavanju pravilnosti iz obilježenog korpusa, dok nenadgledane metode koriste neobilježene primjere iz korpusa za rješavanje problema. Kao i kod sustava temeljenih na pravilima, i kod sustava temeljenih na metodama strojnog učenja krajnji rezultat uvelike je ovisan o domeni teksta koji je korišten kao korpus za učenje.

Strojno učenje oslanja se na vektore značajki izgrađenih iz označenih ili neoznačenih kolekcija dokumenata. Skup značajki koji se koristi ovisi o cilju klasifikacije. Poželjno je odabrati što manji broj značajki koje imaju najveću važnost, odnosno nose što više diskriminatornih informacija. Značajke se prema tipovima vrijednosti mogu podijeliti u diskretne i kontinuirane [2]. Posebna vrsta diskretnih značajki su Booleove značajke koje poprimaju jednu od dvije vrijednosti. Značajke se razlikuju i po poziciji u tekstu pa tako možemo definirati značajke koje se pojavljuju u samoj informacijskoj jedinki i one koje se pojavljuju u njezinoj okolini (kontekstnom prozoru). Također se mogu definirati značajke koje reprezentiraju vezu između susjednih informacijskih jedinica.

Najčešće korištene vrste značajki pri rješavanju problema PKN su [12]:

1. Leksičke značajke – svi leksički atributi riječi – pojavljuju li se u riječi velika slova i na kojim pozicijama, sadrži li riječ brojke itd.;
2. Sintaktičke značajke – koriste se podaci dobiveni označavanjem vrsta riječi (engl. *part-of-speech tagging, POS tagging*) u rečenici;
3. Značajke popisa riječi – određuju nalazi li se neka riječ u popisima naziva poput popisa osobnih imena, popisa organizacija itd.;
4. Značajke sekcije – značajke važne za dijelove teksta – npr naslovi koji imaju svoja specifična pravila pisanja;
5. Značajke vanjskih sustava – izlazi iz drugih sustava mogu se koristiti kao značajke sustava temeljenih na strojnom učenju. Hibridni sustavi za PKN često koriste izlaze sustava temeljenih na pravilima kao značajke za sustave temeljene na strojnom učenju.

INTERNI DOKUMENT

Osobine sustava zasnovanih na strojnom učenju su:

- Zahtijevaju veliku količinu tekstova – nadzirane metode zahtijevaju velike količine unaprijed označenih tekstova za što je potrebno uložiti veliku količinu ljudskog rada. Uz to, obilježeni tekstovi sadržavaju određen broj grešaka što otežava učenje
- Autori sustava ne moraju biti lingvisti čime se smanjuje cijena izrade
- Potrebno je kraće vrijeme za izradu sustava u odnosu na sustave temeljene na pravilima
- Adaptacija na prepoznavanje tekstova iz druge domene je lakša nego kod sustava temeljenih na pravilima
- Lakša adaptacija sustava na prepoznavanje tekstova pisanih drugim jezicima uz uvjet da na drugom jeziku postoji korpus za treniranje
- Preglednost sustava je manja što otežava uklanjanje pogrešaka

U nastavku ćemo opisati metode strojnog učenja koje su od većeg značaja za rješavanje problema PKN.

3.2.1. Model maksimalne entropije

Maksimalna entropija (ME) je vrlo fleksibilna metoda statističkog modeliranja koja se oslanja „budućnost“, „povijest“ i „značajke“. „Budućnost“ čini skup svih mogućih ishoda, u slučaju problema PKN radi se od 29 klasa. „Povijest“ čini skup podataka za treniranje na temelju kojih se izračunavaju vjerojatnosti. „Značajke“ čine skup neovisnih binarnih značajki na temelju kojih se izračunavaju vjerojatnosti *p(f|h)* za svaki ishod *f* iz prostora mogućih ishoda *F* i za svaku povijesti *h* iz prostora svih mogućih povijesti *H* [14]. Osnovna ideja maksimalne entropije je izgraditi model koji uzima u obzir sve dostupne podatke, a izbjegava pretpostaviti nešto što nije poznato [28]. Polazišna je pretpostavka da svi dijelovi postave nezavisno pridonose konačnoj vjerojatnosti događaja. Zbog toga je model maksimalne entropije uspješan u situacijama gdje treba kombinirati nekoliko višeznačnih izvora informacija što je vrlo čest slučaj pri PKN. Prednost metode maksimalne entropije je što omogućava korisniku fokusiranje na pronalaženje značajki koje karakteriziraju problem bez znanja o tome kako se izračunavaju težine [12].

Primjer značajke:

$$g(h,f) = \begin{cases} 1 \text{ ako } pocetnoVelikoSlovo(h) = true \text{ i } f = locationBegin \\ 0 \text{ inače} \end{cases}$$

INTERNI DOKUMENT

S obzirom na dani skup značajki i korpus za učenje proces određivanja maksimalne entropije proizvodi model koji svakoj značajki f_i pridjeljuje težinu α_i. Ti parametri omogućuju izračun uvjetne vjerojatnosti na slijedeći način [29]:

$$P(f|h) = \frac{\prod_i \alpha_i^{g_i(h,f)}}{Z_\alpha(h)}$$

Gdje je $Z_\alpha(h)$ produkt težina svih mogućih značajki:

$$Z_\alpha(h) = \sum_f \prod_i \alpha_i^{g_i(h,f)}$$

Težine α_i određuju se postupkom generaliziranog iterativnog skaliranja (engl. *Generalized Iterative Scaling, GIS*). Ovaj postupak određivanja parametara uvijek konvergira u rješenje [28].

Razmotrimo primjer na korpusu za treniranje *C*:

$$Q \equiv \text{skup prošlosti } h \text{ koje najavljuju prošlost } h+1$$

$$y \equiv \text{neka budućnost iz } F$$

$$J = \frac{|\{(h,f) \epsilon\, C : h \epsilon\, Q \ \wedge f = y\}|}{\{(h,f) \epsilon\, C : h \epsilon\, Q\}}$$

Nije moguće dodijeliti $P(y|h) = J$ preko cijelog skupa *Q* jer vjerojatnost za *y* ovisi o drugim karakteristikama *h*. Model maksimalne entropije rješava ovaj problem tako da zahtjeva da *J* bude prosječna vrijednost $P(y|h)$ u cijelom korpusu za treniranje. GIS algoritam određuje težine α_i tako da zadovolji taj uvjet, odnosno konvergira prema ispunjenju tog uvjeta.

3.2.2. Skriveni Markovljevi modeli

Osnovna je zamisao skrivenih Markovljevnih modela napraviti odvojene dvopojavničke (engl. *bigrams*) statističke modele za svaku pojedinu kategoriju naziva. Osim modela za pojedine kategorije naziva, izgrađuje se i model za nizove pojavnica koji nisu nazivi. Predviđanje o pripadnosti pojedine pojavnice određenoj kategoriji naziva temelji se na prethodnoj pojavnici i prethodnoj kategoriji naziva. Pri izračunu uvjetnih vjerojatnosti kombiniraju se modeli pojedinih kategorija uključujući vjerojatnosti za prelazak iz jednog modela (jedne kategorije naziva) u drugi. Nakon toga se Viterbijevom pretragom [15] pronalazi put najveće vjerojatnosti koji proizvodi ispravnu sekvencu oznaka.

INTERNI DOKUMENT

3.2.3. Stablo odlučivanja

Osnovna ideja je izgraditi stablo koje u svakom trenutku postavlja pitanje W koje reducira neizvjesnost o prostoru budućnosti F u najvećoj mogućoj mjeri. Neizvjesnost se mjeri uvjetnom entropijom *H(F|W)*. Sa računarnog stajališta najpoželjnije je postavljati samo binarna (da-ne) pitanja pa se s ciljem minimiziranja *H(F|W)* traže ona pitanja koja će prostor značajki ugrubo podijeliti na dva jednaka dijela. Izgradnja stabla odabirom pitanja koja vode najvećoj redukciji uvjetne entropije dobro je poznata tehnika. Kritični je zahtjev izgradnja stabla s dovoljno bogatom prošlošću koja omogućava ispitivanje serije informativnih pitanja koja bi reducirala neizvjesnost o polju značajki.

Tri osnovna elementa pri izgradnji stabla odlučivanja jesu:

- Budućnost – čini skup svih mogućih rezultata, u slučaju PKN prema MUC-7 konferenciji radi se od 29 mogućih oznaka;
- Prošlost – informacije dostupne modelu dobivene analizom *n* susjednih pojavnica [12];
- Pitanja – cilj algoritma stabla odlučivanja jest pronaći najbolji niz pitanja koja se postavljaju o prošlosti kako bi se predvidjela budućnost. Hoće li biti postavljeno *m*-to pitanje ovisi o odgovoru na prethodnih *m-1* pitanja

Jednom izgrađeno stablo odluke iznimno se lako koristi: nakon što su postavljena pitanja odgovorena i obavi se obilazak od korijena to listova uzima se distribucija vjerojatnost koja je pohranjena u listu čvora.

3.2.4. Samonadopunjavajući pristup

Samonadopunjavanje (engl. *bootstrapping*) [16] je princip koji se odnosi na samoodržavajuću tehnologiju koja se oslanja na vlastite metode i resurse. To je tehnologija koja započinje s inicijalno malim skupom primjera te postepeno raste u veći i značajniji sustav ili skup podataka. U slučaju samonadopunjavajučeg sustava za PKN radi se o iterativnom procesu iskorištavanja malog skupa inicijalnih naziva u svrhu dobivanja novih [17].

Cilj pristupa jest iskoristiti minimalnu količinu nadziranih primjera kako bi se steklo znanje iz mnoštva neobilježenih primjera. Pristup samonadopunjavanja započinje tako da se srazom popisa imena i teksta prikupljaju informacije o kontekstnoj okolini naziva [11]. Analizom konteksta naziva sustav izvodi pravila u kojima se određena

INTERNI DOKUMENT

kategorija naziva pojavljuje. Pravila se ponovno primjenjuju na korpus, a rezultat je povećan popis imena koji se ponovno koristi za sraz s tekstom. Taj se postupak ponavlja dok se ne zadovoljni neki od kriterija za zaustavljanje.

3.2.5. Meta-učenje

Ovaj pristup zasniva se na ideji kombinacije više klasifikatora ili posebnom učenju jednoga na način da se sam algoritam višestruko primjenjuje na različite podskupove korpusa za učenje [18]. Glavna prednost ovog pristupa jest mogućnost da se slabi klasifikator pretvori u jaki. Glavne tri metode meta-učenja su:

1. Samonadopunjavajuće gomilanje (engl. *bagging, bootstrap aggregation*) – korpus za treniranje se podijeli nekoliko puta koristeći samonadopunjavanje tako da se slabi klasifikator trenira na svakom od dijelova. Za konačnu se klasifikaciju koriste težinske kombinacije različitih predikcija. Ova metoda najbolje radi s nestabilnim klasifikatorima, no nije popularna u rješavanju problema PKN;
2. Slaganje (engl. *stacking)* – kombiniraju se klasifikatori slaganjem jednog na drugi ili kombiniraju korištenjem težinskih funkcija tako da se minimizira prosječna među-validacijska greška [19];
3. Pojačavanje (engl. *boosting*) – slabi klasifikator uči na podacima za učenje kroz nekoliko samonadopunjavajučih rundi te postavlja težine na primjere za učenje. Oni primjeri koji se teže uče dobivaju veće težine, dok lakši primjeri dobivaju manje. Poanta je da se klasifikator usredotoči na primjere koje je teško klasificirati dok se jednostavni primjeri rješavaju ulaganjem manje „truda“. Najpoznatiji algoritam koji pripada u ovu skupinu je algoritam AdaBoost [20].

INTERNI DOKUMENT

4. Primjeri sustava za prepoznavanje naziva

U ovom poglavlju predstaviti ćemo dva sustava za PKN koji su usko povezani s temom rada. Prvo ćemo predstaviti sustav za PKN u hrvatskom jeziku razvijen na Filozofskom fakultetu u Zagrebu pod imenom OZANA (OZnAčivač NAziva). Drugi zanimljiv sustav koji ćemo razmotriti jest *Maximum Entropy Named Entity*, sustav temeljen na modelu maksimalne entropije za engleski jezik.

4.1. Označivač naziva

OZANA [11] je sustav za strojno prepoznavanje i klasifikaciju naziva za hrvatski jezik zasnovan na pravilima. Sustav je izrađen u sklopu doktorske disertacije Bože Bekavca na Filozofskom fakultetu u Zagrebu. Za izradu sustava korišteno je razvojno okruženje *Intex/Unitex* koje je odabrano zbog niza prednosti nad drugim alatima poput grafičkog sučelja za izradu pravila putem grafova, jednostavnosti, pouzdanosti i brze obrade.

Sustav je temeljen na lokalnim gramatikama (engl. *local grammars*). One su konačni preobličivaći (engl. *transducers*) koji opisuju ispravne nizove u tekstu i za njih odabiru odgovarajuće oznake. To je ujedno i osnovna razlika spram konačnih automata – konačni preobličivaći imaju skup izlaznih znakova i funkciju prijelaza koja omogućava ispisivanje izlaznih znakova, čime je omogućeno preoblikovanje teksta umetanjem oznaka. Uporabom lokalnih gramatika cilj je definirati i primijeniti leksičke uvjete u lokalnom okruženju koje sadrži niz od nekoliko riječi. Sustav se može podijeliti u dvije cjeline: dio za predobradu i dio za prepoznavanje i klasifikaciju naziva.

Predobrada za cilj ima adekvatno prirediti ulazni neobilježeni tekst te sljedećem modulu predati obilježeni tekst. Postupak predobrade odvija se u tri koraka:

1. Opojavničenje – rastavljanje
2. Segmentacija na rečenice – izrađeni modul za segmentaciju implementira 7 pravila dovoljnih za točnost veću od 99%
3. Leksička obrada
    a. Leksički resursi
        i. Opći leksikon, Hrvatski morfološki leksikon [21] s 2.126.086 oblika riječi generiranih iz oko 33.500 lema

INTERNI DOKUMENT

ii. Automatski prepoznavanje brojeva – potencijalne inačice brojeva u padežnim oblicima
iii. Automatsko prepoznavanje pridjeva u genitivu – važno jer su često sastavni dio imena organizacija

b. Popisi imena
i. Vlastita imena osoba – Leksička flektivna baza hrvatskih imena i prezimena [22], prikupljena iz javnih izvora te visoko učestala strana vlastita imena osoba prikupljena iz kraćih izdvojenih popisa s padežnim oblicima
ii. Lokacija – prikupljene iz javnih izvora podataka putem Interneta s padežnim oblicima

Prepoznavanje i klasifikacija naziva jest modul koji koristi lokalne gramatike koje se izvode nad obilježenim tekstovima. Gramatike se temelje na strategijama unutarnjih i vanjskih dokaza uz filtriranje lažnih kandidata, a primjenjuju se kaskadno izvođenjem konačnih preobličivača određenim redoslijedom. Modul se sastoji od sljedećih dijelova [23]:

1. Filtriranje lažnih kandidata – izbacivanje pojavnica koje po osobinama upućuju na pripadnost nazivima, ali to nisu
2. I. faza primjene pravila – primjena pravila za prepoznavanje postotaka, novčanih izraza, datuma i vremenskih izraza, organizacija te osoba i lokacija, čvrsta pravila najveće sigurnosti;
3. Filtriranje leksikona – filtriranje visokofrekventnih višeznačnih pojavnica koje otežavaju primjenu olabavljenih pravila;
4. II. faza primjene pravila – primjena olabavljenih pravila koja nastoje prepoznati do tada neprepoznate lokacije i osobe u dovoljno sigurnom kontekstu.

Rezultati obrade su tekstovi u SGML-obliku s obilježenim nazivima prema specifikacijama s konferencije MUC-7. Kao korpus za uvježbavanje korišteni su tekstovi Večernjeg lista (Hrvatski nacionalni korpus [24]) od 1999. do 2003. godine opsega 45.563.824 pojavnica, i tekstovi Vjesnika od 2001. do 2003. godine, opsega 15.193.749 pojavnica. Za testiranje su korišteni tekstovi iz istih listova iz 2004. godine opsega 9.932.498 pojavnica. Za vrednovanje su korišteni tekstovi istih listova iz siječnja 2005.

INTERNI DOKUMENT

Prosječna F-mjera sustava iznosi 92%. Zbog neravnomjerne zastupljenosti kategorija naziva u tekstovima, realnija slika učinkovitosti sustava dobiva se mjerenjem svih naziva koje bi trenutna inačica izrađenim pravilima trebala prepoznati u tekstu. Tako izračunata F-mjera iznosi 90%. Na neinformativnim tekstovima učinkovitost sustava pada.

4.2. Maximum Entropy Named Entity

Maximum Entropy Named Entity (MENE) [25] je PKN sustav Sveučilišta u New Yorku. Sustav se sastoji od C++ i Perl modula koji služe kao omotači oko javno dostupne biblioteke *Maximum Entropy Modeling Toolkit* (MEMN). Sistav MENE natjecao se na konferenciji MUC-7 u hibridnom modelu sa sustavom *Proteus* temeljenom na pravilima.

Zahvaljujući činjenici da model maksimalne entropije može iskoristiti bilo kakvu binarnu informaciju kao značajku MENE se odlikuje velikom fleksibilnosti. MENE koristi nekoliko kategorija značajki:

1. Binarne značajke – iako se sve značajke koje MENE koristi mogu nazvati binarnima, u ovu skupinu ubrajaju se one značajke koje ili vrijede ili ne vrijede za neku pojavnicu. MENE koristi sljedeće binarne značajke:
    a. *2-digit number* (broj od 2 znamenke)
    b. *4-digit number* (broj od 4 znamenke)
    c. *Only digits* (samo znamenke)
    d. *Mixed letters and digits* (miješane znamenke i slova)
    e. *Number with comma* (broj sa zarezom)
    f. *A valid number* („ispravan“ broj)
    g. *All-caps* (sva slova velika)
    h. *Initial Cap* (početno slovo veliko)
    i. *Uncapitalized word* (sva slova mala)
    j. *Internal Capitalization* (unutar riječi postoji veliko slovo)
2. Leksičke značajke – pretražuje se okolina promatrane pojavnice u potrazi za riječima u leksičkom vokabularu. Leksički vokabular stvara se prilikom treniranja umetanjem riječi koje su se pojavile ispred svake budućnosti barem 3 puta u korpusu za treniranje
3. Sekcijske značajke – ove značajke prenose informaciju o dijelu teksta u kojem se pojavnica nalazi, npr. preambula ili tekst

INTERNI DOKUMENT

4. Značajke rječnika – svaka pojavnica se uspoređuje s nekoliko različitih unaprijed ručno izrađenih rječnika. Autori sustava tvrde da su ove značajke ključni element njihovog sustava. MENE koristi sljedeće rječnike:
    a. Osobna imena - 1245 unosa
    b. Imena tvrtki - 10.300 unosa
    c. Imena visokoobrazovnih institucija – 1225 imena
    d. Sufiksi imena tvrtki – 244 unosa
    e. Datumi i vremena – 51 unos
    f. Imena saveznih država – 50 unosa
    g. Svjetske regije – 14 unosa
5. Izlazi drugih sustava – MENE sustav nije samostalno sudjelovao na MUC-7 konferencije već je nastupio kao dio hibridnog sustava. Drugi dio hibridnog sustava činio je *Proteus*, sustav temeljen na pravilima. Izlazi sustava *Proteus* korišteni su kao značajke za sustav MENE.

Jedna od vrlo bitnih karakteristika sustava MENE jest automatski odabir značajki. Prilikom treniranja sustav odbacuje značajke koje manje pridonose ukupnom rezultatu prema nekoliko kriterija. Iako teorijska razmatranja sustava maksimalne entropije kažu da značajke koje ne nose puno informacija ne smanjuju učinkovitost, zbog ograničenja računarnih sustava u praksi dolazi do problema ako se odjednom aktivira više od 40 značajki. Druga prednost manjeg broja značajki jest kraće vrijeme treniranja što u konačnici znači da je moguće izvršiti više testova.

F_1-mjera sustava MENE učenog na službenom korpusu za učenje konferencije MUC-7, a njerena na službenom korpusu za testiranje iznosi 84.22% (preciznost 91% i odziv 78%). Utvrđeno je i da je samome sustavu potrebno barem 20 članaka za postizanje F-mjere od 80.97%.

INTERNI DOKUMENT

5. Model maksimalne entropije za prepoznavanje i klasifikaciju naziva u hrvatskom jeziku

O ovom poglavlju opisati ćemo arhitekturu i karakteristike sustava za PKN razvijenog u okviru ovog rada. Budući da je u trenutku odabira metode rješavanja problema PKN bio dostupan označeni korpus za treniranje bilo je moguće razmatrati i nadzirane metode. Model maksimalne entropije odabran je između modela koji do sada nisu izrađenih za hrvatski jezik (ili nisu bili u fazi izrade) zbog dobrih rezultata u primjeni na drugim jezicima.

Sustav je izrađen u programskom jeziku C#, a prilikom izrade korištene su biblioteke SharpEntropy [27] i *Text Mining Tools* (TMT) [26]. Biblioteka SharpEntropy je C# verzija Java biblioteke, a implementira sve potrebne metode za korištenje modela maksimalne entropije. U ovom radu nije se ulazilo u način na koji se implementira model maksimalne entropije već je biblioteka SharpEntropy korištena kao crna kutija. Biblioteka TMT korištena za opojavničavanje, rastavljanje na rečenice i ekstrakciju morfoloških značajki pojavnica.

Sustav uči iz korpusa označenog prema specifikaciji konferencije MUC-7 (SGML). Jedina iznimka je ta što korpus za učenje mora unaprijed biti segmentiran na rečenice jer sustav za segmentiranje na rečenice biblioteke TMT ne podržava XML, odnosno SGML. Nakon faze učenja sustav je sposoban označavati neoznačeni tekst prema sepcifikaciji konferencije MUC-7.

Model maksimalne entropije oslanja se na niz značajki koje se određuju za svaku pojavnicu u tekstu. Značajke koje su implementirane mogu se podijeliti u sljedeće grupe:

1. binarne značajke,
2. leksičke značajke,
3. morfološke značajke,
4. rječničke značajke.

INTERNI DOKUMENT

5.1. Modul za označavanje binarnih, morfoloških i riječničkih značajki

U fazi učenja, ovaj modul čita rečenicu po rečenicu korpusa za učenje koju zatim rastavlja na pojavnice. Prilikom rastavljanja na pojavnice koriste se gotova rješenja iz biblioteke TMT. Unutar biblioteke podržano je nekoliko načina opojavničavanja, od najjednostavnijeg koji samo rastavlja dani tekst na pojavnice do kompleksnijih koji pomoću regularnih izraza prepoznaju datume, URL-ove i slično. Za potrebe sustava korištena je najjednostavnija varijanta kako bi se modelu maksimalne entropije prepustilo prepoznavanje datuma. Nakon rastavljanja na pojavnice svaka se pojavnica analizira i određuju se aktivirane značajke iz skupa binarnih, morfoloških i rječničkih značajki. Uz značajke, modul za svaku pojavnicu bilježi i oznaku prema MUC-7 koja se nalazi u korpusu za učenje.

Kada se modul koristi u fazi označavanja teksta, prije opojavničavanja se provodi segmentacija na rečenice pomoću alata iz biblioteke TMT. Segmentacija je razmjerno jednostavnim regularnim izrazima. Kao početak rečenice uzima se riječ pisana velikim slovom ispred kojeg se nalazi točka. U slučaju da se točka nalazi unutar navodnika ona se ignorira.

5.2. Modul za određivanje kontekstnog prozora i leksičkih značajki

Modul za određivanje kontekstnog prozora i leksičkih značajki od modula za označavanje binarnih, morfoloških i rječničkih značajki uzima jednu po jednu opojavničenu rečenicu i njezine do tada pronađene značajke, dodaje leksičke značajke te objedinjuje značajke iz kontekstnog prozora u oblik pogodan za implementaciju modela maksimalne entropije biblioteke SharpEntropy koja se koristi u sustavu.

Nakon pronalaska leksičkih značajki koje je detaljnije opisano kasnije, modul uzima značajke od *n* pojavnica prije i *m* pojavnica nakon promatrane pojavnice i dodaje ih promatranoj pojavnici uz indeks pomaka. Razmotrimo slijedeći primjer. Rečenica „Danas je lijep i sunčan dan.“ nakon opojavničenja i podešavanja binarnih značajki izgledati će na način prikazan sljedećom tablicom:

INTERNI DOKUMENT

Tablica 2 - Primjer određivanja značajki uz širinu kontekstnog prozora od jedne riječi

Pojavnica	**Značajke**
Danas	InitCap
je	lowerCase
lijep	lowerCase
i	lowerCase
sunčan	lowerCase
dan	lowerCase
.	isPeriod

Ako je kontekstni prozor obuhvaća 2 riječi prije i 2 riječi poslije promatrane pojavnice, modul će proizvesti skup pojavnica i značajki prikazan tablicom 3.

Tablica 3 - Primjer određivanja značajki uz širinu kontekstnog prozora od pet riječi

Pojavnica	**Značajke**
Danas	InitCap R+1-lowerCase R+2-lowerCase
je	R-1-InitCap lowerCase R+1-lowerCase R+2-lowerCase
lijep	R-2-InitCap R-1-lowerCase lowerCase R+1-lowerCase R+2-lowerCase
i	R-2-lowerCase R-1-lowerCase lowerCase R+1-lowerCase R+2-lowerCase
sunčan	R-2-lowerCase R-1-lowerCase lowerCase R+1-lowerCase R+2-isPeriod
dan	R-2-lowerCase R-1-lowerCase lowerCase R+1-isPeriod
.	R-2-lowerCase R-1-lowerCase isPeriod

INTERNI DOKUMENT

5.3. Bilioteka SharpEntropy

Biblioteka SharpEntropy implementira model maksimalne entropije. Radi se o C# verziji biblioteke MaxEnt toolkit koja je inicijalno razvijena za programski jezik Java. Biblioteka podržava metodu treniranja *Generalized Iterative Scaling* (GIS), a u ovom se radu koristi kao crna kutija. U fazi treniranja biblioteci se prosljeđuje skup povijesti (značajke) i budućnost koja je jedno od traženih stanja koja specificira SGML. U fazi označavanja biblioteci se daje skup prošlosti, a biblioteka izračunava uvjetne vjerojatnosti za sve budućnosti.

S programerskog stajališta, korištenje ove biblioteke je vrlo jednostavno. Potrebno je implementirati nekoliko jednostavnih sučelja koja prilagođavaju podatke formatu koji biblioteka zna interpretirati.

5.4. Viterbijeva pretraga

Modul koji obavlja Viterbijevu pretragu u fazi treniranja izrađuje statistiku prijelaza temeljenu na korpusu za učenje. U fazi označavanja se uvjetne vjerojatnosti za pojedine rečenice predaju modulu koji obavlja Viterbijevu pretragu. Na temelju podataka prikupljenih u fazi učenja, algoritam pronalazi najvjerojatniji skup prijelaza između SGML stanja.

Ovo je vrlo bitan dio sustava jer osigurava da izlaz bude pravilno označeni SGML. Označavanje samo na temelju uvjetnih vjerojatnosti moglo bi proizvesti izlazni niz u kojem bi nakon stanja *locationBegin* slijedilo stanje *dateEnd*, što nije ispravno označen tekst.

5.5. Binarne značajke

Sustav podržava velik broj binarnih značajki koje se jednim dijelom preklapaju, no kako je moguće odabrati koje će se značajke koristiti, to ne predstavlja problem u radu. Ove značajke implementirane su regularnim izrazima, a uglavnom analiziraju pojedine znakove, ili odnose među znakovima unutar jedne pojavnice. Podržane binarne značajke su:

INTERNI DOKUMENT

- allCaps – pojavnica pisana samo velikim slovima;
- capPeriod – veliko slovo iza kojeg slijedi točka;
- capsPeriods – ponavljanje niza veliko slovo – točka;
- capOtherPeriod – riječ pisana velikim slovom koja na kraju ima točku;
- initCap – veliko početno slovo pojavnice;
- lowerCase – sva mala slova u pojavnici;
- innerCap –unutar pojavnice se pojavljuje veliko slovo;
- hasPeriod – pojavnica sadrži zarez;
- hasComma – pojavnica sadrži točku;
- shortWord – pojavnica je kraća od pet znakova;
- oneDigitNumber – pojavnica je broj od jedne znamenke;
- twoDigitNumber – pojavnica je broj od dvije znamenke;
- fourDigitNumber – pojavnica je broj od četiri znamenke;
- onlyDigits – pojavnica se sastoji samo od brojeva;
- hasDigits – pojavnica sadrži brojeve;
- containsDigitAndAlpha – pojavnica sadrži brojeve i znakove;
- containsDigitAndPeriod – pojavnica sadrži brojeve i točku;
- containsDigitAndPeriods – pojavnica sadrži brojeve i točke;
- containsDigitAndDash – pojavnica sadrži brojeve i crticu;
- containsDigitAndSlash – pojavnica sadrži brojeve i kosu crtu;
- containsDigitAndComa – pojavnica sadrži brojeve i zarez;
- containsDigitAndComas – pojavnica sadrži brojeve i zareze;
- validNumber – pojavnica se može parsati u broj;
- containsDigitComaAndPeriod – pojavnica sadrži brojeve, točke i zareze.

5.6. Leksičke značajke

Leksičke značajke opisuju okolinu u kojoj se nalazi promatrana pojavnica. Prilikom treniranja, za svaku moguću budućnost (stanje prema konferenciji MUC-7) bilježe se prethodne i slijedeće pojavnice. Od sakupljenih podataka, odbacuju se sve pojavnice koje se pojave manje od *n* puta, gdje je *n* parametar koji se zadaje prilikom pokretanja programa za treniranje, a ostale pojavnice se, zajedno s frekvencijom pojavljivanja, bilježe u popis.

INTERNI DOKUMENT

Prilikom određivanja značajki, za svaku susjednu (prethodnu i slijedeću) pojavnicu pretražuje se generirani popis te se za svaku od mogućih budućnosti izračunava faktor koji govori koliko se često prethodna pojavnica pojavljuje prije pojedine budućnosti. Kako je dobivena vrijednost kontinuirana, njezina vrijednost se diskretizira dijeljenjem u nekoliko raspona vrijednosti. Rasponi se definiraju tako da se uzima početni prag (zadaje se kao parametar) koji se zatim dijeli s nekim faktorom (također se zadaje kao parametar) kako bi se dobili preostali razredi. Razred u koji ulazi izračunati faktor postavlja se kao značajka.

Uzmimo za primjer rečenicu „Kantina radi do podneva“. Promotrimo pojavnicu „podneva“. Recimo da se u korpusu za učenje pojavnica „do“ pojavljuje ispred stanja timeUnique u 15% slučajeva. Ako je početni prag postavljen na 20%, a faktor kojim se dijeli iznosi 2, pojavnica „do“ uči će u drugi razred stoga će pojavnici „podneva“ biti podešena značajka *lexClass2BeforeTimeUnique*. Spomenuta značajka može se protumačiti kao „ispred ove pojavnice nalazi se pojavnica koja se u 10 do 20 posto slučajeva nalazi ispred pojavnica koje imaju stanje *timeUnique*“.

5.7. Morfološke značajke

Morfološke značajke oslanjaju se na morfološku analizu pojavnica implementiranu u biblioteci TMT. Zbog prirode hrvatskog jezika nije jednoznačno moguće odrediti o kojem tipu i obliku riječi se radi promatrajući samo riječ izvan konteksta. Jednoznačna analiza bila bi moguća kada bi se napravila POS-analiza, no alat koji bi proveo takvu analizu za hrvatski jezik nije javno dostupan. Implementirana morfološka analiza se provodi u dva koraka, u prvom koraku se iz oblika riječi pronađenog u tekstu odrede sve moguće leme riječi. Zatim se za oblik riječi u tekstu i svaku od mogućih lema odredi jedan ili više oblika. Morfološka analiza implementira sljedeće značajke:

- tipImenica – pojavnica je imenica
- tipGlagol – pojavnica je glagol
- tipPridjev – pojavnica je pridjev
- padezNominativ – padež pojavnice je nominativ
- padezGenitiv – padež pojavnice je genitiv
- padezDativ – padež pojavnice je dativ
- padezAkuzativ – padež pojavnice je akuzativ
- padezVokativ – padež pojavnice je vokativ

INTERNI DOKUMENT

- padezLokativ – padež pojavnice je lokativ
- PadezInstrumental – padež pojavnice je instrumental
- brojnostJednina – brojnost pojavnice je jednina
- brojnostMnozina – brojnost pojavnice je množina
- licePrvo – pojavnica je u prvom licu
- liceDrugo – pojavnica je u drugom licu
- liceTrece – pojavnica je u trećem licu
- rodMuski – pojavnica je u muškom rodu
- rodZenski – pojavnica je u ženskom rodu
- rodSrednji – pojavnica je u srednjem rodu

5.8. Rječničke značajke

Rječničke značajke oslanjaju se na unaprijed pripremljene rječnike koji se pretražuju za svaku pojavnicu u tekstu. Ako se pojavnica nalazi u rječniku, podešava se određena značajka. Sustav razvijen u sklopu ovog rada sadrži dva rječnika:

- popis poštanskih ureda,
- popis imena i prezimena.

5.8.1. Popis poštanskih ureda

Popis poštanskih ureda preuzet je sa internetskih stranice Hrvatske poste. Za izradu rječnika korištena su imena mjesta koja su opojavničena. Pojavnicama je zatim određena lema riječi pomoću biblioteke TMT te su takvi rezultati zapisani u rječnik. Prilikom određivanja značajki tekstovna datoteka se učitava u kolekciju koja ima izgrađeni ključ za brzu pretragu. Za svaku pojavnicu iz teksta se određuje se njena lema te se pretražuje popis. U slučaju da se ustanovi da se pojavnica nalazi u rječniku, aktivira se značajka.

5.8.2. Popis osobnih imena i prezimena

Popis imena i prezimena izgrađen je na temelju podataka iz telefonskog imenika. Podaci su prvo filtrirani kako bi se iz njega izbacili telefonski brojevi čiji vlasnici nisu privatne osobe. Zatim je popis parsan kako bi se odbacili suvišni podaci (adresa, mjesto, broj telefona). Za svaku pojavnicu iz parsanog popisa određena je (pomoću biblioteke TMT) lema riječi te je zapisana u kolekciju koja osim pojavnice sadrži i broj pojavljivanja

INTERNI DOKUMENT

te pojavnice u imeniku. Budući da osim same pojavnice raspolažemo i brojem pojavljivanja u imeniku, nije adekvatno koristiti samo jednu binarnu značajku, već je potrebno napraviti razdiobu u razrede. Razdioba u razrede napravljena je po istom načelu kao i kod leksičkih značajki.

INTERNI DOKUMENT

6. Provedeni testovi

Za potrebe testiranja performansi sustava korišten je korpus članaka iz Glasa Slavonije označen sustavom OZANA. Za treniranje je korišten + korpus koji se sastoji od 350 članaka, 7.470 rečenica, odnosno 164.154 pojavnica. Za evaluaciju rezultata korišten je korpus od 100 članaka koji se ne preklapaju s korpusom za treniranje. Korpus za evaluaciju sadrži 1699 rečenica te ukupno 482.540 pojavnica.

Tablica 4 - Broj naziva u korpusu za treniranje i korpusu za evaluaciju

	Korpus za treniranje	Korpus za testiranje
PERSON	3144	775
LOCATION	2243	681
ORGANIZATION	620	157
PERCENT	155	43
MONEY	272	86
DATE	1159	212
TIME	0	0

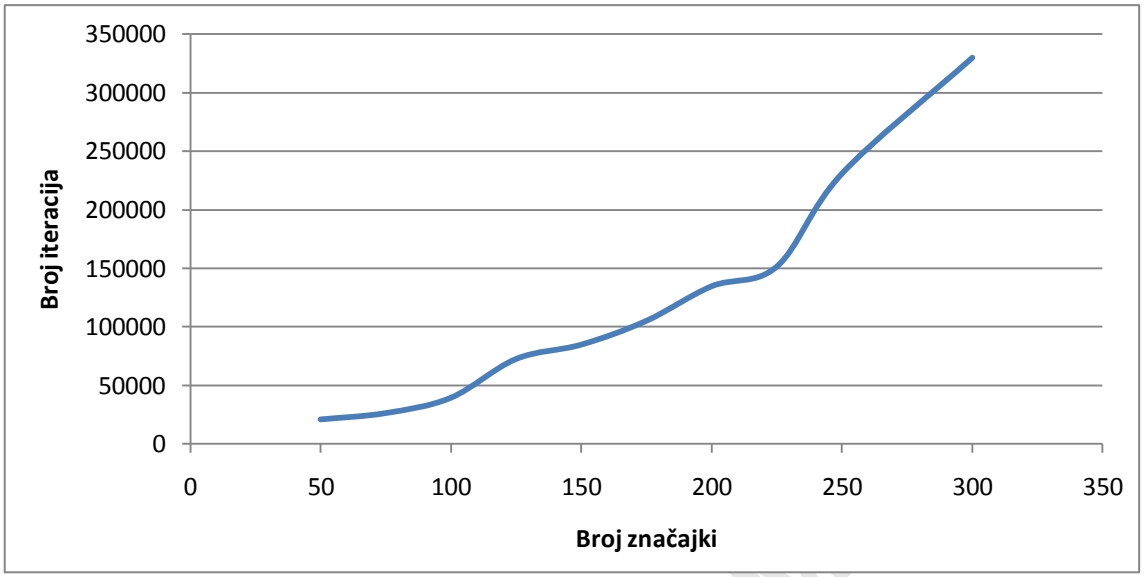
Poželjno bi bilo koristiti korpus ručno označen od strane lingvista. Na prvi pogled jasno je da korišteni korpus prilično odstupa od željenog standarda jer nema označen niti jedan naziv koji označava vrijeme. Kod ostalih naziva F_1 mjera uspješnosti označavanja iznosi manje od 90% [11].

6.1. Složenost treniranja

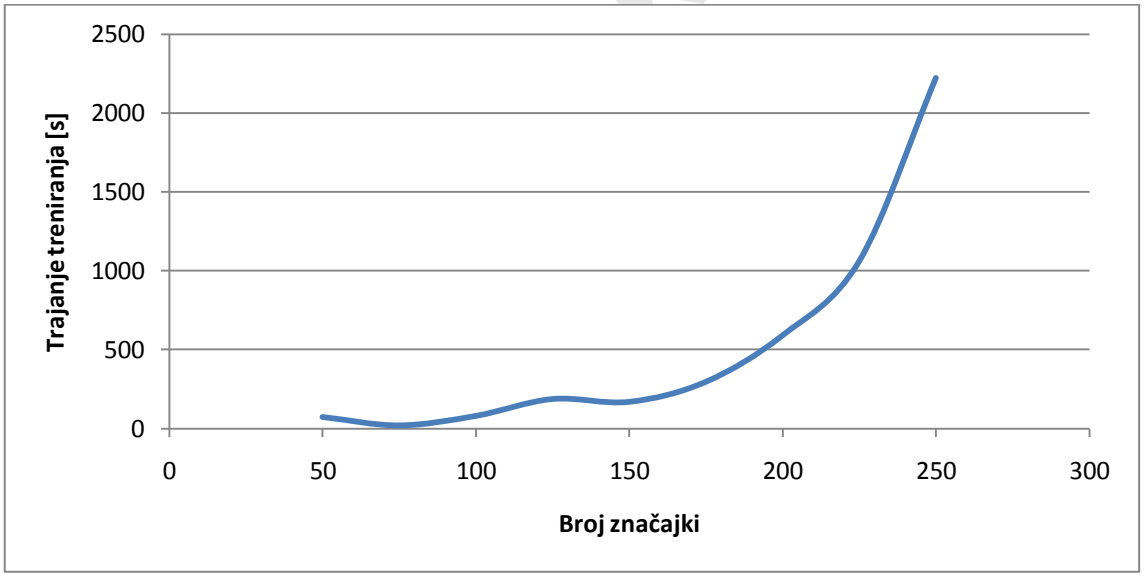
Prilikom treniranja s različitim postavkama (širina konteksta, korištene značajke) potrebno je utvrditi univerzalan uvjet zaustavljanja kako bi se rezultati treniranja s različitim postavkama mogli uspoređivati. Taj uvjet zaustavljanja određen je tako da se prati doprinos izglednosti kroz zadnjih 10 iteracija te se treniranje prekida kada taj doprinos padne ispod određene granice.

Budući da proces treniranja ovisi o korpusu i korištenim značajkama nije moguće matematički odrediti složenost. Slika 1 i Slika 2 prikazuju potreban broj iteracija i potrebno vrijeme za treniranje u ovisnosti o broju korištenih značajki.

INTERNI DOKUMENT



Slika 1 - Ovisnost broja iteracija o broju značajki



Slika 2 - Ovisnost trajanja treniranja o broju značajki

Na temelju podataka iz Slika 1 možemo zaključiti da je porast broja iteracija u odnosu na porast broja značajki gotovo linearan. Iz Slika 2 možemo zaključiti da je vrijeme treniranja raste eksponencijalno sa brojem iteracija. Iz ova dva grafa jasno se vidi da je vrijeme potrebno za jednu iteraciju ovisno o broju značajki koje se koriste.

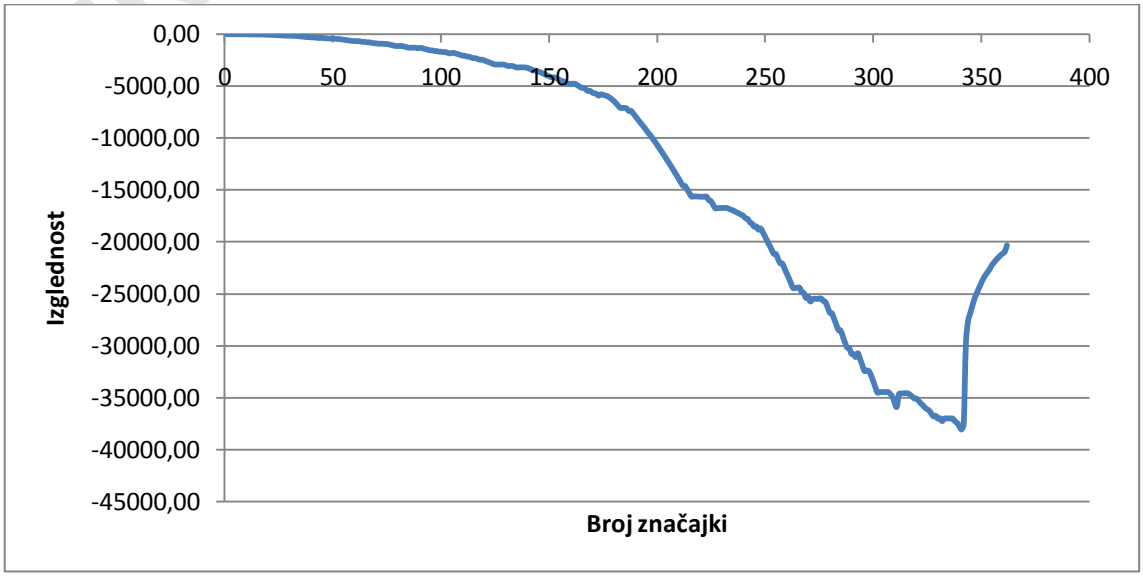
6.2. Odabir značajki

Odabir značajki vrlo je bitan faktor u ostvarivanju dobrih rezultata. Budući da vrijeme potrebno za treniranje raste eksponencijalno s brojem značajki, ključno je odabrati minimalan broj značajki koje daju dobar rezultat. U idealnom slučaju odabrali bismo što više značajki, ali u tom slučaju vrijeme potrebno za treniranje bilo bi predugo.

Optimalan odabir značajki je onaj koji minimizira broj značajki, a maksimizira rezultate. Implementiran je algoritam koji automatski odabire optimalne značajke. U odabir značajki kreće se s praznom listom odabranih značajki, dok se u listi preostalih značajki nalaze sve značajke. Za svaku od značajki u listi preostalih značajki gradi se model sa značajkama iz liste odabranih značajki te se od svih tako izgrađenih značajki odabire onaj koji ima najpovoljnija svojstva. Značajka pridodana listi odabranih značajki koja tvori odabrani model dodaje se u listu odabranih značajki, a miče sa liste preostalih značajki. Ovaj proces se ponavlja te se gradi rang lista značajki. Kao mjera kvalitete koristi izglednost (engl. *likelihood*) koju je vrlo lako moguće izračunati u svakoj iteraciji treniranja. Umjesto izglednosti, kao mjeru moguće je koristiti preciznost, odziv ili F_1 mjeru, no računanje tih mjera traje znatno duže te zbog toga nije implementirano.

Krajnji odabir korištenih značajki prepušten je ručnom odabiru budući da je potrebno utvrditi zlatnu sredinu između broja značajki i dostupnih računalnih resursa za treniranje modela.

Opisani proces proveden je na skupu za treniranje s mogućih 555 značajki.



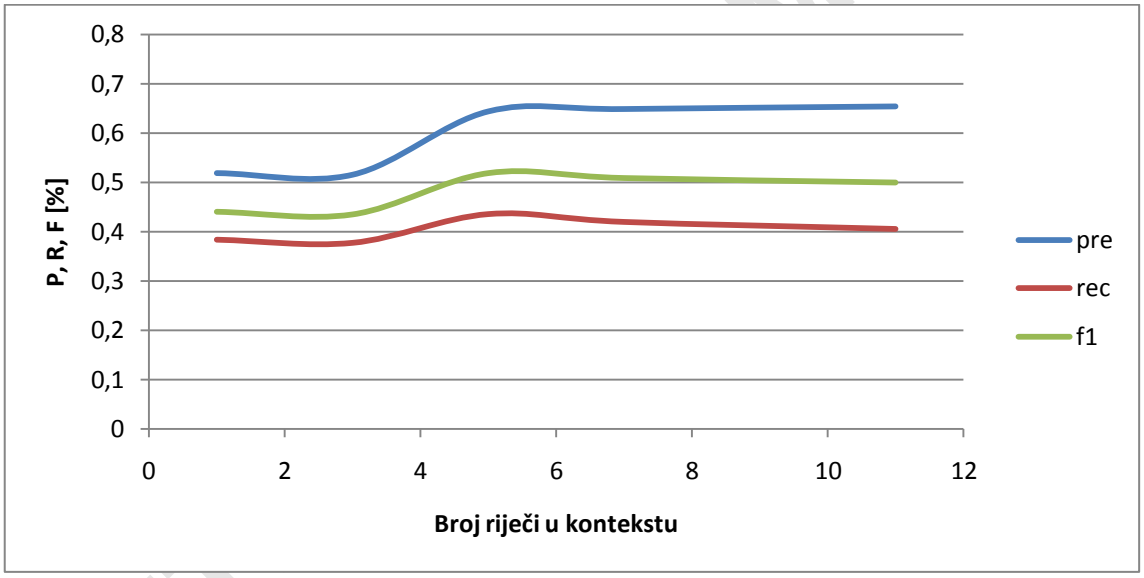
Slika 3 - Ovisnost izglednosti o broju značajki

INTERNI DOKUMENT

Iz grafa je vidljivo da izglednost isprva pada, što je očekivano budući da svaka nova značajka uvodi nova ograničenja u model. No nakon određenog broja značajki izglednost počinje rasti, što nije očekivano budući da kompleksnost modela raste. Upravo se u ovom području nalazi točka s optimalnim brojem značajki.

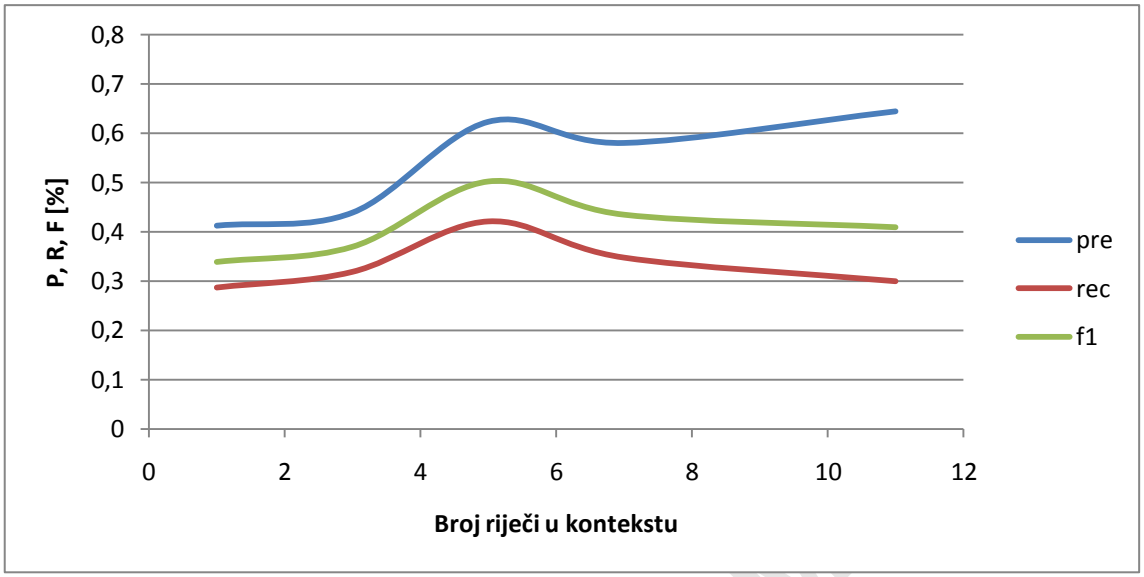
6.3. Utjecaj širine konteksta

Mogući broj značajki linearno raste s brojem riječi u kontekstu koji se razmatra, a samim time i kompleksnost modela. Utjecaj širine konteksta na rezultate vidljiv je iz grafova na slikama 4, 5, 6 i 7:



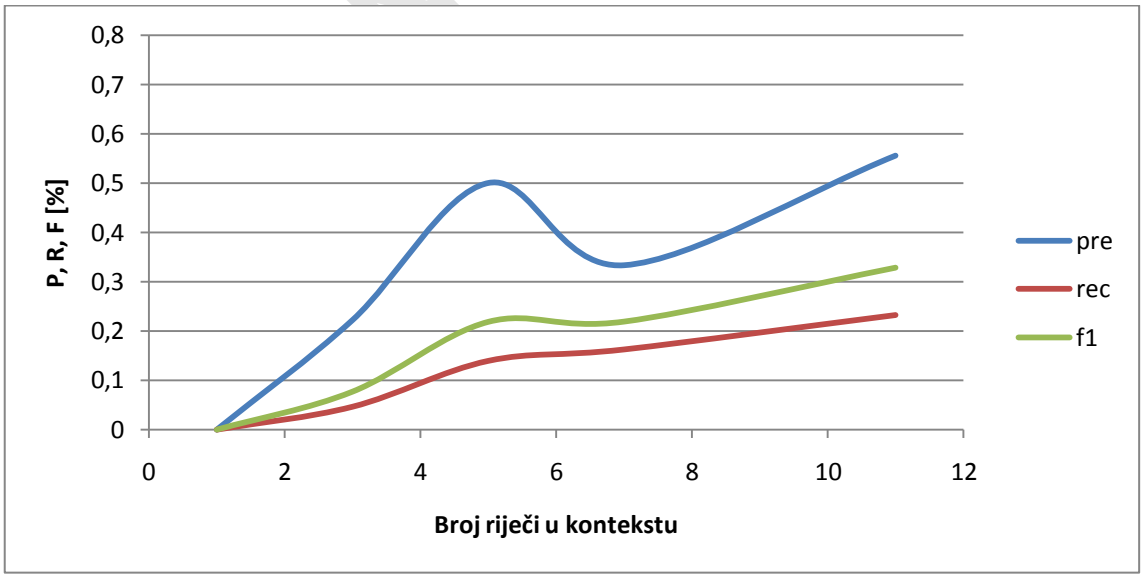
Slika 4 - Ovisnost preciznosti, odziva i F1 mjere cijelog sustava o broju riječi u kontekstu

Iz Slika 4 vidi se da je pet riječi optimalna širina konteksta. Širenjem konteksta preciznost lagano raste, ali odziv i F_1 mjera lagano opadaju.



Slika 5 - Ovisnost preciznosti, odziva i F1 mjere označavanja organizacija o broju riječi u kontekstu

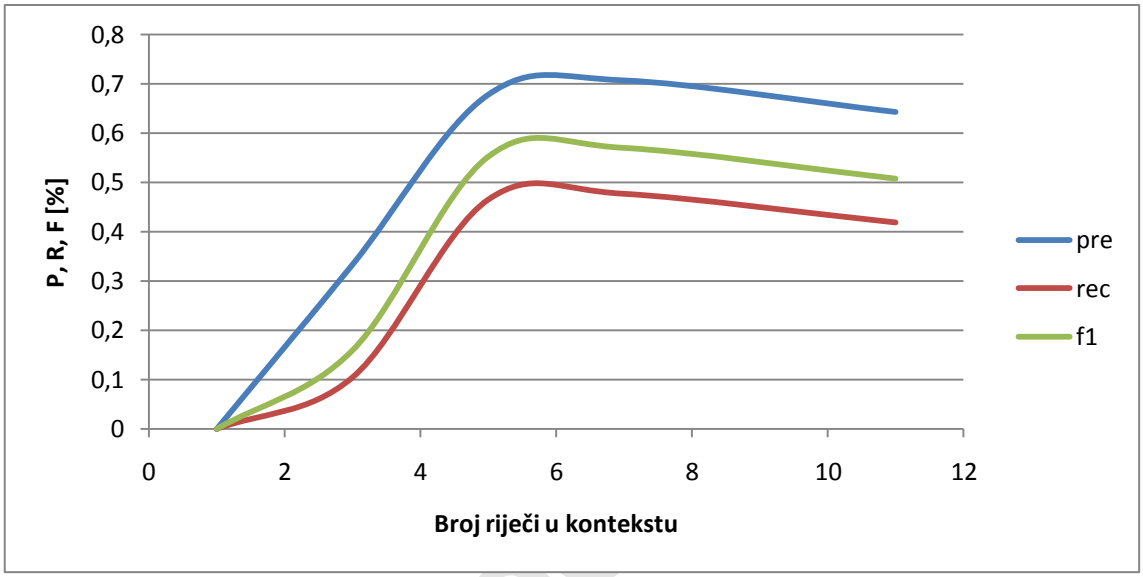
Slika 5 je posebno zanimljiva jer su imena organizacija često sastavljena od više riječi pa je utjecaj širine konteksta ovdje posebno dobro vidljiv. Iako se i ovdje optimalna točka nalazi na širini konteksta od pet riječi, proširenjem konteksta preciznost zamjetno raste.



Slika 6 - Ovisnost preciznosti, odziva i F1 mjere označavanja postotaka o broju riječi u kontekstu

Ovisnost uspješnosti označavanja naziva postotaka (PERCENT) o broju riječi u kontekstu, prikazana slikom 6, je zanimljiva budući da se ponaša drugačije od ostalih naziva. Iako se lokalni maksimum nalazi na širini konteksta od pet riječi, daljnjim širenjem

konteksta postižu se bolji rezultati. Označavanje naziva novčanih valuta (MONEY), prikazana slikom 7, također pokazuje najbolje rezultate pri širini konteksta od pet riječi, no zanimljivo je što daljnjim širenjem konteksta dolazi do opadanja preciznosti što nije slučaj sa ostalim tipovima naziva002E



Slika 7 - Ovisnost preciznosti, odziva i F1 mjere označavanja novčanih valuta o broju riječi u kontekstu

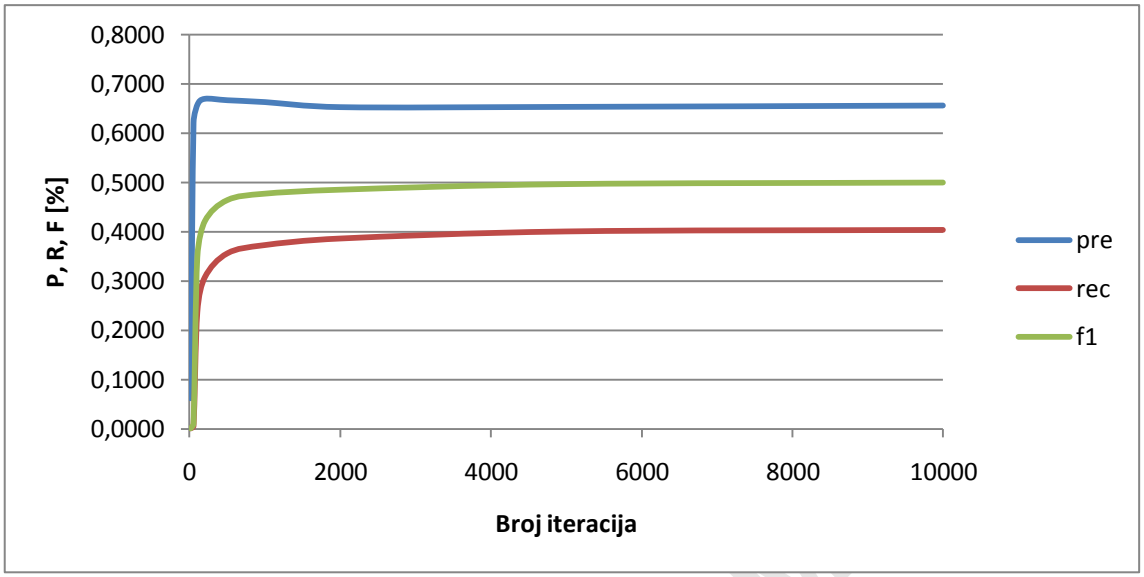
Ponašanje pojedinih tipova naziva ovisno o širini konteksta vrlo je zanimljivo. Iako je jasno vidljivo da je optimalan broj riječi u kontekstu pet, neki tipovi naziva pokazuju bolje, a neki lošije rezultate daljnjim širenjem konteksta. U praksi imamo i vremenska ograničenja, pa je neisplativo koristiti kontekst širi od pet riječi jer je ukupno poboljšanje u rezultatima minimalno, a vrijeme treniranja je znatno duže.

6.4. Utjecaj broja iteracija

U idealnom slučaju broj iteracija treniranja koje želimo napraviti teži u beskonačnost. U praksi poželjno je znati kako se model ponaša ovisno o broju iteracija. To saznanje omogućit će nam da bolje odredimo točku prekida treniranja te tako u istom vremenu provedemo više eksperimenata. S više provedenih eksperimenata moći ćemo bolje utvrditi optimalne parametre te napraviti bolji model.

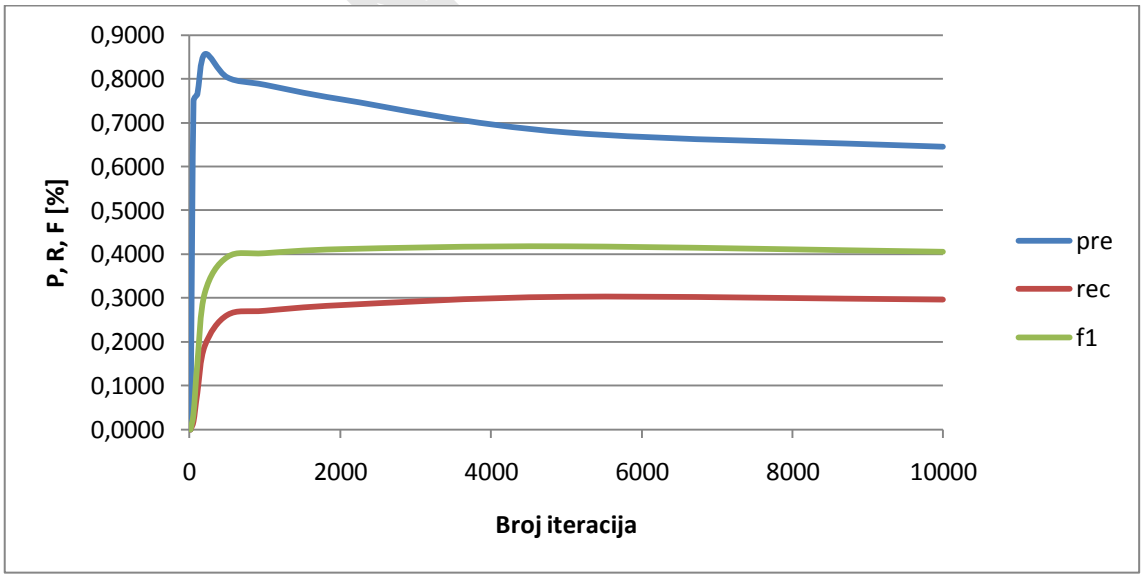
Provedeni su pokusi sa širinom konteksta od pet riječi s korištenjem svih značajki na korpusu za treniranje od 350 članaka. Grafovi na slikama 8, 9 i 10 prikazuju ovisnost preciznosti, odziva i F_1 mjere o broju iteracija.

INTERNI D

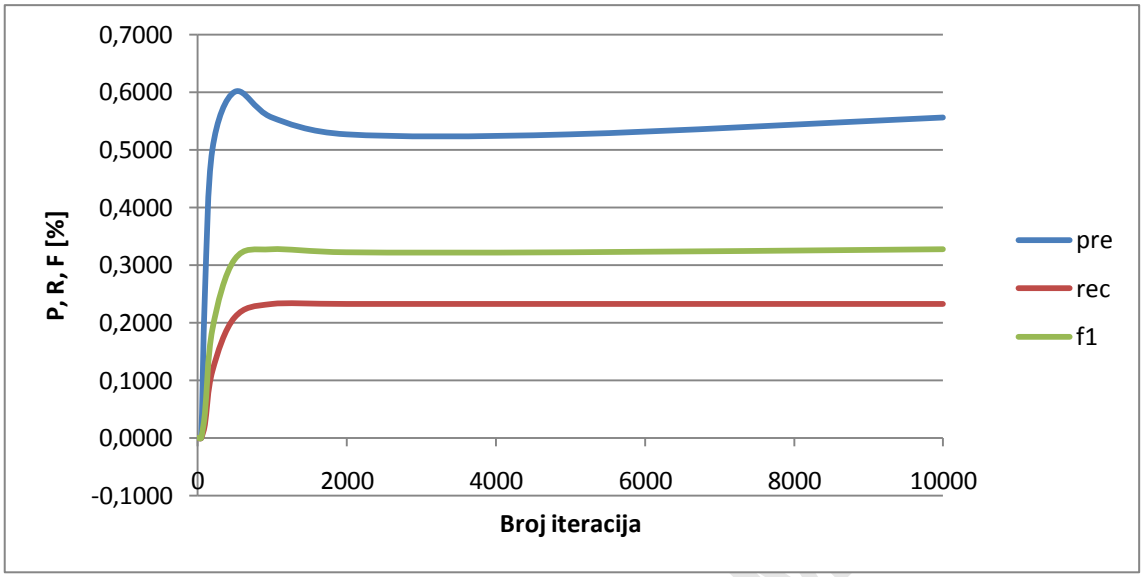


Slika 8 - Ovisnost preciznosti, odziva i F1 mjere cijelog sustava o broju iteracija

Iz grafova možemo zaključiti da se za širinu konteksta od pet riječi zadovoljavajući rezultati postižu već nakon 5000 iteracija. Daljnje treniranje donosi vrlo malo poboljšanje. Na svim grafovima možemo primijetiti da preciznost vrlo brzo naraste, a zatim lagano opada.



Slika 9 - Ovisnost preciznosti, odziva i F1 mjere označavanja naziva organizacija o broju iteracija



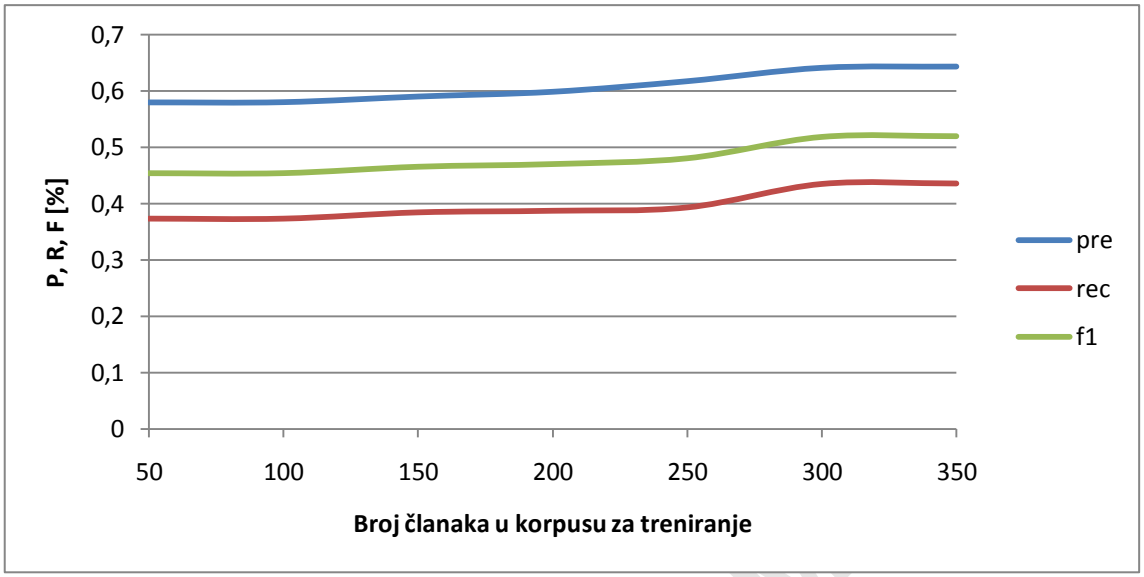
Slika 10 - Ovisnost preciznosti, odziva i F1 mjere označavanja naziva postotaka o broju iteracija

Kao što smo već spomenuli, imena organizacija se često sastoje od više riječi zbog čega ih je teže prepoznati. Na Slika 9 vidimo opadanje preciznosti koje nalikuje na opadanje preciznosti na početku Slika 10. Budući da preciznost na Slika 10 nakon inicijalnog opadanje počinje rasti, možemo nagađati da bi preciznost na Slika 9 počela rasti kada bi napravili dovoljno iteracija.

6.5. Utjecaj veličine korpusa za treniranje

Budući da je korpus za treniranje iznimno važan segment u dobivanju dobrih rezultata, potrebno je utvrditi koja je količina teksta potrebna da bi se postigli dobri rezultati sa što manjim korpusom. Da utvrdimo optimalnu veličinu korpusa za treniranje trenirali smo modele s različitim brojem članaka u korpusu za treniranje. Širina kontekstnog prozora za potrebe testova podešena je na pet riječi, a korištene su sve značajke.

INTERNI DOKUM



Slika 11 - Ovisnost preciznosti, odziva i F1 mjere sustava o broju članaka u korpusu za treniranje

Na temelju dobivenih rezultata zaključujemo da je nakon 300 članaka u korpusu za treniranje dolazi do stagnacije.

6.6. Odabrani model

Na temelju provedenih testova odabran je model koji koristi širinu konteksta od pet riječi treniran na korpusu od 300 članaka. Dobiveni su sljedeći rezultati:

Preciznost: 65.1%

Odziv: 40.4%

F_1 mjera: 49.9%

Ove rezultate treba uzeti s rezervom zbog korpusa na kojem je algoritam trenirao i zlatnog standarda korištenog za uspoređivanje rezultata. Dostupan korpus je označen sustavom OZANA koji prema autorovim testovima označava s F1 mjerom između 90 i 92 posto na korpusu za koji je izrađen. Budući da se korišteni korpus ponešto razlikuje od korpusa korištenog za testiranje sustava OZANA, realno je očekivati da je sustav imao i nešto slabije performanse od deklariranih.

Pri korištenju takvog korpusa nailazimo na nekoliko problema. U fazi treniranja sustav ima problema pri jasnom određivanju parametara zbog netočnosti označavanja. S druge strane, prilikom označavanja i usporedbe sa zlatnim standardom

INTERNI DOKUM

uspješnost je manja zbog netočno označenog zlatnog standarda. U konačnici, sustav je dva puta suočen s istim problemom što mu svakako narušava performanse. Kada bi bio dostupan ručno označeni korpus za treniranje i zlatni standard, performanse sustava bi sigurno osjetno porasle. Za usporedbu, na konferenciji MUC-7 najbolje rezultate ostvario je sustav LTG čija F_1 mjera iznosi 93.39%.

INTERNI DOKUMENT

7. Smjernice za daljnji rad

U ovom radu razmatrani su utjecaji raznih parametara treniranja, ali uvijek je razmatran utjecaj promjene samo jednog parametra dok su ostali parametri bili fiksirani. Bilo bi interesantno i poučno napraviti opširnije testove u kojima bi se mjerili utjecaji promjene više parametara.

Proučavajući sustave za druge jezike temeljene na modelu maksimalne entropije uočeno je da se oni pretežito oslanjaju na rječničke i leksičke značajke. Daljnji rast performansi sustava bio bi moguć kada bi se pročistili korišteni popisi imena i lokacija, te kada bi se napravili novi rječnici. Kao primjer možemo izdvojiti popis titula (gđa, dr, dipl. ing.), popis riječi vezane u organizacije (zavod, društvo, ministarstvo i sl.).

Rezultate ovog sustava trebalo bi temeljito proučiti te pokušati utvrditi koja bi pravila pomogla da se detektiraju tipovi naziva koje sustav s postojećim značajkama nije uspio detektirati. Za ozbiljnije rezultate potrebno bi bilo izraditi kvalitetan, ručno označeni korpus za treniranje, te definirati zlatni standard za tipove teksta koje se želi označavati.

Budući da su najbolje rezultate na konferenciji MUC-7 postigli hibridni sustavi svakako bi trebalo pokušati spojiti sustav temeljen sa pravilima na sustavom temeljenim na modelu maksimalne entropije.

INTERNI DOKUMENT

8. Zaključak

Cilj ovog rada bio je napraviti pregled metoda za strojno prepoznavanje i klasifikaciju naziva te utvrditi je li model maksimalne entropije primjenjiv za prepoznavanje i klasifikaciju naziva u hrvatskome jeziku. Odabrani model implementiran je s potrebnim modulima za treniranje, označavanje teksta te automatski odabir korištenih značajki.

Kroz ovaj rad pokazano je da je metoda maksimalne entropije primjenjiva za označavanje i klasifikaciju naziva, no kako se radi o nadziranoj metodi, krajnji rezultati vrlo su ovisni o korpusu za treniranje i rječničkim značajkama. Korpus za treniranje kao i zlatni standard korišteni u ovom radu daleko su od idealnog budući da se radi o strojno označenim tekstovima u kojima i neupućeni promatrač već na prvi pogled može pronaći greške. Korišteni rječnici također nisu kvalitetno obrađeni, već se radi o strojno prikupljenim podacima. U skladu s time, postignuti odziv i preciznost nisu vrhunski, ali pokazuju da metoda ima velik potencijal.

Dosad dobiveni rezultati opravdavaju daljnji razvoj i ulaganje u potrebne lingvističke resurse, a vrijedilo bi pokušati razviti i hibridni sustav sa nekim sustavom temeljenim na pravilima.

INTERNI DOKUMENT

Sažetak

NASLOV: Metoda maksimalne entropije i njena primjena za označavanje slijednog niza tekstnih podataka

Rad opisuje teorijsku podlogu prepoznavanja i klasifikacije naziva, daje pregled metoda za strojno prepoznavanje s naglaskom na metode strojnog učenja. Opisuje se odabrana programska implementacija sustava za prepoznavanje i klasifikaciju naziva, te binarne, morfološke, leksičke i rječničke značajke korištene za izgradnju modela maksimalne entropije. Opisan je automatski postupak za odabir optimalnih značajki. Napravljena je analiza širine kontekstnog prozora, broja korištenih značajki, veličine korpusa za treniranje te broja iteracija u ovisnosti o uspješnosti označavanja. Na temelju provedenih analiza odabrani su parametri modela koji pokazuje obećavajuće rezultate.

KLJUČNE RIJEČI: prepoznavanje i klasifikacija naziva, maksimalna entropija, strojno učenje, obrada prirodnog jezika, crpljenje obavijesti, hrvatski jezik

INTERNI DOKUMENT

Abstract

TITLE: Maximum Entropy Method and its Application on Text Tagging

This thesis describes theoretical background of Named Entity Recognition and Classification (NERC) and gives an overview of automated recognition methods with emphasis on machine learning methods. A maximum entropy NERC system is described, as well as the binary, morphological, lexical and dictionary features used by the model. The system implements an automated feature selection process. An analysis of performance depending on the context window size, number of features, number of articles in training corpus and number of training iterations is made. Based on the analysis, a model is built that shows promising results

KEYWORDS: named entity recognition and classification, aximum entropy, machine learning, natural language processing, information extraction, Croatian language

INTERNI DOKUMENT

Literatura

1. **Lillian, L.** (2004) *„I'm sorry Dave, I'm afraid I can't do that" : Linguistics, Statistics, and Natural Language Processing circa 2001.*, Computer Science: Reflections on the Field, Reflections from the Field [http://www.cs.cornell.edu/home/llee/papers/cstb.pdf]
2. **Moens, M.F. (**2006), *Information Extraction: Algorithms and Prospects in a Retrieval Context,* Springer
3. **Cunningham, H.** (1999) *Information extraction – a user guid,* Research Memo CS-99-07, Institute for Language, Speech and Hearing (ILASH) and Dept. of Computer Science, University of Sheffield, UK [http://citeseer.ist.psu.edu/cunningham99information.html]
4. **Bošnjak, M.** (2007), *Strojno prepoznavanje naziva tehnikama strojnog učenja.* Diplomski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu
5. **Chinchor N.** (1997), *MUC-7 Named Entity Task Definition (Version 3.5)*, Message Understanding Conference 7 Proceedings [http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/ne_task.html]
6. **Cohen, J.** (1960), *A coefficient of agreement for nominal scales*, Educational and Psychological Measurement Vol.20
7. **Davies, M. Fleiss, J.** (1982), *Measuring Agreement for Multinomial Data,* Biometrics, 38, 1047–1051
8. **Hripcsak, G., Rothschild, A.S.** (2005), *Agreement, the F-Measure, and Reliability in Information Retrieval* [http://www.jamia.org/cgi/content/abstract/12/3/296]
9. **Douthat, A.** (1998), *The Message Understanding Conference Scoring Software User's Manual,* Message Understanding Conference 7 Proceedings [http://www.itl.nist.gov/iad/894.02/related_projects/muc/muc_sw/muc_sw_manual.html]
10. **Settles, B.** (2004), *Biomedical named entity recognition using conditional random fields and rich feature sets.*
11. **Bekavac, B.** (2005), *Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima,* doktorska disertacija, Filozofski fakultet, Sveučilište u Zagrebu
12. **Borthwick, A.** (1999), *A maximum Entropy Approach to Named Entity Recognition*, Ph. D. Thesis, New York University [http://citeseer.ist.psu.edu/borthwick99maximum.html]

INTERNI DOKUMENT

13. **Mitchell, T. M.** (1997), *Machine Learning*, McGraw Hill
14. **Berger, A. L., Della Pietra, S. A., Della Pietra, V. J.** (1996), *A Maximum Entropy Approach to Natural Language Processing*, Columbia University [http://www.aclweb.org/anthology-new/J/J96/J96-1002.pdf]
15. **Forney, G. D.** (1973), *The Viterbi algorithm*, Proceedings of the IEEE 61(3):268–278
16. **Abney, S.** (2002), *Bootstrapping*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia [http://citeseer.ist.psu.edu/abney02bootstrapping.html]
17. **Kozareva, Z.** (2006), *Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists*, In Proceedings of EACL student session (EACL 2006), Trento, Italy. [http://acl.eldoc.ub.rug.nl/mirror/E/E06/E06-3004.pdf]
18. **Roweis, S.** (2003), *Lecture 12: Meta-Learning Methods*, Predavanja s predmeta Machine Learning, University of Torronto [http://www.cs.toronto.edu/~roweis/csc2515-2003/notes/lec12x.pdf]
19. **Tsukamoto, K., Mitsuishim, Y., Sassano, M.** (2002), *Learning with Multiple Stacking for Named Entity Recognition*, Proceeding of the 6th conference on Natural language learning - Vol. 20 [http://citeseer.ist.psu.edu/tsukamoto-learning.html]
20. **Carreras, X., Màrquez, L., Padró, L.** (2002), *Named Entity Extraction using AdaBoost*, Proccedings of CoNLL 2002 Shared Task Contribution [http://citeseer.ist.psu.edu/carreras02named.html]
21. **Tadić, M., Fulgosi, S.** (2003), *Building the Croatian Morphological Lexicon*, In Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages, Budapest [http://www.hnk.ffzg.hr/txts/mtsf4EACL2003.pdf]
22. **Boras, D., Mikelić, N., Lauc, D.** (2003), *Leksička flektivna baza podataka hrvatskih imena i prezimena*, Modeli znanja i obrada prirodnog jezika – Zbornik radova, Radovi Zavoda za informacijske studije (knj. 12)
23. **Bekavac, B., Tadić, M.** (2007), *Implementation of Croatian NERC System*, Balto-Slavonic Natural Language Processing 2007, ACL 2007, Prague [http://acl.ldc.upenn.edu/W/W07/W07-1702.pdf]
24. **Tadić, M.** (1996), *Računalna obradba hrvatskoga i nacionalni korpus*, Suvremena lingvistika, Zagreb

INTERNI DOKUMENT

25. **Borthwick, A., Sterling, J., Agichtein, E., Grishm, R.** (1998), *Description of the MENE Named Entity System as Used in MUC-7*, NYU [http://citeseer.ist.psu.edu/borthwick98nyu.html]
26. **Šilić, A., Šarić, F., Dalbelo Bašić, B., Šnajder, J.** (2007), *TMT: Object-Oriented Text Classification Library*, Proceedings of the 29th International Conference on Information Technology Interfaces
27. **Northedge, R.** (2005), *Maximum Entropy Modeling Using SharpEntropy* [http://www.codeproject.com/KB/cs/sharpentropy.aspx]
28. **Jaynes, E.T.** (1957), *Information theory and statistical mechanics*. Physics Reviews 106
29. **Manning, C.D., Schutze H.** (1999), *Foundations of Statistical Natural Language Processing*, MIT Press

INTERNI DOKUMENT

Dodatak A: Korpus za treniranje

Primjer članka iz korpusa za treniranje:

```
<doc>
    <s><ENAMEX TYPE="PERSON">Ivica Žulj</ENAMEX> dosjetio se kako će
doskočiti kradljivcima ukrasne zvijezde sa svojega Mercedesa :
      Oznaka Osječkog piva s točionika jeftinija od nišana
      VIŠNJEVAC - <ENAMEX TYPE="LOCATION">Osijekom</ENAMEX> i okolicom
već se godinu dana vozi mercedes s neobičnom oznakom na masci automobila
umjesto mercedesove zvjezdice i zaokuplja pozornost svih prolaznika.</s>
    <s>Umjesto nišana, vlasnik automobila<ENAMEX TYPE="PERSON"> Ivica
Žulj</ENAMEX> zvani Žumpi stavio je oznaku s točionika Osječkog piva,
čime izaziva salve smijeha pješaka kada zastane mercedesom na
semaforu.</s>
    <s>Tridesetčetverogodišnji Žumpi vlasnik je "Pecare", višnjevačkog
cafe-bara, pa mu je, radeći iza šanka, jednog dana sinula neobična ideja
kako doskočiti kradljivcima mercedesove zvijezde, koju su mu već triput
ukrali s automobila.</s>
    <s>- Često se kradu zvijezde s mercedesa jer su lijep ukras, a kako
nisu jeftine, dosadilo mi je kupovati ih.</s>
    <s> Gledao sam u taj znak na točioniku i dosjetio se kako će se
zgodno uklopiti s automobilom, a neće ga krasti jer nije atraktivan kao
nišan - kazao nam je Žumpi, za kojeg njegovi sumještani kažu kako mu ta
oznaka i priliči jer ga predstavlja onakvim kakav i jest - vjeran
Osječkom pivu od kada zna za sebe, a i u njegovu se lokalu od hrvatskih
piva toči jedino ono iz osječke Pivovare i strana piva.</s>
    <s>Gosti su naviknuli na njegovu odanost Osječkom pivu, a i mi smo
se u to mogli uvjeriti kada smo ga zatekli za stolom kako s njima u
prijateljskom razgovoru pijucka to najdraže mu hmeljno piće.</s>
    <s>Vlasnik <TIMEX TYPE="DATE">22 godine</TIMEX> starog pivarskog
mercedesa, koji je prepoznatljiv u cijelom gradu, pa zbog njega i gosti
uvijek znaju gdje je trenutačno gazda "Pecare", kaže kako je i kilograme
natukao pijući godinama Osječko pivo, a omiljena mu je pjesma "Voli Ivo
piti pivo", koja kad zasvira, zaigra i Ivino srce. </s>
    <s> Ima Ivo 150 kilograma, a teško mu se i izvagati jer mora tražiti
posebnu vagu koja važe više od onih kućnih.</s>
    <s> Bilo bi lijepo kada bi osječka Pivovara prepoznala Žumpijevu
odanost njihovu pivu, koje pije oduvijek kao lokalpatriot.</s>
    <s> Bezbrojni su primjeri darivanja pojedinih velikih prodavaonica
koje nagrađuju svoje konzumente i kupce pojedinim proizvodima u količini
koja je jednaka njihovoj težini.</s>
    <s>Razloga za nagrađivanje i darivanje ima jer<ENAMEX TYPE="PERSON">
Ivo</ENAMEX> <TIMEX TYPE="DATE">24. siječnja</TIMEX> slavi rođendan pa
bi darivanje bio pravi mali spektakl - ako se pronađe izdržljivija vaga
od one kućne.</s>
    <s>(Autor: <ENAMEX TYPE="PERSON">Darko PEJIĆ</ENAMEX>)</s>
</doc>
```

NEPRIJATELJSKI DOKUMENT