SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1679

STROJNO PREPOZNAVANJE NAZIVA TEHNIKAMA STROJNOG UČENJA

Matko Bošnjak

Zagreb, rujan 2007.

Zahvaljujem se prof. dr. sc. Dalbelo Bašić
na podršci i usmjeravanju u meni, do nedavno, nepoznato područje crpljenja
obavijesti, te mr. sc. Janu Šnajderu i Karli Brkić na besprijekornom lektoriranju.

Posvećeno majci Miri

...i Sonji

unatoč...

Popis tablica

Popis slika

Sadržaj

1. Uvod

Prirodni jezik, govoren, pisan ili oblikovan u znakove, jedna je od ljudskih tvorevina koja čini čovjeka misaonim bićem koje svojim kognitivnim sposobnostima odskače od ostatka živog svijeta. Od 1950. godine, pojavom Turingovog testa [50] prirodni jezik postaje predmetom proučavanja računalne znanosti, uz već poznate formalne jezike. Formalni se jezici, međutim, za razliku od prirodnih, ne koriste za opću već namjensku komunikaciju

Obrada prirodnog jezika (*eng. Natural Language Processing, NLP*) je polje računalne znanosti posvećeno izgradnji sustava[1] koji koriste (prirodne) ljudske jezike kao ulaz i izlaz [30], dakle sustave za koje bi se moglo reći da „razumiju“ prirodni jezik – zadaću koja zasigurno nije laka makar se na prvi pogled činilo da jest [32]. To je polje interdisciplinarne suradnje računalne znanosti, lingvistike, kognitivne znanosti, informacijske znanosti, matematike i statistike.

Tema ovog rada je jedan od fundamentalnih problema obrade prirodnog jezika – prepoznavanje i klasifikacija naziva.

Rad je koncipiran u dvije cjeline: u prvoj se cjelini istražuje teorijska podloga problematike prepoznavanja naziva i pregled dostupnih metoda te konkretnih implementacija dok se u drugoj opisuju implementacija, testiranje te vrednovanje rezultata iste.

[1] računala i/ili programa

2. Teorijska podloga

Prepoznavanje naziva (*eng. Named Entity Recognition, NER*) obuhvaća postupak identifikacije naziva u tekstu, kao i njihovo eksplicitno obilježavanje nakon (ili za) kojega se često obavlja i klasifikacija istih (*eng. Named Entity Classification, NEC*) u unaprijed dogovorene pripadajuće kategorije. U konkretnim se implementacijama najčešće obavljaju obje radnje sjedinjene u jedan postupak koji se tada naziva prepoznavanje i klasifikacija naziva (*eng. Named Entity Recognition and Classification, NERC*), u nastavku rada PKN.

Termin naziv (*eng. Named Entity*) prema konferenciji MUC-7[2] označava skup imena proširenih s određenim vremenskim i brojčanim izrazima. Imena su podskup naziva, izrazi kojima se referira na izvanjezične entitete – osobe, organizacije i lokacije. Međutim nazivi ne moraju nužno biti samo kategorije definirane na konferenciji MUC-7, već i drugi tipovi podataka, u zavisnosti o potrebi[3] (adrese, imena knjiga, imena glumaca, imena bjelančevina itd.).

Glavno svojstvo naziva definiranih prema konferenciji MUC-7 je da sudjeluju u konkretnoj realizaciji jezika odgovaranjem na konkretna pitanja o sadržaju teksta: tko, što, kada, gdje i koliko te time postaju nosioci velike količine obavijesti. Uz to je za brojne jezike utvrđeno da u nekim žanrovima nazivi pokrivaju čak i jednu desetinu cjelokupnog teksta te gotovo da ne postoji tekst na nekom od prirodnih jezika koji ih ne sadrži. Vrlo često se na osnovi prepoznatih naziva može odrediti domena teksta kojoj tekst pripada – važan podatak za mnoge razine obrade jezika.

Gotovi sustavi za PKN postoje za poznatije jezike (npr. engleski, njemački, nizozemski i dr.) te se novi razvijaju za druge. Međutim pri razmatranju ideje o

---

[2] v. poglavlje 2.2
[3] S. Sekine, (2003), Definition of Sekine's Extended Named Entity, http://nlp.cs.nyu.edu/ene/version6_1_0eng.html

preuzimanju takvog ili takvih sustava i njihovoj adaptaciji na neki drugi jezik valja razmisliti. Mogući su, naime, bitno lošiji rezultati uzrokovani jezično-ovisnim razlikama u specifičnim pravilima kodiranja naziva do neizbježno različitih naziva. Čak ni prilagodba sustava izgrađenog za neki jezik blizak hrvatskome (neki od slavenskih jezika) nije moguća iz jednostavne činjenice da takvi sustavi nisu rađeni za slavenske jezike. Navedeno se naravno ponajviše odnosi na sustave izgrađene na ručno kodiranim pravilima dok je za metode temeljene na strojnom učenju adaptacija moguća te je čak i izrada takvog sustava olakšana u odnosu na sustav zasnovan na pravilima.

Sam zadatak naizgled izgleda jednostavno – naivan pristup bi mogao navesti na prepoznavanje svih riječi u tekstu pisanih velikim slovom za kojim slijede mala, no samo problem velikog slova na početku rečenice je prvi veći problem takvog pristupa. Čak i najjednostavniji pristup izrade opsežnog popisa imena iziskivao bi težak i mukotrpan rad čiji bi rezultat neizbježno zastario već sljedeći dan zbog npr. propadanja starih i osnivanja novih tvrtki. Tu je i problem raznih varijacija i akronimskih inačica osnovnog naziva poput *PBZ*, *Privredna banka*, *Privredne banke* koje su sve varijacije osnovnog naziva *Privredna banka Zagreb*. Čak i kada bi bilo moguće obuhvatiti sva imena organizacija, lokacija i osoba, te sve njihove varijacije, preklapanja bi između tih popisa bila tolika da bi to postao velik i teško rješiv problem. Primjerice *Matija Mesić* može biti u popisu imena, ali i u popisu organizacija kao dio naziva *Gimnazija „Matija Mesić“*, *Zagreb* može biti i u popisu lokacija, ali i u popisu organizacija kao dio naziva *Osiguranje Zagreb*. Čak i riječ *Hrvatska* može se, osim u popisu lokacija, naći i u popisu osoba kao dio imena osobe, npr. *Ivan Hrvatska*. Sve navedeno vodi zaključku da je potrebno izgraditi sustav koji bi uvažavao specifičnosti hrvatskog jezika.

Postupak izrade sustava za PKN je interdisciplinarnog karaktera s uporištima u lingvistici i informatici, katkada i statistici. Promatrano sa stajališta znanosti o jeziku, ovaj postupak pripada području računalne lingvistike (*eng. Computational Linguistics*) – dijela znanosti o jeziku koji koristi računalo kao pomoćno sredstvo u istraživanju jezika, dok promatrano sa stajališta računarstva, postupak pripada području obrade prirodnog jezika – dijela računarstva čiji je objekt proučavanja

računalni sustav koji obrađuje prirodni jezik. U nastavku rada, problemu PKN-a će se pristupiti sa stajališta računalne znanosti.

Problem PKN unutar polja obrade prirodnog jezika predstavlja jedan od zadataka iz područja crpljenja obavijesti.

2.1. Crpljenje obavijesti

Crpljenje obavijesti (*eng. Information Extraction*) je prema Moens [36] definirano kao identifikacija i (naknadna ili istodobna) klasifikacija i strukturiranje specifičnih informacija pronađenih u nestrukturiranim izvorima podataka (poput teksta prirodnog jezika) u semantičke klase, čime se informacija čini primjerenijom za daljnju obradu.

PKN je samo jedan od pet zadataka unutar područja crpljenja obavijesti, a izloženi redom kojim se obavljaju, ti zadaci su:

1. Prepoznavanje i klasifikacija naziva (*eng. Named Entity Recognition and Classification, NE*)
   - pronalazak i klasifikacija naziva u tekstu
2. Razrješavanje koreferencija (*eng. Coreference Resolution, CO*)
   - identifikacija veza među entitetima u tekstu – jedna je od osobina teksta da se na iste izvanjezične entitete najčešće ne referira istim nazivima već se zamjenjuju skraćenim oblicima ili zamjenicama koje razrješavanje koreferencija dovodi u svezu s imenima iz različitih rečenica
3. Izrada obrasca elementa (*eng. Template Element Production, TE*)
   - dodavanje deskriptivne informacije nazivima koristeći razrješavanje koreferencija – na osnovi obavijesti iz prvih dvaju koraka, združuje prikupljene podatke iz kojih se izrađuju obrasci s opisnim informacijama entiteta.
4. Konstruiranje odnosa među obrascima (*eng. Template Relation Construction, TR*)
   - izgradnja relacija između entiteta iz prethodnog koraka.
5. Izrada obrasca scenarija (*eng. Scenario Template Production, ST*)
   - združivanje rezultata trećeg i četvrtog koraka u određeni scenarij događaja.

Primjer

U svrhu jasnije predodžbe zadataka crpljenja informacija, razmotren je sljedeći primjer – adaptacija primjera iz [16]:

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 sljedeći:

- NE (pravokutnici pune crte) obilježava prisutne nazive raketa, utorak, dr. Ludoga Znanstvenika, Dr. Znanstvenik i Mi Gradimo Rakete Inc.
- 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.

Crpljenje obavijesti ne valja brkati s pronalaženjem obavijesti (*eng. Information Retrieval*) koje za zadaću ima pronalaženje važnih (cijelih!) dokumenata, za razliku od crpljenja obavijesti čija je zadaća pronalazak važnih podataka unutar dokumenata tj. traženje specifične obavijesti u tekstu [16].

Područje crpljenja obavijesti počelo se intenzivno razvijati pod utjecajem konferencija o razumijevanju poruka.

2.2. Konferencije o razumijevanju poruka

Spomenuta konferencija MUC-7 zadnja je od konferencija o razumijevanju poruka (*eng. Message Understanding Conferences, MUC*) koje su devedesetih godina prošlog stoljeća postavile brojne standarde i znatno pridonijele razvoju područja crpljenja obavijesti. Organizator konferencija je američka agencija *The Defense Advanced Research Projects Agency (DARPA)* koja je postavila cilj vrednovanja sustava za crpljenje obavijesti (konferencije su bile natjecateljskog karaktera). Od ukupno 7 održanih konferencija, PKN je u posebnom žarištu bila na

konferencijama MUC-6 i MUC-7. Na posljednjoj konferenciji MUC-7 održanoj 1998. sudjelovalo je 12 natjecatelja/sustava od kojih je većina koristila ručno izrađena pravila.

Prema MUC specifikacijama, nazivi su definirani kao vlastita imena i određeni izrazi za iznose te su kategorizirani na sljedeći način:

- Imena osoba, organizacija i lokacija
- Vremenski i datumski izrazi
- Brojčani izrazi – postotci i novčani izrazi

Kategorijama tj. vrstama entiteta pripadaju i odgovarajuće oznake, redom: ENAMEX, TIMEX i NUMEX.

Prema zadacima prepoznavanja gore navedenih entiteta, sustav za PKN mora proizvesti jedinstven, jednoznačan tekst za bilo koji relevantan niz znakova u tekstu, čak i onda kada pravi odgovor nije na prvi pogled očigledan. Dogovoreni jezik za obilježavanje je SGML[4] (*Standard Generalized Markup Language*), a pronađeni entiteti trebaju biti obilježeni kao elementi s pripadajućim atributima po sljedećem uzorku:

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

npr.

<ENAMEX TYPE="LOCATION">Sjeverna Koreja</ENAMEX>

oznaka_entiteta govori o kojoj se vrsti entiteta radi, a atribut *tip_entiteta* dodatno specificira entitet.

Svi potencijalni elementi i atributi definirani na MUC-7 konferenciji su:

1. ENAMEX vrste entiteta
   - ORGANIZATION – tvrtke, državne institucije i druge organizacija
   - PERSON – vlastita imena i prezimena
   - LOCATION – imena politički ili geografski definiranih lokacija (gradovi, pokrajine, države, vode, planine, itd.)

[4] http://www.w3.org/MarkUp/SGML/

2. TIMEX vrste entiteta
   - DATE – potpuni i nepotpuni izrazi za datume
   - TIME – potpuni i nepotpuni izrazi za vremena unutar dana
3. NUMEX vrste entiteta
   - MONEY – novčani izrazi
   - PERCENT – postotni izrazi

Primjer

Primjer teksta [4] obilježenog po MUC-7 specifikacijama može se vidjeti na slici 1.

I Program osiguranja izvoza zabilježio je **<TIMEX TYPE="DATE">**lani **</TIMEX>** veliki rast. U **<TIMEX TYPE="DATE">**2004. godini**</TIMEX>** osiguran je promet od **<NUMEX TYPE="MONEY">**580 milijuna kuna **</NUMEX>**, što je povećanje**<NUMEX TYPE="PERCENT">**180 posto **</NUMEX>** prema **<TIMEX TYPE="DATE">**prethodnoj godini**</TIMEX>**, a odobreno je 357 zahtjeva, što je povećanje od **<NUMEX TYPE="PERCENT">** 306 posto**</NUMEX>**. **<TIMEX TYPE="DATE">**Lani**</TIMEX>** je **<ENAMEX TYPE="ORGANIZATION">**HBOR**</ENAMEX>** osigurao izvoz 67 izvoznika, za razliku od 35 u **<TIMEX TYPE="DATE">**2003. godini**</TIMEX>**.

Slika 1. Tekst obilježen po MUC-7 specifikacijama

Na konferencijama MUC razvijen je i potpuno automatizirani program za ocjenjivanje (*eng. scoring software*) kako ljudskog tako i strojnog obilježavanja. Utvrđeno je da je i ljudsko rješavanje PKN problema podložno pogreškama te da ovisi o razini ekspertize čovjeka koji vrši obilježavanje. Navedena tvrdnja je dokazana mjerenjem nad dvama osobama koje su u konačnici postigle F-mjere[5] od 97,60% i 96,95%..

[5] v. poglavlje 2.4

Konferencije MUC više se ne održavaju, ali postoji nekoliko konferencija koje se smatraju konferencijama nasljednicama: CoNNL (*Conference on Computational Natural Language Learning*) i ACE (*Automated Content Extraction*).

CoNNL konferencije održavaju se godišnje s određenom temom, a 2002. i 2003. godine tema je bila „Jezično neovisno prepoznavanje i klasifikacija naziva“ (*eng. Language-independent NERC*). Godine 2002. jezici od interesa bili su španjolski i nizozemski, a 2003. engleski i njemački.

Jedan je od glavnih ciljeva ACE programa, čiji je pokrovitelj NIST (*National Institute of Standards and Technology*), jest „detekcija i praćenje naziva“ (*eng. Entity Detection and Tracking*).

Među ostalim konferencijama i natjecanjima koji su popularizirali PKN problem, tu su i konferencija MET (*The Multilingual Entity Task*) na kojoj su promatrani jezici bili španjolski, japanski i kineski te natjecanje IREX (*Information Retrieval and Extraction Exercise*) na kojemu su tema bili crpljenje i pronalaženje obavijesti (uključivo i PKN) za japanski jezik.

2.3. Primjena

Neke od primjena sustava za PKN su:

- generiranje metapodataka namijenjenih objavljivanju na Internetu
- kvalitetnije Internet tražilice koje dohvaćaju preciznije rezultate o traženom pojmu npr. ukoliko se traži grad *Clinton, South Carolina* dobiveni rezultati ne obuhvaćaju stranice povezane s istoimenim predsjednikom SAD-a
- sažimanje dokumenta s obzirom na neku temu
- velika ušteda ljudskog rada pri automatskom generiranju dokumentacijskih i knjiških indeksa – većina stvari koje bi ušle u indeks su nazivi
- sustav za PKN korak je pretprocesiranja pri pojednostavljenju zadataka poput strojnog prevođenja
- ključna komponenta kompleksnijih sustava za crpljenje obavijesti

- od 1998 se također pojavio veliki interes za prepoznavanje naziva u domenama molekularne biologije i bioinformatike gdje tražene nazive predstavljaju imena gena i genskih produkata [41]

2.4. Metrika

Mjere koje se koriste za vrednovanje uspješnosti sustava u zadatku crpljenja obavijesti [36], a time i standardne mjere za vrednovanje PKN zadatka, tipične se mjere iz područja klasifikacije teksta: preciznost, odziv i F-mjera.

Preciznost (*eng. precision*) je mjera koja se definira kao omjer svih sustavom pronađenih točnih naziva i svih naziva koje je sustav ponudio kao rezultat. Ova je mjera pokazatelj točnosti sustava te govori koliko je sustav ponudio pogrešnih odgovora.

$$P = \frac{\text{Sustavom pronadjeni tocni nazivi}}{\text{Svi sustavom pronadjeni nazivi}} \tag{2.1}$$

Odziv (*eng. recall*) je mjera koja se definira kao omjer naziva koje je sustav ispravno pronašao i svih točnih naziva u tekstu. Ova je mjera pokazatelj sveobuhvatnost sustava, odnosno ocjenjuje koliko je sustav potpun u pronalaženju relevantnih informacija.

$$R = \frac{\text{Sustavom pronadjeni tocni nazivi}}{\text{Svi tocni nazivi u tekstu}} \tag{2.2}$$

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

$$F - mjera = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{2.3}$$

Parametar β odražava relativnu važnost preciznosti i odziva. Ukoliko preciznost i odziv imaju jednaku težinsku vrijednost, tada je $\beta = 1$ te F-mjera postaje mjera pod nazivom harmonijska sredina ili F_1-mjera:

$$F_1 - mjera = \frac{2 \times P \times R}{P + R} \tag{2.4}$$

Česta je praksa da se navedene mjere izražavaju u postotcima. U nastavku rada će se pod pojmom F-mjera zapravo podrazumijevati F_1-mjera što je praksa u radovima iz ovog područja.

Primjer

Ako u tekstu postoji 300 naziva, a sustav je pronašao 220 od kojih je 170 ispravno, tada je preciznost sustava P = 0,77, odziv R = 0,57, a F_1-mjera = 0,66.

3. Sustavi za prepoznavanje i klasifikaciju naziva

Metodologija koja se koristi za rješavanje PKN problema može se raščlaniti na dva osnovna pristupa: prvi se temelji na pravilima, a drugi na metodama strojnog učenja. Osnovna je razlika ova dva pristupa ta što se sustavi temeljeni na pravilima primarno koriste jezičnim činjenicama, dok se sustavi temeljeni na strojnom učenju primarno koriste algoritmom obrade i adekvatnim izborom značajki.

Uz navedena dva osnovna pristupa, prisutan je i treći, hibridni pristup koji ih objedinjuje.

3.1. Sustavi zasnovani na pravilima

Ručno izrađeni sustavi koji se snažno oslanjaju na intuiciju ljudskih dizajnera zovu se sustavi zasnovani na pravilima (*eng. rule based*). Takvi se sustavi najčešće modeliraju regularnim gramatikama.

3.1.1. Regularne gramatike, jezici i izrazi

Regularne gramatike su gramatike koje mogu opisati regularne jezike, a regularni jezici su jezici koji se mogu prikazati regularnim izrazima (*eng. Regular Expressions, RegEx*) ili konačnim automatom (*eng. Finite State Automaton, FSA*).

Deterministički konačni automat (DKA) je definiran kao uređena petorka :

$$(Q, \Sigma, \delta, q_0, F) \tag{3.1}$$

gdje su:

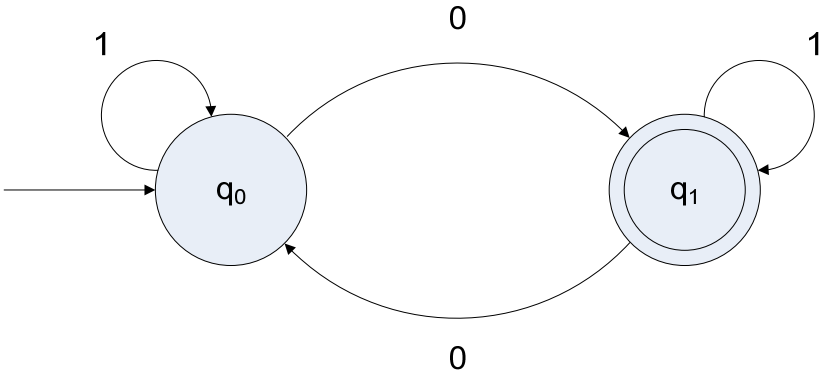
$Q \equiv$ konačni skup stanja

$\Sigma \equiv$ konačni skup znakova, alfabet

$\delta \equiv$ prijelazna funkcija $\delta : Q \times \Sigma \to Q$

q_0 ≡ početno stanje, $q_0 \in S$

F ≡ skup završnih stanja, $F \subseteq S$



Slika 2. Primjer deterministočkog konačnog automata

Slika 2. prikazuje deterministički konačni automat koji prihvaća nizove nula i jedinica s neparnim brojem nula, npr. 11101, 1011010, itd.

Automati su važna klasa generatora jezika, posebno pogodnih za računalnu primjenu. Dovoljno su izražajni za modeliranje pravila sustava. Koriste se kao prepoznavatelji jezika pri izvedbi sustava za PKN. Beskontekstne gramatike su znatno snažniji, ali ujedno i složeniji formalizam za obradu jezika pa ne postoji dovoljna opravdanost za njihovu primjenu u rješavanju problema PKN.

Uporaba regularnih gramatika za obradu jezika u slučaju prepoznavanja i klasifikacije naziva primjerenija je kao rješenje iz sljedećih razloga [4]:

1. Jednostavnost prikaza i izvedbe – automati su iznimno jednostavni mehanizmi s čitljivim i preglednim zapisom pravila
2. Brzina obrade regularnim gramatikama je mnogo veća u odnosu na beskontekstne gramatike. Učinkovite su sa stajališta računalnih resursa. Eksperimentalno je dokazano da je sustav zasnovan na konačnom automatu iznimno robustan.
3. Postoji velik broj razrađenih alata za obradu prirodnog jezika koji koriste regularne izraze.

3.1.2. Osobine sustava zasnovanih na pravilima

Osobine sustava zasnovanog na pravilima jesu [4], [8]:

- Ne zahtijevaju prethodno obilježene tekstove, samo tekstove na osnovu kojih će se izraditi pravila.
- Čitljivost, tj. preglednost sustava je veća – važna osobina za uočavanje i otklanjanje pogrešaka koje se sustavno pojavljuju.
- Neke klase uzoraka je iznimno teško prepoznati metodama strojnog učenja, a relativno lako regularnim izrazima.
- Sustavi zasnovani na ručno izrađenim pravilima mogu dati najbolje rezultate, naravno u ovisnosti o količini i kvaliteti uloženog rada.
- Autori sustava moraju biti lingvisti, a samim time cijena sustava znatno raste jer se oslanja na stručnost računalnih lingvista koji su skup resurs
- Performanse su visoko ovisne o stručnosti računalnih lingvista i o količini utrošenog rada [8]
- Potrebno je duže vrijeme za razvoj sustava. U usporedbi sa sustavima temeljenim na strojnom učenju, vrijeme izgradnje pravila je znatno duže.
- Adaptacija sustava za prepoznavanje tekstova iz druge domene može biti zahtjevna. Tekstovi iz druge domene mogu imati bitno drugačije osobine, naravno, adaptacija mora biti ručna.
- Adaptacija sustava za prepoznavanje drugih jezika iznimno je teška ili čak nemoguća. Potrebno je izraditi nova pravila i nove leksikone.

Međutim, kako ništa nije savršeno, tako ni regularni izrazi ne mogu u potpunosti riješiti problem PKN. Skoro svako pravilo za nazive uvijek će imati mnogobrojne iznimke [8]. Općenito je nemoguće, s obzirom na dano vremensko ograničenje, kodirati svaku iznimku koja se uoči prilikom oblikovanja sustava za PKN, pogotovo iznimke koje ne postanu očite sve do testiranja sustava. Uz to svaki različiti tip dokumenta može imati svoje posebnosti poput stereotipiziranih naslova (pravila za takve naslove bi pomogla na tim naslovima, ali ne i na naslovima drugih domena/tekstova).

3.2. Sustavi zasnovani na metodama strojnog učenja

Strojno učenje je prema općoj definiciji *proučavanje i izgradnja računarskih sustava koji automatski poboljšavaju svoje performanse kroz iskustvo* [35]. Za razliku od sustava zasnovanih na pravilima, takvi sustavi stječu potrebnu količinu znanja učeći uzorke na primjerima iz obilježenog ili neobilježenog korpusa za uvježbavanje. Sustavi koji su naučeni na nekom korpusu za učenje zatim se mogu koristiti kao označivači koji će moći obilježiti prethodno neviđene podatke, tj. koriste se kao klasifikatori koji će te podatke moći klasificirati. Dakako, baš kao i za sustave zasnovane na pravilima, na konačne rezultate sustava bitno utječe domena teksta koja je služila za prikupljanje znanja o jeziku.

Na osnovi korpusa za uvježbavanje, za svaku se pojavnicu[6] w_i izračunava vjerojatnost pridruživanja jedne od mogućih oznaka kategorije naziva, c_i. Jednostavno izračunavanje $p(c_i|w_i)$ za svaku pojavnicu, neovisno o drugima, nedvojbeno bi postiglo loš rezultat jer se na taj način ne bi u obzir uzimalo supojavljivanje drugih pojavnica u kontekstu. Stoga se izračunava vjerojatnost označavanja u ovisnosti o kontekstu duljine n oko pojavnice w_i, $p(c_i|w_{i+n},...,w_i,...,w_{i-n})$. Osim promatranja same pojavnice i njenog konteksta, većina sustava koristi i druge jezične i nejezične osobine teksta – značajke (*eng. features*).

3.2.1. Odabir značajki

Pristup strojnog učenja oslanja se na značajke vektora izgrađenih iz označenih (tj. klasificiranih) ili neklasificiranih kolekcija dokumenata. Ovisno o zadatku klasifikacije, odabire se skup značajki. Obično se ne koriste sve značajke prisutne u tekstu, već se selektira određeni broj važnih za dotični zadatak crpljenja obavijesti, kako bi se reducirala kompleksnost izračuna pri učenju klasifikatora, te kako bi se ujedno sačuvalo što je više moguće diskriminatornih informacija.

[6] v. poglavlje 0

Za sustave otvorene adaptaciji na druge domene i/ili jezike važno je i da značajke budu generičke, tako da budu uporabljive kroz različite domene i/ili jezike te da se njihove vrijednosti mogu automatski detektirati. Značajke se prema Moens [36] po tipovima vrijednosti mogu podijeliti na diskretne i kontinuirane. Posebna vrsta diskretnih značajki jesu značajke tipa *boolean* tj. tipa koji podrazumijeva jednu od dvije osnovne vrijednosti: 0 ili 1. Značajke se mogu razlikovati i po poziciji u tekstu. Prvo, mogu se definirati značajke koje se pojavljuju u samoj informacijskoj jedinki (pojavnici) poput kompozicije slova i brojki u entitetu. Drugo, mogu se definirati one značajke koje su u susjedstvu promatrane jedinke tj. u njenom kontekstnom „prozoru“[7]. Treće, mogu se definirati značajke koje reprezentiraju vezu između dvije jedinice, te u konačnici značajke koje podrazumijevaju gledanje šireg konteksta u kojemu se informacijska jedinica može pojaviti (šireg u smislu do granica dokumenta ili kolekcije dokumenata). Pomoću takvih se značajki rješava npr. problem ponavljanja naziva.

Skup značajki (*eng. set of features*) specifičan je za svaku pojedinu vrstu obrade teksta. Značajke tako mogu biti npr grafijske, interpunkcijske, pripadnost popisu imena, vrsta riječi i sl.

Najčešće korištene značajke primjenjive na problem PKN su:

- Leksičke značajke (*eng. Lexical Features*)
  - Svi leksički atributi riječi – je li riječ pisana velikim početnim slovom, je li je cijela pisana malim slovima ili je cijela pisana velikim slovima, sadrži li riječ velika slova unutar strukture, sadrži li brojke, u kojem uzorku sadrži brojke, koji su afiksi riječi tj. njeni sufiksi i prefiksi, zatim korijen riječi, rod, broj i padež – jednostavne morfosintaktičke značajke.
- Sintaktičke značajke (*eng. Syntactic Features*)
  - Najčešće korištena značajka iz ove grupe značajki je označavanje vrsta riječi (*eng. part-of-speech tagging, POS tagging*) koje je značajno za određivanje vrijednosti drugih značajki
- Značajke popisa/rječnika (*eng. Dictionary Features*)

[7] Pojam kontekstni prozor se koristi kao istoznačnica pojma kontekst

  - Značajke koje određuju nalazi li se riječ ili pojam u popisima ili rječnicima naziva poput popisa osobnih imena, popisa organizacija, popisa lokacija i dr.

Neki sustavi [7] izdvajaju još i sljedeće značajke:

- Značajke sekcije (*eng. Section Features*)
  - Značajke važne za određene tipove odjeljaka teksta – naslov, preambula ili nešto drugo sa specifičnim pravilima pisanja.
- Značajke vanjskih sustava (*eng. External Systems Features*)
  - Izlazi drugih sustava mogu se koristiti kao skup značajki koje se zatim kombiniraju s ostalim značajkama sustava. Time se postiže efektivno korištenje znanja drugih sustava uz mogućnost ispravljanja njihovih inherentnih pogrešaka[8]

Sve navedene značajke odnose se na samu pojavnicu kao informacijsku jedinku, ali također i na njeno susjedstvo – susjedne riječi u kontekstnom prozoru, tj. fiksan broj riječi lijevo i desno od same pojavnice.

Koriste se različite strategije odabira značajki (*eng. feature selection strategies*) koje najpovoljnije utječu na konačan rezultat obrade. Idealan je slučaj kada minimalan broj odabranih relevantnih značajki utječe na rezultat PKN. Primjeri strategija odabira značajki su: postavljanje praga na broj pojavljivanja svake značajke u tekstu, algoritamsko ocjenjivanje važnosti značajki i jednostavno gomilanje svih značajki čiji se broj može popeti i iznad nekoliko stotina tisuća [33].

Nakon odabira značajki, pristupa se modeliranju klasifikacijskog procesa koji se zatim koristi za predviđanje klasa novih, neviđenih primjera. Postoje dvije osnovne grane pristupa metodama strojnog učenja, a to su nadzirane i nenadzirane metode uz mogućnost kombinacije više različitih klasifikatora u neki od algoritama meta-učenja.

[8] v. poglavlje 4.1.2.1

3.2.2. Nadzirane metode

Nadzirane metode po definiciji zahtijevaju bazu klasificiranih primjera za postupak učenja sustava. Baza tih primjera je u ovome slučaju ručno obilježen (mali) korpus nad kojim metoda nastoji generalizirati obilježene primjere u svrhu izrade funkcije ili pravila koji se mogu primijeniti na prethodno neviđenim podacima. Polazišna zamisao ovih metoda jest da obilježavanje korpusa košta manje nego ručna ekstrakcija pravila.

Nadzirane su metode popularne u području pronalaženja obavijesti za zadatak klasifikacije dokumenata u kategorije koristeći riječi dokumenata kao značajke, a također i u području crpljenja obavijesti za klasifikaciju manjih jedinica sadržaja u kategorije.

Matematički gledano, problem PKN može se reducirati na problem klasifikacije – dodjele jedne od $4n+1$ oznaka svakoj riječi, gdje je n broj kategorija naziva. Za svaku se zasebnu kategoriju može dodijeliti jedno od 4 stanja: *start* (početna riječ višerječnog naziva), *continue* (riječ između početne i završne riječi višerječnog naziva), *end* (završna riječ višerječnog naziva) i *unique* (riječ-naziv). Uz te četiri oznake, riječ se još može obilježiti i kao *other* i time označiti da nije dio naziva.

Najpoznatije metode nadziranog učenja korištene u području crpljenja obavijesti [36] su:

- Potporni vektorski strojevi (*eng. Support Vector Machines, SVM*)
  - Metoda generaliziranog linearnog klasifikatora koja pronalazi diskriminatornu funkciju između dvaju klasa. Simultano minimizira empirijsku pogrešku klasifikacije te maksimizira geometrijsku marginu (hiperravninu) koja odjeljuje pozitivne i negativne primjere. Često se koristi s jezgrenim funkcijama [25]
- Uvjetna nasumična polja (*eng. Conditional Random Fields, CRF*)
  - Statistička metoda temeljena na modelima neusmjerenih grafova. Može se smatrati generalizacijom HMM-a i ME-a, ali s prednošću da ublažava pretpostavku nezavisnosti koju HMM modeli zahtijevaju. Dodatno, zaobilazi problem pristranosti oznake, slabost koju iskazuju ME i HMM modeli [53]. CRF je metoda koja je u zadnjih par godina uzela zamaha u IE zajednici.

- Relacijsko učenje (*eng. Relational Learning*)
  - Odnosi se na sve tehnike koje uče strukturne definicije koncepta iz polaznih primjera čija je kompletna struktura klasificirana. Poznata potkategorija relacijskog učenja je induktivno logičko programiranje (*eng. Inductive Logic Programming, ILP*).
- Učenje zasnovano na transformacijama (*eng. Transformation-based Learning, TBL*)
  - Algoritam koji se ravna pogreškama (*eng. error-driven*), sastoji se od dva glavna koraka: prvo se započinje s inicijalnom klasifikacijom nad podacima te se zatim predlažu najbolja transformacijska pravila. Pravila se evaluiraju te se selektiraju promjene u klasifikaciji koje maksimalno smanjuju broj pogrešaka
- Učenje zasnovano na memoriji (*eng. Memory-based Learner, MBL*)
  - Sprema primjere za učenje i njihove pripadajuće klase i uspoređuje nove primjere sa svakim od spremljenih izračunom sličnosti ili distance te dodjeljuje klasu najsličnijih primjera novome
- Model maksimalne entropije (*eng. Maximum Entropy Models, ME*)
- Skriveni Markovljevi modeli (*eng. Hidden Markov Models, HMM*)
- Pravila i stabla odluke (*eng. Decision Rules and Trees*)

Detalji navedenih metoda strojnog učenja dostupni su u [36].

Od navedenih metoda za PKN problem su od najvećeg značaja (kako uporabnog tako i povijesnog) metode ME, HMM i stabla odluke koje će biti pobliže opisane u nastavku.

3.2.2.1 Model maksimalne entropije

Maksimalna entropija [8] je fleksibilna metoda statističkog modeliranja koja dodjeljuje vjerojatnost ishoda za svaku pojavnicu prema njenoj povijesti i aktiviranim značajkama, a onda ih poslije multiplikativno kombinira s obzirom na kontekstno okruženje. Polazišna pretpostavka metode je da svi dijelovi konteksta nezavisno pridonose konačnoj vjerojatnosti događaja. Metoda je uspješna u situacijama gdje treba kombinirati nekoliko višeznačnih izvora informacija.

Prostor mogućih ishoda (*eng. space of possible futures*) čine svi mogući rezultati modela, u ovom slučaju $4n+1$ različitih mogućih oznaka. ME izračunava vjerojatnost $p(f|h)$ za svaki ishod f iz prostora svih mogućih ishoda F i za svaku

povijest *h* iz prostora svih mogućih povijesti *H*. Povijest čine svi uvjetni podaci koji omogućavaju dodjelu vjerojatnosti prostoru ishoda. Za problem PKN, povijest se može promatrati kao informacija izvediva iz korpusa za učenje, temeljena na kontekstnom okruženju trenutne pojavnice.

Izračun *p(f|h)* u ME modelu ovisi o skupu binarnih značajki koje pomažu u predviđanju ishoda. Primjer značajke je:

$$g(h,f)=\begin{cases}1 & \text{ako trenuta\_pojavnica\_pisana\_velikim\_pocetnim\_slovom}(h)=\text{true i} \\ & f=\text{location\_start} \\ 0 & \text{inace}\end{cases} \tag{3.2}$$

Ukoliko binarna funkcija tr*enutna_pojavnica_pisana_velikim_-pocetnim_slovom* vrijedi nad trenutnom pojavnicom, ta pojavnica ima veću vjerojatnost određivanja kao početak naziva lokacije od prosječne vrijednosti.

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. Te težine zatim omogućavaju izračun uvjetne vjerojatnosti na sljedeći način:

$$P(f \mid h)=\frac{\prod_i \alpha_i^{g_i(h,f)}}{Z_\alpha(h)} \tag{3.3}$$

$$Z_\alpha(h)=\sum_f \prod_i \alpha_i^{g_i(h,f)} \tag{3.4}$$

Navedena jednadžba opisuje uvjetnu vjerojatnost ishoda s danom poviješću kao produkt težina svih značajki koje su se aktivirale na paru $\langle h,f \rangle$, normaliziran produktom težina svih mogućih značajki. Težine α_i se određuju postupkom generaliziranog iterativnog skaliranja (*eng. Generalized Iterative Scaling*, GIS). Tehnika određivanja maksimalne entropije garantira očekivanu vrijednost α_i jednaku empirijski očekivanoj vrijednosti α_i korpusa za učenje za svaku značajku f_i.

Prednost ME metode je što omogućava korisniku fokusiranje na pronalaženje značajki koje karakteriziraju problem bez uplitanja u rutinu izračuna težina [8].

Sam proces označavanja ili dekodiranja novog teksta je relativno jednostavan proces. Počinje se opojavničenjem[9] teksta nakon kojeg slijede razne predobrade poput sraza teksta i rječnika i sl. Za svaku se pojavnicu provjerava koje se sve značajke aktiviraju te se kombinira α_i vrijednost aktiviranih značajki prema jednadžbi (3.3). Naposljetku se izvršava Viterbijeva pretraga [23] koja pronalazi put najveće vjerojatnosti kroz rešetku uvjetnih vjerojatnosti koja ne proizvodi neispravnu sekvencu oznaka (da se ne bi dogodio neispravan poredak poput *location_begin organization_continue*).

3.2.2.2 Skriveni Markovljevi modeli

Osnovna zamisao skrivenih Markovljevih modela primijenjenih na PKN problem je izrada zasebnih dvopojavničkih tj. bigramskih (*eng. bigrams*) statističkih jezičnih modela za svaku pojedinu kategoriju naziva. Uz navedeno se izgrađuje i model koji predviđa kategoriju sljedećeg naziva na temelju prethodne riječi i prethodne kategorije naziva.

Jednostavna verzija HMM sustava se temelji na sljedećim dvjema jednadžbama:

$$P(NC \mid NC_{-1}, w_{-1}) = \frac{c(NC, NC_{-1}, w_{-1})}{c(NC_{-1}, w_{-1})} \tag{3.5}$$

$$P(\langle w, f \rangle \mid \langle w, f \rangle_{-1}, NC) = \frac{c(\langle w, f \rangle, \langle w, f \rangle_{-1}, NC)}{c(\langle w, f \rangle_{-1}, NC)} \tag{3.6}$$

gdje su:

NC ≡ trenutna kategorija naziva

NC_{-x} ≡ kategorija naziva x-te prethodne riječi

$c(W)$ ≡ broj koliko puta se događaj W pojavio u korpusu za treniranje

[9] v. poglavlje 0

$w \equiv$ riječ

$f \equiv$ značajka

Jednadžbe opisuju model koji predviđa kategoriju naziva ako mu je dana kategorija prethodnog naziva i prethodna riječ, te model koji predviđa idući bigram riječ-značajka ako mu je dan prethodni bigram riječ-značajka i trenutna kategorija naziva. Zatim se radi Viterbijeva pretraga za sekvencom kategorija naziva koja bi dodijelila najveću vjerojatnost korpusu za učenje.

Značajke u HMM sustavu su uobičajene, npr. je li riječ pisana velikim početnim slovom, pripada li početku rečenice, je li član neke od listi (imena, prezimena, lokacija...) [7].

Primjer

Uz odgovarajući bi se korpus mogla održati i sljedeća nejednakost:

$$P(\langle antić, VPS\rangle \mid \langle ante, VPS\rangle_{-1}, organizacija) < P(\langle antić, VPS\rangle \mid \langle ante, VPS\rangle_{-1}, osoba)$$ [10]

Na kraju, može se očekivati da bi Viterbijeva pretraga odabrala kategoriju „osoba“ spram kategorije „organizacija“.

U idealnom bi slučaju bio dostupan dovoljno veliki korpus za treniranje svakog događaja čiju je uvjetnu vjerojatnost potrebno izračunati te dovoljno uzoraka za samo izračunavanje. Međutim, to u stvarnosti nije moguće ostvariti pa se pojavljuju problemi s nepoznatim riječima.

Problem nepoznatih riječi (prethodna, trenutna ili obje riječi nepoznate) rješava se modelima ustručavanja (*eng. back-off model*) koji impliciraju odvojeno učenje nepoznatih modela riječi iz dijela korpusa za zaglađivanje (*eng. smoothing*) ili učenje nepoznatih riječi. Ukoliko model naiđe na nepoznatu riječ, može ustuknuti

[10] VPS – veliko početno slovo

pred manje moćnim i manje deskriptivnim modelima koji bi mogli razriješiti problem.

Primjer sustava temeljenog na HMM-u je BBN-ov IdentiFinder koji je osvojio treće mjesto na konferenciji MUC-7 s F-mjerom od 90,44%.

3.2.2.3 Stabla odlučivanja

Pojednostavljeno gledano, stablo odlučivanja (*eng. Decision Tree*) sastoji se od tri osnovna elementa:

- Budućnost
  - jednako kao kod modela maksimalne entropije: mogući rezultati modela stabla odlučivanja, *4n+1* različitih mogućih oznaka koje tvore prostor budućnosti.
- Prošlost
  - informacije dostupne modelu, dobivene npr. analizom prethodne, trenutne i sljedeće riječi, međutim ne postoji razlog da širina promatrane prošlosti ne bude veća [8]
- Pitanja
  - srž algoritma – cilj algoritma je pronaći najbolju sekvencu pitanja koja se postavljaju o prošlosti u svrhu određivanja budućnosti. Valja obratiti pozornost da je pri određivanju te sekvence vjerojatnost postavljanja *m*-tog pitanja određena odgovorima na prethodnih *m-1* pitanja

Sustav pri analizi konteksta svake pojavnice dodjeljuje vjerojatnosti supojavljivanja s ostalima, uzimajući u obzir kontekst susjednih pojavnica. Pri izračunu vjerojatnosti za svaku se pojavnicu testira vjerojatnost supojavljivanja sa susjednom i onom prije susjedne. Također se mogu uzimati i ostale osobine pojavnica kao što je na primjer veliko/malo slovo.

Postavlja se pitanje kako efikasno izgraditi stabla koja su vremenski i memorijski učinkovita. Osnovna je ideja 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:

$$\mathbf{W} = \{\text{Svi moguci odgovori na pitanje w}\} \tag{3.7}$$

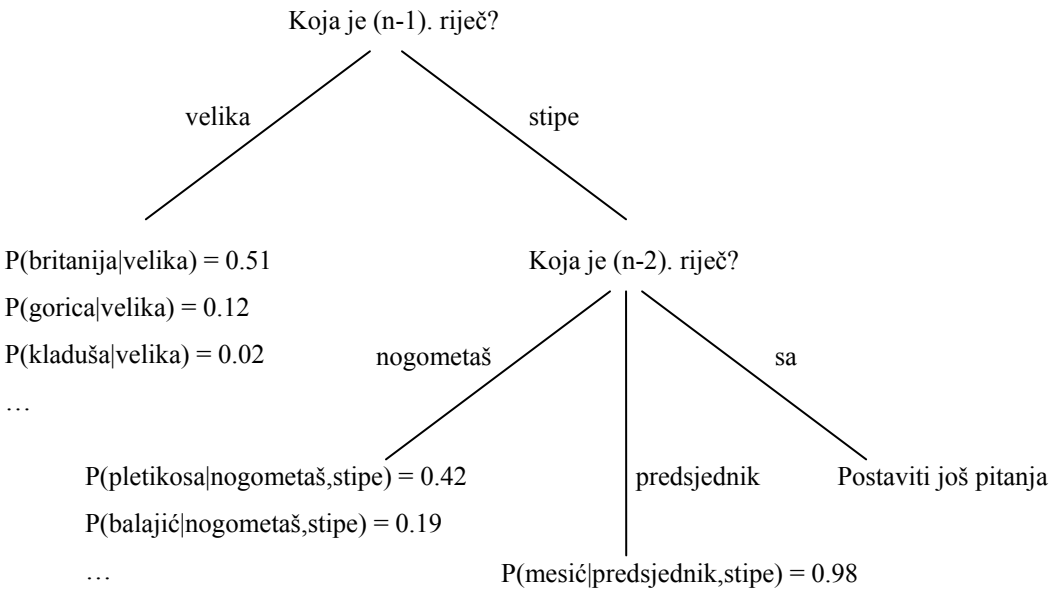
$$H(F \mid W) = -\sum_{q \in \mathbf{W}} P(q) \sum_{f \in \mathbf{F}} P(f \mid q) \log P(f \mid q) \tag{3.8}$$

Problem s ovom metodom je što se u traženju najboljeg pitanja u bilo kojem čvoru stabla žele odabrati pitanja *W* za koja je |**W**| velik s obzirom da generalno takva pitanja vode nižim vrijednostima uvjetne entropije *H(F|W)*. Međutim, s računalnog stajališta želimo zaobići nepotrebne fragmentacije podataka tako da preferiramo manji *W*. Rješenje problema je postavljati samo binarna (da-ne) pitanja. S ciljem minimiziranja *H(F|W),* traže se pitanja koja će prostor značajki ugrubo podijeliti na pola između da i ne pitanja.

Izgradnja stabla selektiranjem pitanja koja vode najvećoj redukciji uvjetne entropije dobro je poznata tehnika. Kritični zahtjev je izgradnja stabla s dovoljno bogatom prošlošću koja omogućava ispitivanje serije informativnih pitanja koja bi reducirala neizvjesnost o polju značajki.

Kritično pitanje je pružiti stablo s dovoljno bogatom prošlošću tako da može pitati seriju informativnih pitanja koja mogu reducirati neizvjesnost o polju značajki. Jednom izgrađeno, stablo odluke se iznimno lako koristi (Slika 3.):

Primjer



Slika 3. Primjer dijela izgrađenog stabla odluke

Kada su postavljena pitanja riješena i kada se obavi obilazak od korijena do listova stabla, uzima se distribucija vjerojatnosti koja je pohranjena u listu čvora kao $P(v_i|v_{i-2}v_{i-1})$.

Sekine u svom poznatom modelu stabla odluke [40] upotrebljava samo tri osobine: POS, osobine riječi i popise imena.

3.2.3. Nenadzirane metode

Za razliku od nadziranih metoda kojima je potreban obilježeni korpus za uvježbavanje, nenadzirane metode pokušavaju pronaći uzorke za prepoznavanje klasa u neobilježenom korpusu. Osnovna polazišna misao ovih metoda je da je ručno označavanje korpusa usko grlo nadziranih metoda strojnog učenja zbog cijene izrade i općenite nedostupnosti većih količina istih. S druge se pak strane može vrlo lako doći do velikih količina neobilježenih tekstova koji su pritom i znatno jeftiniji.

Nenadzirane metode čine tzv. slabo nadzirane[11] (*eng. weakly supervised*) i potpuno nenadzirane metode. Razlika između navedenih je u tome da slabo nadzirane metode uče iz ograničenog početnog skupa obilježenih podataka (*eng. seeds*) i velike količine neobilježenih na kojima sustav inkrementalno povećava svoje performanse do postizanja adekvatne točnosti na testnom korpusu, dok se potpuno nenadzirane metode[12] oslanjaju isključivo na neobilježene podatke.

Predstavnici slabo nadziranih metoda su:

- Ekspanzija (*eng. Expansion*)
  - iterativna ekspanzija skupa za učenje sličnim primjerima
- Samotreniranje (*eng. Self-training*)
  - inkrementalno učenje klasifikatora temeljenog na početnom skupu za učenje i skupu neobilježenih primjera koji su obilježeni trenutnim klasifikatorom

[11] tj. polu-nadzirane (*eng. semi-supervised*)

[12] Jedna od potpuno nenadziranih metoda je i klasteriranje (*eng. clustering*). Riječ je o metodi koja pronalazi slične uzorke i skuplja ih u grozdove (*eng. clusters*) ili grupe iz kojih se izvlače korisni zaključci od sličnostima i razlikama uzoraka. Međutim ta se metoda rijetko koristi za rješavanje problema PKN, a više za druge probleme crpljenja obavijesti poput razrješavanja koreferencija.

- Ko-treniranje (*eng. Co-training*)
  - dva ili više klasifikatora se treniraju koristeći isti skup za učenje, ali s nezavisnim skupovima značajki
- Aktivno učenje (*eng. Active learning*)
  - sustav pažljivo odabire primjere koji nose najviše informacija o klasifikaciji koje zatim čovjek klasificira

Navedeni sustavi su predstavnici slabo nadziranih metoda koje se oslanjaju na tehnologiju samonadopunjavanja.

3.2.3.1 Samonadopunjujući pristup

Samonadopunjavanje (*eng. Bootstrapping*) [1] je princip koji se odnosi na samoodržavajuću tehnologiju koja se oslanja na vlastite metode i resurse[13]. To je tehnologija koja započinje s inicijalno malim skupom primjera te postepeno raste u veći i značajniji sustav i/ili skup podataka.

Konkretno u slučaju sustava za PKN, samonadopunjavanje je iterativni proces iskorištavanja malog skupa inicijalnih naziva u svrhu dobivanja novih [28].

- Cilj pristupa:
  - iskoristiti minimalnu količinu nadziranih primjera
  - steći znanje iz mnoštva neobilježenih primjera
- Generalni plan:
  - inicijalizirati početni skup primjera za treniranje inicijalnog modela
  - klasificirati korpus s inicijalnim modelom
  - dodati najpouzdanije klasificirane primjere korpusu za učenje i iterirati

Pristup samonadopunjavanja udružen s nekoliko početnih pravila i popisa poznatih imena funkcionira tako da započinje srazom popisa imena i teksta prikupljajući informacije o kontekstnoj okolini naziva [4]. Analizom konteksta naziva sustav izvodi pravila u kojima se određena kategorija naziva pojavljuje.

[13] http://encarta.msn.com/dictionary_1861591886/bootstrap.html

Pravila se ponovno primjenjuju na korpus, a rezultat je povećan popis imena koji se ponovno koristi za sraz[14] s tekstom i tako iterativno dok se ne zadovolji neki od kriterija za zaustavljanje.

Primjer

Prikazani primjer je adaptacija primjera iz [4].

Pod pretpostavkom da se na popisu vlastitih imena za hrvatski sustav za PKN nalazi vlastito ime *Ivan*, sustav srazom popisa i korpusa nailazi na rečenicu:

Ivan Kljajić je uspješno završio drugu etapu utrke.

Na osnovu jednostavnog pravila *VLASTITO_IME VELIKO_SLOVO*, sustav zaključuje da je *Kljajić* prezime te ga pohranjuje u popis prezimena i ponovno obavlja sraz s tekstom koristeći novostečeno znanje. Pronalaskom rečenice:

Generalni direktor Poznate Tvrtke Marko Kljajić jučer je dao otkaz.

pravilo *VELIKO_SLOVO PREZIME* daje signal sustavu da je pojavnica *Marko* ime koje se dodaje u popis. Nakon ponovnog sraza teksta i popisa, sustav nailazi na rečenicu:

Dr. Marko Perić jučer je predstavio svoju teoriju ljudskog ponašanja…

Sustav sada „primjećuje“ da se u korpusu često uz vlastito ime i prezime pojavljuje titula *Dr.* te izvodi pravilo *Dr. VELIKO_SLOVO VELIKO_SLOVO*. Novostečeno pravilo se zatim ponovno koristi u srazu s tekstom za ekstrakciju novih naziva i novih pravila.

Samonadopunjavanje je moguće koristiti i uz neku drugu metodu, inače metodu nadziranog učenja koja se oslanja na samonadopunjavanje – početnim popisom se obilježi tekst nad kojim se onda vrši učenje te se naučeni klasifikator ponovno koristi za obilježavanje teksta nad kojim se algoritam ponovno uči, i tako

[14] Usporedba i označavanje

iterativno (načelo samotreniranja). Jedan od takvih primjera je dan u [52] gdje se samonadopunjavanje koristi zajedno s algoritmom Ripper [13] u svrhu slabo nadgledane ekstrakcije popisa geografskih lokacija s Interneta, koristeći tražilicu AltaVista.

Iz navedenoga se lako primjećuje da pristup samonadopunjavanja ne zahtijeva velike količine ugrađenog znanja, štoviše, pristup je siromašan znanjem (*eng. knowledge-poor*), domenski neovisan (*eng. domain independent*), pa i jezično neovisan (*eng. language independent*) te se zato često koristi za prikupljanje nepoznatih činjenica iz teksta.

3.2.4. Meta-učenje

Postoji još jedan popularan pristup pod nazivom meta-učenje [39] (*eng. meta-learning*) koji se zasniva na ideji kombinacije više klasifikatora i/ili posebnom učenju jednoga na način da se sam algoritam višestruko primjenjuje na različite podskupove korpusa za učenje. Glavna prednost ovih metoda je mogućnost da se slabi klasifikator pretvori u jaki ili odlični. Glavne tri metode meta-učenja su:

- Samonadopunjavajuće gomilanje (*eng. Bagging, Bootstrap aggregation*)
  - Korpus za treniranje se podijeli nekoliko puta koristeći samonadopunjivanje tako da se slabi klasifikator trenira na svakom od dijelova. Za konačnu klasifikaciju se koriste težinske kombinacije različitih predikcija. Radi najbolje s nestabilnim klasifikatorima – slabi klasifikatori s visokom varijancom. Nije toliko popularan pristup za rješavanje problema PKN.
- Slaganje (*eng. Stacking*)
  - Kombinacija višestrukih klasifikatora slaganjem klasifikatora jednog na drugi ili njihovo kombinirano težinsko korištenje – za svaki se klasifikator određuje težina tako da se minimizira prosječna *leave-one-out* među-validacijska (*eng. cross-validation*) greška. Popularan pristup za rješavanje problema PKN [49].
- Pojačavanje (*eng. Boosting*)
  - Slabi klasifikator se uči na podacima za učenje kroz nekoliko samonadopunjujućih rundi te postavlja težine na primjere za učenje. Oni primjeri koji se teže uče dobivaju veće težine, dok lakši primjeri dobivaju manje. Svrha je da se klasifikator „koncentrira“ na primjere koje je teško klasificirati, dok se jednostavni problemi jednostavno rješavaju, ulaganjem manje „truda“. Jedan od također popularnih pristupa. Ovoj grupi pripada često korišten AdaBoost algoritam [10].

3.2.5. Osobine sustava zasnovanih na metodama strojnog učenja

Osobine sustava zasnovanih na strojnom učenju su:

- Zahtijevaju veliku količinu tekstova,
  - Obilježenih tekstova (nadzirane metode) – potreban je ljudski rad, a uz to obilježeni tekstovi mogu sadržavati pogreške zbog ljudskog čimbenika. Takve pogreške mogu bitno narušiti učinkovitost sustava, a često ih je teško pronaći u korpusu za uvježbavanje
  - Neobilježenih tekstova (nenadzirane metode) – velike količine neobilježenih tekstova su lakše dostupne i ne zahtijevaju ulaganje ljudskog rada. Međutim takve metode upravo zbog neobilježenosti teksta u pravilu postižu slabije rezultate od nenadziranih
  - Ostaje otvoreno pitanje koliko velik korpus mora biti
- Autori sustava ne moraju biti lingvisti čime se smanjuje cijena izrade sustava
- Potrebno je kraće vrijeme za razvoj sustava. Sustav sam zaključuje o nazivima i/ili pravilima.
- Lakša je adaptacija sustava za prepoznavanje tekstova iz druge domene, uz uvjet da postoji velika količina (obilježenih ili neobilježenih) tekstova u korpusu za uvježbavanje, gdje sustav ima dovoljno primjera iz više domena.
- Lakša adaptacija sustava za prepoznavanje tekstova pisanih drugim jezicima uz uvjet da postoji adekvatan korpus i da je sustav dovoljno konfigurabilan
- Čitljivost, odnosno preglednost pravila sustava je manja, ponekad nikakva, zavisno o odabranoj metodi. Pravila sustava su najčešće sadržana u matričnom i/ili numeričkom obliku koji za čovjeka nije čitljiv.

3.3. Hibridni sustavi

Hibridni sustavi su sustavi koji objedinjuju sustave zasnovane na pravilima i sustave zasnovane na metodama strojnog učenja. Pozitivna osobina modularno oblikovanih sustava (kako onih zasnovanih na pravilima tako i onih zasnovanih na metodama strojnog učenja) jest mogućnost međusobne kombinacije u svrhu

povećanja učinkovitosti. Takvi sustavi su sustavi slabije integracije koji često funkcioniraju tako da izlaz jednog postaje ulaz drugog sustava. Primjer takvog hibridnog sustava je *MENE+Proteus* sustav[15]. Druga mogućnost jest izrada usko povezanog (*eng. tightly integrated*) hibridnog sustava u kojemu se ne mogu jasno razdvojiti sastavni sustavi. Primjer takvog sustava je LTG sustav[16]. Također su moguće i višestruke kombinacije različitih klasifikatora poput ME, HMM i sustava zasnovanog na pravilima, poput sustava opisanog u [42] koji postiže F-mjeru od 0,89 na službenom testnom korpusu s konferencije MUC-7.

Praksa je pokazala da hibridni sustavi temeljeni na pravilima i metodama strojnog učenja daju najbolje rezultate.

[15] v. poglavlje 4.1.2.1
[16] v. poglavlje 4.1.3

4. Primjeri sustava za prepoznavanje naziva

U ovom poglavlju dano je nekoliko općepoznatih primjera sustava za PKN za engleski jezik, opisan je za sada jedini sustav za PKN za hrvatski jezik, više takvih sustava za razne jezike, te nekoliko obećavajućih sustava za višejezično i jezično neovisno PKN.

4.1. Engleski jezik

Za engleski je jezik izgrađeno najviše sustava za PKN zbog opće prihvaćenosti toga jezika kao internacionalnog jezika te zbog dostupnosti velikih obilježenih korpusa[17]. Opisani su najpoznatiji/najznačajniji sustavi za PKN uz dodatak jednog sustava za automatsku ekstrakciju naziva.

4.1.1. IsoQuest

IsoQuest [29] je sustav zasnovan na pravilima namjenski razvijen za konferencije MUC-a na osnovi komercijalne aplikacije NetOwl[18] s 90% gotovih pravila komercijalnog alata, a 10% namjenski izrađeno. Koristi se kontekstnim informacijama kao i leksikonom kada nema dovoljno dokaza za klasifikaciju iz teksta. Sustav vodi računa o identifikaciji aliasa zamjenskih imena, akronima ili skraćenih inačica punog naziva. Problem višeznačnosti naziva i problem postojanja više pravila koja obavljaju klasifikaciju istog naziva u različite kategorije sustav rješava fazom nadmetanja pravila (*eng. rule competition phase*). U toj se fazi oslanja na brojčane težinske vrijednosti pridodane svakom pravilu. Iznimke i specifične slučajeve koji prema MUC specifikaciji nisu nazivi, a pravilima sustava bi bili prepoznati, rješava odvojeno eksplicitnim navođenjem iznimaka u tablici.

[17] MUC korpusi, CoNNL korpusi, i dr.
[18] http://www.netowl.com/

IsoQuest sustav je na konferenciji MUC-7 osvojio drugo mjesto s F-mjerom od 91,6%.

4.1.2. MENE

Maximum Entropy Named Entity ili MENE [7] je PKN sustav Njujorškog sveučilišta (*New York University*, NYU) čija se arhitektura sastoji od C++ i Perl modula koji služe kao omotači (*eng. wrappers*) oko javno dostupnog ME alata MEMT (*Maximum Entropy Modeling Toolkit*) za izračunavanje α vrijednosti iz jednadžbe (3.3) iz para datoteka kreiranih MENE sustavom. Algoritam učenja MENE sustava sažet je na slici 4. [8]

1. Definiraj korpus za učenje C
2. Opojavniči ga
3. Izradi datoteku kandidata značajki, uključujući leksičke značajke izrađene iz korpusa
4. Izvrši predizračun informacija iz datoteke pojavnica za statistiku prošlosti
5. Odredi koliko je puta pojedina značajka g_i aktivirana na C ($\#g_i$)
6. Ukloni značajke kojima je $\#g_i<m$
7. Izradi datoteku „očekivanja“ s očekivanim vrijednostima K_i za sve značajke g_i nad korpusom za treniranje $K_i = \frac{\# g_i}{|C|}$
8. Izradi datoteku „događaja“ koja sadrži sve značajke koje se aktiviraju za svaki par $\langle h, f \rangle$ za sve $h \in C$ i sve f
9. Izračunaj sve težine α_i za svaku značajku g_i koristeći MEMT toolkit s datotekama očekivanja i događaja kao ulaznim podacima.

Slika 4. Algoritam za učenje MENE sustava

Postignuta je velika fleksibilnost sustava zahvaljujući činjenici da može iskoristiti gotovo bilo kakvu binarnu funkciju za značajku kao funkciju povijesti i ishoda trenutne pojavnice. MENE koristi sljedeće vrste značajki:

- Leksičke značajke (*eng. Lexical Features*)
- Sekcijske značajke (*eng. Section Features*)
- Značajke rječnika (*eng. Dictionary Features*)
- Značajke vanjskih sustava (*eng. External Systems Features*)

Iz algoritma prikazanog na slici 4. jasno se vidi da je odabir značajki realiziran postavljanjem praga po broju aktiviranja značajki. Prag je u slučaju učenja MENE sustava na korpusu za učenje postavljen na tri. Algoritam odabira značajki funkcionira u potpunosti bez ljudske intervencije.

Proces označavanja je jednostavan te je identičan procesu objašnjenom u poglavlju 3.2.2.1.

F-mjera MENE sustava učenog na službenom korpusu za učenje konferencije MUC-7, primijenjenog na službenom korpusu za testiranje iznosi 84,22%, preciznosti 0,91 i odziva 0,78. Utvrđeno je da je samome sustavu potrebno barem 20 članaka za postizanje F-mjere od 80,97%, dok u sprezi s drugim sustavima količina istih varira o jačini sustava na koji se MENE veže [8]. Na konferenciji MUC-7, MENE je nastupao u hibridnom modelu s Proteus sustavom.

4.1.2.1 MENE+Proteus

Hibridni sustav nastao na Njujorškom sveučilištu, službeni natjecatelj istog sveučilišta na konferenciji MUC-7. Sustav MENE u ovom slučaju preuzima izlaz bitno naprednije inačice sustava Proteus zasnovanog na pravilima te ga koristi u sklopu posebno definiranih značajki vanjskih sustava (*eng. External System Features*) tako da može mijenjati izlaz početne aplikacije. Interesantno je da je MENE sustav uočio i ispravio (premostio) prirođenu pogrešku Proteus sustava koji je imao tendenciju odsijecanja višerječnih naziva na prvoj riječi [7]. Može se zaključiti da, uz dobre podatke za učenje, MENE može odrediti i ispraviti slabosti ručno kodiranih sustava. Sustav je zauzeo četvrto mjesto na konferenciji MUC-7 s F-mjerom od 88,8%.

4.1.3. LTG

LTG [34] je hibridni sustav za PKN koji objedinjuje ručno kodirana pravila i model maksimalne entropije. Temelj sustava čine modularni alati razvijani tijekom nekoliko godina unutar *Language Technology Group* iz Edinburgha.

Ključna karakteristika sustava jest da se procesiranje entiteta odvija u fazama. Prepoznavanje TIMEX i NUMEX entiteta odvija se odvojeno od ENAMEX entiteta zbog veće strukturiranosti prvih pa ih je jednostavnije obilježiti koristeći gramatička pravila.

ENAMEX entiteti se procesiraju u sljedećim fazama:

1. Pravila sigurnog okidanja (*eng. Sure-fire Rules*)

   Skup pravila koja se oslanjaju na kontekst te se primjenjuju u slučaju sigurnog i sugestivnog konteksta[19]. Pronađeni se nazivi tretiraju kao mogući, a ne kao definitivni. Odgađa se označavanje organizacija s konjunkcijskim veznikom „i“ i organizacija na početku rečenice. Imena lokacija u popisu imena se obilježavaju samo ako su u kontekstu koji čvrsto sugerira lokaciju (visoka preciznost od 0,98, nizak odziv od 0,42).

2. Djelomično slaganje 1 (*eng. Partial Match 1*)

   Generiranje mogućih kraćih inačica naziva čuvajući poredak riječi. Skup slabijih pravila koji se šalju preduvježbanom ME modelu koji ocjenjuje djelomično slaganje te donosi odluku o obilježavanju na temelju informacija o kapitalizaciji, činjenici je li riječ identificirana kao naziv negdje drugdje u tekstu, položaja riječi u rečenici i dr.

3. Olabavljivanje pravila (*eng. Rule Relaxation*)

   Korištenje olabavljenih pravila niže preciznosti koja se koriste zajedno s popisima (popisi se tek sada koriste!) lokacija, organizacija i imena, koristi se

[19] Signali za tvrtke, titule za osobe, sigurni konteksti temeljeni na POS i jednostavnom semantičkom obilježavanju

gramatika imena za obilježavanje nizova pojavnica koji izgledaju kao imena. Razrješava se problem imena organizacije s konjukcijskim veznikom „i“ kao i pripadnost modifikatora koji se nalazi na početku rečenice koji može, a ne mora pripadati složenom nazivu (npr. *Suspended Ceiling Contractors Ltd.)*. Klasifikacijski nejasni nazivi gdje kontekst ne pomaže.

4. Djelomično slaganje 2 (*eng. Partial Match 2*)

   Još jedan ME korak sličan prethodnom – kada su iscrpljeni resursi o unutarnjim[20] i vanjskim[21] dokazima kao i popisi imena, obilježavaju se nejasni slučajevi, razrješava se interentitetska konjunkcija tako da je se tretira kao da nije konjunkcija, a obrađeni se podaci u završnici potpomažu s ME

5. Dodjeljivanje naslovima (*eng. Title Assignment*)

   Zadnja ME faza upravlja prepoznavanjem i klasifikacijom naziva pronađenih u naslovima dokumenata[22] na temelju prepoznatih naziva u tijelu i podnaslovu teksta.

LTG sustav je pobjednik konferencije MUC-7 s F-mjerom od 93,39%.

Analizirajući sustav, Borthwick u [8] raspravlja o performansama sustava navodeći da je nejasno dolazi li prednost sustava iz statističkih ili iz faza primjene pravila. Arhitektura sustava navodi na mogućnost da je za prednost pred drugim sustavima odgovorna jača integracija spomenutih faza.

4.1.4. KnowItAll

Dijelovi sustava za PKN, kao i odvojeni sustavi mogu funkcionirati kao sustavi za ekstrakciju novih naziva koji se zatim mogu spremati u popise naziva.

[20] Unutarnji dokaz se izvodi iz niza riječi koje obuhvaćaju naziv, npr. „d.d.“, „O.Š.“ i dr.
[21] Vanjski dokaz je klasifikacijski kriterij pribavljen kontekstom u kojem se naziv pojavljuje, npr. „zaposlenik tvrtke Pliva“ gdje riječ „tvrtke“ upućuje na kategoriju organizacija
[22] Naslovi dokumenata su najčešće pisani velikim slovima (verzalima) te stoga pružaju malo obavijesti za prepoznavanje

Automatizirani proces ekstrakcije većih kolekcija novih naziva zapravo je cilj implementacije sustava u ovom radu, pa je stoga opisan i jedan takav sustav za automatsku ekstrakciju, KnowItAll.

KnowItAll [22] je sustav za automatsku ekstrakciju velikih količina naziva (u daljnjem tekstu činjenica) s Interneta na nenadgledan, domenski neovisan i skalabilan način. Primjeri naziva koje je moguće ekstrahirati ovakvim sustavom su npr. popisi političara, gradova, glumaca i dr. Ekstrahiranim se kandidatima automatski testira vjerojatnost koristeći PMI statistiku računatu tretiranjem WWW-a kao masivnog tekstnog korpusa. Sustav asocira vjerojatnost svakoj ekstrahiranoj činjenici omogućavajući time balansiranje između preciznosti i odziva.

Jedini ulaz u KnowItAll sustav su predikati, točnije skup predikata (unarnih i n-arnih) koji određuje fokus sustava. Primjer jednog takvog predikata koji pronalazi glavne izvršne direktore (*eng. Chief Executive Officer, CEO*) tvrtki je prikazan na slici 5.

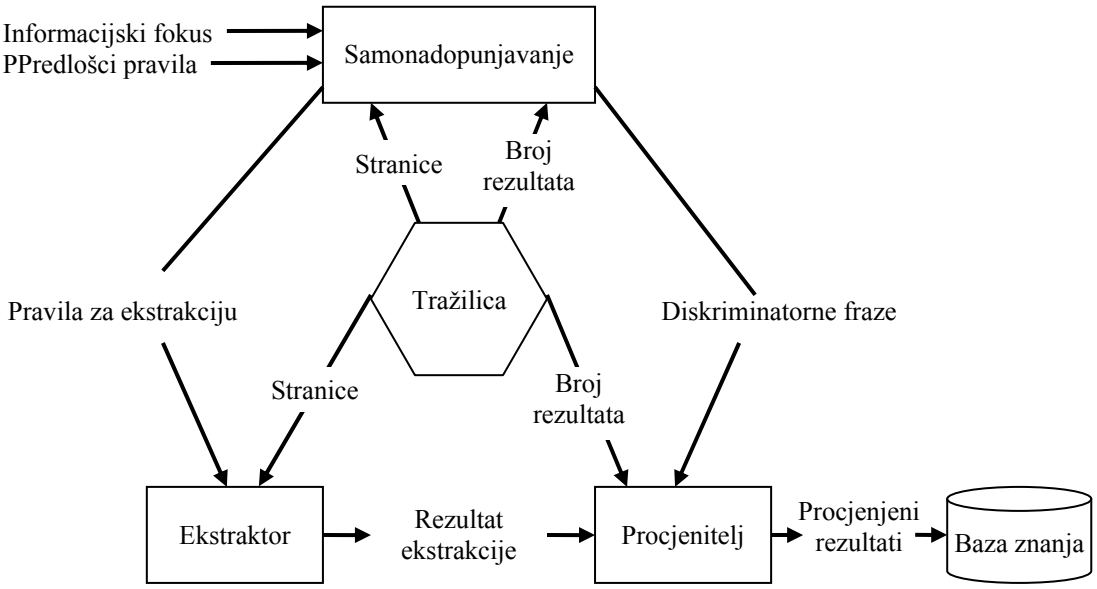
Predikat:	CeoOf(Person, Company)
Uzorak:	NP1 „„“ P2 NP3
Ograničenja:	properNoun(NP1)
	P2=“CEO of“
Povezivanja:	CeoOf(NP1, NP3)
Ključne riječi:	„CEO of“ NP3

Slika 5. Primjer pravila za ekstrakciju

Sustav koristi princip samonadopunjavanja na temelju domenski neovisnih uzoraka za ekstrakciju kako bi za svaki od predikata u fokusu kreirao ekstrakcijska pravila i diskriminatorne fraze.

Prema tvrdnjama autora [22], procedura samonadopunjavanja je u potpunosti automatski realizirana što je u kontrastu s ostalim metodama koje zahtijevaju skup ručno kreiranih primjera, međutim vidljivo je da je kao početni uzorak ipak potreban skup predikata.

Dva glavna modula sustava su ekstraktor (*eng. Extractor*) i procjenitelj (*eng. Assessor*). Ekstraktor stvara upite iz ključnih riječi svakog pravila, šalje ih tražilici (trenutačno korišten Google API) i primjenjuje pravila za ekstrakciju informacija s rezultirajućih WWW stranica. Procjenitelj izračunava vjerojatnost korektnosti činjenice te vrši filtriranje prema vjerojatnostima prije dodavanja u bazu znanja. Ovaj modul temelji svoje izračunavanje vjerojatnosti na broju rezultata koje vraća korištena Internet tražilica kako bi izračunao PMI između ekstrahirane činjenice i skupa automatski generiranih diskriminatornih fraza koje su vezane na klasu činjenice.



Slika 6. Shema sustava KnowItAll

Prema shemi sustava (Slika 6.), Algoritam samonadopunjavanja stvara pravila za ekstrakciju i diskriminatorne fraze za svaki predikat u fokusu. Zatim se stvara lista upita prema tražilicama koji su asocirani s pravilima za ekstrakciju te izvršava glavna petlja. Na početku petlje sustav odabire upite, dajući prednost predikatima i pravilima koji su bili najproduktivniji u prethodnoj iteraciji glavne petlje. Ekstraktor šalje odabrane upite tražilici, ekstrahira informacije iz dobivenih WWW stranica te tako dobivene kandidate za činjenice šalje procjenitelju. Procjenitelj izračunava vjerojatnost korektnosti svake ekstrakcije te ju dodaje u bazu

podataka. Petlja se ponavlja dok nisu izvršeni svi upiti ili dok se ekstrakcija novih informacija ne ocijeni neproduktivnom.

Preciznost osnovnog sustava za kategoriju grad iznosi 0,83, za film 0,59 za kategoriju znanstvenik 0,77. Navedene preciznosti variraju zavisno o uporabi dodatnih algoritama u radu [22].

4.2. Hrvatski jezik

Tadić u svom radu [45] iz 2000. godine piše kako razvijenost jezičnih alata i resursa za hrvatski jezik nažalost nije na zadovoljavajućoj razini. Međutim, od objavljivanja ove tvrdnje do danas, dosta je napravljeno u tom području. Od važnosti za ovaj rad valja spomenuti nedavno (2005. godine) izrađen sustav za automatsko prepoznavanje i klasifikaciju naziva: OZANA.

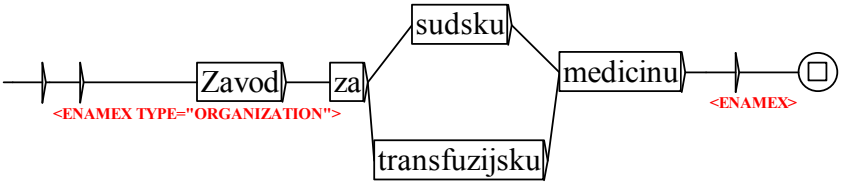
4.2.1. OZANA

OZnAčivač NAziva ili OZANA [4] je sustav za strojno prepoznavanje i klasifikaciju naziva za hrvatski jezik zasnovan na pravilima, izrađen na Filozofskom fakultetu u Zagrebu, autora Bože Bekavca. Sustav je izrađen u *Intex/Unitex*[23] razvojnom okruženju, jednom od računalnojezikoslovnih alata koji je odabran na temelju niza prednosti pred drugim alatima, poput grafičkog sučelja za izradu pravila putem grafova, jednostavnosti, pouzdanosti i brzine obrade.

Srž sustava je temeljena na lokalnim gramatikama (*eng. local grammars*). Lokalne gramatike su konačni transduktori (Slika 7.) koji opisuju ispravne (*eng. well-formed*) nizove u tekstu i za njih odabiru odgovarajuće oznake. To je ujedno i osnovna razlika spram konačnih automata – konačni transduktori imaju skup izlaznih znakova i funkciju prijelaza koja omogućava ispisivanje izlaznih znakova, čime je omogućeno preoblikovanje teksta umetanjem oznaka (izlaznih znakova) u tekst.

[23] http://intex.univ-fcomte.fr/INTEX.htm

Uporabom lokalnih gramatika, cilj je definirati i primijeniti leksičke uvjete u lokalnom okruženju koje sadrži niz od nekoliko riječi.



Slika 7 . Primjer konačnog transduktora [4]

Sustav se ugrubo može podijeliti na dva dijela: dio za predobradu i dio za prepoznavanje i klasifikaciju naziva.

4.2.1.1 Predobrada

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

1. Opojavničenje
   - rastavljanje teksta na pojavnice[24]
2. Segmentacija na rečenice
   - izrađeni modul za segmentaciju implementira 7 pravila dovoljnih za točnost veću od 99% (izmjereno na tekstovima Večernjeg lista iz travnja 2003. godine)
3. Leksička obrada

   a. Leksički resursi
   - Opći leksikon, Hrvatski morfološki leksikon (HML) [46] s oko 2.126.086 oblika riječi generiranih iz oko 33.500 lema
   - Automatsko prepoznavanje brojeva – potencijalne inačice brojeva u padežnim oblicima
   - Automatsko prepoznavanje pridjeva u genitivu – važno jer su često sastavni dio imena organizacija

[24] v. poglavlje 0

b. Popisi imen

- Vlastita imena osoba - Leksička flektivna baza hrvatskih imena i prezimena [6], prikupljena iz javnih izvora te visoko učestala strana vlastita imena osoba prikupljena iz kraćih izdvojenih popisa s padežnim oblicima generiranim pomoću *Perl* skripti
- Lokacije (*eng. gazetteer*) - prikupljen iz javnih izvora podataka putem Interneta s padežnim oblicima generiranim pomoću *Perl* skripti

Navedene akcije se obavljaju bez razrješenja mogućih višeznačnosti.

4.2.1.2 Prepoznavanje i klasifikacija naziva

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 transduktora određenim redoslijedom [4]. Modul se sastoji od predmodula i tri podmodula [4], [5]:

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

Rezultat obrade su tekstovi u XML obliku s obilježenim nazivima prema specifikacijama s konferencije MUC-7.

4.2.1.3 Rezultati obrade

Prije samih rezultata, navedeni su korišteni korpusi pri izradi sustava:

- Korpus za uvježbavanje
  - Tekstovi Večernjeg lista (Hrvatski nacionalni korpus [44]), od 1999. do 2003. godine opsega 45.563.824 pojavnica, i tekstovi Vjesnika od 2001. do 2003. godine, opsega 15.193.749 pojavnica
- Korpus za testiranje
  - Tekstovi istih dnevnih listova iz 2004. godine opsega 9.932.498 pojavnica
- Korpus za vrednovanje
  - Tekstovi istih dnevnih listova iz siječnja 2005.

Pravila sustava primijenjena na korpus za vrednovanje daju sljedeće rezultate [4]:

Tablica 1. Rezultati obrade na korpusu za vrednovanje

	Osobe	**Organizacije**	**Lokacije**	**Postotci**	**Valute**	**Datumi**
Preciznost	0,95	0,93	0,98	0,99	0,99	0,94
Odziv	0,69	0,86	0,93	0,99	0,99	0,90
F-mjera	0,79	0,89	0,95	0,99	0,99	0,92

Prosječna F-mjera sustava iznosi 0,92. 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 0,90.

Na neinformativnim tekstovima, učinkovitost sustava pada. Analiza na proznim tekstovima[25] daje sljedeće rezultate [5]:

[25] Viktor Žmegač, (1998), Bečka moderna
Zvonko Maković, (1997), Vilko Gecan

Tablica 2 . Rezultati obrade na neinformativnim tekstovima

	Osobe	**Organizacije**	**Lokacije**	**Postotci**	**Valute**	**Datumi**
Preciznost	0,65	0,69	0,61	0,95	0,92	0,91
Odziv	0,35	0,38	0,31	0,66	0,61	0,53
F-mjera	0,46	0,49	0,41	0,78	0,73	0,67

4.3. Drugi jezici

U nastavku su navedeni primjeri PKN sustava za 16 jezika uz pripadajuću referencu na članak u kojemu je sustav pobliže opisan. Navedene su samo osnovne značajke sustava i mjera uspješnosti (ako je dostupna).

- Arapski [31]
  - podudaranje uzoraka (*eng. pattern matching*) i morfološka analiza, F-mjera 0,85
- Bugarski [37]
  - sustav zasnovan na pravilima, kaskadne regularne gramatike, uporaba leksikona i popisa, F-mjera 91,76%
- Francuski [3]
  - posebna stabla odluke, semantička klasifikacijska stabla (*eng. Semantic Classification Trees, SCT*), nema podataka o F-mjeri
- Grčki [26]
  - sustav zasnovan na pravilima, F-mjera 0,83
- Japanski [2]
  - ekstrakcija naziva, SVM metoda sa znakovno temeljenim prepoznavanjem odsječaka (*eng. chunking*), F-mjera 87,2%
- Katalonski [11]
  - Dvojezični model koji koristi AdaBoost algoritam sa skupom binarnih klasifikatora. Uči se sustav za PKN za španjolski, zatim se koriste ili prijevodni rječnici za leksičke značajke ili dvojezični model koji koristi kros-lingvističke značajke. Nakon toga slijedi samonadoupnjavanje na katalonskim tekstovima, F-mjera 91,18%

- Kineski [54]
  - CHINERS sustav, PKN sustav za sportsku domenu, automatska izrada DKA iz pravila prepoznavanja, plitki parser (*eng. shallow parser*) uz segmentaciju riječi, F-mjera 0,84
- Korejski [12]
  - HMM model, eksperimenti s ko-treniranjem, F-mjera 0,67
- Nizozemski [19], [9]
  - hibridni sustav koji koristi popise naziva, gramatike imena i algoritam indukcije pravila (Ripper [13]) koji koristi pet značajki, ručno odabiranje dobrih kontekstom generiranih pravila, F-mjera oko 0,67
  - traže se konteksti riječi s popisa unutar neobilježenog korpusa te se koriste stabla odlučivanja (IGTree), F-mjera 0,71
- Njemački [47]
  - inkrementalna metoda (samonadopunjavanje), analiza sintaktičkih, tekstualnih konteksta i morfološka analiza, novi konteksti se koriste za pronalazak novih naziva, 65% korektno označenih naziva
- Portugalski [21]
  - HMM, TBL i SVM, SVM postiže F-mjeru od 88,11%
- Španjolski [10]
  - AdaBoost algoritam, popis okidajućih riječi i opći popisi naziva, dva klasifikatora, jedan za prepoznavanje (F-mjera 82,47%), drugi za klasifikaciju (F-mjera 87,84%)
- Švedski [17]
  - kombinacija ručno pisanih pravila i samonadopunjavanja; započinje se s manjim leksikonima koji se povećavaju procesiranjem velikih količina tekstova, F-mjera 0,61
- Turski [51]
  - n-gramski model jezika sa skrivenim Markovljevim modelima koristeći leksičke, kontekstualne, morfološke i modele oznake entiteta, F-mjera 91,56%
- Ukrajinski [27]
  - učenje uzoraka s Interneta (Google) uz uporabu Levenshteinove mjera sličnosti te glasanje (*eng. voting*) i slaganje triju klasifikatora (naivni Bayes, stabla odlučivanja i 2-nn), F-mjera oko 0,50
- Vijetnamski [48]
  - učenje potpornih vektorskih strojeva (SVM) nad obilježenim korpusom uz korištenje popisa naziva i segmentaciju riječi, F-mjera 87,75%

4.4. Višejezični i jezično neovisni sustavi

Jedna od iznimno poželjnih osobina sustava za prepoznavanje i klasifikaciju naziva je i mogućnost korištenja istog na tekstovima pisanim drugim jezicima. Tu osobinu nije moguće postići sustavima zasnovanim na pravilima bez dodatnog rada eksperata, ali ju je zato lakše postići nekom od metoda strojnog učenja, samim odabirom pogodne metode ili izradom korpusa na drugom jeziku na kojemu bi se metoda mogla istrenirati.

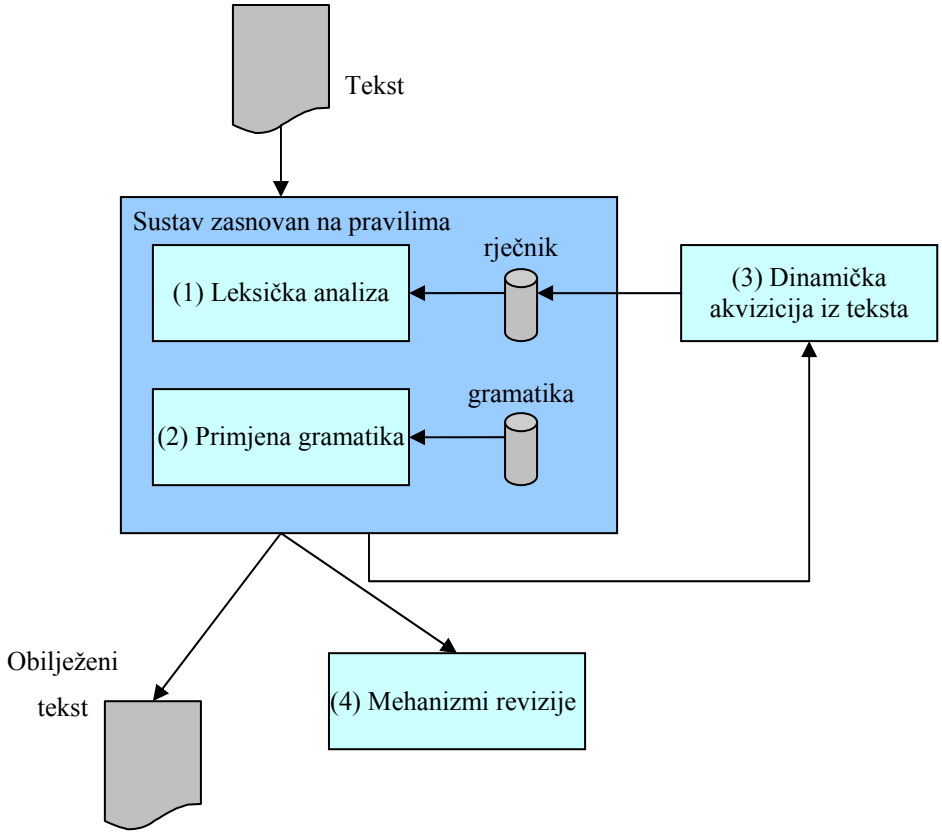
Ciljevi takvih sustava su rukovanje osnovnim jezično ovisnim dokazima, učenje malih listi naziva (oko 100 imena), procesiranje velikih i malih tekstova, postizanje dobre skalabilnosti po klasama i inkrementalno učenje koje sprema naučene informacije za buduću uporabu.

U ovoj kategoriji sustava se pojavilo nekoliko radova koji obećavaju višejezične i/ili jezično neovisne sustave.

U [38] je predstavljena višejezična arhitektura za prepoznavanje i klasifikaciju naziva u više od 12 jezika: arapski, kineski, engleski, francuski, njemački, japanski, finski, malgaški, perzijski, poljski, ruski, španjolski i švedski. Arhitektura sustava sastoji se od četiri izvora znanja, uglavnom različita za svaki jezik (Slika 8.):

1. Popisi
   - Djelomično dijeljen resurs između sličnih jezika: romanski, indoeuropski jezici i dr.
2. Gramatike
   - Koristi se Intex/Unitex za francuski i engleski jezik, morfološki analizator Chasen za azijske jezike
3. Sposobnosti učenja
   - Učenje koncepata u svrhu proširenja skupa izraza koje pravila mogu identificirati. Leksikoni se iskorištavaju kao domene za dinamički pronalazak novih entiteta
4. Sposobnosti revizije
   - Revizija oznaka u određenim tekstovima

Arhitektura opisanog sustava prikazana je na slici 8.:



Slika 8. Arhitektura sustava [38]

Resursi za druge jezike (ruski – ćirilica, arapski i perzijski – arapski sustav pisanja) su trenutno u fazi definiranja i adaptacije. Sustav je za sada testiran jedino za engleski i francuski. Na engleskim korpusima s konferencije MUC-6 rezultati su za preciznost 0,86, odziv 0,95 što čini F-mjeru od 0,90

Među višejezičnim i jezično neovisnim sustavima, često su popularne metode nenadziranog odnosno slabo nadziranog učenja kao metode koja omogućavaju jezičnu neovisnost čak i bez potrebe obilježenog korpusa.

Jedan od takvih sustava [14] ostvaruje PKN na rumunjskom, engleskom, grčkom, turskom i hindi jeziku s F-mjerama redom 0,74, 0,54, 0,55, 0,53 i 0,41.

Sustav koristi kontekstualne uzorke i unutarnje dokaze (sufikse i prefikse) kao nezavisne izvore dokaza kroz strukturu trie (*eng. reTRIEval*) [24] za morfološku i kontekstualnu statistiku kroz sljedeće korake:

0. korak – izgraditi inicijalnu listu naziva za treniranje reprezentacija klasa (entiteta)
1. korak – čitajući tekst izgraditi lijevi i desni morfološki i kontekstni trie
2. korak – trenirati trieve na podacima, uvoditi informacije iz treniranja i nanovo procijeniti internu distribuciju trieva samonadopunjavanjem
3. korak – identificirati i klasificirati nazive u tekstu koristeći klasifikatore - trieve
4. korak – osvježiti nazive i kontekste koristeći novoekstrahirane informacije

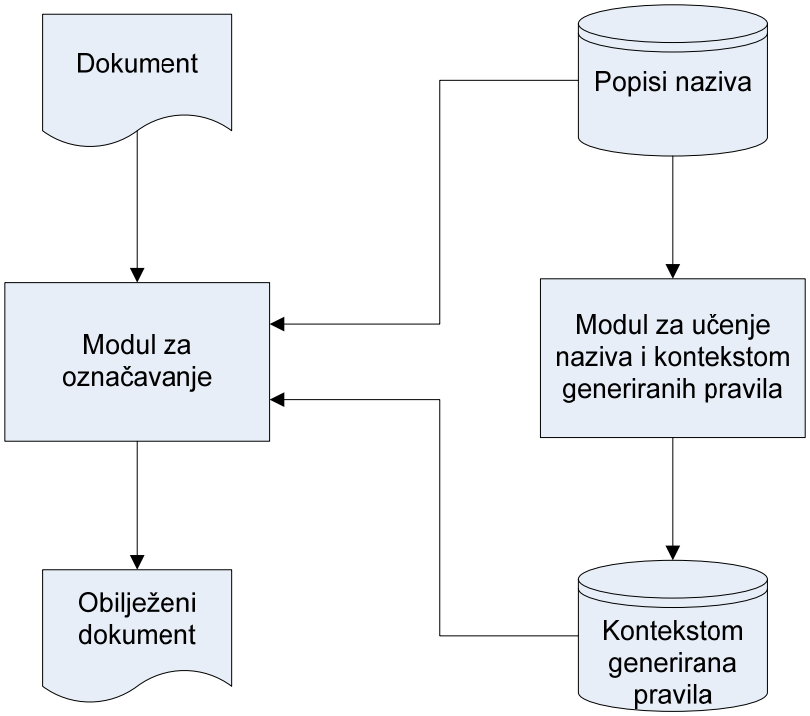
Prošireni model istih autora [15], uz drugačiju metodu zaglađivanja distribucije putanja TRIE-a, novu segmentacijsku metodu i drugačiju metodologiju označavanja, za španjolski i nizozemski postiže F-mjere od 0,77 odnosno 0,72.

5. Samonadopunjujući sustav za ekstrakciju naziva

U okviru ovog rada izgrađen je sustav za PKN za hrvatski jezik koji se temelji na popisima naziva i sustav za ekstrakciju naziva. Kao rezultat proučavanja više metoda strojnog učenja za rješavanje PKN problema, odabran je samonadopunjujući pristup i to u smjeru ekstrakcije novih, nepoznatih naziva iz neobilježenog korpusa (zbog nedostupnosti velikog obilježenog korpusa) koristeći mali skup provjerenih naziva. Umjesto fiksnog korpusa nad kojim se vrši ekstrakcija, odabran je korpus koji sačinjavaju rezultati upita WWW (*World Wide Web*) tražilici.

Realizirani sustav može se klasificirati kao hibridni sustav zbog malog broja ugrađenih pravila, poput pravila koje nalaže da nakon prepoznatog imena dolazi prezime i sličnih pravila navedenih u nastavku. Sustav je nenadziran, točnije slabo nadziran jer koristi mali broj naziva kao ulaz u algoritam, siromašan je znanjem te se može primijeniti na više jezika i time spada u grupu jezično neovisnih sustava (naravno, ne u potpunosti jezično neovisnih s obzirom da nema ugrađenih metoda rješavanja specifičnih problema jezika poput arapskog i kineskog).

Implementacija se sastoji od dva osnovna modula: modula za označavanje i modula za učenje naziva i kontekstom generiranih pravila, prikazanih na slici 9.

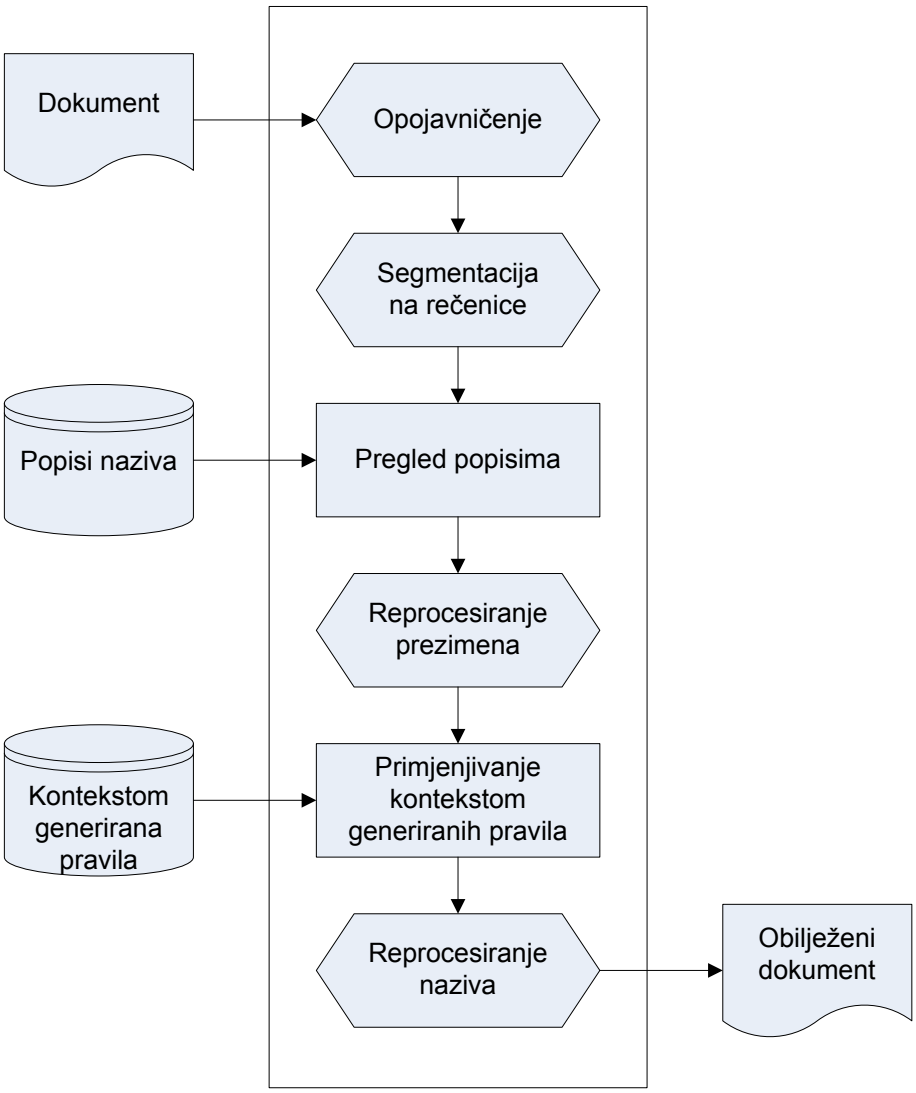


Slika 9. Shema sustava za prepoznavanje naziva

5.1. Modul za označavanje

Zadatak modula za označavanje je obilježavanje prepoznatih naziva u neobilježenom dokumentu koji je modulu predstavljen u obliku čistog teksta u UTF-8 formatu[26], bez ikakvih dodatnih oznaka. Konačni je rezultat dokument obilježen u skladu s MUC specifikacijama. Modul je građen prema uzoru na modul za označavanje opisan u [19]. Shema modula prikazana je na slici 10.

[26] 8-bit UCS/Unicode Transformation Format, enkodiranje Unicode standarda za reprezentaciju i transformaciju teksta, http://www.utf-8.com/



Slika 10. Shema modula za označavanje

U nastavku će se detaljnije opisati svaki od postupaka modula te će se rezultati obrade prikazati kroz sljedeći odabrani primjer:

...tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež. Zarez i Padež su svoje otkriće...

5.1.1. Opojavničenje

Pojam pojavnica ili token definiran je kao skup znakova koji se nalaze između dva znaka koji služe kao graničnici. Skup znakova koji čine pojavnicu mora

biti iz točno definirane abecede: za hrvatski jezik to je skup svih znakova hrvatske abecede kojoj su pridodane znamenke i crtica. Opojavničenje ili tokenizacija (*eng. tokenization*) je stoga postupak rastavljanja teksta na pojavnice, dakle postupak prepoznavanja granica riječi.

U implementaciji se koristi klasa *UTF8Tokenizer()* iz biblioteke TMT *(eng. Text Mining Tools*) [43] koja ulazni tekst razlaže na pojavnice tako da pojavnicom proglasi sve što se nađe između dvije bjeline i/ili znaka interpunkcije uključivši i crticu, bez obzira bila ona u službi spojnice[27] ili ne. Navedena klasa također rješava problem decimalnih brojeva čineći npr. 2,4 jednom pojavnicom u konačnici, bez rastavljanja na 2 i 4. Rezultat postupka je struktura u kojoj je jasno obilježeno kojoj klasi pripada pojavnica (riječ, broj, bjelina, znak interpunkcije).

Primjer

Postupak opojavničenja nad primjerom utvrđenim na početku poglavlja daje sljedeći rezultat:

...tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež. Zarez i Padež su svoje otkriće...

Punom crtom su označeni rezultantne pojavnice, istočkanom crtom su označeni znakovi interpunkcije dok su bjeline ostale netaknute.

[27] Npr. „gore-dolje“ bit će rastavljeno na „gore“ i „dolje“

5.1.2. Segmentacija na rečenice

Segmentacija na rečenice (*eng. sentence segmentation*)[28] postupak je rastava teksta na rečenice. Taj je postupak potreban zbog činjenice da nazivi nikada ne prelaze granice rečenice te zbog identifikacije i obrade pojavnica koje se nalaze na prvom mjestu u rečenici. Segmentacija ne mora biti savršena jer ne spada u primarne ciljeve pa su pogreške pri segmentaciji prihvatljive ukoliko se pojavljuju zbog primjene opuštenijih pravila – bolje je segmentirati šire nego umetati oznake za kraj rečenice unutar stvarne rečenice i time potencijalno naškoditi prepoznavanju.

Za hrvatski jezik postoji gotov sustav za segmentaciju[29] s deklariranom točnošću od 99.5%. Međutim tako kompleksan i precizan sustav nije zapravo ni potreban zbog gore navedenih razloga.

U tu svrhu implementirana je segmentacija na rečenice sa sljedećim jednostavnim pravilima:

- znakovi interpunkcije „!“ , „?“ i „…“ označavaju kraj rečenice
- točka ispred popisa od oko 40 kratica ne označava kraj rečenice (titule, opće kratice, kratice za oznaku tipa tvrtke, neke rimske brojke, kratice dana u tjednu), u popisu nisu navedene kratice poput „itd.“ i „sl.“ koje se često javljaju na kraju rečenice
- točka iza brojke nije kraj rečenice jer može označavati redni broj ili datum
- točka unutar kratica pisanih velikim slovom nije završetak rečenice (iako zadnja točka to može biti)
- točka između znamenaka nije kraj rečenice (decimalna točka, datum) – ovaj slučaj rješava tokenizator koji čini tu točku dijelom brojke
- svaku ostalu točku označi kao kraj rečenice

[28] Poznato još kao i razrješavanje rečenične granice (eng. Sentence Boundary Disambiguation)
[29] D. Boras, (1998), Teorija i pravila segmentacije teksta na hrvatskom jeziku, doktorska disertacija, Filozofski fakultet Sveučilišta u zagrebu

Primjer

Rezultat segmentacije na rečenice nad zadanim primjerom su dvije rečenice:

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež.

Zarez i Padež su svoje otkriće…

Točke unutar titula „prof.“ „dr.“ i „sc.“ se ne tretiraju kao kraj rečenice, međutim točka nakon „Nenad Padež“ (označena istočkanom crtom) ne spada niti u jedno pravilo pa se na kraju označava kao kraj rečenice[30].

5.1.3. Pregled popisima

Idući korak je pregled popisima tj. sraz opojavničenog i rečenično segmentiranog teksta s popisima imena, organizacija i lokacija navedenim redoslijedom. Pri pregledu se ignoriraju mogući znakovi interpunkcije tako da će naziv „A. G. Matoš“ biti prepoznat ako je napisan kao „A G Matoš“, ali se poštuje jednaka tipografija pojavnica koje se uspoređuju, dakle ako se u tekstu pojavi pojavnica *nada*, s imenom *Nada* u popisu imena prvotna pojavnica se neće obilježiti. Trenutno u sustavu nije podržano podudaranje s morfološkim varijantama (oblicima riječi).

Primjer

Pod pretpostavkom da se u popisu imena nalazi i pojavnica „Stjepan“, rezultat obrade je:

[30] Trotočke u primjeru nisu dio primjera nego pokazatelj njegove djelomičnosti te stoga nisu označene

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež.

Zarez i Padež su svoje otkriće…

Punom crtom je označena pripadnost klasi osoba.

5.1.4. Reprocesiranje prezimena

U svrhu povećanja preciznosti prepoznavanja, dodan je postupak reprocesiranja koji se temelji na pretpostavci da je pojavnica nakon imena napisana velikim početnim slovom prezime. To se prezime zatim ekstrahira te se ponovno izvršava pregled teksta, ali sada s tim popisom prezimena. Ovaj je korak uvjetovan pojavom da se u tekstu nakon predstavljanja osobe imenom i prezimenom u nastavku teksta istu tu osobu češće referira samim prezimenom nego imenom zbog kraćeg pisanja, te se tim korakom osigurava prepoznavanje daljnje reference na tu osobu.

Ovaj postupak predstavlja jedno od ugrađenih konkretnih pravila čime sustav postaje hibridnog tipa.

Primjer

Rezultat nakon reprocesiranja prezimena je:

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež.

Zarez i Padež su svoje otkriće…

Kraj imena Stjepan pronađenog u prethodnom koraku, označenog istočkanom crtom, pronađeno je prezime Zarez koje se zatim obilježava u čitavom tekstu (puna crta).

5.1.5. Primjenjivanje kontekstom generiranih pravila

Jedan od produkta modula za učenje su i kontekstom generirana pravila koja se javljaju u tri oblika: *dr sc **, ** je izjavio* i *d d * d d*. Kontekstom generirana pravila su značajna jer se koriste za prepoznavanje naziva.

Prvi i drugi oblik (konteksti oblika *X* * i * *X* gdje *X* označava konkretnu pojavnicu ili više njih) kontekstom generiranih pravila na mjesto znaka * primaju jednu ili više pojavnica pisanih velikim početnim slovom desno odnosno lijevo od osnove pravila. Prihvaćanje samo pojavnica pisanih velikim početnim slovom je odabrano kao privremeno rješenje za implementaciju koja ne podržava prepoznavanje veće jezične strukture. Prepoznavanje veće jezične struktura (npr. imenske sintagme) se aproksimira navedenim pravilom zbog nemogućnosti ostvarenja iste zbog nepostojanja označivača vrste riječi ili nekih drugih resursa koji bi mogli još poboljšati tu osnovnu aproksimaciju. Ta aproksimacija, naravno ne pokriva slučajeve poput Zavod za elektroniku (jer nakon pojavnice *Zavod* pisane velikim početnim slovom slijedi pojavnica *za* pisana malim početnim slovom) i sl.

Treći oblik kontekstom generiranih pravila (*X* * *X*) na mjesto znaka * prima bilo koji niz pojavnica, bez obzira na veliko ili malo početno slovo, a u slučaju više uzastopnih osnovnih dijelova istog pravila, poštuje se načelo odabira najduljeg pogotka (*eng. longest match*). Npr. ukoliko se koristi pravilo *d d * d d* te se pojavi niz znakova *d d d d D d a d d d d*, na mjesto znaka * se ekstrahira niz *d d D d a d d*.

Primjena ovih pravila je uvedena zbog obilježavanja neviđenih naziva u tekstu koji se pokoravaju pravilu, a ne nalaze se na nekom od popisa. Zato je važan njihov dobar odabir[31]. Važno je napomenuti da se pravila primjenjuju na tekst bez obzira na to je li neki od naziva prethodno već obilježen tj. pravilo ima prednost nad popisima.

[31] v. poglavlje 5.2.2

Primjer

Pod pretpostavkom da se u popisu kontekstom generiranih pravila nalazi i pravilo *dr sc **, rezultat primjene takvog pravila je:

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež.

Zarez i Padež su svoje otkriće…

Iscrtkanom crtom označeno je pronađeno pravilo, punom crtom označeni je ime uz pravilo.

5.1.6. Reprocesiranje naziva

Sve što su pravila do sada prepoznala potrebno je nanovo procesirati (reprocesirati) jer postoji mogućnost da se nešto od pravilima prepoznatih naziva ponavlja u nastavku teksta bez konteksta koji bi mogao upućivati na njih. Takvi bi nazivi u slučaju nepostojanja ovoga koraka ostali neobilježeni. Uz to se provjerava i djelomično slaganje osobnih imena i imena organizacija tako da se generiraju poredci susjednih pojavnica s lijeva na desno (v. primjer) uz kraćenje u akronime s izbacivanjem pojavnica veličine manje od 3 slova[32].

Primjer

Reprocesiranje imena i organizacija kao rezultat daje:

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež.

Zarez i Padež su svoje otkriće…

[32] npr. a, o, po, i, uz, za, i dr.

Pronađeno ime *Nenad* (istočkana crta) upućuje na prezime *Padež* koje se zatim obilježava drugdje u tekstu (puna crta). Uz pretpostavku da je u tekstu korištenjem kontekstom generiranih pravila pronađeno ime organizacije *Privredna Banka Zagreb*, korak reprocesiranja naziva generira sljedeće djelomične nazive *Privredna Banka*, *Banka Zagreb* (bez *Privredna Zagreb*!) te kraticu *PBZ*.

Naposljetku je potrebno oblikovati rezultat u željenu formu u skladu sa specifikacijama s konferencija MUC. Izrađena je i XSL[33] (*eXtensible Stylesheet Language*) datoteka za formatiranje izgleda XML dokumenta.

Primjer

Krajnji rezultat obrade primjera je XML datoteka prikazana na slici 11.:

```
…tvrde prof. dr. <ENAMEX TYPE="PERSON">Stjepan Zarez</ENAMEX> i dr. sc.
<ENAMEX TYPE="PERSON">Nenad Padež</ENAMEX>. <ENAMEX
TYPE="PERSON">Zarez</ENAMEX>  i <ENAMEX TYPE="PERSON">Padež</ENAMEX>
su svoje otkriće…
```

Slika 11. Primjer krajnjeg rezultat metode modula za označavanje

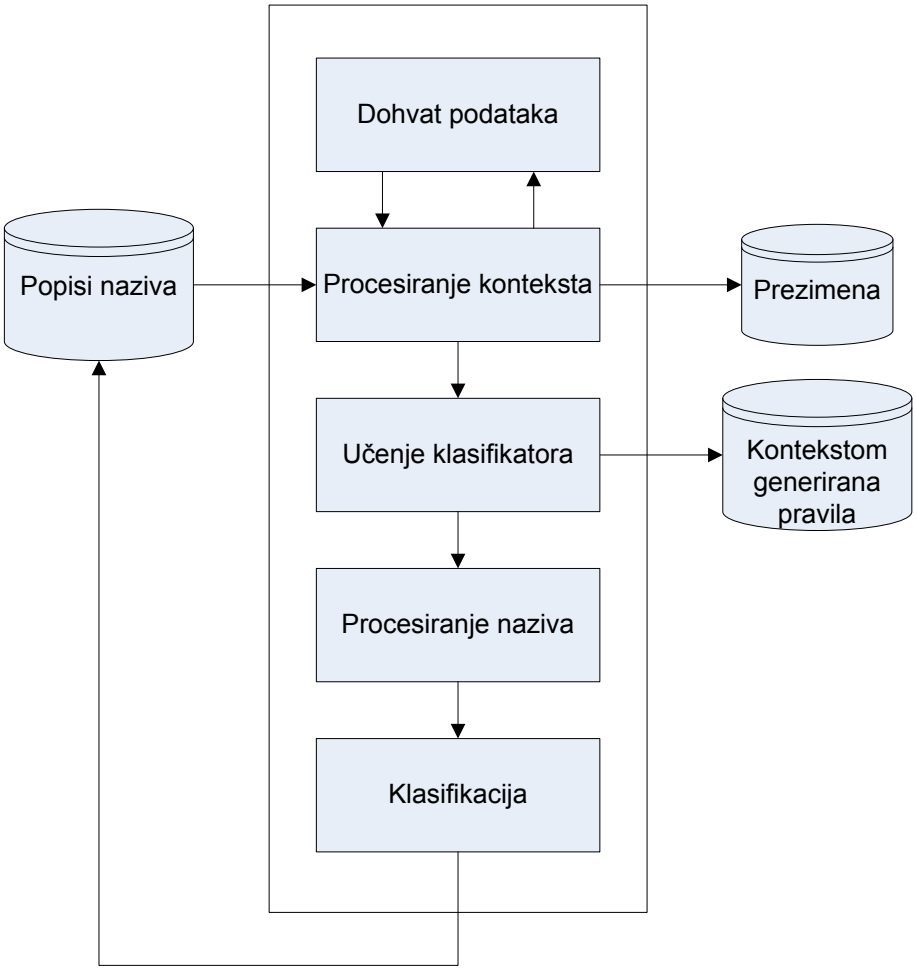
se u Internet pretraživaču u konačnici prikazuje kao[34]:

…tvrde prof. dr. Stjepan Zarez i dr. sc. Nenad Padež. Zarez i Padež su svoje otkriće…

[33] obitelj transformacijskih jezika koji opisuju načine na koji će se XML datoteka formatirati ili transformirati, http://www.w3.org/Style/XSL/
[34] uz izrađenu XSL opisnu datoteku

5.2. Modul za ekstrakciju naziva i kontekstom generiranih pravila

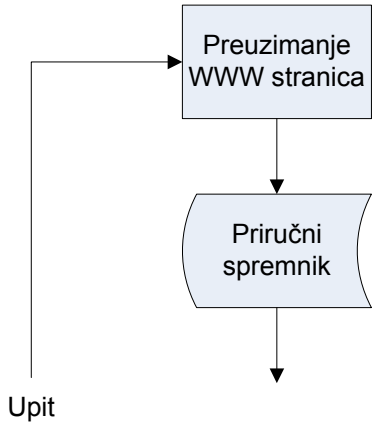
Zadatak modula za učenje naziva i kontekstom generiranih pravila jest, počevši od početnog popisa naziva (smještenih u tri kategorije: osobe, organizacije i lokacije), a koristeći korpus, automatski ekstrahirati kontekste, adekvatno ih ocijeniti te uz pomoć odabranih konteksta dalje ekstrahirati nove nazive. Te je nazive zatim potrebno posebno ocijeniti i dodati u popise naziva, i tako po mogućnosti iterativno. Modul je izgrađen kombinacijom ideja opisanih u [52] i [22].



Slika 12. Shema modula za ekstrakciju naziva i kontekstom generiranih pravila

5.2.1. Dohvat podataka

Kao ulazni podatak, sustavu je na raspolaganju najveći postojeći korpus – WWW (*eng. World Wide Web*), ogromni sustav povezanih hipertekstualnih dokumenata, točnije jedan njegov dio, onaj koji je indeksiran tražilicom koja se koristi za dohvat potrebnih informacija. Shema ovog podmodula prikazana je na slici 13.



Slika 13. Shema podmodula za dohvata podataka

U sustavu se za ekstrakciju podataka koriste tražilica Google[35] za dohvat potrebnih informacija te hrvatska tražilica Pogodak.hr[36] za ekstrakciju morfoloških oblika pronađenih naziva/imena jer ista tražilica podržava pretragu po morfološkim oblicima koristeći ugrađeni algoritam za nalaženje podniza zajedničkog svim oblicima riječi (*eng. stemmer*). Naravno, moguće je koristiti bilo koju drugu tražilicu uz izradu adekvatnih procedura i odgovarajuće modifikacije, zavisno o specifičnostima koje tražilica podržava ili ne podržava u usporedbi s tražilicom Google. Konkretno, Google omogućava korištenje * operatora koji označava jednu ili više riječi prije ili nakon neke pojavnice tako da je jednostavnim formiranjem

[35] www.google.hr
[36] www.pogodak.hr

upita poput *Privredna * Zagreb* moguće izvršiti pretragu za pojmovima koji se nalaze između pojavnica *Privredna* i *Zagreb* (u ovom slučaju, najvjerojatnije samo pojam *banka*). Dohvat podataka temelji se na spomenutom operatoru te bi korištenje tražilice koja ga ne podržava zahtijevalo ponovno kodiranje određenih rutina.

Preuzimanje podataka može se u načelu ostvariti koristeći Google-ov API (*Application Program Interface*), sučelje za komunikaciju s tražilicom. Međutim, Google je zatvorio pristup[37] API-u tako da se moralo pristupit sporom postupku preuzimanja pojedinih stranica. Kako automatsko postavljanje upita Google-u nije dozvoljeno, uvedena je nasumična vremenska zadrška iznosa između 10 i 20 sekundi prije svakog preuzimanja stranica.

Postupak preuzimanja stranice odvija se preko rutine koja šalje zahtjev za adresom stranice (*Uniform Resource Locator, URL*) preko protokola HTTP (*HyperText Transfer Protocol*) i preuzima stranicu koja se upućuje na proces obrade u narednom dijelu. Postupak je vidljiv na slici 13.

U sustav je implementiran i jednostavan priručni spremnik (*eng. cache*) koji služi za spremanje dohvaćenih stranica. Priručni se spremnik koristi radi uštede resursa i ubrzavanja rada sustava jer nema potrebe ponovno skidati već skinute stranice. Time se također smanjuje fluktuiranje rezultata same tražilice – rezultati koji su inače dinamični zbog čestog osvježavanja Google-ove baze.

Uz tražilicu Google, interesantne su još i Altavista zbog *NEAR* operatora koji bi se mogao koristiti za aproksimaciju * operatora tražilice Google te Pogodak.hr, Pogodak.si i Pogodak.ba zbog mogućnosti pretrage po morfološkim oblicima pojma koji se traži.

Promjena jezika od interesa (jezika za koji se želi izvršiti ekstrakcija naziva) se vrši promjenom linije koda u kojoj se prepravljaju parametri adrese tražilice Google i izradom adekvatnih početnih popisa naziva.

---

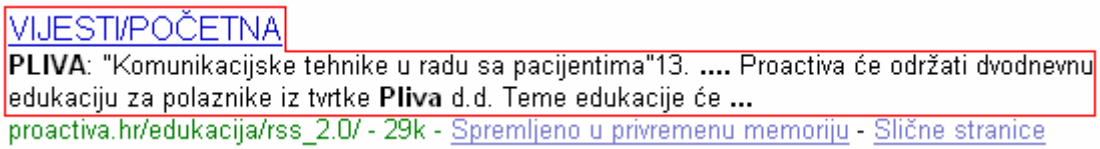
[37] Kontakti sa Google-om u svezi korištenja API-a nisu bili uspješni, međutim navedeni API ionako ne bi mogao poslužiti zbog ograničenja od 1000 upita na dan

5.2.2. Procesiranje konteksta

Postupak procesiranja konteksta je prvi proces modula za učenje naziva i kontekstom generiranih pravila kojemu je zadatak na temelju početnog skupa naziva (popis imena, popis organizacija i popis lokacija) ekstrahirati i adekvatno ocijeniti kontekste u kojima se nazivi s popisa nalaze.

5.2.2.1 Ekstrakcija konteksta

Proces ekstrakcije konteksta započinje korištenjem rutina za dohvat podataka u svrhu preuzimanja rezultata Google-ove pretrage nad svakim od naziva iz početnih popisa naziva. Nakon preuzimanja stranice, ista se obrađuje i iz nje se uzimaju relevantne informacije koje u ovom slučaju predstavljaju označeni tekst iz Google-ovog isječka rezultata (*eng. snippet*) prikazan na slici 14.


VIJESTI/POČETNA
PLIVA: "Komunikacijske tehnike u radu sa pacijentima"13. **....** Proactiva će održati dvodnevnu edukaciju za polaznike iz tvrtke **Pliva** d.d. Teme edukacije će **...**
proactiva.hr/edukacija/rss_2.0/ - 29k - Spremljeno u privremenu memoriju - Slične stranice

Slika 14. Google-ov isječak rezultata

Isječak se sastoji od naslova (plava hiperveza) i teksta koji zatim prolaze kroz proces opojavničenja i segmentiranja na rečenice[38]:

VIJESTI/POČETNA

PLIVA: "Komunikacijske tehnike u radu sa pacijentima"13. ...Proactiva će održati dvodnevnu edukaciju za polaznike iz tvrtke Pliva d.d. Teme edukacije će ...

[38] Valja primijetiti da u ovom slučaju sustav neće označiti kraj rečenice niti ispred riječi „Teme“ niti iza brojke „12“ zbog oslabljenih pravila, makar ta točka zaista i predstavlja kraj rečenice, ali će zato označiti „…“ kao kraj rečenice

Unutar tako obrađenog isječka se pronalazi traženi naziv te se izolira njegov kontekstni prozor *C* veličine *n*:

$$C = W_{i-n}\ W_{i-(n-1)}\ \ldots\ W_{i-1}\ W_i\ W_{i+1}\ \ldots\ W_{i+(n-1)}\ W_{i+n} \tag{5.1}$$

Tražena pojavnica zamjenjuje se znakom *:

$$C = W_{i-n}\ W_{i-(n-1)}\ \ldots\ W_{i-1}\ *_i\ W_{i+1}\ \ldots\ W_{i+(n-1)}\ W_{i+n} \tag{5.2}$$

Te se tako dobiveni niz riječi razlaže na kombinacije *K* s time da se čuva poredak riječi. Efektivno, traži se podskup uređenog skupa C na sljedeći način:

$$K = \{podskup(C, x, y) \mid i - n \le x \le i \le y \le i + n \wedge x \ne y\} \tag{5.3}$$

gdje relacija *podskup(C, x, y)* vraća podskup skupa *C* od *x*-tog do *y*-tog znaka, a *i* je indeks znaka * u skupu *C*.

Za svaku kombinaciju konteksta $k \in K$ bilježi se koliko se puta kontekst *k* pojavio u čitavom postupku (time se određuje frekvencija pojavljivanja konteksta *k* u korpusu promatranih Google isječaka). U implementaciji veličina kontekstnog prozora *n* jednaka je 2. Naposljetku se vrši i zbrajanje frekvencija konteksta koji sadrže iste riječi pisane na različit način (npr. *Grad* * i *grad* *).

Čitav proces izvršava se višedretveno, kroz tri dretve – svaka za jedan popis naziva.

Cijena ekstrakcije konteksta za *n* naziva iz svake od tri kategorije, preuzimajući *x* stranica po nazivu iznosi:

$$N_1 = 3xn \tag{5.4}$$

Primjer

Pronalaženje naziva i ekstrakcija kontekstnog prozora veličine 2 u isječku gore navedenog primjera:

...za polaznike <u>*iz tvrtke*</u> **Pliva** <u>*d.d.*</u> *Teme edukacije...*

lijevi kontekst desni kontekst

rezultira kontekstnim prozorom:

*iz tvrtke * d d*

Ovaj se prozor zatim razlaže na sljedeće kombinacije:

*iz tvrtke * d d, iz tvrtke * d, iz tvrtke *,*

*tvrtke * d d, tvrtke * d, tvrtke *, * d d, * d*

kojima se za svako ponavljanje konteksta u korpusu povećava frekvencija pojavljivanja.

5.2.2.2 Ekstrakcija prezimena

Iz svih konteksta koji se dobiju prethodnim korakom iz popisa imena vrši se ekstrakcija prezimena ugrađenim jednostavnim pravilom koje nalaže da pojavnica nakon imena pisana velikim početnim slovom predstavlja prezime. Pojavnice ekstrahirane tim pravilom postaju kandidati za prezimena kojima se bilježi frekvencija pojavljivanja u skupu svih konteksta dobivenih od skupa imena.

Dobiveni kandidati se pri ekstrakciji filtriraju tako da se uklone moguće stop riječi[39]. Nakon ekstrakcije, kandidati se filtriraju tako da se obrišu svi oni čija je frekvencija pojavljivanja manja od nekog praga *p* (u sustavu konkretno *p*=5).

Preostali kandidati se filtriraju tako da se uklanjaju nazivi koji se nalaze na početnim popisima te se uklanjaju duplikati već pribilježeni na popisu imena. Preostali kandidati za prezimena proglašavaju se prezimenima i pohranjuju u datoteci.

[39] Koriste se postojeći popisi stop riječi za hrvatski i engleski jezik

Primjer

Pod pretpostavkom mogućeg konteksta:

*izjavio je * Antić*

sustav bi ekstrahirao *Antić* kao kandidata za prezime. Ukoliko se kandidat pojavio više od 5 puta u svim kontekstima, proglašava se prezimenom (ukoliko nije stop riječ ili nije na nekom od drugih popisa).

5.2.2.3 Filtriranje konteksta

U nastavku obrade vrše se višestruka filtriranja skupa konteksta. Filtriraju se jednostavni konteksti koji sadrže najviše jednu riječ sa svake strane (konteksti oblika $* X$, $X *$, $X * X$ koji na mjestu X sadrže najviše jednu riječ) tako da se uklanjaju oni konteksti kojima se kao riječ pojavljuju stop riječi i prezimena ekstrahirana u prethodnom koraku.

Zatim se uklanjaju konteksti kojima je frekvencija pojavljivanja jednaka jedan čime se gubi oko 80% konteksta.

Nakon toga slijedi „kažnjavanje“ konteksta [52] prema sljedećoj formuli:

$$s'(r,c_i) = s(r,c_i) - \sum_{j \neq i} s(r,c_j) a(c_j,c_i) \tag{5.5}$$

gdje su:

$s(r,c_i)$ ≡ frekvencija naziva r u kategoriji (klasi) i

$$a(c_1,c_2) = \frac{velicina\_popisa\_za\_klasu(c_2)}{velicina\_popisa\_za\_klasu(c_1)} \tag{5.6}$$

Trenutno je u sustavu veličina a postavljena na vrijednost 1.

Bit „kažnjavanja“ je smanjenje frekvencije konteksta koji su zajednički za sve popise tj. konteksta koji nemaju diskriminatornu vrijednost jer ne upućuju na neku posebnu kategoriju.

Primjer

Kontekst „*centar* *“ jedan je od konteksta koji se pojavljuje na sva tri popisa. Na popisu osoba pojavio se 4 puta, na popisu organizacija 2, a na popisu lokacija 133 puta. „Kažnjavanje“ za rezultat ima smanjenje frekvencija na: -131, -135 i 127.

Navedeni kontekst sada gubi značaj (ako ga je imao) u kategorijama osoba i organizacija dok mu je frekvencija na popisu lokacija neznatno smanjena. Time se sačuvalo njegovo značenje za kategoriju lokacija.

U konačnici se obavlja sortiranje konteksta po frekvenciji pojavljivanja i odsijecanje prvih **c** konteksta koji se zatim upućuju na ocjenjivanje potrebno za učenje klasifikatora.

5.2.3. Učenje klasifikatora

Korištenjem odabranih *c* konteksta uči se naivni Bayesov klasifikator (*eng. Naive Bayes Classifier, NBC*) koji će se na kraju modula koristiti za klasifikaciju novih naziva. Prvo je potrebno ocijeniti početne nazive koji su predstavljeni kao ulaz algoritmu.

5.2.3.1 Ocjenjivanje naziva

U trenutnoj fazi, odabranih *c* konteksta postaju pravila, točnije kontekstom generirana pravila koja se u nastavku koriste za ocjenjivanje početnog skupa naziva iz popisa naziva. Proces ocjenjivanja konteksta je u načelu višedretven te podržava proizvoljan broj dretvi manji od broja naziva u popisu, ali se trenutno koriste samo dvije zbog smanjenja opterećenja tražilice Google.

Početne nazive potrebno je ocijeniti da bi se na temelju dobivenih ocjena mogao izgraditi klasifikator koji će na temelju ocjena novih naziva moći klasificirati te nazive u pripadajuće kategorije[40].

Kao ocjena se koristi statistička mjera uzajamna obavijesnost (*eng. Pointwise Mutual Information, PMI*) koja se često pojavljuje u radovima iz područja crpljenja obavijesti:

$$PMI = \log_2 \frac{P(x,y)}{P(x)P(y)} \tag{5.7}$$

gdje su:

$P(x)$, $P(y)$ ≡ vjerojatnosti događaja x odnosno y

$P(x,y)$ ≡ vjerojatnost događaja x i y

Točkasta uzajamna obavijesnost je dakle količina obavijesti koju pruža odvijanje događaja kojega predstavlja y o odvijanju događaja kojeg predstavlja x. Ukratko, navedena mjera pokazuje koliko jedan pojam govori o drugome. PMI je dobra mjera nezavisnosti, jer vrijednosti blizu 0 indiciraju nezavisnost frekvencija, ali loša mjera zavisnosti jer ovisi o frekvenciji individualnih pojmova. Problem te mjere su rijetki podaci koji dobivaju veću vrijednost zbog niskih frekvencija pojavljivanja.

Oblik uzajamne obavijesnosti koji se koriti u radu, a po uzoru na [22] je:

$$PMI(NE, C) = \frac{\#hits(NE + C)}{\#hits(NE)} \tag{5.8}$$

gdje su:

$\#hits(P)$ ≡ funkcija koja vraća broj rezultata tražilice na pojam P

NE ≡ naziv

[40] v. poglavlje 5.2.5

C ≡ kontekstom generirano pravilo

$NE{+}C$ ≡ kontekstom generirano pravilo u kojem je znak * supstituiran nazivom NE

Samo ocjenjivanje se konkretno provodi slanjem upita sadržaja *NE+C* i *NE* tražilici. Iz dobivenih preuzetih stranica se ekstrahira broj rezultata koji se onda koriste u formuli (5.8). Ukupna cijena PMI ocjenjivanja u broju upita koji se šalju tražilici za ocjenjivanje *n* naziva s *c* konteksta u svakoj od tri kategorije naziva je:

$$N_2 = 3n(c+1) \tag{5.9}$$

Formula (5.8) zbog nedostatka funkcije logaritmiranja kao rezultat vraća iznimno niske nenegativne vrijednosti (reda veličine 10^{-4} - 10^{-5}, u zavisnosti o podacima). PMI uz stanovit oprez pri uporabi, ponajprije pri niskoučestalim jezičnim jedinicama, daje nadasve upotrebljive rezultate.

Pri procesu ocjenjivanja, za svako se kontekstom generirano pravilo bilježi uz koliko je ukupno naziva rezultat pretrage bio veći od nule. Ona pravila koja uz manje od jedne trećine naziva vraćaju rezultat pretrage jednak nuli izbacuju se iz popisa pravila. Navedeno filtriranje je uvedeno kako bi se spriječilo usvajanje pravila koja reagiraju samo na određena imena poput *sv * i Jakov*. Odabrana vrijednost od jedne trećine može se, naravno, promijeniti.

5.2.3.2 Odabir primjera

PMI ocjena tretira se kao značajka koja se koristi kao ulaz NBC-u. Kako bi se NBC naučio klasificirati, prvo je potrebno adekvatno odabrati primjere koji će biti prezentirani algoritmu za učenje. Kako je ovo nenadzirana tj. slabo nadzirana metoda, odabir je potrebno automatizirati. Početni skup naziva koristi se za izradu dvaju skupova pozitivnih i negativnih primjera. Jedan skup koristi se za određivanje praga PMI koji dijeli pozitivne od negativnih primjera za učenje, a drugi se koristi za

određivanje aposteriornih uvjetnih vjerojatnosti NBC-a[41]. Dva potrebna skupa biraju se tako da se najprije odredi parametar *k* koji govori koliko primjera će se iz svakog popisa naziva izdvojiti za svaki od skupova:

$$k = \min(\frac{|P|}{2}, 20) \tag{5.10}$$

$$|P| = \min_{i=1}^{3}(|P_i|) \tag{5.11}$$

gdje je:

$|P_i|$ ≡ broj naziva u popisu kategorije *i*

Parametar *k* ograničen je maksimalnim iznosom 20 zbog uštede resursa i skraćivanja vremena izvođenja. Nakon određivanja parametra *k*, skupovi primjera biraju se na sljedeći način:

$$\begin{aligned} T_i^+ &\subset P_i \quad \text{tako da} \quad \left|T_i^+\right| = k \\ T'^+_i &\subset P_i \quad \text{tako da} \quad \left|T_i^+\right| = k \wedge T_i^+ \cap T'^+_i = \{\varnothing\} \end{aligned} \tag{5.12}$$

$$\begin{aligned} T_i^- &\subset \bigcup_j P_j \quad \text{tako da} \quad i \neq j \wedge \left|T_i^+\right| = k \\ T'^-_i &\subset \bigcup_j P_j \quad \text{tako da} \quad i \neq j \wedge \left|T_i^+\right| = k \wedge T_i^+ \cap T'^+_i = \{\varnothing\} \end{aligned} \tag{5.13}$$

gdje je:

i=1,2,3

T_i^+, T'^+_i ≡ skupovi pozitivnih primjera za učenje

T_i^-, T'^-_i ≡ skupovi negativnih primjera za učenje

Skup *X* bira se pseudoslučajno. U slučaju pozitivnih primjera bira se tako da se iz popisa kategorije za koju se primjeri traže nasumce odabere *k* primjera dok se u

[41] v. poglavlje 5.2.3.3

slučaju negativnih primjera, *k* primjera traži iz svih drugih kategorija. Dakle, dio pozitivnih primjera jedne klase postaje skup negativnih primjera druge klase [22]. Drugi skup primjera bira se na potpuno jednak način, s time da se skupovi biraju tako da budu disjunktni. Drugi skup u nastavku je označen s T'^{+}_{i}, T'^{-}_{i}.

Nad negativnim skupovima vrši se ocjenjivanje, a nad pozitivnima ne jer su pozitivni ocijenjeni u prethodnom koraku. Cijena ovog ocjenjivanja je:

$$N_3 = 6kc \tag{5.14}$$

5.2.3.3 Učenje parametara Bayesovog klasifikatora

PMI ocjene koriste se kao značajke koje predstavljaju ulaz u NBC temeljenom na sljedećoj formuli:

$$P(\varphi \mid f_1, f_2, ... f_n) = \frac{P(\varphi)\prod_i P(f_i \mid \varphi)}{P(\varphi)\prod_i P(f_i \mid \varphi) + P(\neg\varphi)\prod_i P(f_i \mid \neg\varphi)} \tag{5.15}$$

Formula (5.15) iskazuje da je vjerojatnost da je činjenica φ ispravna s obzirom na značajke $f_1, f_2, ... f_n$, a pod pretpostavkom nezavisnosti značajki[42].

Pretvaranje PMI ocjene u uvjetne vjerojatnosti potrebne formulu (5.15) je neposredno [22]. Prvo se koristi prvi skup pozitivnih i negativnih primjera za određivanje praga PMI ocjene. Spomenuti prag razdvaja pozitivne od negativnih primjera svake od kategorija, za svako od kontekstom generiranih pravila. Prag se postavlja tako da se jednostavno izračuna aritmetička sredina najmanje PMI ocjene pozitivnih primjera i najveće PMI ocjene negativnih primjera (računanje centroida primjera bi u slučajevima velikih odstupanja ocjena moglo dati loše vrijednosti praga):

$$prag_i = \frac{\min_{p=1}^{k}\{PMI(NE_p, C_i)\} + \max_{q=1}^{k}\{PMI(NE_q, C_i)\}}{2} \quad , NE_p \in T_i^+, NE_q \in T_i^- \tag{5.16}$$

[42] Unatoč čestom nepoštivanju pretpostavke u praksi, klasifikator daje izvanredne rezultate
C. Elkan, (1997), Naïve Bayesian Learning

Zatim se koristi drugi skup pozitivnih i negativnih primjera za određivanje aposteriornih uvjetnih vjerojatnosti i to brojanjem pozitivnih i negativnih primjera čiji je PMI iznad ili ispod praga za svaki od kontekstnih pravila odabrane kategorije naziva:

$$P(PMI > prag \mid klasa) = \frac{|X|}{|T'^{+}_{i}|} \quad , X \subseteq T'^{+}_{i} , \forall x \in X : PMI(x, C_i) > prag_i \quad (5.17)$$

$$P(PMI > prag \mid \neg klasa) = \frac{|Y|}{|T'^{-}_{i}|} \quad , Y \subseteq T'^{-}_{i} , \forall y \in Y : PMI(y, C_i) > prag_i \quad (5.18)$$

te:

$$\begin{aligned} P(PMI \leq prag \mid klasa) &= 1 - P(PMI > prag \mid klasa) \\ P(PMI \leq prag \mid \neg klasa) &= 1 - P(PMI > prag \mid \neg klasa) \end{aligned} \quad (5.19)$$

Kada su sve aposteriorne uvjetne vjerojatnosti izračunate, proces učenja je završen.

Jedan od problema NBC-a je polarizacija procijenjenih vjerojatnosti blizu nule ili jedinice koje onda vrlo često nisu točne. Međutim, klasifikator je iznenađujuće efikasan zato što treba samo donijeti jednostavnu odluku o klasifikaciji.

5.2.3.4 Daljnje filtriranje konteksta

Promatranjem kontekstom generiranih pravila nakon procedure ocjenjivanja utvrđeno je da se katkada može dogoditi da aposteriorna uvjetna vjerojatnost nekog kontekstom generiranog pravila, $P(PMI > prag \mid klasa)$, bude jednaka nuli. To se događa kada se navedeno pravilo kao kontekst nalazi samo uz rijetke nazive te kada se vrši ocjenjivanje nad skupom primjera koji ne sadrži takav rijedak naziv. Takva pravila prema formuli (5.15) za rijetke nazive daje konačnu vjerojatnost jednaku nuli, bez obzira na ostale parametre formule, što vodi nepotrebnom smanjenju odziva. Zato se takva pravila uklanjaju sa skupa kontekstom generiranih pravila

Primjer takvog pravila *je * Zelina*, pravilo će uglavnom vratiti naziv *Ivan*. Ukoliko se slučajno Google-u pošalje upit za *Ana Zelina*, kako je *P*=0, ime *Ana* će se automatski odbaciti.

5.2.4. Procesiranje naziva

Postupak procesiranja naziva je proces ekstrakcije novih naziva koristeći prethodno odabrana kontekstom generirana pravila. Dobiveni nazivi se filtriraju i prepuštaju klasifikatoru na daljnju klasifikaciju.

5.2.4.1 Ekstrakcija naziva

Ponovno koristeći Google kao izvor podataka, kontekstom generirana pravila se šalju kao upiti te se iz rezultirajućih stranica tj. isječaka rezultata ekstrahiraju novi nazivi. Za pravila oblika X * i * X se ekstrahiraju sve pojavnice desno, odnosno lijevo od pojavnice X pisane velikim početnim slovom te se tretiraju kao jedan kandidat za naziv (samo u slučaju kategorija lokacije i organizacije, za kategoriju osobe se ekstrahira samo jedna pojavnica). Za pravilo oblika X * Y se kandidatom tretira sve između pojavnica X i Y, bez obzira na tipografiju. Za svakog se ekstrahiranog kandidata bilježi frekvencija pojavljivanja, tj. koliko je puta taj kandidat pronađen u korpusu isječaka. Ekstrakcija naziva implementirana je kao višedretveni proces.

Cijena ekstrakcije novih naziva je:

$$N_4 = 3xc \tag{5.20}$$

Primjer

Slanjem upita Google-u sa sadržajem *dr sc* *, jedan od mogućih vraćenih rezultata je:

...i športa. Prof. dr. sc. Dragan Primorac. Osobni podaci....

iz čega se ekstrahira *Dragan*, kandidat za kategoriju osobe.

5.2.4.2 Filtriranje kandidata

Korak filtriranja kandidata za naziv sličan je koraku filtriranja kandidata za kontekstom generirana pravila:

- filtriranje stop riječi (hrvatskih i engleskih, moguće i drugih jezika)
- filtriranje postojećih naziva iz prethodnih popisa naziva
- brisanje naziva s frekvencijom pojavljivanja jednakom jedan
- brisanje naziva duljine jedan (obično inicijali)
- uzimanje prvih *d* naziva po frekvenciji pojavljivanja

Nakon filtriranja (koje se može smatrati *offline* ocjenjivanjem) vrši se ocjenjivanje kandidata koristeći PMI mjeru između skupa novih naziva (kandidata) i skupa kontekstom generiranih pravila.

5.2.5. Klasifikacija naziva

Za sam postupak klasifikacije potrebne su vrijednosti PMI mjere za pojedine nazive. Ocjenjivanje kandidata za nazive u potpunosti je jednako procesu ocjenjivanja naziva u poglavlju 5.2.3.1.

Cijena ovog ocjenjivanja je:

$$N_5 = 3d(c+1) \tag{5.21}$$

Ocijenjeni kandidati se zatim propuštaju kroz naivni Bayesov klasifikator naučen u 5.2.3 koji će ocijeniti koje kandidate proglasiti nazivima, a koje odbaciti koristeći formulu (5.15).

Dobivene PMI vrijednosti se provjeravaju s pragovima te se formiraju značajke – aposteriorne uvjetne vrijednosti. Apriorne vjerojatnosti da je kandidat za naziv u kategoriji φ i da nije su jednake tako da $P(\varphi)$ i $P(\neg\varphi)$ nestaju iz formule (5.15). U konačnici se izračunava vjerojatnost da zadani naziv pripada kategoriji, $P(\varphi,f_1,f_2,...f_n)$, te ukoliko prelazi postavljeni prag (u implementaciji se koristi prag jednak 0,5), proglašava se pozitivnim nazivom i dodaje odgovarajućem popisu naziva.

Otvorena je mogućnost sljedeće iteracije koja bi se zasnivala na popisima proširenima novim nazivima.

Iz kurioziteta, modulu je još dodana procedura za dohvat morfoloških oblika pronađenih naziva koristeći neku od tražilica koja podržava korjenovanje riječi (u ovome slučaju tražilica Pogodak.hr).

5.3. „Cijena“ izvođenja

Zbrajanjem formula (5.9), (5.14), (5.20) i (5.21) formira se „cijena“ izvođenja sustava izražena u broju upita koji se šalju tražilici Google:

$$N_{uk} = 3\left[n(x+c+1)+c(2k+x)+d(c+1)\right] \tag{5.20}$$

gdje su:

n ≡ broj početnih naziva

x ≡ broj stranica koje se skida

c ≡ broj kontekstom generiranih pravila

d ≡ broj željenih naziva

k ≡ broj primjera za učenje klasifikatora

Formula (5.20) predstavlja maksimalan broj upita koji može biti poslan, u praksi je taj broj redovito niži zbog mogućeg manjeg broja upita u pojedinim podmodulima sustava. Npr. pri početnom preuzimanju stranica, ukoliko je traženi pojam rijedak, nije moguće skinuti punih x stranica nego manje. Isto tako vrijedi i za ocjenjivanje naziva koji se ocjenjuju s filtriranim kontekstima kojih u konačnici može biti manji broj od c.

Koristeći formulu (5.20) lako se može izračunati broj upita koji se šalje tražilici, a time i aproksimirati vrijeme izvođenja (pod pretpostavkom dovoljno brze veze prema Internetu čime bi se smanjio utjecaj prijenosa podataka) koje iznosi:

$$t \approx \frac{N_{uk}}{8}\,\text{min} \tag{5.21}$$

Važno je napomenuti da navedena procjena ne uzima u obzir brzinu računala nego se isključivo temelji na implementiranoj vremenskoj zadršci. Čitav je proces spor isključivo zbog vremenske zadrške pri preuzimanju stranica te se formula (5.21) izvodi iz aritmetičke sredine aproksimiranih vremena za preuzimanje stranica s maksimalnim i minimalnim brojem dretvi.

Konačna procjena bi bila složenija, međutim izvedena je vrijednost dovoljna jer daje približan vremenski iznos za koji je kroz nekoliko pokretanja sustava utvrđeno da je zaista blizak stvarnom vremenu izvođenja.

Primjer

Za n=15 (k=7), x=5, c=10, d=25, dobiva se N_{uk}=2115, što čini procijenjeno vrijeme izvođenja približno jednakim 4 sata i 20 minuta. Navedeni parametri su korišteni pri eksperimentima s drugim jezicima[43]. Formula (5.20) kao rezultat vraća 2115 dok je pri pokusu s engleskim jezikom preuzeto konkretnih 1977 stranica. Odstupanje od 138 upita je jasno vidljivo pomnijim promatranjem pojedinih faza rada sustava. Vrijeme izvođenja pokusa je bilo približno jednako 4 i pol sata, što je usporedivo da procijenjenim vremenom izvođenja.

Za n=30 (k=15), x=10, c=20, d=1000, dobiva se N_{uk}=68190 što čini procijenjeno vrijeme izvođenja približno jednakim 6 dana.

Ukoliko su željene stranice spremljene u priručni spremnik, ne postoji vremenska zadrška pri preuzimanju i čitav se proces odvija u bitno kraćem vremenu koji zavisi o brzini obrade i količini podataka koje valja obraditi. Tako se prvi primjer izvršava za oko 2 do 3 minute.

Iz navedenoga se vidi koliku vremensku uštedu čini priručni spremnik.

[43] v. poglavlje 6.6

6. Testiranje i vrednovanje rezultata

U ovom poglavlju biti će prikazani rezultati testiranja sustava za ekstrakciju naziva. Sustav za obilježavanje neće se testirati zbog nedostatka tekstova za testiranje, te zbog činjenice da bi rezultati neizbježno bili loši – po pretpostavci, umjerena do visoka preciznost, ali vrlo nizak odziv zbog nedovoljno velikih popisa i zbog nedostatka neke druge metode strojnog učenja[44] koja bi samo obilježavanje odradila kvalitetnije. Naglasak ovoga rada je bio na ekstrakciji novih naziva, bez korištenja ikakvih dodatnih saznanja i bez uplitanja u rad metode.

Prikazani rezultati u nastavku su vrednovani mjerama opisanim u poglavlju 2.4 s time da su se odziv, a time i F-mjera računali s obzirom na nazive koje je sustav odbacio nakon postupka klasifikacije. Iz tih podataka dobivaju se svi potrebni podaci za izračun željenih mjera. Važno je napomenuti da se broj željenih i broj ekstrahiranih naziva razlikuju jer broj željenih naziva označava količinu naziva koji će se klasificirati, a broj ekstrahiranih označava količinu naziva koja je zadovoljila klasifikator.

Vrednovanje konačnih popisa je izvršeno ručno, koristeći raspoložive resurse s Interneta. Pod pojedinim kategorijama dozvoljeni su svi nazivi određeni MUC-7 specifikacijama, bez obzira da li su bili ekstrahirani koristeći diferencirane ili nediferencirane početne popise.

Nediferencirani početni popisi su popisi u kojima prevladava uglavnom jedna potkategorija željene kategorije i oni su korišteni u skoro svim eksperimentima[45]. Tako se popis imena sastojao od poznatih hrvatskih imena, popis organizacija od poznatih hrvatskih tvrtki, a popis lokacija od hrvatskih gradova. Popisi su izrađeni koristeći raspoložive resurse s Interneta.

---

[44] v. poglavlje 3.2
[45] osim u eksperimentu u poglavlju 6.5

Pri vrednovanju naziva poštivala su se slijedeća pravila:

- Dozvoljeni su nazivi bez hrvatskih dijakritika (nazivi u kojima su *č*, *ć*, *š*, *đ*, *ž* zamijenjeni u *c*, *c*, *s*, *d*, *z*)
- Prihvaćane su samo cjelovite organizacije (dakle ne *Com* umjesto *T-Com* i sl.)
- Nisu prihvaćani nazivi jedan iza drugog poput *Zagreb Hrvatska* i sl.
- Prihvaćena su imena i prezimena nehrvatskog podrijetla

Eksperimenti su rađeni na uglavnom malim početnim popisima zbog vremenske zahtjevnosti izvođenja. Uz to je zbog istog razloga i lakše evaluacije tražen i manji broj naziva *d*.

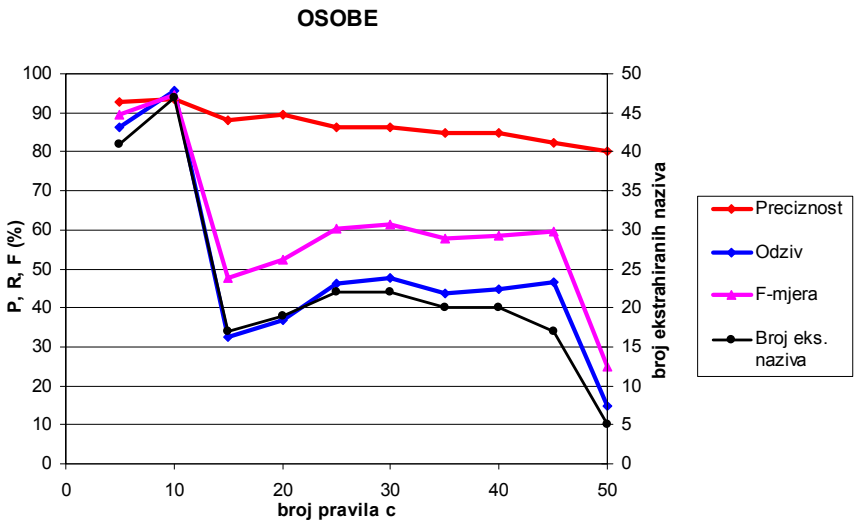
Važno je napomenuti da se metoda oslanja o odabir primjera[46] koji je pseudoslučajan, te da je za potrebe eksperimentiranja početno „sjeme“ pseudoslučajnosti postavljeno na konstantnu vrijednost tokom svih eksperimenata. Odabir početnog sjemena može imati značajan utjecaj na konačne rezultate.

6.1. Utjecaj broja kontekstom generiranih pravila

U ovom se eksperimentu provjerava utjecaj broja kontekstom generiranih pravila na preciznost, odziv, F-mjeru i broj ekstrahiranih naziva. Broj konteksta *c* se kreće od 5 do 50 s korakom 5. Ekstrahira se 50 naziva.

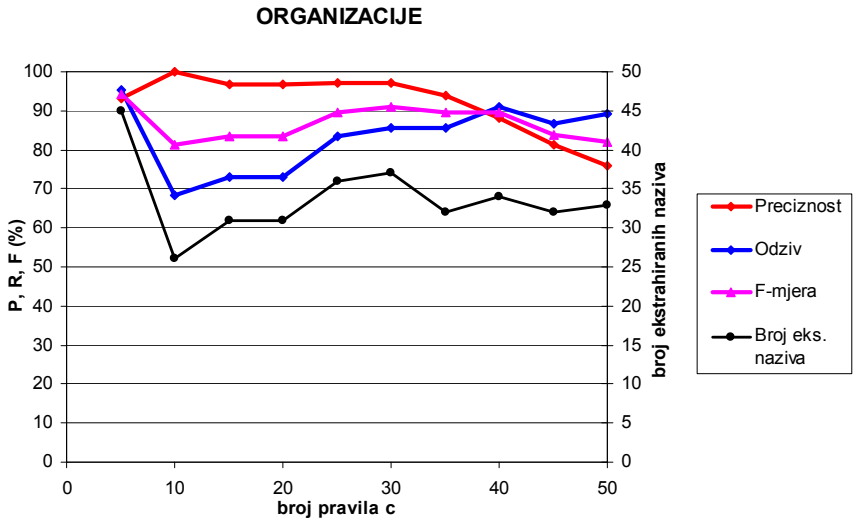
Očekivano je da s većim brojem pravila koja sudjeluju u klasifikaciji preciznost i broj ekstrahiranih naziva padaju.

[46] v. poglavlje 5.2.3.2



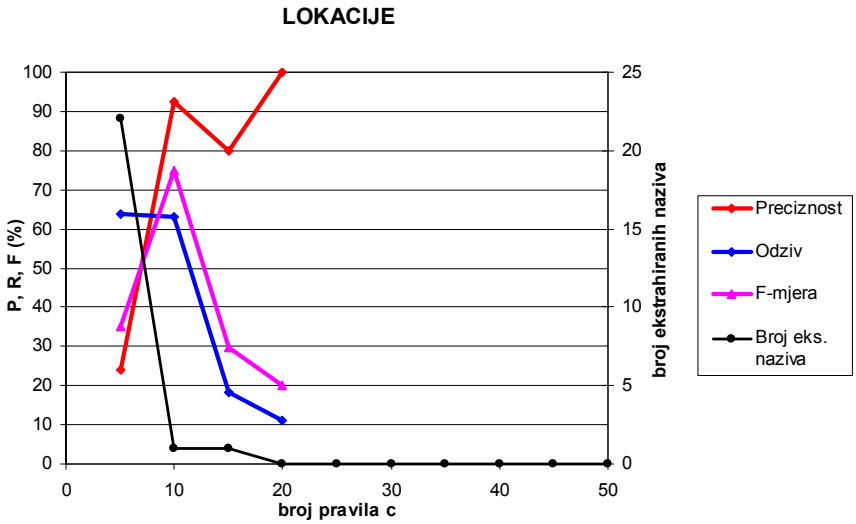
Slika 15. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju kontekstom generiranih pravila za kategoriju osobe

Kategorija osobe pokazuje dobru preciznost koja se blago spušta s maksimalnih 92,68% na 80% pri 50 korištenih pravila. Promjena odziva je nepravilna i uglavnom niska, s maksimumom od 95,65% pri 10 korištenih pravila, a minimumom od 14,81% pri 50 korištenih pravila.. Nizak odziv se odrazio i na broj ekstrahiranih pravila koji se spušta s maksimalnih 41 na 5. Optimalan broj pravila po F-mjeri iznosi 10.



Slika 16. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju kontekstom generiranih pravila za kategoriju organizacije

Kategorija organizacije pokazuje najbolje rezultate unatoč većem padu preciznosti na 75,75% pri 50 korištenih pravila. Odziv je umjereno visok s minimumom u 68,42%. Broj ekstrahiranih naziva je najveći među svim kategorijama. Optimalan broj pravila po F-mjeri iznosi 40.



Slika 17. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju kontekstom generiranih pravila za kategoriju lokacije

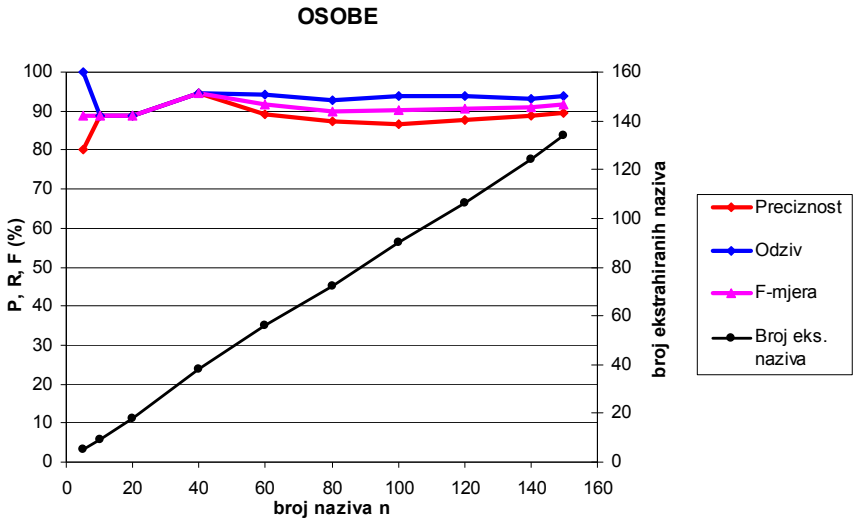
Kategorija lokacije daje najlošije rezultate, štoviše, nakon oko 25 pravila više se ne ekstrahiraju novi nazivi. Preciznost je najveća pri korištenih 20 pravila. Odziv značajno opada sa 63,63% na 20% (pri 20 pravila). Optimalan broj pravila po F-mjeri iznosi 10.

Globalno gledano, optimalan broj pravila iznosi 10 i kao takav korišten je u daljnjim eksperimentima.

6.2. Utjecaj broja naziva

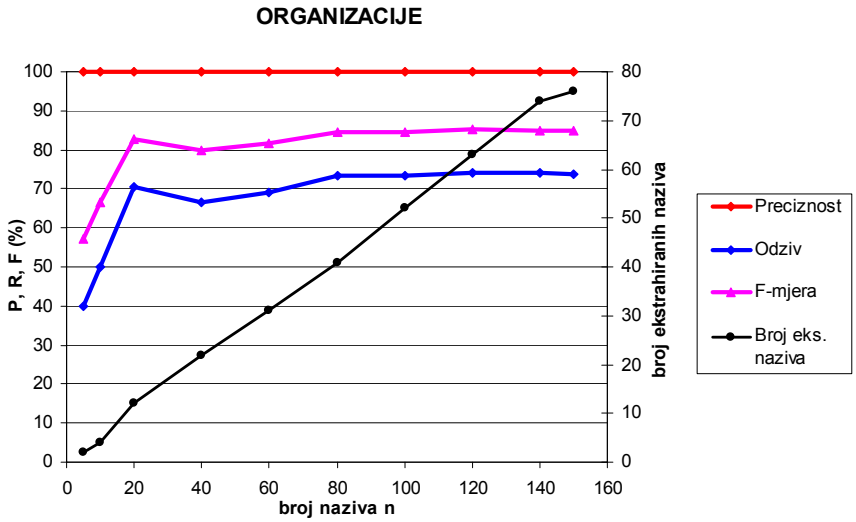
Ovaj eksperiment provjerava utjecaj broja željenih naziva na preciznost, odziv, F-mjeru i broj ekstrahiranih naziva. Broj naziva *n* kreće se od 10, 20 do 160 s korakom 20. Poželjna je što pravilnija karakteristika tj. da mjere što manje odstupaju od konstantnog iznosa jer broj željenih naziva ne bi trebao značajno utjecati na

konačni rezultat. Također je poželjna linearna karakteristika broja ekstrahiranih naziva s nagibom što bližim jedan.



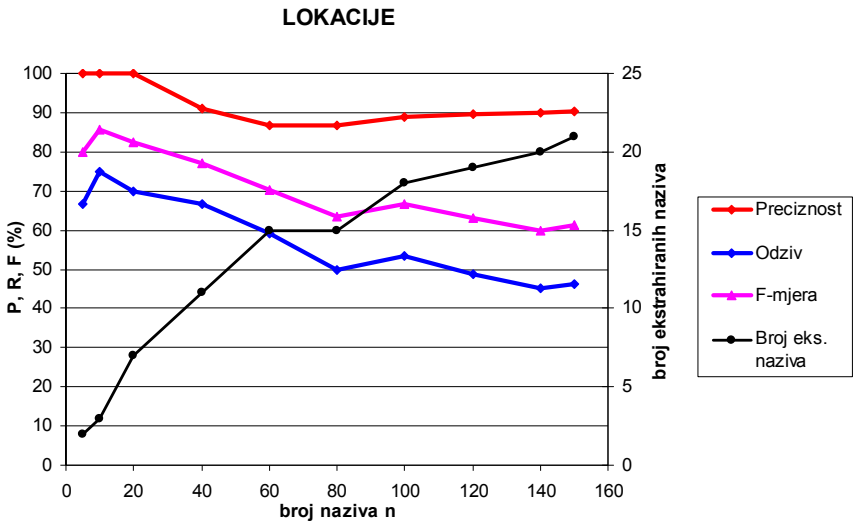
Slika 18. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o željenom broju naziva za kategoriju osobe

Kategorija osobe daje odlične rezultate za sve tri mjere. Preciznost je visoka s minimumom od 86,66% pri 100 željenih naziva. Odziv je također visok s minimumom od 92,64% pri 80 željenih naziva. Ukupna karakteristika mjera ove kategorije je približno konstantna s manjim odstupanjima što je i željen rezultat. Broj ekstrahiranih naziva se ponaša linearno s visokim brojem ekstrakcija.



Slika 19. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o željenom broju naziva za kategoriju organizacije

Kategorija organizacije kroz cijeli eksperiment daje savršenu preciznost od 100% što je izuzetan rezultat koji ipak treba gledati s dozom opreza. Odziv varira od niskih 40% do umjerenih 74,11%. Broj ekstrahiranih naziva linearno ovisi o željenom broju naziva s nagibom manjim od jedan. F-mjera pokazuje manja odstupanja od konstantnog iznosa od 80 željenih naziva.



Slika 20. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o željenom broju naziva za kategoriju lokacije

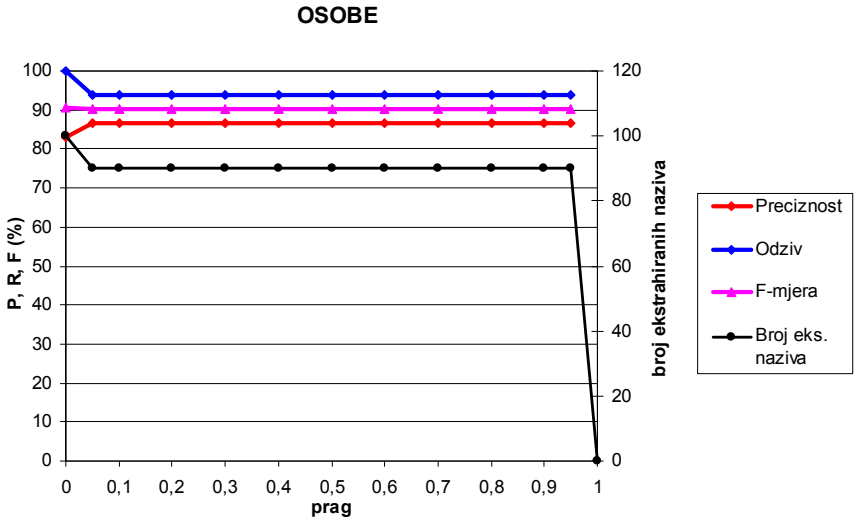
Kategorija lokacije pokazuje visoku preciznost, ali s padajućim odzivom što se u konačnici manifestira opadanjem F-mjere u ovisnosti o broju željenih naziva. Broj ekstrahiranih naziva odstupa od linearnog rasta te je u konačnici iznimno mali od tek 21 ekstrahiranog naziva od željenih 160.

Izmjereno ponašanje sustava u ovisnosti o broju naziva je zadovoljavajuće za kategorije osobe dok je eksperiment pokazao da je kategorija lokacije osjetljiva na željeni broj naziva što nije dobro svojstvo.

6.3. Utjecaj praga Bayesovog klasifikatora

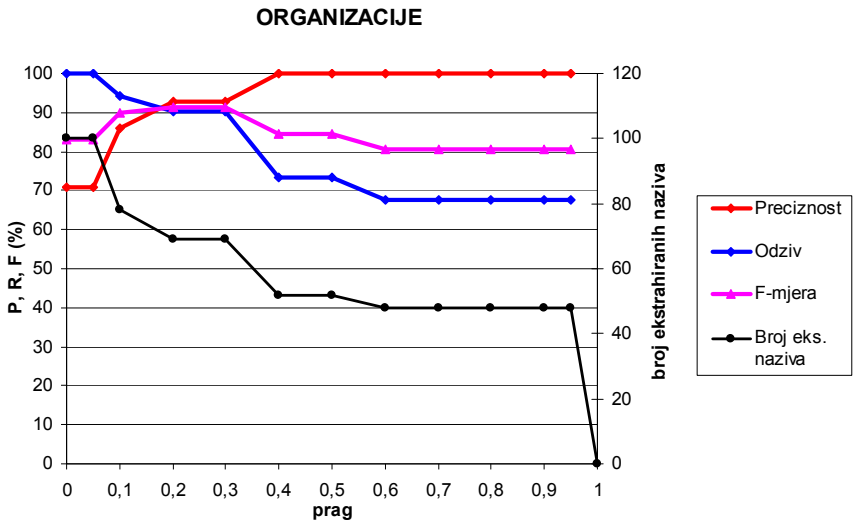
Ovaj eksperiment provjerava utjecaj broja praga naivnog Bayesovog klasifikatora koji je potrebno prijeći za pozitivnu klasifikaciju na preciznost, odziv, F-mjeru i broj ekstrahiranih naziva. Prag se kreće od 0 do 1 s korakom od 0,1 uz 0,05 i 0,95. Broj željenih naziva iznosi 100.

Poželjno je da se promjenom praga mogu mijenjati željeni preciznost i odziv, a da je F-mjera po mogućnosti nepromjenjiva. Očekivano je da se s porastom vrijednosti praga, vrijednost preciznosti povećava, a vrijednost odziva smanjuje tj. očekivan je njihov obrnuto proporcionalan odnos.



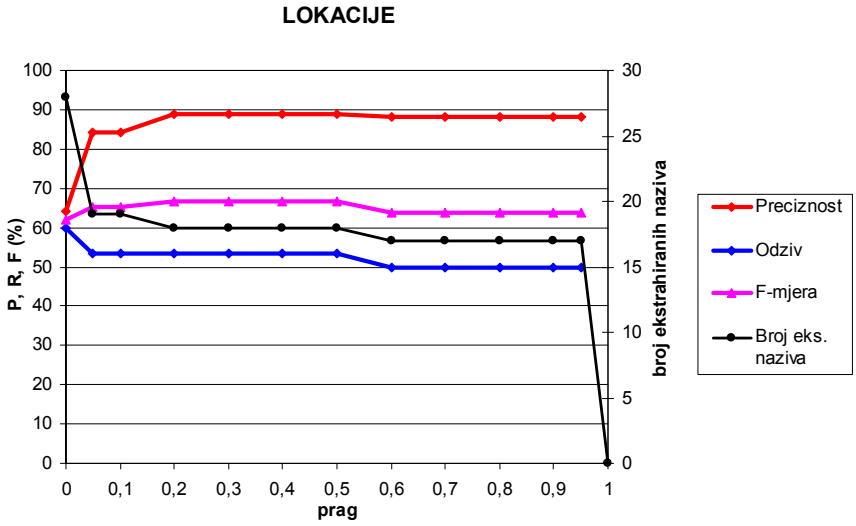
Slika 21. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o pragu naivnog Bayesovog klasifikatora za kategoriju osobe

Na kategoriju osobe prag, zanemarujući vrijednosti do 0,05, prag uopće ne utječe što znači da je klasifikator odabrane nazive klasificirao s iznimno visokom vjerojatnošću koja je vrlo bliska jedinici. Promjena do praga 0,05 pokazuje nekolicinu naziva s jako niskom vjerojatnošću. Broj ekstrahiranih naziva je također konstantan.U konačnici, naivni Bayesov klasifikator u slučaju ove kategorije polarizira vjerojatnosti naziva ili jako blizu nuli ili jako blizu jedinici.



Slika 22. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o pragu naivnog Bayesovog klasifikatora za kategoriju organizacije

Od svih kategorija, prag najviše utječe na kategoriju organizacije što je vidljivo iz obrnuto proporcionalnog odnosa preciznosti i odziva. Preciznost se kreće od 71% do 100%, a odziv od 100% do 67,60%. Broj ekstrahiranih naziva prati porast preciznosti odnosno pad odziva. U ovome slučaju, vrijednost praga omogućava promjenu preciznosti i odziva.

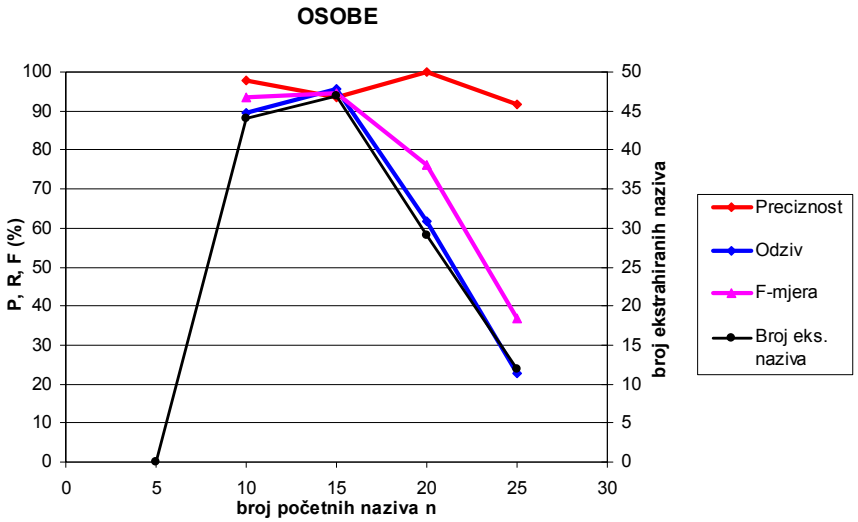


Slika 23. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o pragu naivnog Bayesovog klasifikatora za kategoriju lokacije

Na kategoriju lokacije, prag praktički ne utječe. Jedine promjene se događaju do vrijednosti 0,1 gdje preciznost poraste za 20%, a odziv opada za oko 6% te između 0,5 i 0,6 gdje preciznost neznatno opada uz slabiji pad odziva. Broj ekstrahiranih naziva prati promjenu odziva.

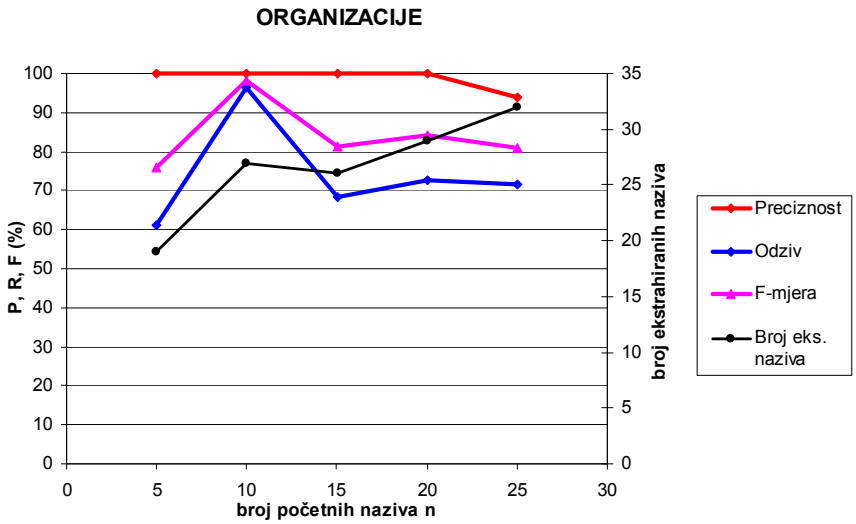
6.4. Utjecaj broja početnih naziva

U ovom se eksperimentu provjerava utjecaj broja kontekstom generiranih pravila na preciznost, odziv, F-mjeru i broj ekstrahiranih naziva. Broj konteksta *c* se kreće od 5 do 50 s korakom 5. Broj željenih naziva iznosi 50. Poželjno je da s većim brojem početnih naziva raste preciznost.



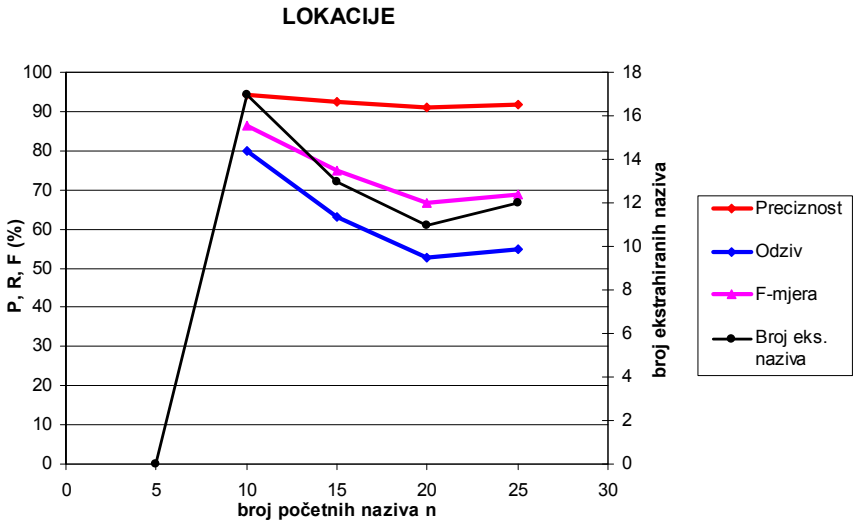
Slika 24. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju naziva u početnim popisima za kategoriju osobe

U slučaju kategorije osobe, preciznost oscilira približno oko 95% uz znatan pad odziva na 22,91%. Sustav ne vraća niti jedan novi naziv za 5 početnih naziva po popisu. Optimalan iznos početnih naziva je 15, s F-mjerom od 89,58%. U slučaju 15 početnih naziva, broj ekstrahiranih naziva iznosi 44.



Slika 25. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju naziva u početnim popisima za kategoriju organizacije

Kategorija organizacije pokazuje konstantan iznos preciznosti od 100% do 25 početnih naziva gdje pada na 93,75%. Odziv prvo poraste s 91,29% na visokih 96,42% što predstavlja izuzetan rezultat. Zatim pada na približno 70%. Začudo, broj ekstrahiranih naziva raste s povećanjem broja početnih naziva. Optimalan broj početnih naziva iznosi 10 s F-mjerom od 98,18%.



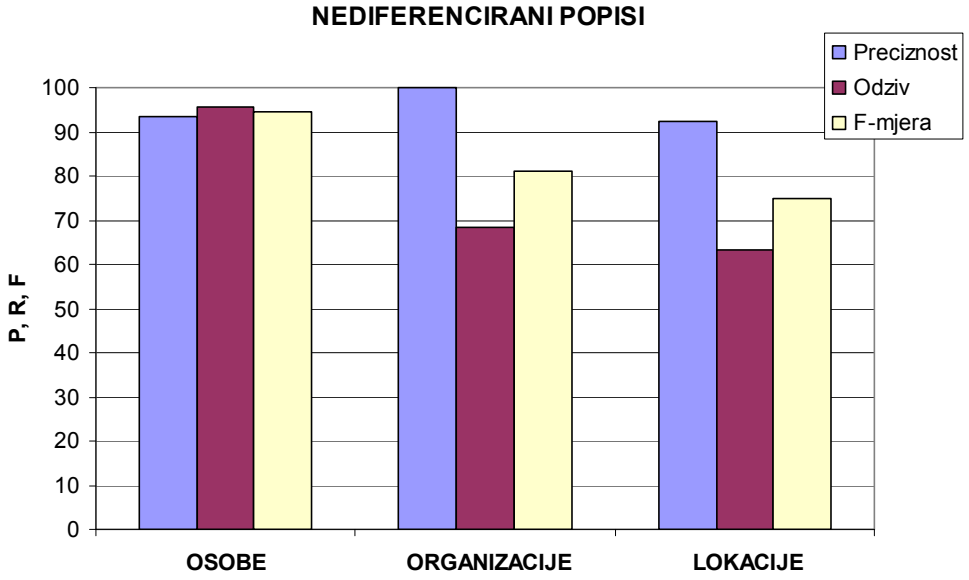
Slika 26. Ovisnost preciznosti, odziva, F-mjere i broja ekstrahiranih naziva o broju naziva u početnim popisima za kategoriju lokacije

Kategorija lokacije pokazuje slabiji pad preciznosti uz jače opadanje odziva koji u stopu prati i pad broja ekstrahiranih naziva. Optimalan broj početnih naziva je 10 pri kojemu je F-mjera 86,48%.

Promatrajući rezultate eksperimenata, može se zaključiti kako povećanjem broja početnih naziva, F-mjera sustava pada. Međutim za takav se zaključak treba ipak izvršiti više eksperimenata s kontroliranijim popisima od onih korištenih u ovom eksperimentu.

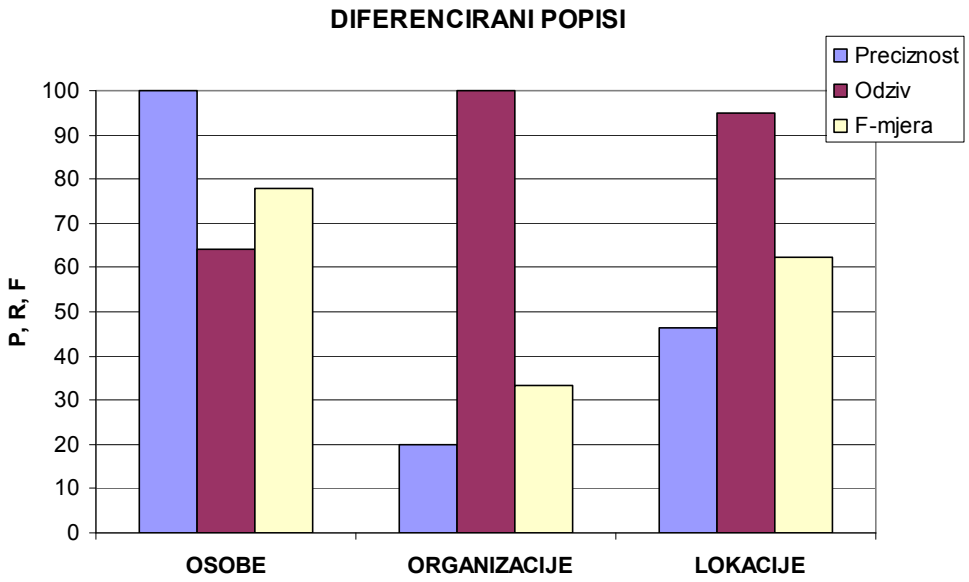
6.5. Utjecaj diferencijacije popisa

Cilj ovog eksperimenta je utvrditi utjecaj diferencijacije popisa na konačne rezultate. Popis osoba sastoji se od jednako zastupljenog broja prezimena i imena, popis organizacija od vladinih organizacija, zavoda, tvrtki do škola i fakulteta, te popis lokacija od voda, otoka i planina do geopolitičkih i geoloških lokacija. Broj željenih naziva postavljen je na 50.



Slika 27. Rezultati eksperimenta s nediferenciranim popisima

Nediferencirani popis pokazuje visoku preciznost, iznad 90% u sve tri kategorije, s visokim odzivom u kategoriji osobe te nižim odzivima u preostale dvije kategorije. Konkretni ekstrahirani rezultati mogu se vidjeti u dodatku B.



Slika 28. Rezultati eksperimenta s diferenciranim popisima

Diferencirani popis daje najvišu preciznost od 100% u kategoriji osobe što ukazuje na neosjetljivost zajedničkog pojavljivanja imena i prezimena u popisu. Preciznost u kategoriji organizacije je izrazito niska, tek 20% uz maksimalan odziv od 100% što može biti objašnjeno nemogućnošću pronalaska dovoljno diskriminatornih konteksta koristeći tako različite nazive. Štoviše, dobiveni konteksti poput * *Zagreb* i * *hrvatski* nisu uopće diskriminatorni niti za jednu kategoriju. Kategorija lokacije pokazuje nešto višu, ali svejedno nisku, preciznost od 46,34% uz visoki odziv od 95%. Konteksti za kategoriju lokacije također nisu diskriminatorni, ali unatoč tome daju bolje rezultate.

Iz navedenoga se može zaključiti kako sustav postiže visoku preciznost samo za nediferencirane popise, dok za diferencirane, preciznost pada, barem u kategorijama organizacije i osobe, na iznimno niske vrijednosti. Za diferencirane popise sustav nije u mogućnosti naći diskriminatorna pravila koja se mogu upotrijebiti nad svim nazivima iz kategorije.

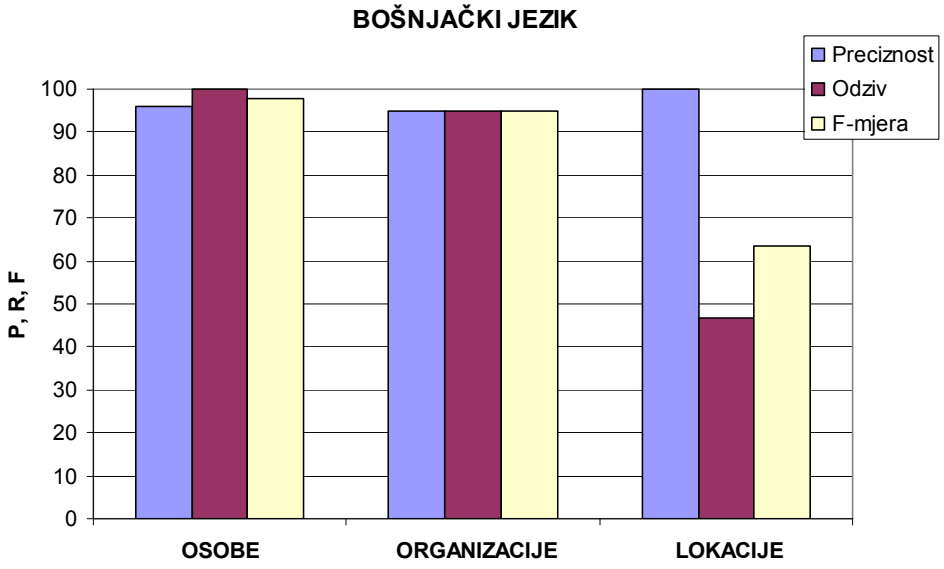
6.6. Eksperimenti s drugim jezicima

Kako je u sustav ukomponirana tek nekolicina pravila koja su uglavnom jezično neovisna, moguće je iskoristiti sustav i nad drugim jezicima. Eksperimenti su izvršeni s dva jezika susjednih država: bošnjačkim i slovenskim, te s engleskim jezikom.

Početni popisi naziva se sastoje od 15 naziva karakterističnih za svaki jezik. Broj kontekstno generiranih pravila postavljen je na 10, a broj ekstrahiranih naziva na 25, zbog lakše provjere istih. Provjera dobivenih rezultata je ručna, koristeći resurse tražilice Google.

6.6.1. Bošnjački

Popisi imena sastavljeni su koristeći resurse s Interneta pronađene tražilicom Google. Popis imena osoba se sastoji od većinom muslimanskih imena[47], lokacije su gradovi u Bosni i Hercegovini dok su organizacije slučajno odabrane organizacije



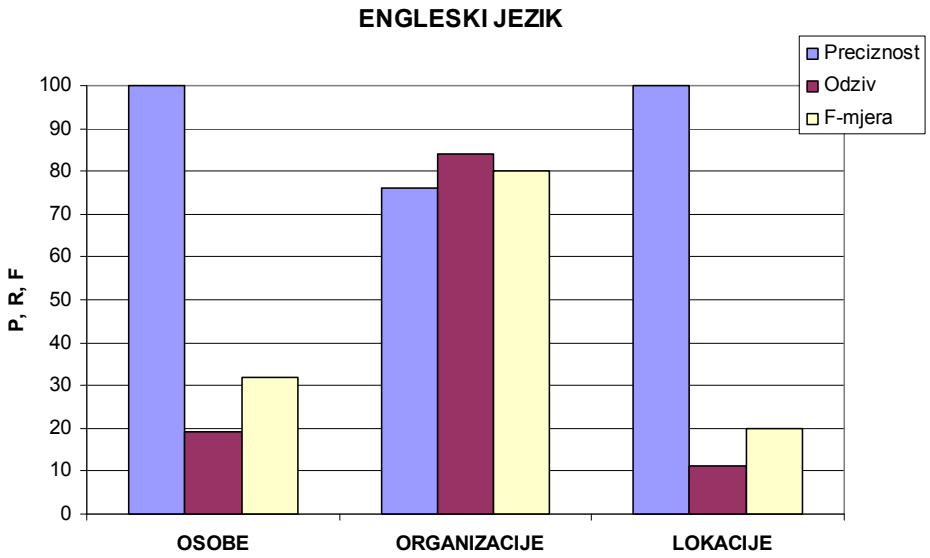
Slika 29. Rezultati eksperimenta s bošnjačkim jezikom

[47] Muslimanska imena: Uloga socijalističke revolucije u bošnjačkoj porodici, http://www.bhdani.com/arhiva/182/t18208.shtml

Preciznost sve tri kategorije je iznenađujuće visoka, veća od 90%, s 100%-tnom preciznošću kategorije lokacija. Odziv je za prve dvije kategorije također visok, dok je za lokacije ispod 50%. Dobiveni rezultati se pripisuju kvalitetnim pravilima za prve dvije kategorije, dok su pravila za lokacije bila nediskriminatorna te su stoga dala niži odziv. Globalno gledano, kvaliteta dobivenih naziva je više nego zadovoljavajuća. Broj ekstrahiranih naziva po kategorijama redom iznosi 25, 19 i 7. Konkretni ekstrahirani rezultati mogu se vidjeti u dodatku B.

6.6.2. Engleski

Popis imena osoba sastoji se od najpopularnijih imena[48] u Sjedinjenim Američkim Državama 2000. godine, popis organizacija od imena trenutno aktivnih tvrtki[49], a popis lokacija od imena većih gradova dobiven kombiniranjem više izvora.



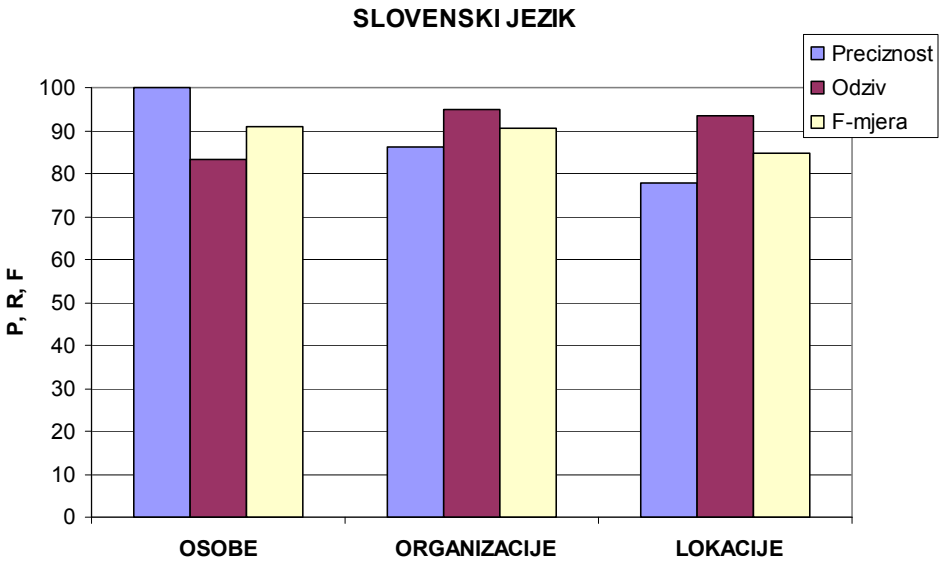
Slika 30. Rezultati eksperimenta s engleskim jezikom

[48] http://en.wikipedia.org/wiki/List_of_the_most_popular_names_in_the_2000s_in_the_United_States
[49] http://en.wikipedia.org/wiki/List_of_United_States_companies

Preciznost kategorija osobe i lokacije je izvrsna, dok kategorija organizacije pokazuje ne tako lošu preciznost od 75%. Međutim, odzivi kod engleskog jezika su iznimno niski za osobe i lokacije, dok je za organizacije odziv začudo puno viši. To se pripisuje visoko diskriminatornim pravilima za organizacije, za razliku od pravila za lokacije i osobe koja nisu zasebno diskriminatorna, ali zajedno daju rezultate visoke preciznosti. Broj ekstrahiranih naziva po kategorijama redom iznosi 4, 21 i 2. Konkretni ekstrahirani nazivi i konteksti mogu se vidjeti u dodatku B.

6.6.3. Slovenski

Popis imena dobiven je iz istog izvora kao i za engleski jezik, dok su popisi organizacija i lokacija dobiveni koristeći resurse dostupne preko tražilice Google.



Slika 31. Rezultati eksperimenta sa slovenskim jezikom

Preciznost kategorije osoba je 100%-tna dok je preciznost organizacija i lokacija manja od 90%. Začuđujuće je visok odziv za svaku od kategorija koji u globalu iznosi više od 80%. Uvidom u ekstrahirana pravila, primjećuje se da su pravila za kategorije osobe i organizacije visoko diskriminatorna. Broj ekstrahiranih naziva po kategorijama iznosi redom 15, 22 i 18. Konkretni ekstrahirani nazivi i konteksti mogu se vidjeti u dodatku B.

Nakon izvođenja eksperimenata, može se zaključiti da odabrana nenadzirana metoda daje prilično dobre (ponekada čak i predobre) rezultate za kategorije osobe i organizacije, uz malo lošije rezultate za kategoriju lokacije. Uvidom u konkretne nazive i rezultate došlo se do slijedećih važnih zaključaka:

- Najbolji rezultati postignuti su za kategorije osobe i organizacije, a nešto lošiji za lokacije. Razlog tome je postojanje (ili ekstrahiranje) više diskriminatornih pravila za prve dvije kategorije
- Metoda kao rezultat uglavnom daje rjeđe nazive što je izravno posljedica korištenja uzajamne obavijesnosti kao ocjene zavisnosti odnosno nezavisnosti kontekstnog pravila i naziva.
- Kontekstom generirana pravila koja sustav daje kao rezultat uglavnom su diskriminatorna, te ukoliko i nisu, zajedno s drugim diskriminatornim ili nediskriminatornim pravilima mogu dati odlične rezultate s obzirom na posebnu strukturu korpusa koji se koristi.

7. Smjernice za daljnji rad

Kako je ovo prvi rad u području prepoznavanja i klasifikacije naziva i ekstrakcije naziva za hrvatski jezik temeljen na strojnom učenju, postoji dosta mjesta za njegovo poboljšanje:

- Modul za obilježavanje
  - ubrzati obradu
  - sustav preoblikovati tako da se koristi regularnim izrazima zbog brzine obrade i mogućnosti lakše integracije možebitnih ručnih pravila u sustav
  - omogućiti korištenje gramatika regularnih izraza, ukoliko bi postojala nakana da se sustav primjeni na različitim jezicima tako da se omogući korištenje jezično ovisnih pravila
  - po uzoru na [19] uvesti gramatiku imena koja će omogućiti bolje prepoznavanje imena i prezimena uključujući i male riječi koje dolaze između imena i prezimena (van, de, itd.)
- Modul za ekstrakciju naziva i kontekstom generiranih pravila
  - ubrzati proces ekstrakcije
  - istražiti mogućnosti povećanja odziva
  - izvršiti veći spektar eksperimenata s trenutno podržanim i drugim/novim klasifikatorima
  - isprobati neku drugu mjeru
  - iskoristiti druge značajke osim PMI mjere, poput leksičkih značajki
  - adekvatno adaptirati sustav i pokrenuti ga nad fiksnim korpusom ili koristiti nekih od stalnijih izvora teksta (stranice dnevnih novina)
  - razmisliti o integraciji ovakvog sustava sa sustavom zasnovanim na pravilima u svrhu poboljšavanja rezultata ili razrješenja višeznačnosti

8. Zaključak

Cilj ovog rada bio je istražiti i napraviti sustavan pregled tehnika za strojno prepoznavanje naziva i na temelju provedenog istraživanja implementirati neku od tehnika. Odabrana je slabo nadzirana, samonadopunjujuća metoda orijentirana prema ekstrakciji novih naziva iz malog skupa početnih. Također je izgrađen i modul za obilježavanje teksta temeljen na popisima naziva i kontekstom generiranih pravila dobivenih iz modula za učenje. Odabrana metoda se pokazala uspješnom, s obzirom da ne zahtijeva nikakvo prethodno znanje o području. Eksperimentalni rezultati su vrlo dobri s obzirom na slabo nadziran karakter metode – preciznost je zadovoljavajuće visoka, a odziv je umjereno dobar. Eksperimenti su potvrdili jezičnu neovisnost metode na latiničnim jezicima sa sličnim tipografskim pravilima, konkretno na bošnjačkom i slovenskom uz odlične rezultate, te na engleskom uz zadovoljavajuće rezultate. Još jedna prednost sustava je da zbog karakterističnosti PMI mjere vrši ekstrakciju rijetkih naziva.

Daljnjim istraživanjima sustav bi se mogao usavršiti te bi mogao pridonijeti u poboljšanju performansi nekog od sustava zasnovanih na pravilima ili sustava zasnovanih na metodama strojnog učenja.

Sažetak

Rad opisuje teorijsku podlogu prepoznavanja i klasifikacije naziva, s naglaskom na metode strojnog učenja i odabranu implementaciju sustava za ekstrakciju naziva. Sustav se sastoji od modula za označavanje i modula za ekstrakciju naziva i kontekstom generiranih pravila. Modul za označavanje vrši prepoznavanje i označavanje naziva koristeći popise imena i kontekstom generirana pravila koja su rezultat modula za ekstrakciju naziva i kontekstom generiranih pravila. Modul za ekstrakciju naziva i kontekstom generiranih pravila ekstrahira nove nazive koristeći male popise početnih naziva i tražilicu Google. Početni nazivi se u obliku upita šalju tražilici Google, iz čijih se rezultata obradom dobivaju konteksti koji se nakon više stupnjeva filtriranja i ocjenjivanja koriste kao pravila. Pravila upitima Google-u dohvaćaju nove, neviđene nazive. Srž ovog modula za ekstrakciju je naivni Bayesov klasifikator. Na kraju su predstavljeni obećavajući eksperimentalni rezultati.

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

Abstract

This thesis describes theoretical background of Named Entity Recognition and Classification (NERC) with an emphasis on machine learning NERC and the chosen implementation of system for Named Entity Extraction. The system consists of a tagging module and a module for named entity and context generated rules extraction. Tagging module performs named entity recognition and tagging by using gazetteers (lists of names) and context generated rules generated by the named entity and context generated rules extraction module. The second module extracts new named entities using small lists of seeds and Google search engine. Queries of seeds are being sent to Google and resulting pages processed for contexts. Those contexts are filtered, scored and then used as rules to query Google for new, unseen entities. The core of this module is Naïve Bayes Classifier (NBC). In the end, promising experimental results are presented.

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

Literatura

[1] S. Abney, (2002), *Bootstrapping*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, str. 360-367, [http://citeseer.ist.psu.edu/abney02bootstrapping.html]

[2] M. Asahara, Y. Matsumoto, (2003), *Japanese Named Entity extraction with redundant morphological analysis*, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, str. 8-15, [http://cl.aist-nara.ac.jp/papers/2003/masayu-a/NAACL-2003.pdf]

[3] F. Béchet, A. Nasr, F. Genet, (2000), *Tagging unknown proper names using decision trees*, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics table of contents, Hong Kong, str. 77-84, [http://citeseer.ist.psu.edu/541058.html]

[4] B. Bekavac, (2005), *Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima*, doktorska disertacija, Filozofski fakultet, Zagreb

[5] B. Bekavac, M. Tadić, (2007), *Implementation of Croatian NERC System*, Balto-Slavonic Natural Language Processing 2007, ACL 2007, Prague, str. 11-18, [http://acl.ldc.upenn.edu/W/W07/W07-1702.pdf]

[6] D. Boras, N. Mikelić, D. Lauc, (2003), *Leksička flektivna baza podataka hrvatskih imena i prezimena*, Modeli znanja i obrada prirodnog jezika – Zbornik radova, Radovi Zavoda za informacijske studije (knj. 12), str. 219-237

[7] A. Borthwick, J. Sterling, E. Agichtein, R. Grishman, (1998), *NYU: Description of the MENE Named Entity System as Used in MUC-7*, [http://citeseer.ist.psu.edu/borthwick98nyu.html]

[8] A. Borthwick, (1999), *A maximum Entropy Approach to Named Entity Recognition*, Ph. D. Thesis, New York University. [http://citeseer.ist.psu.edu/borthwick99maximum.html]

[9] S. Buchholz, A. van den Bosch, (2000), *Integrating seed names and n-grams for a named entity list and classifier*, In Proceedings of the Second International Conference on Language Resources and Evaluation, str. 1215-1221, Athens, Greece, [http://citeseer.ist.psu.edu/buchholz00integrating.html]

[10] X. Carreras, L. Màrquez, L. Padró, (2002), *Named Entity Extraction using AdaBoost*, Proccedings of CoNLL 2002 Shared Task Contribution, [http://citeseer.ist.psu.edu/carreras02named.html]

[11] X. Carreras, L. Màrquez, L. Padró, (2003), *Named entity recognition for Catalan using Spanish resources*, Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1, Budapest, Hungary, str. 43-50, [http://citeseer.ist.psu.edu/carreras03named.html]

[12] E. Chung, Y. G. Hwang, M. G. Jang, (2003), *Korean named entity recognition using HMM and CoTraining model*, Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11, Sapporo, Japan, str. 161-167, [http://acl.ldc.upenn.edu/W/W03/W03-1121.pdf]

[13] W. W. Cohen, (1995), *Fast Effective Rule Induction*, Proceedings of the 12th International Conference on Machine Learning, str. 115-123, Tahoe City, CA, [http://citeseer.nj.nec.com/cohen95fast.html]

[14] S. Cucerzan, D. Yarowsky, (1999), *Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence*, In Proceedings 1999 Joint SIGDAT Conference on EMNLP and VLC, [http://citeseer.ist.psu.edu/cucerzan99language.html]

[15] S. Cucerzan, D. Yarowsky, (2002), *Language Independent NER using a Unified Model of Internal and Contextual Evidence*, Proceedings of CoNLL-2002, Taipei, Taiwan, 2002, str. 171-174 [http://citeseer.ist.psu.edu/cucerzan-language.html]

[16] H. Cunningham, (1999), *Information extraction – a user guide*, 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]

[17] H. Dalianis, E. Åström, (2001), *SweNam-A Swedish Named Entity recognizer Its construction, training and evaluation*, Technical report, TRITA-NA-P0113, IPLab-189, NADA, KTH, [http://citeseer.ist.psu.edu/dalianis01swenam.html]

[18] H. C. Daumé III, (2006), *Practical Structured Learning Techniques for Natural Language Processing*, Ph.D. Thesis, University of Southern California, Los Angeles, CA. [http://pub.hal3.name/daume06thesis.ps]

[19] F. De Meulder, W. Daelemans, V. Hoste, (2002), *A Named Entity Recognition System for Dutch*, Computational Linguistics in the Netherlands 2001, page 77-88. [http://www.cnts.ua.ac.be/Publications/2002/DDH02/]

[20] F. De Meulder, W. Daelemans, (2003), *Memory-Based named Entity Recognition using Unannotated Data*, In Proceedings of CoNNLL-2003, [http://citeseer.ist.psu.edu/demeulder03memorybased.html]

[21] J. C. Duarte, R. L. Milidiu, (2007), *Machine Learning Algorithms for Portuguese Named Entity Recognition*, Monografias em Ciência de Computação nº 09/07, ISSN 0103-9741, [ftp://ftp.inf.puc-rio.br/pub/docs/techreports/07_09_duarte.pdf]

[22] O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. D. S. Weld, A. Yates, (2005), *Unsupervised named-entity extraction from the web: an experimental study*, Artificial Intelligence archive, Volume 165 , Issue 1 (June 2005), str. 91-134, [http://www.cs.washington.edu/homes/etzioni/papers/knowitall-aij.pdf]

[23] G. D. Forney, (1973), *The Viterbi algorithm*, Proceedings of the IEEE 61(3):268–278

[24] W. B. Frakes, (1992), *Information Retrieval: Data Structures and Algorithms*, Prentice Hall PTR

[25] H. Isozaki, H. Kazawa, (2002), *Efficient Support Vector Classifiers for Named Entity Recognition*, Proceedings of the 19th international conference on Computational linguistics - Volume 1, str. 1-7, [http://citeseer.ist.psu.edu/isozaki02efficient.html]

[26] V. Karkaletsis, C D. Spyropoulos, G. Petasis, (1998), *Named Entity Recognition from Greek Texts: the GIE Procject*, Advances in Intelligent Systems: Concepts, Tools and Applications, ed. S.Tzafestas, Kluwer Academic Publishers, Ch. 12, str. 131-142, [http://citeseer.ist.psu.edu/karkaletsis98named.html]

[27] S. Katrenko, P. Adriaans, (2007), *Named Entity Recognition for Ukranian: A Resource-Light Approach*, Balto-Slavonic Natural Language Processing 2007, ACL 2007, Prague, str. 88-93, [http://acl.ldc.upenn.edu/W/W07/W07-1712.pdf]

[28] Z. Kozareva, (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]

[29] G. R. Krupka, K. Hausman, (1998), *IsoQuest, Inc.: Description of the NetOwl™ Extractor System as Used for MUC-7*, Proceedings of the 7th Message Understanding Conference (MUC-7), Fairfax, Virginia

[30] L. Lillian, (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, str.111-118, [http://www.cs.cornell.edu/home/llee/papers/cstb.pdf]

[31] J. Maloney, M. Niv, (1998), *TAGARAB: A Fast, Accurate Arabic Name Recogniser Using High Precision Morphological Analysis*, Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, str. 8-15, [http://citeseer.ist.psu.edu/maloney-tagarab.html]

[32] C. Manning, H. Schütze, (1999), *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA

[33] J. Mayfield, P. McNamee, C. Piatko, (2003), *Named entity recognition using hundreds of thousands of features*, Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, Edmonton, Canada, str. 184-187, [http://citeseer.ist.psu.edu/645219.html]

[34] A. Mikheev, C. Grover, M. Moens, (1998), *Description of the LTG System Used for MUC-7*, Proceedings of the 7th Message Understanding Conference (MUC-7), Fairfax, Virginia, [http://citeseer.ist.psu.edu/mikheev98description.html]

[35] T. M. Mitchell, (1997), *Machine Learning*, McGraw Hill

[36] M.F. Moens, (2006), *Information Extraction: Algorithms and Prospects in a Retrieval Context*, Springer

[37] P. Osenova and S. Kolkovska, (2002), *Combining the named-entity recognition task and NP chunking strategy for robust pre-processing*, In Proceedings of the Workshop on Linguistic Theories and Treebanks, 20-21 Sept., Sozopol, Bulgaria, [http://citeseer.ist.psu.edu/osenova02combining.html]

[38] T. Poibeau, (2003), *The Multilingual Named Entity Recognition Framework*, In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003.) Budapest, Hungary [http://citeseer.ist.psu.edu/poibeau03multilingual.html]

[39] S. Roweis, (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]

[40] S. Sekine, R. Grishman, H. Shinnou, (1998), *A Decision Tree Method for Finding and Classifying Names in Japanese Texts*, In Proceedings of the Sixth Workshop on Very Large Corpora, [http://citeseer.ist.psu.edu/sekine98decision.html]

[41] B. Settles, (2004), *Biomedical named entity recognition using conditional random fields and rich feature sets*, Proc. JNLPBA/BiONLP2004, str.104–107., [http://pages.cs.wisc.edu/~bsettles/pub/bsettles-nlpba04.pdf]

[42] R. Srihari, C. Niu, W. Li, (2001), *A hybrid approach for named entity and sub-type tagging*, In: Proc. 6th Applied Natural Language Processing Conference. (2001) [http://citeseer.ist.psu.edu/srihari01hybrid.html]

[43] A. Šilić, F. Šarić, B. Dalbelo Bašić, J. Šnajder, (2007), *TMT: Object-Oriented Text Classification Library*, Proceedings of the 29th International Conference on Information Technology Interfaces, str. 559-566

[44] M. Tadić, (1996), *Računalna obradba hrvatskoga i nacionalni korpus*, Suvremena lingvistika 41-42, Zagreb, str. 603–612

[45] M. Tadić, (2000), *Information Retrieval Meets Human Language Technology*, CUC2000 Zbornik, CD-ROM, CARNet, Zagreb, [http://www.carnet.hr/CUC/cuc2000/radovi/f4.html]

[46] M. Tadić, S. Fulgosi, (2003), *Building the Croatian Morphological Lexicon*, In Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages (Budimpešta 2003), ACL, str. 41-46, [http://www.hnk.ffzg.hr/txts/mtsf4EACL2003.pdf]

[47] C. Thielen, (1995), *An Approach to Proper Name Tagging for German*, In Proc. EACLSIGDAT -95., [http://citeseer.ist.psu.edu/thielen95approach.html]

[48] Q. Tri Tran, T. X. Thao Pham, Q. Hung Ngo, D. Dinh, N. Collier, (2007), *Named entity recognition in Vietnamese documents*, Progress in Informatics, No.4, March 2007, str. 5-13, [http://www.nii.ac.jp/pi/n4/4_5.pdf]

[49] K. Tsukamoto, Y. Mitsuishim, M. Sassano, (2002), *Learning with Multiple Stacking for Named Entity Recognition*, Proceeding of the 6th conference on Natural language learning - Volume 20, str. 1-4, [http://citeseer.ist.psu.edu/tsukamoto-learning.html]

[50] A. M. Turing, (1950), *Computing machinery and intelligence*, Mind, 59, 433-460, [http://www.loebner.net/Prizef/TuringArticle.html]

[51] G. Tür, (2000), *A Statistical Information Extraction System for Turkish*, Ph. D. Thesis, Bįlkent University, [http://citeseer.ist.psu.edu/553165.html]

[52] O. Uryupina, (2003), *Semi-supervised learning of Geographical gazetteers from the Internet*, In Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references - Volume 1 str 18 – 25, Edmonton, [http://www.coli.uni-saarland.de/~ourioupi/ws913.pdf]

[53] H. M. Wallach, (2004), *Conditional Random Fields: An Introduction*, Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, [http://www.inference.phy.cam.ac.uk/hmw26/papers/crf_intro.ps]

[54] T. Yao, W. Ding, G. Erbach, (2003), *CHINERS: A Chinese Named Entity Recognition System for the Sports Domain*, Second ACL SIGHAN Workshop on Chinese Language Processing, http://citeseer.ist.psu.edu/yao03chiners.html]

Dodatak A: Programska podrška

Implementacija sustava izvedena je u programskom jeziku *C++*, a razvijena je korištenjem razvojne okoline *Microsoft Visual Studio 2005*. Za prevođenje, implementacija zahtijeva C++ biblioteke *boost*[50] i *asio*[51] koje realiziraju višedretvenost i dohvat podataka koristeći mrežna sučelja.

Uz tekst rada priložen je CD s izvornim kodom, kratkim uputama za prevođenje, uputama za korištenje, izvršnom datotekom, cjelokupnim priručnim spremnikom korištenim kroz sve eksperimente, rezultati eksperimenata kao i ovaj rad u *pdf* i *doc* formatu.

[50] http://www.boost.org/
[51] http://asio.sourceforge.net/

Dodatak B: Rezultati nekih eksperimenata

U nastavku se nalaze rezultati nekih od eksperimenata u obliku konkretnih naziva i konteksta koje je sustav ekstrahirao.

Tablica 3. Nazivi i konteksti ekstrahirani eksperimentom s nediferenciranim popisima

Osobe	Organizacije	Lokacije
Ekstrahirani nazivi		
Alen Barbara Benedikta Bernarda Blaža Damir Darja Dragan Draženka Frane Franje Galeb Imam Ime Jurčić Krševana Lovre Luci Luku Milko Miomir Mladenka Nedeljko Pichlera Roka Ruza Sanader Smiljko Vinku Vladis Vlaha Zdeslav Zdravko Zenun Zivko Zlatan Zoran Zvanje Čedomilu Đula Đurđana Štefica Žagar Žarko Želimir Željana Želko	Aastra Automehanika Autotrans Banka Kovanica Belupo Brodomerkur Ghetaldus Indeco Ivkom Jadroplov Kaplast Konzum Labud Marituna Radež Sladorana Tankerkomerc Vage Varkom Vemat Vindija Wienerberger Zabat Zagorje Tehnobeton Čistoća Đuro Đaković Aparati	Bjelovar Daruvar Koprivnica Kutina Ludbreg Opatija Samobor Slavonski Brod Virovitica Vrbovec Zagreb Hrvatska Zaprešić Županja
Ekstrahirani konteksti		
* i ja * prezime Vela * dr * dr sc * sc * sv * sveta * sveti *	* Zagreb * d * d d * d o tvrtke *	* hrvatska * tel centar * d d * fakultet * grad * klub * o o * škola *

Tablica 4. Nazivi i konteksti ekstrahirani eksperimentom s bošnjačkim jezikom

Osobe	Organizacije	Lokacije
Ekstrahirani nazivi		
Fehim Fehima Haris Husein Huseina Jasminka Kasima Lejla Mevlida Muftije Muhameda Mustafom Mustafu Nejra Nikola Porodice Reis Reisom Salvatore Sejfudina Selima Senaid Sulejman Vildana Šerif	Aluminij Bosfin Konjuh Pinuspro Raiffeisen Bankk Sejari Svjetlostkomerc Termotehnika Turkish Ziraat Bank Bosnia Tuzlanska Banka Unigradnja Unionkomerc Unipromet Unitrade Vegafruit Vranica Zvečevo Lasta Šipad Komerc Žica	Bihać Gradačac Kiseljak Lukavac Tešanj Zavidovići Zenica
Ekstrahirani konteksti		
* ef * i ja dr * ja sam * mr * ovo je * zove se * zovem se *	* d d * dd * dd Sarajevo firme *	* BiH * Bosna * Bosna i * tel bb * d d * novi * o o * općine *

Tablica 5. Nazivi i konteksti ekstrahirani eksperimentom s engleskim jezikom

Osobe	Organizacije	Lokacije
Ekstrahirani nazivi		
Calvin Stuart Timothy Vivian	Ameriprise Cargill Carl Zeiss Meditec Corning Cotton Girindus Holding International Play Leesport Merriam Webster Nanometrics Neuronics Pharma Rheinmetall Roscor Sandvine Siemens Soccergirl Software Student Swinerton	Auckland Houston
Ekstrahirani konteksti		
* J * R Daryl * Dr * artist * pictures * posted by * professor * saint *	* AG * Corporation * Financial * Inc * Incorporated * Ltd * company	* State * Texas * area * city * hotel * hotels * real University of * city of * greater *

Tablica 6. Nazivi i konteksti ekstrahirani eksperimentom sa slovenskim jezikom

Osobe	Organizacije	Lokacije
Ekstrahirani nazivi		
Alojzij Avguštin Gregor Gregorja Krištofa Mihaela Milček Mitjan Neža Rupel Vaupotič Vinko Vladimirja Wohlmann Zinko	Abanka Vipa Delo Halcom Impol Komunaprojekt Kontrola Lama Lipa Ajdovščina Merkur Mladina Mlinopek Mlinotest Mobitel Novolit Podravka Steklarna Hrastnik Telekom Slovenije Vegrad Zaloker Zaloker Zavarovalnica Zavarovalnica Maribor Zavarovalnica Triglav	Bohinj Brežice Dravinjske Jesenice Kidričevo Krško Lendava Ljubljana Slovenija Ljubljana Črnuče Novo Ptuj Radovljica Tržič Velike Lašče Vrhnika Šempeter Vrtojba Šentjur Škofja Loka
Ekstrahirani konteksti		
* s p Institut * cerkev sv * dr * imenom * prof dr * sv * svetega * sveti *	* d * d d podjetja * podjetje *	* Slovenija * tel center * cesta * klub * mestna občina * o o * občina * občine * ulica *