MATEMATIČKI FAKULTET
UNIVERZITET U BEOGRADU

Jelena Tomašević

XML baze podataka u upravljanju leksičkim resursima

Magistarski rad

Mentor: dr Gordana Pavlović-Lažetić

Beograd
2008.

Predgovor

Ova magistarska teza nastala je kao rezultat rada na ispitivanju mogućnosti i karakteristika skladištenja leksičkih resursa i manipulisanja njima u okviru izvornih XML baza podataka. Sprovedeno je više eksperimenata nad pojedinačnim resursima kao i nad njihovim integracijama koji su uključili različite vrste operacija. Posebna pažnja posvećena je analizi leksičko-semantičke mreže reči — *Wordnet*. Rezultati koji su dobijeni kao posledica ove analize prikazani su u radu [13] koji je prihvaćen za predstavljanje na konferenciji "Sixth Language Technologies Conference" u Ljubljani, koja će biti održana 15. oktobra 2008. godine. Sprovedeni su i eksperimenti sa klasifikacijom tekstova zasnovanom na *Wordnet*-u.

Ovaj magistarski rad je organizovan u osam glava. U uvodnoj glavi, dat je pregled XML-a i XML baza podataka kao najboljeg načina predstavljanja i skladištenja podataka koji nisu dobro struktuirani. Prikazan je i uopšten pregled leksičkih resursa. Istaknuti su osnovni ciljevi i plan istraživanja. Detaljniji prikaz XML-a, XML baza podataka i leksičkih resursa dat je u okviru naredne tri glave. Glave 5, 6 i 7 predstavljaju centralni deo rada i u okviru njih su opisani eksperimenti koji su sprovedeni nad leksičkim resursima. U okviru glave 5, opisana je biblioteka XQuery funkcija "biogfun" za manipulisanje integracijom resursa *Wordnet*, kolekcije biografija matematičara iz Spomenice Matematičkog fakulteta i rečnika vlastitih imena. Glava 6 daje prikaz analize *Wordnet*-a, određivanje najproduktivnijeg koncepta i poređenje srpskog i engleskog *Wordnet*-a. U tom cilju razvijena je biblioteka XQuery funkcija "wnfun". U glavi 7, prikazani su rezultati klasifikacije tekstova iz dnevnog lista Politika, zasnovane na srpskom *Wordnet*-u. Na kraju, u okviru zaključka, sumirani su dobijeni rezultati i prikazani su mogući pravci u daljem radu.

Zadovoljstvo mi je da ovom prilikom zahvalim svom mentoru Gordani Pavlović-Lažetić, koja mi je pružila izuzetnu pomoć prilikom izrade ove teze. Svojim idejama ona je najviše doprinela njenom izgledu, a korisnim savetima i sugestijama pomogla da tekst postane bolji. Posebno sam zahvalna na velikom razumevanju, podršci i poverenju koje mi je ukazala u toku rada na tezi. Zahvaljujem i članovima komisije za pregled i ocenu rada — prof. Dušku Vitasu i prof. Ivanu Obradoviću, čiji me je naučni rad inspirisao i veoma doprineo izgledu ove teze. Posebnu zahvalnost dugujem prof. Vitasu, koji je svojim sugestijama i mnogim dobronamernim savetima najviše doprineo mom odabiru sadašnjeg poziva i bavljenju naukom.

Ovoj tezi i mom zadovoljstvu u njenom pisanju veoma su doprineli promišljeni kriticizam i sugestije mnogih prijatelja i kolega. Veliku zahvalnost dugujem svojim kolegama i pre svega prijateljima Mileni i Peđi, koji su mi od samog početka pružali nesebičnu i veliku podršku. Kao pažljivom čitaocu ovog rada,

zahvaljujem kolegi Đorđu Stakiću, koji je nizom korisnih sugestija pomogao da rad izgleda bolje. Zahvalna sam i kolegama Filipu, Mladenu, Sani i Vesni na korisnim diskusijama i razmeni iskustva na različitim problemima.

Najveću zahvalnost dugujem svojoj majci i svom bratu za bezgraničnu požrtvovanost i darovanje svog vremena koje mi je omogućilo da se posvetim naučnom radu, pa i da napišem ovu tezu. Takođe, veliku zahvalnost dugujem svom vereniku Igoru, na razumevanju kada sam vikende provodila radeći umesto da budem sa njim. Veliku zahvalnost dugujem i njegovim roditeljima i bratu, koji su mi u poslednje vreme pružali veoma značajnu podršku. Svim svojim prijateljima sam zahvalna na velikom razumevanju i savetima.

Jelena Tomašević

Beograd, septembar 2008.

Sadržaj

1

Uvod

U savremenom svetu količina informacija koja se nudi je ogromna i često se spominje preopterećenje informacijama (eng. information overload) koje savremeni čovek nosi kao "krst napretka". Nepregledne količine podataka svakog dana nastaju i nestaju, i veliki broj ljudi se jednostavno gubi u tom sivilu prekomernih informacija. Sve veći broj istraživanja opominje da kvalitet savremenog načina života upravo zavisi od toga kako se nosimo sa prekomernim informacijama koje nas svaki dan zapljuskuju. Dobra organizovanost podataka je iz tog razloga od ključnog značaja. Podatke je potrebno efikasno skladištiti i imati efikasan pristup njima. Oni se mogu razlikovati prema svojoj struktuiranosti i u zavisnosti od toga mogu se skladištiti na različite načine.

Jedan od načina skladištenja i efikasnog upravljanja podacima je u okviru relacionih baza podataka koje su dosta dobro razvijene. One su pogodne za čuvanje dobro struktuiranih podataka. Primer takvih podataka bio bi telefonski imenik kod koga je svako polje definisano tako da ima jasnu strukturu, na primer ime, adresu i broj telefona. Međutim, sve češće se javljaju podaci koji nemaju tako jasno definisanu strukturu, takozvani polustruktuirani podaci. To su podaci koji mogu sadržati polja koja nisu poznata u trenutku projektovanja dokumenta, podaci kod kojih se iste vrste podataka mogu predstaviti na različite načine. Primer polustruktuiranih podataka bio bi telefonski imenik koji ima strukturu gore pomenutog imenika, ali sa značajnom količinom mogućih varijacija unutar te strukture. Na primer, osim imena, adrese i broja telefona, neka polja mogu imati veći broj adresa, veći broj brojeva telefona, mogu imati broj fax-a, broj licence ili neki dodatni tekst. Za takve podatke koji nisu dobro struktuirani, relacione baze podataka nisu najpodesnije.

1.1 Upravljanje polustruktuiranim podacima

Najbolji način za predstavljanje polustruktuiranih podataka jeste XML (eXtensible Markup Language). To je jezik markiranja sličan HTML-u, slobodan i proširiv. XML etikete nisu unapred definisane tako da se mogu same osmisliti u zavisnosti od dokumenta koji se njima predstavlja. XML dokumenti upadaju u dve široke kategorije: podatak-centrični (eng. data-centric) i dokument-centrični (eng. document-centric). Podatak-centrični dokumenti su struktuirani, postoji jasno definisan redosled podataka i oni se projektuju

uglavnom za mašinsko procesiranje pre nego za ljudsku upotrebu. Takvi dokumenti XML koriste uglavnom samo za prenos podataka. Dokument-centrični dokumenti su polustruktuirani dokumenti sa neregularnim sadržajem. Oni su uglavnom namenjeni za ljudsku upotrebu.

Sa sve većim rastom Internet-a raste i popularnost XML-a a samim tim i potreba za skladištenjem XML podataka. Razlikujemo XML-proširene baze podataka i izvorne XML baze podataka. XML-proširene baze podataka su relacione baze podataka koje imaju dodatnu mogućnost preslikavanja XML dokumenata u njih same, odnosno tabele u okviru relacionih shema. Teško je uklopiti bogatu strukturu XML-a u krute relacione tabele bez deljenja dokumenta u veoma male standardne jedinice koje se mogu prikazati kao ćelije u tabeli. Zbog toga čak i jednostavna XML shema proizvodi veliki broj tabela ili tabele sa velikim brojem nedostajućih vrednosti. Strukturne informacije u shemi baziranoj na drvetima, kakva je XML shema, dobijaju se spajanjem tabela u relacionoj shemi. XML upiti se konvertuju u SQL upite nad relacionim tabelama i tako se čak i jednostavni XML upiti transliraju u skupu seriju spajanja tabela odgovarajuće relacione baze.

S obzirom na ove argumente, javila se potreba za traženjem direktne implementacije upravljanja XML dokumentima bez njihovog preslikavanja u relacije fiksne strukture. Izvorne XML baze podataka su baze podataka koje smeštaju XML dokumente u njihovoj "izvornoj" formi čuvajući pri tome njihovu drvoliku strukturu. Izvorne XML baze podataka najviše se koriste za upravljanje dokument-centričnim XML dokumentima, integrisanim podacima i polustruktuiranim podacima. Više o XML-u i XML bazama podataka biće reči u poglavljima 2 i 3.

1.2 Leksički resursi

Jedni od tipičnih predstavnika dokument-centričnih dokumenata jesu leksički resursi. Leksički resursi, kao posebna vrsta jezičkih resursa, sadrže različite vrste lingvističkih informacija. Neki od primera su korpusi, razne vrste elektronskih rečnika, leksičko-semantičke mreže tipa *Wordnet*, rečnici vlastitih imena i drugo. Oni igraju ključnu ulogu u mnogim procesima obrade prirodnih jezika.

U okviru grupe za obradu prirodnih jezika na Matematičkom fakultetu u Beogradu, razvija se više leksičkih resursa važnih u procesu obrade srpskog jezika.

Prvi važan resurs je korpus, koji danas postoji za sve evropske jezike, a omogućava istraživačima da razgledaju strukturu i ponašanje određenog jezika. Slikovito rečeno, korpus je neka vrsta lingvističkog mikroskopa. Korpus našeg jezika nastao je na inicijativu Odbora za standardizaciju srpskog jezika.

Posebno je zanimljiv resurs koji se zove "paralelni korpusi". Zahvaljujući njemu moguće je uporedno sagledati rečenice ili delove teksta u različitim prevodima, na primer, uporediti isti deo romana "Majstor i Margarita" u prevodu na svim slovenskim jezicima. Paralelni korpus pruža značajne olakšice u prevođenju i osnov je višejezičke leksikografije.

Morfološki elektronski rečnik savremenog srpskog jezika je još jedan važan resurs. To nije rečnik koji se čita već je namenjen automatskoj obradi teksta i značajan je za lingvistička i leksikografska istraživanja jer u realnom vremenu daje precizne podatke do kojih bi se moralo, uz pomoć olovke i papira, tragati

danima. Ovaj rečnik omogućava da se pronađu svi oblici reči, najfrekventnija reč (u korpusu našeg jezika to je "i") ili spoj od dve ili tri reči u nekom književnom delu. Na primer, u Andrićevoj "Travničkoj hronici" najfrekventniji trigram, spoj od tri reči, je "kao da je". Značajno je i što je ovakav rečnik napravljen na istom principu za većinu evropskih jezika, što olakšava komparativna lingvistička proučavanja [46].

Jedan od savremenih elektronskih jezičkih resursa koji se razvija za veliki broj jezika, uključujući i srpski, jeste leksičko-semantička mreža poznata kao *Wordnet*. U *Wordnet*-u rečnik je organizovan po konceptima, tj. značenjima koja su međusobno povezana mrežom semantičkih i leksičkih relacija.

Ovi resursi omogućavaju proučavanje jezika na jedan drugačiji način, a mogu i da otkriju ponešto o prirodi književnih remek-dela. Najfrekventnije reči i njihovi spojevi možda mogu da ukažu na stilske odlike jednog pisca, da otkriju psihološki profil autora ili zašto određeno delo čitalac doživljava kao dinamično ili statično. Matematički pogled na jezike možda gradi Vavilonsku kulu uz pomoć računara otkrivajući da ono što se čini strahovito različito ima mnogo zajedničkog jer sistemi svih jezika opisani su sličnim formalnim modelima i vrlo su jake veze između matematike i jezika [46]. Više o leksičkim resursima biće reči u poglavlju 4.

1.3 Ciljevi i plan istraživanja

Jedan od najčešćih standarda za prikaz leksičkih resursa je XML. U toku procesa obrade prirodnih jezika, važnu ulogu igra efikasnost dobijenih informacija iz leksičkih resursa. Osnovni ciljevi istraživanja opisanog u ovom radu su:

- Ispitivanje mogućnosti i karakteristika skladištenja i manipulisanja leksičkim resursima u izvornim XML bazama podataka. Ove baze podataka omogućavaju jednostavne operacije pretraživanja jednog ili više resursa, po različitim kriterijumima, kao i semantički kompleksne operacije nad leksičkim resursima kao što su klasifikacija ili ekstrakcija informacija;

- Istraživanje novih pristupa procesima obrade prirodnih jezika koje može da sugeriše koncept izvornih XML baza podataka kao sredstva za jednostavnu integraciju i manipulaciju raznorodnih leksičkih resursa. U tom cilju sprovedeni su eksperimenti nad pojedinačnim resursima i njihovim integracijama (rečnik vlastitih imena, *Wordnet*, korpus) koji su uključili jednostavne operacije, i eksperimenti sa klasifikacijom tekstova baziranom na *Wordnet*-u.

Glavni deo rada posvećen je rezultatima dobijenim za kompleksni operator nad leskičkim resursima - klasifikaciji tekstova zasnovanoj na *Wordnet*-u.

Srpski Wordnet razvijen je kao deo *Balkanet* projekta čija je glavna aktivnost razvoj mreža tipa *Wordnet* za balkanske jezike pojedinačno (bugarski, grčki, rumunski, srpski, turski i češki).

U *Wordnet*-u, koncepti su organizovani hijerarhijski u nivoe, počev od opštih ka specifičnim. Koncepti koji nisu ni suviše opšti ni suviše specifični i koji se nalaze negde na sredini hijerarhije, označeni su kao "bazični koncepti". U cilju formalizacije pojma "bazični koncept", i njegove eksploatacije u realizaciji

kompleksnih operatora nad tekstom, u ovom radu se definišu mere "produktivnosti" koncepta koje određuju koliko neki koncept efektivno predstavlja hijerarhiju kojoj pripada. U tom smislu, "bazični koncepti" mogu biti okarakterisani visokom vrednošću mere produktivnosti i za nih se može smatrati da najbolje opisuju hijerarhiju kojoj pripadaju. Zato se mogu koristiti, između ostalog, za realizaciju operatora klasifikacije tekstova. Rad zasnovan na ovim rezultatima, prihvaćen je za predstavljanje i objavljivanje u zborniku radova na konferenciji "Sixth Language Technologies Conference" u Ljubljani, oktobra 2008 [13].

Klasifikacija tekstova ima za zadatak pridruživanje tekstualnom dokumentu jedne ili više prethodno definisanih kategorija, na osnovu njegovog sadržaja. Najproduktivniji koncepti i hijerarhije sa korenom u tim konceptima, mogu voditi ka novom pristupu klasifikacije tekstova. Ideja je da se odabrani najproduktivniji (ili neki drugi) koncepti mogu pridružiti klasama koje su unapred odabrane i koje će ti koncepti predstavljati. U poglavlju 7 su prikazani rezultati ovakvog algoritma klasifikacije na kolekciji članaka iz dnevnog lista Politika sa rubrikama sport, politika i ekonomija. U skladu sa ovim rubrikama odabrane su i klase. Klasi *Sport* pridruženi su koncepti iz hijerarhija sa korenima u konceptima "takmičenje", "igra" i "sport", klasi *Politika* pridruženi su koncepti iz hijerarhije sa korenom u konceptu "društvena grupa" filtrirani po domenu "politics" i klasi *Ekonomija* pridruženi su koncepti iz hijerarhije sa korenom u konceptu "svojina" filtrirani po domenima "economy", "banking", "money" i "commerce".

Ako se klasifikacija izvrši na ovaj način onda i druge operacije, na primer izdvajanje informacija, mogu da se obave efikasnije. Upit može da se proširi literalima iz hijerarhije nekog dobro izabranog produktivnog koncepta, a izdvajanje informacija može da se sprovede samo nad tekstovima odgovarajuće klase.

2

XML

XML je skaraćenica od "eXtensible Markup Language" i predstavlja osiromašenu verziju SGML-a (Standard Generalized Markup Language), mnogo većeg meta jezika koji je postojao i pre pojave veba. SGML definiše gramatiku za sve jezike označavanja dokumenata i SGML dokumenti nose svoju gramatičku definiciju sa sobom u obliku DTD-a (Document Type Definition). DTD specifikuje sve oznake koje se koriste u jednom dokumentu i određuje im semantiku. HTML je jedna od aplikacija SGML-a. On je razvijen kao jezik čija je osnovna namena da opiše izgled jedne veb strane i njegova glavna karakteristika je bila jednostavnost. Jedan od razloga zbog kog je Internet doživeo veliku popularnost jeste upravo ta jednostavost HTML-a. HTML je čvrsto određen skup oznaka i njegovi dokumenti ne moraju sa sobom da nose DTD, jer predstavljaju fiksnu, neproširivu kategoriju odnosno imaju fiksnu gramatiku. Upravo iz tog razloga, on je lak za učenje i lak za pisanje aplikacija koje ga prevode ("renderuju" HTML). Drugu stranu medalje predstavlja to što se HTML često ne može koristiti u nekim specijalnim slučajevima. Proširivi SGML bi u ovakvim situacijama mogao da pomogne, ali on je suviše glomazan da bi se lako učio i implementirao. Umesto implementacije celog SGML-a na vebu, svetski veb konzorcijum (World Wide Web Consortium - W3C) je predložio osiromašenu verziju SGML-a - XML. XML se može shvatiti i prihvatiti kao vrsta SGML Lite-a, namenjenog premošćavanju jaza između komplikovanog SGML bogatstva i HTML-ove lakoće korišćenja na vebu, i ne samo na vebu. XML je kao i SGML jezik, meta jezik. Ali dok promena HTML-a zahteva zvaničnu promenu standarda, XML je predviđen da bude proširiv. Richard Light, u svojoj knjizi, za XML kaže: "XML nudi 80% dobrih strana SGML-a sa 20% njegove složenosti" [6].

XML međutim, nije zamena za HTML. U okviru budućeg razvoja veba, tendencija je da se HTML koristi za formatiranje i prikazivanje a XML za opisivanje podataka. Ipak, XML u velikoj meri prevazilazi tu svoju osnovnu namenu. Postao je dominantan standard kako za struktuiranje tako i za skladištenje podataka ali i za razmenu podataka između heterogenih sistema.

XML ima mnogo prednosti kao format za opisivanje podataka u odnosu na ostale formate:

- Sintaksa etiketa nije fiksirana. Autor ima slobodu da kreira etikete za potrebe određenje aplikacije. Semantika etiketa nije ograničena ali je za-

visna od konteksta u kom aplikacija koristi dokument;

- Fleksibilan je i proširiv, odnosno daje mogućnost dodavanja novih informacija i dopušta postojanje različitih tipova podataka u okviru jednog dokumenta;
- Opisuje podatke stavljanjem akcenta na to šta podaci jesu a ne kako oni izgledaju;
- Ima format koji je čitljiv za čoveka pa je lakše locirati i ispravljati greške. Ima drvoliku strukturu koju je lako pratiti.
- Ima mogućnost internacionalnog korišćenja zahvaljujući činjenici da koristi Unicode kodnu šemu;
- Nezavisan je od platforme odnosno od softvera i hardvera koji se koristi;
- Postoji veliki broj gotovih aplikacija za procesiranje XML-a koje se mogu koristiti.

U isto vreme, XML format ima i slabosti kao na primer to da je manipulisanje podacima često sporije nego kod tradicionalnih formata, a optimizacija je kompleksnija zahvaljujući bogatstvu i velikoj izražajnoj moći upitnih jezika koje koristi. XML dokument mora biti dobro formiran, odnosno mora da zadovolji veoma precizna i stroga pravila nametnuta XML standardom.

Sintaksa XML-a smatra se jednostavnom i vrlo strogom. Pravila su laka za učenje i korišćenje pa je i kreiranje softvera koji čita i manipuliše XML-om vrlo jednostavno. Neka od sintaksnih pravila XML-a su da svi XML elementi moraju imati zatvorene etikete, etikete su osetljive na veličinu slova (eng. case sensitive), svi XML elementi moraju biti ispravno ugnježdeni, moraju imati koreni element, svi XML dokumenti moraju imati jedinstven par etiketa za definisanje korenog elementa, svi drugi XML elementi moraju biti unutar korenog elementa, svi elementi mogu imati pod-elemente (umetnute elemente) koji moraju biti korektno ugnježdeni unutar roditeljskog elementa.

Kako XML-om može da se opiše struktura podataka to treba da postoji način da se ta struktura specifikuje. Postoje različiti mehanizmi koji se koriste da bi se specifikovalo koji se elementi mogu pojaviti u dokumentu, u kakvom redosledu se mogu pojaviti, kakva su njihova ograničenja i drugo. U tu svrhu koriste se DTD i XML sheme. XML dokument sa korektnom sintaksom smatra se dobro formiranim, a ako još odgovara DTD-u ili XML shemi smatra se valjanim.

2.1 Definisanje tipa dokumenta

Definisanje tipa dokumenta (eng. Document Type Definitions — DTD) predstavlja originalni način za specifikovanje strukture XML dokumenta. On ima različitu sintaksu od XML-a i koristi se da specifikuje dopuštene gradivne blokove i redosled elemenata u XML dokumentu. Jednostavni gradivni blokovi XML dokumenta sa tačke gledišta DTD-a su:

- Elementi;
- Etikete — glavni gradivni elementi XML dokumenta;

- Aributi — navode se u okviru etiketa;
- Entiteti — promenljive koje se koriste da definišu opšti tekst. Predefinisani entiteti u XML-u su na primer `<(<)`, `> (>)`, `& (&)`, `&quot;` (");
- PCDATA (Parsed Character DATA) —- tekst između otvorene i zatvorene etikete XML elementa koji će se analizirati pomoću parsera;
- CDATA (Character DATA) — tekst koji se neće analizirati pomoću parsera i etikete unutar teksta ne označavaju obeležavanje.

DTD može da bude deklarisan unutar XML dokumenta a može da bude deklarisan i kao spoljašnja referenca. Na primer, neka je dat sledeći fragment XML dokumenta:

```
<student studbr="st0001">
 <ime>Igor Jovanovic</ime>
 <starost>21</starost>
</student>
```

DTD koji opisuje strukturu ovog dela XML dokumenta je:

```
<!ELEMENT student (ime, starost)>
<!ATTLIST student studbr CDATA>
<!ELEMENT ime (#PCDATA)>
<!ELEMENT starost (#PCDATA)>
```

Ovim DTD-om je specifikovano da element `student` ima dva elementa koji su njegova deca, `ime` i `starost`, sadrži karakterske podatke i atribut `studbr` koji takođe sadrži podatke predstavljene karakterima.

Alternativa DTD-u za opis strukture XML dokumenta koju podržava W3C je XML Shema. DTD obezbeđuje manju kontrolu nad XML dokumentima u odnosu na XML sheme ali je zato jednostavniji.

Jedan od načina da se prikaže XML dokument je korišćenjem nekog od razgledača: Mozilla, Opera, Internet Explorer, Netscape 6+ i drugi. XML dokument se može prikazati i pomoću XSL-a (the eXtensible Stylesheet Language) koji je daleko sofisticiraniji od CSS-a (Cascading Style Sheet) čija je osnovna namena dodavanje stila u veb dokument. Jedan od načina da se prikaže XML korišćenjem XSL-a je transformacija XML-a u HTML pre prikazivanja razgledačem.

2.2 Postavljanje upita nad XML dokumentom

Ponekad je neophodno da se izdvoji podskup podataka smeštenih u okviru XML dokumenta. Veliki broj jezika je kreiran za postavljanje upita nad XML dokumentima uključujući XML-QL, XPath, XQL, XQuery. XPath je W3C preporuka a sa pojavom XQuery postaje još popularniji. Oba ova jezika mogu se koristiti za dobijanje i manipulisanje podacima iz XML dokumenata.

2.2.1 XPath — XML Path Language

XPath je jezik koji omogućava pristup delovima XML dokumenta. On koristi sintaksu sličnu URL-u (Uniform Resource Locator) ili hijararhijskim putanjama korišćenim za adresiranje dela programskog sistema. On tretira XML dokument kao drvo međusobno povezanih grana i čvorova. Čvor XML dokumenta može biti element, atribut, instrukcija za obradu, komentar, tekstualni sadržaj, imenovani domen (eng. namespace) ili sam dokument. Model XPath drveta se ne bazira na samim čvorovima, već na njihovim međusobnim vezama, na primer načinu na koji su elementi povezani jedni sa drugim, načinu na koji su atributi povezani sa elementima i tako dalje. XPath takođe podržava korišćenje funkcija za međusobnu interakciju odabranih podataka iz dokumenta. Podržava funkcije za pristup informacijama o čvorovima dokumenta kao i manipulaciju niskama, brojevima ili logičkim vrednostima. XPath je proširiv i funkcijama koje sam korisnik može da doda biblioteci funkcija koja je podrazumevano dostupna. XPath koristi kompaktnu sintaksu različitu od XML-a u cilju omogućavanja korišćenja XPath-a unutar URI-a i XML vrednosti atributa (ovo je bitno za na primer XML sheme i XSLT koji koriste XPath unutar atributa). XPath radi sa apstraktnim, logičkim strukturama XML dokumenta pre nego sa površnom sintaksom. On je dizajniran da radi sa jednim XML dokumentom. Vrednost vraćena XPath upitom je skup od jednog ili više čvorova.

Jednostavni primeri XPath upita nad jednostavnim fragmentom XML dokumenta su:

Primer 2.1 *Selektovati sve elemente* `ime` *koji su deca korenog elementa* `student`.

```
/student/ime
```

Primer 2.2 *Selektovati sve elemente* `starost` *u dokumentu.*

```
//starost
```

Primer 2.3 *Selektovati sve elemente koji su deca korenog elementa* `student`.

```
/student/*
```

Primer 2.4 *Selektovati sve* `studbr` *atribute elemenata* `student` *u dokumentu.*

```
/student[@studbr]
```

Primer 2.5 *Selektovati sve elemente* `starost`.

```
//*[name()='starost']
```

Primer 2.6 *Selektovati sve pretke od svih elemenata* `starost` *koji su deca od elementa* `student` *(što znači da treba selektovati elemente* `student`*).*

```
/student/starost/ancestor::*
```

2.2.2 XQuery — XML Query Language

XQuery je jezik koji je projektovan sa namerom da obezbedi upitni jezik koji ima istu širinu funkcionalnosti i skriveni formalizam kao SQL nad relacionim bazama podataka. XQuery ima sofisticirani sistem tipova podataka baziran na XML shemi i omogućava manipulisanje čvorovima dokumenta kao XPath.

Takođe, model podataka XQuery jezika nije projektovan da radi samo sa jednim dokumentom već i sa dobro formiranim delovima dokumenta, nizom dokumenata ili nizom delova dokumenata. To je funkcionalni jezik u kome je svaki upit iskaz. Iskazi u XQuery-u upadaju u 6 širokih tipova: iskazi putanje, konstruktori elemenata, FLWR iskazi, uslovni iskazi, kvantifikovani iskazi i iskazi u okviru kojih se koriste operatori i funkcije. Sintaksa i semantika ovih tipova iskaza značajno varira, što je posledica mnogih različitih uticaja na projektovanje XQuery jezika.

Iskazi putanja

XQuery obezbeđuje iskaze putanja koje su nadskup od onih u XPath-u.

Primer 2.7 *Iz dokumenta koji sadrži zaposlene i njihovu mesečnu zaradu, izdvojiti godišnju zaradu za zaposlenog sa imenom "Marko".*

```
//zaposleni[ime="Marko"]/zarada * 12
```

Primer 2.8 *U dokumentu "zoo.xml" pronaći sve slike u poglavljima od 2 do 5.*

```
document("zoo.xml")//poglavlje[2 TO 5]//slika
```

Konstruktori elemenata

Ponekad je neophodno za upit da kreira ili generiše elemente. Takvi elementi se mogu generisati direktno u upitu u okviru iskaza nazvanog konstruktori elemenata.

Primer 2.9 *Generisati elemente* `<zaposleni>` *koji imaju "zapid" atribute. Vrednost atributa i sadržaj elementa su specifikovani promenljivom* `$id` *koja je dodeljena u nekom drugom delu upita.*

```
<zaposleni zapid = {$id}>
             {$ime}
             {$posao}
       </zaposleni>
```

FLWR iskazi

FLWR se izgovara kao "flower". Ovaj iskaz je upit koji se sastoji od FOR, LET, WHERE i RETURN klauze. FOR klauza je iterativna konstrukcija koja promenljivoj dodeljuje niz vrednosti vraćenih upitom. To je često XPath iskaz. LET klauza slično dodeljuje vrednosti promenljivim ali umesto serije dodeljivanja vrši se samo jedno dedeljivanje. WHERE klauza sadrži jedan ili više uslova koje čvorovi koji su vraćeni LET ili FOR klauzama, treba da zadovolje. RETURN klauza generiše rezultat FLWR iskaza, što može biti bilo koji niz čvorova ili prostih vrednosti. Ova klauza se izvršava jednom za svaki čvor koji je vraćen FOR i LET klauzom i nakon prolaska kroz WHERE klauzu. Ono što se vrati njenim izvršenjem predstavlja rezultat ovog izraza.

Primer 2.10 *Izlistati sve naslove knjiga čiji je izdavač "Morgan Kaufmann" u 1998. godini.*

```
FOR $b IN document("bib.xml")//knjiga
        WHERE $b/izdavac = "Morgan Kaufmann"
        AND $b/godina = "1998"
        RETURN $b/naslov
```

Primer 2.11 *Izlistati sve izdavače koji su izdali više od 100 knjiga.*

```
       <veliki_izdavaci>
           {
             FOR $p IN distinct(document("bib.xml")//izdavac)
             LET $b := document("bib.xml")//knjiga[izdavac = $p]
             WHERE count($b) > 100
             RETURN $p
           }
        </ veliki_izdavaci >
```

Uslovni iskazi

Uslovni iskazi ocenjuju test iskaze i onda vraćaju jedan od dva rezultujuća iskaza. Ako je vrednost test iskaza tačno onda se vraća kao rezultat vrednost prvog rezultujućeg iskaza, u suprotnom, vraća se vrednost drugog.

Primer 2.12 *Napraviti listu svih knjiga uređenih po naslovu. Za beletristiku, uključiti izdavača a za sve ostale autora.*

```
FOR $k IN //knjiga
        RETURN
          <knjiga>
            {$k/naslov,
             IF ($k/@zanr = "Beletristika")
             THEN $k/izdavac
             ELSE $k/autor
            }
          </knjiga>
         SORTBY (naslov)
```

Kvantifikovani iskazi

XQuery podržava konstrukcije koje su ekvivalentne kvantivikatorima koji se koriste u matematici i logici. SOME klauza je ekvivalent kvantifikatoru "postoji" i koristi se da se ispita da li u nizu vrednosti postoji bar jedan čvor koji zadovoljava uslov. EVERY klauza je ekvivalent kvantifikatoru "za svaki" i koristi se da se ispita da li svi čvorovi u skupu vrednosti zadovoljavaju odgovarajući uslov.

Primer 2.13 *Pronalaći naslove svih knjiga u kojima su "jedrenje" i "surfovanje" pomenuti u nekom paragrafu.*

```
FOR $k IN //knjiga
        WHERE SOME $p IN $b//paragraf SATISFIES
        (contains($p, "jedrenje") AND contains($p, "surfovanje"))
        RETURN $k/naslov
```

Primer 2.14 *Pronalaći naslove knjiga u kojima se "jedrenje" pominje u svakom paragrafu.*

```
FOR $k IN //knjiga
        WHERE EVERY $p IN $k//paragraf SATISFIES
           contains($p, "jedrenje")
        RETURN $k/naslov
```

Iskazi koji u sebi uključuju korisnički definisane funkcije

Osim toga što je podržana centralna biblioteka funkcija sličnih onima u XPath-u, XQuery takođe daje mogućnost korisnicima da definišu funkcije koje će proširiti ovu biblioteku.

Primer 2.15 *Napisati funkciju koja za knjigu za koju su date informacije o ceni i popustu (u procentima), izračunati cenu sa popustom.*

```
declare function local:minCena($cena as xs:decimal?,
             $popust as xs:decimal?) AS xs:decimal?
{
let $pop := ($cena * $popust) div 100
return ($cena - $pop)
};
(: Primer poziva ove funkcija je::)
<minCena>
{
   local:minCena($knjiga/cena,$knjiga/popust)
}
</minCena>
```

2.3 XML modeli za opisivanje podataka

XML dakle obezbeđuje novi tip modela za opisivanje podataka koji je predstavljen uređenim drvetom sa čvorovima kojima su dodeljeni tipovi i imena i kome su podaci predstavljeni samo u listovima. Za razliku od relacionih modela podataka, ne postoji jedinstveni XML model podataka. Ipak, svi XML modeli podataka su izvedeni iz osnovnog modela. Osnovni model uključuje različite tipove čvorova koje ćemo predstaviti sa osvrtom na sledeći primer [4]:

Primer 2.16 *XML dokument koji predstavlja deo menija nekog restorana*

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE meni>
<meni datum='5.10.2007'>
    <jelo>
        <ime>Domaca pileca supa</ime>
        <cena>100.00</cena>
        <opis>
        cinija pilece supe sa biberom
```

```
        </opis>
        <kalorije>650</kalorije>
    </jelo>
    <jelo>
        <ime>Francuski tost</ime>
        <cena>60.50 din</cena>
        <opis>
        tanki parcici napravljeni od domaceg hleba
        </opis>
        <kalorije>300</kalorije>
    </jelo>
    <jelo>
        <ime>Bakin dorucak</ime>
        <cena>195.95 din</cena>
        <opis>
        dva jaja, slanina ili kobasica, tost
        </opis>
        <kalorije>350</kalorije>
    </jelo>
</meni>
```

Različiti tipovi čvorova osnovnog modela su:

- Čvor **elementa**. On ima tip (na primer `cena`, `jelo` ili `meni`), može imati uređenu listu dece (za čvor elementa tipa `jelo` to su elementi tipa `ime`, `cena`, `opis`, `kalorije`) kao i neuređen skup atributa u formi (ime_atributa = vrednost_atributa) parova, na primer, "datum='5.10.2007'". Primer čvorova ovog tipa je:

  ```
  <cena>195.95 din</cena>
  <meni datum='5.10.2007'>...</meni>
  ```

- Čvor **dokumenta**. To je specijalna vrsta čvora elementa predstavljena preko `<!DOCTYPE>` čvora. Ima tip (u ovom primeru to je `meni`) ali nema atribute. Ima tačno jedan čvor elementa koji je izveden iz njega (njegovo dete), koji mora imati isti tip kao čvor dokumenta;

- Čvor za **procesiranje instrukcija** kao na primer `<?xml version...?>`;

- **Komentari** (u formi `<!--komentar-->`);

- **Podaci** koji se javljaju samo u listovima, na primer "dva jaja, slanina ili kobasica, tost" ili "Bakin dorucak".

XML se takođe može posmatrati i kao aptstraktni tip podataka (ADT — Abstract Data Type) koji je veoma bogat i sadrži stringove i identifikatore.

Postoje i XML modeli podataka koji su bazirani na upitnim jezicima koje koriste, XQuery i XPath. Takozvani XDM, XQuery 1.0 i XPath 2.0 modeli podataka, su W3C preporuka od kraja 2005. godine.

3

XML baze podataka

3.1 Da li je XML baza podataka?

Kada je reč o XML-u, pitanje koje se nameće je: "Da li se XML dokument može smatrati bazom podataka?"

Ako se pod bazom podataka podrazumeva bilo kakva kolekcija podataka, onda se XML u tom striktnom smislu, može smatrati bazom podataka. Na taj način posmatrano, svaka datoteka bi mogla predstavljati bazu podataka jer svaka od njih u sebi čuva neku vrstu podataka. XML međutim ima neke osobine koje mu u tom smislu daju prednosti u odnosu na druge dokumente. Samo-opisiv je (etikete opisuju strukturu i tip podataka), prenosiv je (Unicode) i može predstaviti podatke u drvolikoj strukturi ili u obliku grafa.

Mnogo korisnije pitanje je da li XML i tehnologije koje ga okružuju mogu predstavljati sistem za upravljanje bazama podataka? Ono što ide u prilog tome je da XML može obezbediti većinu stvari koje su karakteristične za baze podataka: skladište podataka (XML dokumenti), sheme (DTD, XML sheme), upitne jezike (XQuery, XPath, XQL, XML-QL, QUILT), interfejse (SAX, DOM, JDOM), i drugo. Ono što ne ide u prilog tome jeste da njemu nedostaje mnogo stvari zajedničkih bazama podataka a to su: efikasno skladištenje, indeksi, bezbednost, transakcije i integritet podataka, pristup od strane više korisnika, trigeri, upiti nad više dokumenata odjednom i tako dalje.

Dakle, u situacijama kada se radi sa malom količinom podataka, kada postoji svega nekoliko korisnika i kada su skromni zahtevi za performansama, XML dokumenti se mogu smatrati bazom podataka. Primer su INI datoteke (datoteke koje se koriste za konfigurisanje inicijalnih podešavanja programa). To je datoteka koja se čita i piše linearno i to samo kada se aplikacija startuje ili kada se završava. Finiji primer je telefonski imenik (ime, prezime, broj telefona, adresa i tako dalje). Međutim kako postoje baze podataka koje su jeftine i lake za korišćenje (Access), nema puno razloga da se u te svrhe koristi XML. Jedina prava prednost jeste prenosivost podataka ali u većini slučajeva ona nije i dovoljna.

U drugim situacijama koje su mnogo brojnije, kada postoji mnogo korisnika, kada su striktni zahtevi za integritetom podataka i kada postoje zahtevi za dobrim performansama, XML dokumenti ne ispunjavaju uslove koji su neophodni da bi se koristili kao baze podataka.

3.2 Vrste XML dokumenata

Svi XML dokumenti, prema stepenu rigidnosti svoje strukture, upadaju u dve široke kategorije: podatak-centrični (eng. data-centric) i dokument-centrični (eng. document-centric) [1].

3.2.1 Podatak-centrični dokumenti

Podatak-centrični dokumenti su dokumenti koji XML koriste za čuvanje i razmenu podataka. Oni su projektovani tako da se koriste uglavnom za obradu od strane računara pre nego za ljudsku upotrebu. Činjenica da se podaci neko izvesno vreme pamte i razmenjuju u XML formatu, za aplikaciju koja koristi te podatke nije od značaja. Primeri takvih podataka su jelovnik u nekom restoranu, redosled prodaje proizvoda u nekoj prodavnici, plan letenja aviona na aerodromu i tako dalje. Ovi podaci se karakterišu prilično regularnom strukturom, fino-zrnastim podacima (najmanje nezavisne jedinice podataka su na nivou PCDATA, elementa ili atributa) i jasno određenim redosledom sa malo ili bez mešovitog sadržaja. Ovakvi podaci su najčešće smešteni u nekoj relacionoj bazi podataka i javlja se potreba za transferom podataka iz relacione baze u XML dokument, iz XML dokumenta u relacionu bazu podataka ili u oba smera. Primer ovakve vrste dokumenta je:

```
<meni datum='5.10.2007'>
    <jelo>
        <ime>Domaca pileca supa</ime>
        <cena>100.00</cena>
        <kalorije>650</kalorije>
    </jelo>
    <jelo>
        <ime>Francuski tost</ime>
        <cena>30.50 din</cena>
        <kalorije>300</kalorije>
    </jelo>
    <jelo>
        <ime>Bakin dorucak</ime>
        <cena>200.00 din</cena>
        <kalorije>350</kalorije>
    </jelo>
</meni>
```

U opštem slučaju, svaki veb sajt koji dinamički konstruiše HTML dokument popunjavajući šablon podacima iz baze podataka može biti zamenjen nizom podatak-centričnih XML dokumenata i jednim ili više XSLT (Extensible Stylesheet Language Transformations) dokumentom. XSLT je jezik koji omogućava transformaciju XML dokumenta u neku drugu formu (na primer u HTML ili XHTML dokument).

3.2.2 Dokument-centrični dokumenti

Dokument-centrični dokumenti su projektovani tako da se uglavnom koriste za ljudsku upotrebu. Primeri su knjige, elektronska pošta, reklame i gotovo svaki

ručno napravljeni XHTML dokument. Oni se karakterišu manje regularnom ili neregularnom strukturom, krupno zrnastim podacima (najmanja nezavisna jedinica može biti na nivou elementa sa mešovitim sadržajem, a može biti i na nivou celog dokumenta) i dosta mešanim sadržajem. Redosled elemenata vrlo često nije od značaja.

Ovakvi dokumenti su najčešće ručno pisani u XML-u ili nekom drugom formatu (RTF, PDF, SGML) a onda konvertovani u XML. Primer ovakvog dokumenta je:

```
<Proizvod>
  <Ime>KIRKOLINA  caj za mrsavljenje</Ime>
  <Proizvodjac>Kirka-Pharma</ Proizvodjac>
  <Opis>
    <Paragraf>
      Predstavlja mesavinu lekovitog bilja koje kombinovanim
      dejstvom regulisu promet materija u organizmu,
      ubrzavaju sagorevanje masnih naslaga i uticu na
      smanjenje telesne tezine. <i>Krusina, sena, zova</i>
      stimulisu metabolizam, podsticu probavu, smanjuju nadutost.
      <i>Breza, pirevina, rastavic</i> eliminisu nakupljene
      toksicne materije, poboljsavaju cirkulaciju.
      <i>Maticnjak</i> oslobada od stresa koji je cesto uzrok
      nekontrolisanog konzumiranja hrane. <i>alfija</i>
      kao izuzetni antiseptik stiti od mogucih infekcija
      i utice na jacanje organizma.
    </Paragraf>
    <Paragraf>
      <b>Moete:</b>
    </Paragraf>
    <List>
      <Item>
        <Link URL="Naruci.html">Naruci caj</Link>
      </Item>
      <Item>
        <Link URL="Kirkolina.htm">Vise o ovom proizvodu</Link>
      </Item>
      <Item>
        <Link URL="Katalog.zip">Katalog nasih proizvoda</Link>
      </Item>
    </List>
    <Paragraf>
      Ovaj caj kosta <b>samo 500 dinara.</b>
    </Paragraf>
  </Opis>
</Proizvod>
```

U praksi nije uvek jasna razlika između ova dva tipa dokumenata. Tako na primer, podatak-centrični dokument kao na primer faktura, može sadržati krupno-zrnaste podatke, neregularno struktuirane. S druge strane, dokument-centrični dokument kao na primer korisnička uputstva mogu sadržati fino-zrnaste,

regularno struktuirane podatke kao na primer ime autora i datum revizije. Stroga podela između podatak-centričnih i dokument-centričnih dokumenata nije uvek moguća, tako da je ponekad teško odrediti model korišćenja XML-a. U specifičnim slučajevima kada je potrebno koristiti dokumenta oba tipa potrebno je napraviti hibridni model kakav je predložio Dare Obasanjo [3].

3.3 Prenos podataka između XML dokumenta i baze

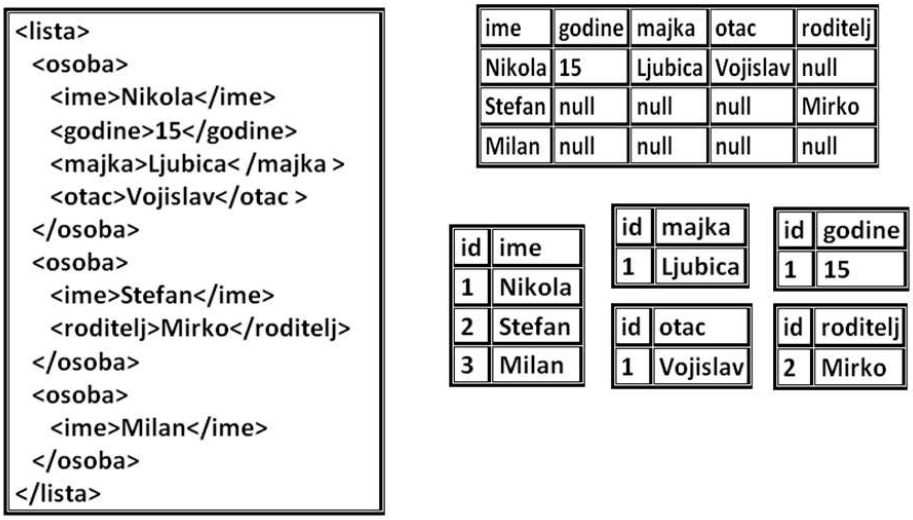
Jedna od popularnih tehnika upravljanja XML podacima jeste njihovo preslikavanje u već postojeće (relacione) baze podataka. Neophodno je dakle izvršiti preslikavanje XML sheme (DTD, XML Shema) u shemu baze podataka. Za ovaj prenos podataka projektuje se poseban softver koji može biti ugrađen u neki od srednjih aplikativnih slojeva (eng. middleware) ili je pak softver za prenos sastavni deo same relacione baze podataka kada kažemo da je baza podataka XML-proširena.

Prilikom prenosa podataka iz XML dokumenta u relacionu bazu podataka prihvatljivo je da se odbace neke informacije o dokumentu (ime i DTD) kao i njegova fizička struktura (definicija i korišćenje entiteta, CDATA sekcije, način korišćenja binarnih podataka i drugo). U nekim situacijama je čak prihvatljivo da se odbaci i deo logičke strukture dokumenta (instrukcije za procesiranje i redosled pojavljivanja elemenata sa istog nivoa). Pri prenosu u suprotnom smeru, iz relacione baze u XML dokument, dobijeni dokument najverovatnije neće sadržati CDATA sekcije a redosled elemenata na istom nivou zavisiće od redosleda u kome ih baza podataka daje na izlazu. Kao posledica ignorisanja informacija o dokumentu i njegovoj fizičkoj strukturi dolazi do toga da kada se XML dokument transformiše u relacione podatke, a zatim se na osnovu dobijenih podataka pokuša rekonstrukcija polaznog dokumenta, dobija dokument koji se znatno razlikuje od polaznog XML dokumenta. U tom slučaju se kaže da sistem ne omogućava kružno putovanje (eng. round trip) XML dokumenta.

Softver za prenos podataka svoj rad zasniva na preslikavanju između dokumenata i baze odnosno između elemenata XML dokumenta i tabela i atributa relacione baze. Međutim, zbog fleksibilne prirode XML-a, ovakvo preslikavanje vrlo često ima kao rezultat ili tabele sa velikim brojem nedostajućih vrednosti ili veoma veliki broj tabela (slika 3.1). Teško je uklopiti bogatu strukturu XML-a u krute relacione tabele bez deljenja dokumenta u veoma male standardne jedinice koje se mogu prikazati kao ćelije u tabeli. Zbog toga čak i jednostavna XML shema proizvodi veliki broj tabela. Strukturne informacije u shemi baziranoj na drvetima dobijaju se spajanjem tabela u relacionoj shemi. XML upiti se konvertuju u SQL upite nad relacionim tabelama i tako se čak i jednostavni XML upiti transliraju u skupu seriju spajanja tabela odgovarajuće relacione baze.

U [1] Ronald Bourret je definisao dve vrste preslikavanja sheme XML dokumenta u shemu relacione baze podataka. To su preslikavanje bazirano na tabelama i objektno-relaciono preslikavanje.

Preslikavanje bazirano na tabelama se koristi kod mnogih proizvoda srednjeg aplikativnog sloja za preslikavanje XML dokumenta u relacionu bazu podataka. XML dokumenti se modeluju kao jedna tabela ili skup tabela.



Slika 3.1: Relaciona reprezentacija dela XML dokumenta

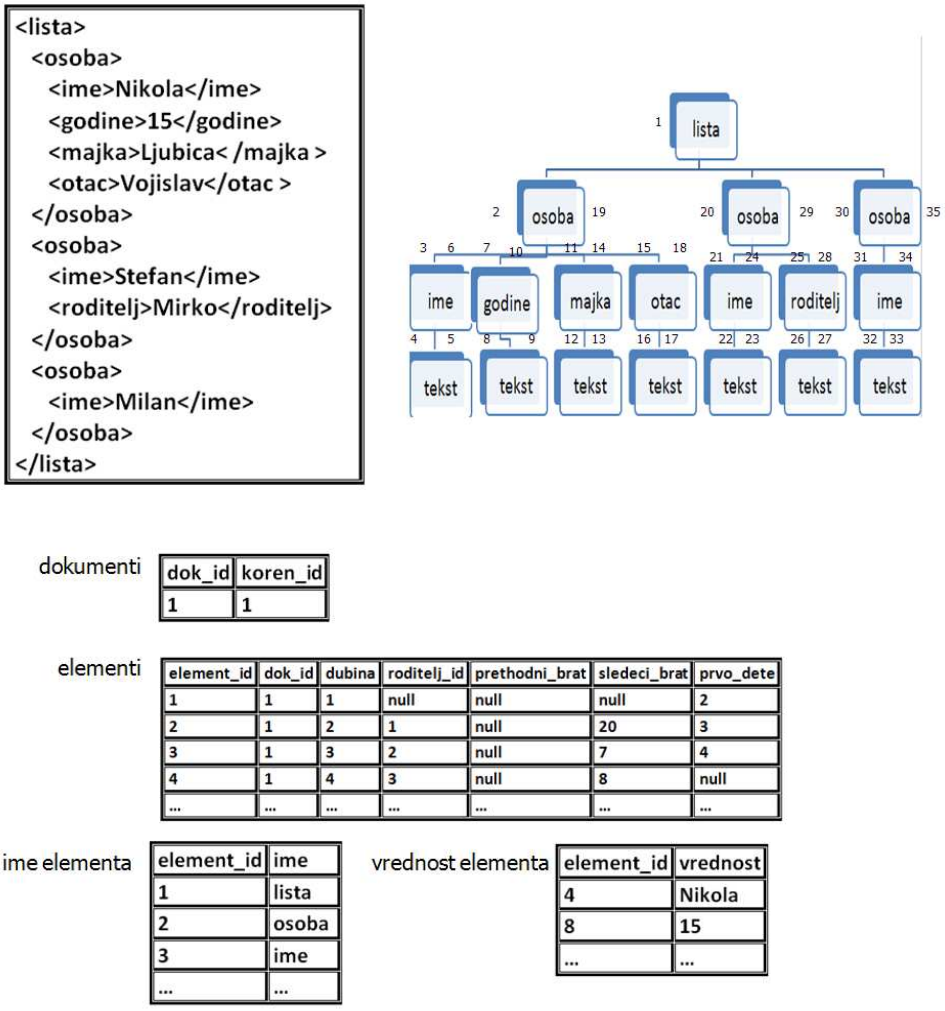
Objektno-relaciono preslikavanje se koristi od strane svih XML-proširenih relacionih baza podataka i nekih proizvoda srednjeg aplikativnog sloja. Ovom metodom, XML dokument se modeluje kao stablo objekata koji su specifični za podatke u tom dokumentu. Kod ovog modela se tipovi elemenata sa atributima, sadržajem ili kombinovanim sadržajem u opštem slučaju modeluju kao klase a prosti elementi, atributi i sami podaci se modeluju kao skalarni podaci klasa. Model se zatim preslikava na relacionu bazu podataka korišćenjem standardnih objektno-relacionih tehnika koje su detaljno opisane u [7]. Primer je dat na slici 3.2. Ovu vrstu preslikavanja treba razlikovati od DOM (Document Object Model) koji modeluje sam dokument i isti je za sve XML dokumente. Objektno-relaciono preslikavanje modeluje podatke u dokumentu i razlikuje se za skupove XML dokumenata koji se uklapaju u zadatu XML shemu.

Primenom ove vrste preslikavanja, struktura dokumenta mora tačno da se uklapa u strukturu koju preslikavanje očekuje. Kako ovo često nije slučaj, korišćenjem XSLT-a dokument se prvo transformiše u odgovarajuću strukturu koju preslikavanje očekuje pa se tek onda prenosi u bazu. Slično, dokument koji se dobije iz baze podataka, pre upotrebe prvo se transformiše u strukturu koju očekuje aplikacija.

3.3.1 Problemi koji se javljaju pri prenosu podataka

Neki od najčešćih problema koji se javljaju pri prenosu podataka između XML dokumenata i baze su sledeći [8]:

- **Tipovi podataka.** U opštem slučaju, XML nema ugrađenu podršku za tipove podataka i sa izuzetkom CDATA entiteta, ceo XML dokument predstavlja običan tekst. To znači da softver za prenos podataka mora da izvrši konverziju teksta u druge tipove podataka pri čemu je broj formata koje softver prepoznaje ograničen. Jedno od rešenja ovog problema je dodavanje XML dokumentu tipova podataka preko rezervisanih atributa.



Slika 3.2: Primer arhitekture bazirane na modelu

Pored toga, XML shemom se dokumentu na jednostavan način, pored opisa strukture, može dodati i deklaracija tipova.

- **Binarni podaci.** U XML dokumentu binarni podaci se mogu čuvati na dva načina: kao CDATA elementi ili korišćenjem BASE64 kodiranja. I jedan i drugi pristup su problematični sa stanovišta softvera za prenos podataka. Ne postoji standard u XML-u za indikaciju da neki element sadrži BASE64 kodirane podatke pa se može desiti da softver ne prepozna da se radi o binarno kodiranim podacima. Kritično je i to što se notacija vezana za BASE64 kodiranje i za CDATA elemente najčešće zanemaruje prilikom prenosa.

- **NULL vrednosti.** XML podržava NULL vrednosti preko koncepta opcionih atributa i elemenata. Pri preslikavanju strukture XML dokumenta treba voditi računa da se opcioni elementi i atributi preslikavaju samo na one kolone u tabelama koje dozvoljavaju NULL vrednosti. Isto važi i pri prenosu u suprotnom smeru.

- **UNICODE podrška.** S izuzetkom nekih kontrolnih karaktera, XML može sadržati samo UNICODE karaktere. Najveći broj relacionih baza podataka nudi veoma slabu ili nikakvu podršku za UNICODE i obično zahtevaju da se unapred izvrši posebna konfiguracija.

- **Procesirajuće instrukcije i komentari.** Oni nisu deo podataka u XML dokumentu i najčešće se zanemaruju prilikom prenosa. Mogu se pojaviti bilo gde u okviru XML dokumenta i zbog toga se ne uklapaju ni u jedno preslikavanje.

- **Pamćenje markera.** Nekad je korisno da se elementi sa kompleksnim sadržajem zapamte takvi kakvi su bez dalje obrade. U XML dokumentima se markeri pamte korišćenjem referenci entiteta. Ovo se može iskoristiti i kod baze podataka ali problem je sa SQL-om koji nije u stanju da raspoznaje reference entiteta.

3.4 XML baze podataka

Prilikom korišćenja XML podataka veoma je bitno izabrati način na koji će ti podaci biti trajno sačuvani. U takvim situacijama, do izražaja dolaze prednosti koje DBMS (DataBase Managment System) može da ponudi aplikacijama. DBMS nudi veliki broj servisa koji su suviše važni i kritični da bi bili prepušteni aplikacijama da ih same implementiraju a tu se pre svega misli na sigurnost, transakcije, rad većeg broja korisnika, programske interfejse i drugo. Međutim, kada se radi o korišćenju DBMS-a za čuvanje XML podataka javlja se veliki broj problema koje je neophodno rešiti.

Michael Champion, u radu [9], daje najbitnije razlike između relacionih podataka i XML podataka. Relacioni podaci su normalizovani, imaju jednostavnu strukturu i podeljeni su na veći broj relacija. XML podaci nisu normalizovani, imaju hijerarhijsku strukturu i uvek su monolitni. Michael Champion je takođe izvršio podelu mogućih rešenja za čuvanje XML podataka na relacione, XML-proširene (post-relacione) i izvorne XML baze podataka.

3.4.1 Relacione baze podataka

Relaciona rešenja predlažu pristup po kome XML podatke treba normalizovati na tabele i kolone pri čemu svaka ćelija sadrži atomični tekstualni podatak. Za sam postupak normalizacije se može koristiti pristup koji je opisan u [10]. Generalno svaki XML dokument se može normalizovati za smeštanje u relacionu bazu podataka. Ovo rešenje ima smisla samo kod dobro struktuiranih XML dokumenata. Što je dokumet manje struktuiran, ima više kompleksnih elemenata i elemenata sa kompleksnim sadržajem, to će broj tabela u bazi drastično rasti tako da će na kraju relacioni model XML dokumenta izgubiti svaki praktični smisao.

3.4.2 XML-proširene (post-relacione) baze podataka

Post-relaciona rešenja su nastala u pokušaju da se relacione baze podataka prilagode zahtevima koje je pred njih postavio objektno orijentisani pristup. Ta nova rešenja u velikoj meri olakšavaju upravljanje XML podacima. Većina relacionih DBMS je uvela LOB (Large OBjects) tipove podataka koji omogućavaju smeštanje velikih količina podataka u pojedinačnim ćelijama. Osim toga proizvođači DMBS-a (a i sam SQL standard) su dodali podršku za ćelije koje mogu da sadrže podatke koji se ponavljaju kao i mogućnost pretrage celog teksta (eng. full-text) odnosno pretrage svih reči sadržanih u tekstu. Pored ovoga razvijen je veliki broj dodatnih alata koji omogućavaju lakše manipulisanje XML podacima. Kada se uz pomoć pretrage celog teksta pronađe podatak koji je od interesa, uz pomoć ovih alata se podatak može predstaviti u XML obliku i može se manipulisati njime korišćenjem DOM-a ili XPath-a. Međutim ni jedan od ovih sistema ne omogućava kompletno kružno putovanje (eng. round trip) proizvoljnog dokumenta iz baze i u bazu.

3.4.3 Izvorne XML baze podataka

Izvorne XML baze podataka (eng. Native XML Database — NXD), za razliku od prethodnih rešenja koja pokušavaju samo da modifikuju postojeće relacione baze, su baze podataka projektovane specijalno za upravljanje XML dokumentima. One održavaju prirodnu drvoliku strukturu ovih dokumenata i karakterišu se velikom fleksibilnošću.

Izvorne XML baze podataka po definiciji (http://www.xmldb.org/) podrazumevaju:

- Definisanje (logičkog) modela za XML dokumente i smeštanje i dobijanje dokumenata u skladu sa tim modelom. Minimalno, model mora uključiti elemente, atribute, PCDATA i redosled dokumenta. Primeri takvih modela su XPath model podataka, XML Infoset (XML Information Set), i modeli implicirani sa DOM (Document Object Model) i događajima u SAX (Simplified API for XML);

- XML dokument im je osnovna jedinica (logičkog) skladištenja, kao što je to vrsta u tabeli kod relacionih baza;

- Nema zahteva za postojanjem specifičnog fizičkog modela skladištenja. Na primer, baza može biti izgrađena na relacionoj, hijararhijskoj ili objektno-

orijentisanoj bazi, ili može koristiti odgovarajući format za smeštanje kao što su indeksovane, kompresovane datoteke.

Zaključak koji se može izvesti iz ove definicije je da su izvorne XML baze specijalizovane za smeštanje XML dokumenata, smeštajući pri tome sve komponente XML modela netaknutim. XML dokumenti se smeštaju i XML dokumenti se dobijaju iz baze. XML baze podataka ne moraju biti samostalne baze i ne moraju smeštati XML podatke u pravoj izvornoj formi (kao tekst). Format u kome se vrši skladištenje podataka ili fizički model nisu od značaja za kategorizaciju baze podataka.

Treba imati na umu da izvorne XML baze podataka nemaju tendenciju da zamene postojeće baze podataka već da pomognu u efikasnom upravljanju XML dokumentima. U tabeli 3.1 dato je poređenje relacionih i XML baza podataka [2].

Relaciona baza	**XML baza**
Osnovu čini tabela	Osnovu čini kolekcija
Relaciona tabela sadrži slogove (koji predstavljaju vrste u tabeli) definisane po istoj shemi	Kolekcija sadrži XML dokumente istog modela
Relacioni slog predstavlja listu nesortiranih vrednosti	XML dokument ima strukturu drveta sa međusobno povezanim čvorovima
SQL upit vraća nesortiran skup slogova	XQuery upit vraća sortiranu sekvencu čvorova

Tabela 3.1: Poređenje relacionih i XML baza podataka

Arhitektura izvornih XML baza podataka može upasti u dve široke kategorije: bazirana na tekstu i bazirana na modelu [1].

Izvorne XML baze bazirane na tekstu čuvaju XML podatke kao tekst. Ono što je karakteristično za ove XML baze je izuzetno dobro indeksiranje koje omogućava da se veoma brzo referencira bilo koji XML dokument ili neki njegov deo. Baze bazirane na tekstu zbog toga postižu izuzetne performanse kada se podacima pristupa u skladu sa predefinisanom hijerarhijom.

Izvorne XML baze bazirane na modelu formiraju interni objektni model na osnovu XML dokumenta i onda skladište taj model. Najčešće su izgrađene tako da koriste neku relacionu bazu kao sredstvo za fizičko skladištenje podataka (videti primer na slici 3.2). U tom slučaju performanse ovih baza u velikoj meri zavise od relacionih baza koje rade u pozadini. Ako koriste neki drugi sistem za skladištenje podataka onda se obično koriste fizički pokazivači između XML dokumenata čime se obezbeđuju performanse koje su veoma slične performansama baza zasnovanih na tekstu.

Osobine izvornih XML baza podataka

Iako nisu sve izvorne XML baze podataka iste, postoje neke osobine koje karakterišu svaku od njih kao što su: podrška kolekciji dokumenata, transakcija, sigurnost, višekorisnički pristup, programski interfejs, upitni jezici i tako dalje.

Kolekcije dokumenata

Mnoge izvorne XML baze podataka podržavaju logički model grupisanja dokumenata, nazvan "kolekcije". Kolekcija igra ulogu sličnu tabeli u relacionim bazama ili direktorijumu u programskim sistemima. Na taj način se obezbeđuje da upiti koji se postavljaju budu ograničeni na određeni skup dokumenata. Ono što kolekciju razlikuje od tabele u relacionim bazama je da ne zahtevaju sve izvorne XML baze da shema bude pridružena kolekciji. To znači da se kod takvih baza može smestiti bilo koji XML dokument u kolekciju, bez obzira na shemu. Upiti se i dalje mogu postavljati za sva dokumenta u kolekciji. Izvorne XML baze koje podržavaju ovu osobinu zovu se shema-nezavisne.

Ovo je osobina koja bazi daje veliku fleksibilnost. Na žalost, ovo može biti i loša osobina jer postoji opasnost od narušavanja integriteta podataka. Zbog toga se u situacijama kada je integritet veoma važan, ne preporučuje korišćenje baza koje su shema-nezavisne.

Kod nekih izvornih XML baza, može se definisati i hijararhija kolekcija odnosno u okviru neke kolekcije može se definisati neka druga kolekcija dokumenata.

Upitni jezici

Skoro sve izvorne XML baze podataka podržavaju jedan ili više upitnih jezika. Najpopularniji je XPath (sa proširenjem za upite kroz više dokumenata) i XQuery, deklarativni upitni jezik preporučen od strane W3C.

Ažuriranje i brisanje baze

Ažuriranje predstavlja veliku slabost postojećih izvornih XML baza podataka. Postoji veliki broj strategija: od jednostavnog brisanja i zamene, preko korišćenja DOM stabla pa do definisanja posebnih jezika za modifikaciju. Mnogi proizvodi zahtevaju dobijanje dokumenta iz baze, njegovo ažuriranje korišćenjem nekog od XML API (XML Application Programming Interface) i onda njegovo vraćanje u bazu. Nekoliko proizvoda imaju razvijene odgovarajuće jezike koji omogućavaju ažuriranje dokumenta unutar baze. U te svrhe, neke izvorne XML baze podataka podržavaju XML:DB XUpdate, jezik koji je XML-zasnovan. U okviru njega se koristi XPath za identifikovanje nekog skupa čvorova. Zatim se vrši specifikacija da li se ti čvorovi žele ubaciti u bazu, izbrisati ili se želi ubacivanje novih čvorova pre ili posle tog identifikovanog skupa čvorova.

Bez obzira na to, ažuriranje će biti problematično sve dok XQuery ne doda skup proširenja za ažuriranje koji je predložen od strane članova W3C grupe. Do tada, DOM manipulacija će biti najzastupljeniji metod ažuriranja korišćen od strane izvornih XML baza podataka.

Transakcije, zaključavanje i konkurentnost

U okviru izvornih XML baza podataka, zaključavanje je uvek na nivou čitavog dokumenta a ne na nivou jednog čvora pa višekorisnička konkurentnost može biti na relativno niskom nivou. Ovo zavisi od aplikacije i od toga šta jedan dokument predstavlja (poglavlje korisničkog vodiča ili sve podatke o nekoj kompaniji, finansijski ugovor i tako dalje).

Problem sa zaključavanjem jednog čvora je u implementaciji. Zaključavanje jednog čvora obično zahteva zaključavanje njegovih roditelja pa njihovih roditelja i tako do korena drveta što vodi ka zaključavanju celog dokumenta.

Programski interfejsi (Application Programming Interfaces — APIs)

Što se tiče programskog interfejsa uglavnom se razvijaju interfejsi koji su veoma slični ODBC-u (Open DataBase Connectivity) ali se razlikuju od proizvoda do proizvoda. Najpoznatiji su XML:DB API i XQJ (XQuery api for Java). Oni imaju metode za konektovanje na bazu, izvršavanje upita i dobijanje rezultata. Rezultati se obično vraćaju kao XML niska, DOM drvo ili SAX parser. Postoje pokušaji da se razvije univerzalni programski interfejs koji bi bio nezavisan od proizvođača.

Kružno putovanje (Round-Tripping)

Bitna osobina izvornih XML baza podataka je mogućnost kružnog putovanja XML dokumenata. To znači da se neki XML dokument može smestiti u bazu i da se isti taj može dobiti nazad. To je bitno za dokument-centrične aplikacije a manje bitno za podatak-centrične aplikacije. Stepen izraženosti ove osobine zavisi od baze, da li je tekstualno bazirana ili zasnovana na modelu. U opštem slučaju tesktualno bazirane baze podržavaju kružno putovanje samog dokumenta dok baze zasnovane na modelu podržavaju kružno putovanje na nivou modela.

Indeksi

Kao relacione baze tako i izvorne XML baze koriste indekse za lociranje podataka. To obezbeđuje izuzetno dobre performanse kada se podacima pristupa u skladu sa nekom projektovanom hijerarhijom. Međutim svako odstupanje od ove hijerarhije dovodi do drastičnog pada performansi. Zbog toga izvorne XML baze dosta koriste indekse i obično indeksiraju sve elemente. Ovim se smanjuje vreme pristupa podacima ali zato vreme ažuriranja podataka postaje kritično jer treba održavati i veliki broj indeksa. Ono što karakteriše izvorne XML baze je da su performanse prilikom pristupanja neindeksiranim podacima veoma loše.

U opštem slučaju, postoje tri tipa indeksa [1]:

- Vrednosni indeksi — indeksiraju tekst i vrednosti atributa i koriste se za izvršavanje upita oblika "Pronaći sve elemente ili atribute čija je vrednost 'Santa Cruz'";
- Strukturalni indeksi — indeksiraju lokaciju elemenata i atributa i koriste se za izvršavanje upita oblika "Pronađi sve City elemente čija je vrednost 'Santa Cruz'";
- Indeksi celog teksta — indeksiraju pojedinačne tokene u tekstu i vrednosti atributa i koriste se za izvršavanje upita oblika "Pronađi sva dokumenta koja sadrže reč 'Santa Cruz'" ili zajedno sa strukturalnim indeksima "Pronađi sva dokumenta koja sadrže reč 'Santa Cruz' unutar elementa Address".

  Većina izvornih XML baza podataka podržava i vrednosne i strukturalne indekse a neke baze podržavaju i indekse celog teksta.

Normalizacija i referencijalni integritet

Rizik od dupliranih, redudantnih podataka prisutan je kod izvornih XML baza kao i kod relacionih baza podataka, povećavajući time rizik od nekonzistentnosti podataka. Kao ni kod relacionih baza tako ni kod izvornih XML baza ne postoji mehanizam koji od korisnika zahteva da izvrši normalizaciju. Korisnik sam treba da odluči ima li smisla uložiti napor u proces normalizacije ili ne, što zavisi od količine podataka koji se preklapaju u dokumentima. Proces normalizacije kod izvornih XML baza je u mnogome sličan kao onaj kod relacionih baza: potrebno je projektovati dokument tako da nema podataka koji se ponavljaju. Jedna bitna razlika je da XML a samim tim i izvorne XML baze podržavaju viševrednosne elemente dok relacione baze ne podržavaju viševrednosne atribute. Relacije jedan-prema-više u XML-u se mogu prikazati direktno u okviru jednog dokumenta kao deca jednog roditelja bez redudantnosti, što bi u relacionim bazama moglo samo pomoću više tabela. Često je dovoljno normalizaciju izvršiti samo do nekog razumnog nivoa [1].

Pod integritetom se smatraju uslovi koje treba da zadovolje podaci u bazi da bi stanje baze bilo valjano. Pri svakom ažuriranju baze podataka sistem za upravljanje bazama podataka treba da proveri eksplicitno ili implicitno zadate uslove integriteta. Pod referencijalnim integritetom ili integritetom obraćanja kod relacionih baza podrazumeva se provera da svaki strani ključ pokazuje na validan primarni ključ dok kod izvornih XML baza se podrazumeva mehanizam koji obezbeđuje da veze u dokumentu (XLink) ukazuju na validne dokumente ili njihove delove.

Referencijalni integritet se može odnositi na interne veze (kada se veza u dokumentu odnosi na drugi deo istog dokumenta) i eksterne veze (kada se veza u dokumentu odnosi na neki drugi dokument koji pripada ili ne pripada istoj bazi).

Mali broj izvornih XML baza podržava referencijalni integritet i to onaj koji se odnosi na interne veze. Većina njih ga uglavnom podržava samo parcijalno. To je zbog toga što većina izvornih XML baza obavljaju validaciju dokumenta samo kada se dokument ubacuje u bazu. Dakle, ako se ažuriranje vrši na nivou celog dokumenta (dokument se briše i na njegovo mesto se ubacuje novi), onda je validacija dokumenta dovoljna da obezbedi referencijalni integritet koji se odnosi na interne veze. Ali ukoliko se ažuriranje vrši na nivou čvora (ubacivanje, menjanje ili brisanje čvora) onda bi baze morale da preduzmu dodatni posao kako bi se garantovao integritet. Ovako nešto podržano je kod veoma malog broja izvornih XML baza.

Referencijalni integritet koji se odnosi na eksterne veze gotovo da nije podržan (mnoge baze ne podržavaju ni eksterne veze). Razlog tome je što ako veza pokazuje na dokument koji nije deo baze (na primer na neku veb stranu na Internetu) nema načina na koji bi baza mogla kontrolisali takve dokumente. Jedini stepen u kome bi se ovaj integritet mogao postići, jeste provera ispravnosti veza koje pokazuju na druge dokumente ali u okviru iste baze.

3.5 Kako izabrati najbolje rešenje?

Pri izboru baze podataka prvo pitanje na koje treba dati odgovor jeste "Zbog čega će se koristiti baza podataka i na kakav način će se koristiti?" Što se

tiče aplikacija koje mogu koristiti izvorne XML baze podataka, nema nikakvih ograničenja sem da moraju da koriste XML. Takođe, ne postoje striktna pravila o tome koje vrste aplikacija bi trebalo a koje ne bi trebalo da koriste ovu vrstu baza podataka niti postoji alat koji bi automatski mogao da izanalizira XML dokument i odredi vrstu baze podataka koja bi najviše odgovarala upravljanju tom vrstom dokumenata. Takva vrsta odluke ostavljena je samom korisniku. Ipak, postoje neka opšta pravila koja treba poštovati.

Možda najznačajniji faktor u izboru baze podataka je da li će se u nju smeštati podaci ili dokumenti. Kao neko opšte pravilo može se reći da se za podatak-centrične dokumente koriste relacione baze podataka a za dokument-centrične XML dokumente izvorne XML baze. Međutim ovo nije apsolutno pravilo jer na izbor baze podataka ne utiče samo tip XML dokumenata već i način njihovog korišćenja.

Generalno gledano, izvorne baze podataka su izuzetno pogodne za skladištenje nestruktuiranih podataka kod kojih preslikavanja u relacionu bazu podataka nemaju nikakvog smisla. Izvorne baze podataka su takođe pogodne za smeštanje polustruktuiranih podataka. To su podaci koji imaju regularnu strukturu ali ona toliko varira da njihovo preslikavanje u relacionu bazu podataka dovodi do kreiranja velikog broja tabela (što je neefikasno sa stanovišta performansi) ili do velikog broja kolona sa nedostajućim vrednostima (neefikasno sa stanovišta zauzetog prostora). Ovo se pre svega odnosi na velike XML dokumente sa velikim brojem kompleksnih elemenata i sa rekurzivnim strukturama. S druge strane relacione baze su najpogodnije sa struktuirane podatke i za polustruktuirane podatke koji se lako mogu preslikavati.

Što se performansi tiše, na prvi pogled izvorne baze podataka imaju prednost. Međutim to važi samo dok se podacima pristupa u skladu sa definisanom hijerarhijom. U svim ostalim slučajevima performanse drastično opadaju i lošije su nego kod relacionih baza podataka.

Izvorne XML baze su pogodne i u situacijama kada je potrebno izvršiti integraciju podataka. One omogućavaju mnogo veću fleksibilnost od smeštanja podataka u relacione tabele. To je dobro u sutacijama u kojima je potrebno obezbediti smeštanje podataka koji odgovaraju nepoznatoj vrsti sheme. Ovo može biti i problem jer se na taj način može ugroziti integritet podataka. Ove baze takođe podržavaju i promene u shemi kao i podatke bez sheme što je bitno u situaciji kada podaci koji se integrišu nisu pod našom direktnom kontrolom.

Ažuriranje podataka je takođe kritična stavka kod izvornih baza podataka jer ažuriranje podrazumeva da će se pristupati celom XML dokumentu i pored toga što je od interesa samo određeni njegov deo. S druge strane postojanje velikog broja indeksa (radi poboljšanja performansi) dovodi do toga da je ažuriranje vremenski jako zahtevno.

Izvorne XML baze podataka kao prednost nude podršku za XML upite oblika: "Daj mi sve dokumente u kojima treći paragraf nakon početka sekcije sadrži boldovanu reč". Ovakve upite je teško postaviti u jezicima kao što je SQL a veoma lako nekim od upitnih jezika podržanih izvornim XML bazama podataka.

Izvorne baze podataka obično podatke mogu da vrate samo u obliku XML dokumenta ili kolekcije XML dokumenta. Ako aplikacija zahteva podatke u nekom drugom formatu onda mora da se izvrši prevođenje tog XML dokumenta u oblik koji aplikacija očekuje.

Izvorne baze podataka nude veoma slabu podršku za normalizaciju podataka

i gotovo nikakvu podršku za referencijalni integritet.

Najpoznatiji sistemi za upravljanje izvornim XML bazama podataka su: eXist, Berkeley DB XML, Oracle XML DB, MarkLogic Server i drugi.

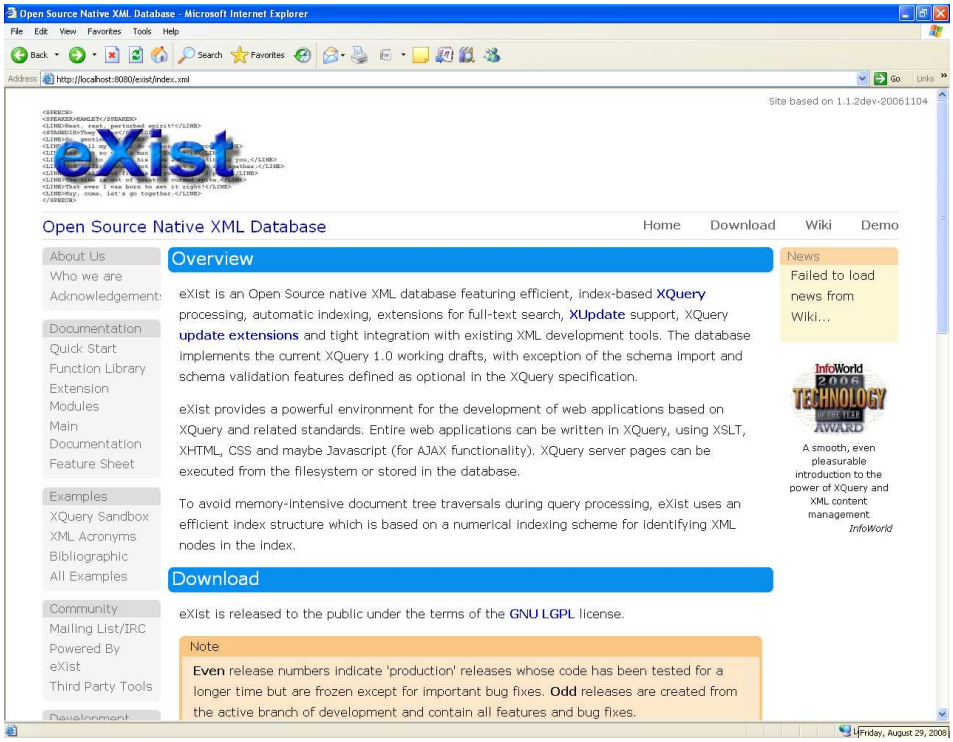
3.6 eXist

Jedan od predstavnika izvornih XML baza podataka jeste eXist DB - Open Source Native XML Database. On jednostavno može da bude integrisan u druge aplikacije koje koriste i obrađuju XML. Baza podataka je potpuno napisana u Javi. Može biti isporučena i korišćena na više načina, kao samostalan proces, unutar nekog servleta ili direktno ubačena u aplikaciju.

eXist pruža mogućnost čuvanja XML dokumenata u kolekcijama uređenim hijararhijski. Korišćenjem proširene XPath sintakse, korisnik može da postavi upit nad određenim delom kolekcije, pa čak i nad svim dokumentima sadržanim u bazi. eXist-ov sistem za postavljanje upita je efikasan i baziran na indeksu. Poboljšana šema indeksiranja omogućava bržu identifikaciju strukturalnih relacija između čvorova, kao što su roditelj-dete ili predak-naslednik relacije. Algoritmi bazirani na vezama između putanja omogućavaju procesiranje velikog broja upita koristeći samo informaciju o indeksu. Pristup aktuelnim čvorovima nije potreban za ovu vrstu izraza. Takođe, eXist pruža veliki broj ekstenzija standardnog XPath jezika, kako bi efikasno procesirao upite nad celim tekstom, uključujući pretragu po ključnoj reči i regularne izraze.

Trenutno, eXist baza najbolje odgovara malim ili velikim kolekcijam XML dokumenata koji se povremeno ažuriraju.

Prilikom standardne instalacije, eXist baza podataka podržana je interfejsom koji predstavlja veb aplikaciju pod Jetty vebserverom. Za pristup ponuđenom interfejsu potrebno je u veb razgledaču ukucati sledeću veb adresu: http://localhost:8080/exist/index.xml. Na slici 3.3 prikazan je izgled početne stranice eXist interfejsa. Na ovoj stranici može se pronaći celokupna dokumentacija sa primerima i pristup administrativnom delu pomoću koga je omogućeno dodavanje novih korisnika, kao i kreiranje novih kolekcija.



Slika 3.3: Početna stranica eXist interfejsa

4

Leksički resursi

Jezički resursi sadrže širok spektar različitih lingvističkih informacija u zavisnosti od njihove prirode i funkcije kojoj su namenjeni. Posebna vrsta jezičkih resursa su leksički resursi koji mogu biti različitih tipova. Neki od primera su prosta lista reči, mašinski čitljivi rečnici, leksikoni, korpusi, rečnici stručnih izraza, azbučni registri reči, skupovi slika, video snimaka i drugo [41]. U zavisnosti od broja jezika na koje se odnose, leksički resursi mogu biti podeljeni na jednojezične, dvojezične i višejezične. Kod dvojezičnih i višejezičnih leksičkih resursa, reči među različitim jezicima mogu biti povezane ili ne. Moguće je izgraditi leksički resurs koji se sastoji od različitih rečnika istog jezika. Na primer, jedan rečnik se može odnositi na uopštene reči a jedan ili više rečnika se mogu odnositi na reči koje pripadaju specijalnim domenima. Najčešći standardi za prikaz leksičkih resursa su XML, SGML i RDF (Resource Description Framework).

Leksički resursi za srpski jezik se razvijaju u okviru Grupe za jezičke tehnologije na Matematičkom fakultetu Univeziteta u Beogradu (Grupa) već duži niz godina, tako da je danas na raspolaganju veliki broj različitih resursa, razvijenih u značajnom obimu [28], [30]. Pored korpusa srpskog jezika, kao i višejezičnih paralalenih korpusa, od posebnog značaja su sistem morfoloških rečnika srpskog jezika (SMR) razvijenih u okviru mreže RELEX [21], kao i semantička mreža za srpski jezik (*Srpski Wordnet* — SWN) razvijena u okviru međunarodnog projekta Balkanet. S obzirom na to da su ovi resursi nastajali tokom dužeg vremena, oni su razvijani u okviru različitih projekata i stoga neminovno unutar različitih konceptualnih i tehnoloških okvira. Iako je Grupa pri tome ulagala velike napore da stepen koherentnosti i standardizovanosti resursa bude što veći, određena mera heterogenosti se nije mogla izbeći. Ovi leksički resursi razvijeni su na osnovu sasvim različitih modela pa samim tim sadrže i različite vrste leksičkih informacija.

Pored već pomenutih resursa, u Grupi se koriste i razvijaju i grafovi, koji se u lingvističkim softverima koriste za formalizaciju lingvističkih fenomena i za obradu (parsiranje) teksta, a pored njih, i dvojezične, paralelne liste, kao pomoćni resurs pri pretraživanju i prevođenju. Konačno, Grupa učestvuje i u razvoju višejezične ontologije vlastitih imena (Prolex [22]), organizovane oko koncepta vlastitog imena, kao jedinstvenog koncepta u različitim jezicima. Naime, u višejezičnom kontekstu, opis vlastitih imena ne može se svesti samo na elektronski rečnik, zbog kompleksnosti semantičkih veza koje ih povezuju.

Pored različitih formata resursa, poseban problem bili su i različiti kodni rasporedi koji su se vremenom javljali u resursima, počev od takozvanog aurora zapisa, u kome su slova ć, č, š, ž, đ, dž, lj i nj kodirana ACCII karakterima cx, cy, sx, zx, dx, dy, lx i nx, preko ISO 8859-2 i ISO 2 8859-5 koda, pa do Unicode-a. Da bi se rešili ovi problemi heterogenosti, nastalo je integrisano i prilagodljivo softversko rešenje, nazvano WS4LR (Work Station for Lexical Resources) kojim je omogućeno upravljanje i rad pojedinačnim resursima, kao i njihovo integrisanje [29].

4.1 Elektronski rečnik

Elektronski rečnik ili e-rečnik je rečnik koji je predstavljen u elektronskoj formi i koji je namenjen ekskluzivno automatskoj transformaciji teksta (za razliku od mašinski čitljivih rečnika koji su namenjeni korišćenju od strane čoveka). Između ostalog, njegova glavna svrha je korišćenje u procesu obrade prirodnog jezika. Kao što je već pomenuto, u okviru Grupe za obradu prirodnih jezika na Matematičkom fakultetu Univeziteta u Beogradu, razvijen je sistem morfoloških rečnika srpskog jezika kao deo mreže RELEX [30].

4.1.1 Sistem morfoloških rečnika srpskog jezika

Sistem morfoloških rečnika SrpDic srpskog jezika, sastoji se od nekoliko osnovnih delova: DELAS, koji predstavlja rečnik prostih reči u osnovnom obliku (prostih lema), DELAC, rečnika složenih reči (tj. kontingentnih niski prostih reči) i DELAF, rečnika oblika prostih reči, kao i morfoloških gramatika koje omogućavaju prepoznavanje "nepoznate" reči, tj. reči koja nije prepoznata na osnovu postojećih rečnika. Aktuelni obim SrpDic obuhvata oko 70.000 prostih reči iz kojih je generisan rečnik DELAF sa preko 1.000.000 oblika prostih reči [30]. Svakom zapisu u rečniku prostih lema (DELAS) pridružena je informacija o vrsti reči i, ako je potrebno, kod flektivne klase, precizan opis promene reči. Elementima rečnika DELAS se mogu dodati morfosintaktičke, sintaksičke ili semantičke kao i informacije o izgovoru. Tako je, na primer [30], pridev `devojcyin` u rečniku DELAS zapisan sa:

```
devojcyin, A1+Pos+Ek (1)
```

što znači da se radi o pridevu koji pripada flektivnoj klasi A1, koji je prisvojan (+Pos), ekavskog izgovora (+Ek). Informacije iz sistema rečnika SrpDic mogu se pomoću sistema Intex [21] koristiti za formulisanje kompleksnih upita za pretraživanje tekstova. Na primer, upitom: `<A+Pos-Ek>` će se dobiti svi prisvojni pridevi u tekstu koji ne pripadaju ekavskom izgovoru. Zapisi u DELAS rečniku mogu sadržati i derivacione relacije kojima se povezuju reči koje pripadaju istom derivacionom gnezdu. Ovaj tip informacija se razdvaja znakom podvake (_). Na primer:

```
devojcyin, A1+Pos+Ek_N=4ka (2) devojka, N618+Hum+Ek_A=2cyin
```

Informacije koje se nalaze iza podvlake u pridevu devojcyin (A) povezuju ga sa imenicom devojka (N) tako što se poslednja četiri karaktera (cyin) zamene slovima ka. U drugom redu pokazano je kako se, na sličan način, imenica devojka povezuje sa pridevom devojcyin. Sem toga, morfosintaktičkim informacijama

(ispred kojih stoji znak plus), može se opisati i tip derivacione veze između dve leme. U primeru (2), iz oznake +Pos vidi se da je pridev devojčin prisvojni pridev imenice devojka. Ove informacije iz rečnika DELAS mogu se koristiti za lematizaciju tekstova na osnovu proizvoljno izabrane leme iz jednog derivacionog gnezda, uz pomoć konačnih transduktora [30].

4.2 Rečnik vlastitih imena

Rečnik vlastitih imena nastao je kao deo Prolex projekta [22] u cilju projektovanja i implementacije višejezičnog rečnika vlastitih imena i relacija među njima. Od 1996. godine u okviru ovog projekta razmatrana su vlastita imena, posebno toponimi i imena stanovnika, i istaknuta je potreba da se sva vlastita imena povežu zajedno. Kreirana je višejezična baza vlastitih imena, nazvana Prolexbase, sa lingvističkim informacijama korisnim u procesu obrade prirodnih jezika.

Za reprezentaciju vlastitih imena i relacija među njima korišćen je XML zbog prednosti koje ima kao što su omogućena integracija i razmena informacija među lingvističkim podacima [27]. Jedan odlomak ovog rečnika dat je na slici 4.1.

Kao što se može primetiti, rečnik se sastoji iz dva dela, jednog koji se odnosi na relacije i drugog koji se odnosi na jezike koji su u njemu definisani.

Prvi deo, koji se odnosi na relacije, ima koren u elementu *relationships* i sastoji se iz:

- liste elemenata tipa *pivot* koji predstavljaju apstraktnu notaciju za definisanje opštih relacija između vlastitih imena;
- liste elemenata tipa *predication* koji povezuju dva pivota sa određenim iskazom određenog jezika;
- liste elemenata tipa *type* pri čemu je svaki tip koren hijararhijski uređenog skupa tipova i
- elementa tipa *Wordnet* koji beleži veze sa *Wordnet*-om.

Svaka od ovih listi elemenata može biti prazna.

Element *pivot* ima jedinstveno određen identifikator i on predstavlja koncept koji mora postojati u bar jednom od definisanih jezika. Zbog toga u okviru svakog *pivot* elementa mora postojati element *concept* za koji je definisan jezik u okviru atributa *language* i vlastito ime u okviru atributa *prolexeme*. Svaki pivot, odnosno koncept, može se odnositi na samo jedno vlastito ime u okviru jednog jezika ili na više njih ali svako u okviru različitog jezika. Sama vlastita imena su opisana u drugom delu XML dokumenta.

Element *prediction* definiše vezu između dva pivota (koncepta) koja je definisana u okviru nekog jezika. Svaki element ovog tipa ima obavezno element *pReference* koji definiše jezik i iskaz ili vezu među vlastitim imenima određenog jezika. Ovaj element u okviru *prediction* odgovara elementu *concept* u okviru elementa *pivot*. Može se primetiti da atribut *language* u okviru elemenata *concept* i *pReference* ima veliku ulogu kod aplikacija koje služe za prevođenje iz jednog jezika u drugi. Zaista, na ovaj način pristup vlastitom imenu iz jednog u drugi jezik obavlja se automatski.

```
<root>
  <relationships>
    <pivot @num="400", @essence="historical", @type="city", @wordNet="05558236n">
      <canonical @pivot="410" @register="diachronic"/>
      <concept @language="english", @prolexeme="500" />
    </pivot>
    <pivot @num="600", @essence="historical", @type="country", @wordNet="05557178n">
      <concept @language="english", @prolexeme="800" />
    </pivot>
    <predication @pivot1="400", @pivot2="600"><pReference @language="english", @predicate="500"/>
    </predication>
    <type @name="Toponym"><type @name="Country"/><type @name="City"/></type>
    .....
    <wordNet ><lli @num="05558236n"/><lli @num="05557178n"/></wordNet>
  </relationships>
  <languages>
    <language @name="english">
      <prolexemes>
      <prolexeme @num="500", @name="Paris", @determination="no", inflection="89", @pivot="400">
      <derivatives>
      <derivative @name="Parisian", @category="3", @inflection="96" >
      <instances>
      <instance @name="Parisian", @morphology="S" />
      <instance @name="Parisians", @morphology="P" />
      </instances>
      </derivative>
      .....
      </derivatives>
      <instances><instance @name="Paris", @morphology="S" /></instances>
      </prolexeme>
      <prolexeme @num="800", @name="France", @determination="no", inflection="89", @pivot="600">
      <derivatives>
      <derivative @name="French", @category="3", @inflection="96" >
      <instances><instance @name="French", @morphology="S"/>... </instances>
      </derivative>
      .....
      </derivatives>
      <instances><instance @name="France", @morphology="S"/></instances>
      </prolexeme>
      </prolexemes>
      <predicates><predicate @num="500", @name="capital", @grammar="12"/>... </predicates>
      .....
    </language>
  </languages>
</root>
```

Slika 4.1: Primer rečnika vlastitih imena u XML-u za koncepte "Paris" i "France"

U primeru na slici 4.1 element *predication* pokazuje da je Pariz glavni grad Francuske jer u okviru elementa *pReference* atribut *predicate* ima vrednost 500 a to odgovara elementu *predicate* sa imenom *capital* koji je definisan u drugom delu XML dokumenta. Ovakvim predstavljanjem pivota, kada imamo jedan pivot možemo lako dobiti sve pivote sa kojima je ovaj u vezi kao i listu svih veza (jedna veza za jedan jezik).

Drugi deo XML dokumenta, sa korenom u *languages*, sadrži informacije o bar jednom jeziku. Svaki jezik ima ime i sadrži skup vlastitih imena u okviru elementa *prolexemes* i njihove opise. Takođe sadrži i skup veza ili odnosa između vlastitih imena za taj jezik, definisanih u okviru elementa *predicates*.

Rečnik vlastitih imena ima široku primenu u procesu obrade prirodnih jezika kao što su automatsko prevođenje, izdvajanje informacija, višejezično poravnanje teksta i tako dalje.

4.3 Wordnet

Wordnet koji je danas poznat kao *Prinstonski Wordnet* (PWN) razvili su Džordž Miler i njegov tim sa ciljem da se koristi kao jedna vrsta mentalnog leksikona u okviru psiholingvističkih projekata [15]. U okviru tradicionalnih rečnika, leksički pojmovi su alfabetski uređeni i za svaki od njih data je definicija za svako od mogućih značenja. Za razliku od toga, kod *Wordnet*-a, sve reči kojima se može izraziti neki pojam, grupisane su zajedno u skup sinonima (eng. synset) predstavljajući tako jedan koncept. PWN predstavlja skup približno 100.000 koncepata povezanih semantičkim relacijama u semantičku mrežu. Projekat *EuroWordnet* (EWN) [26] [12] je dao projektu *Wordnet* novu dimenziju uvodeći višejezičnost u semantičku mrežu. Vokabulari sedam evropskih jezika su prvo organizovani na sličan način kao PWN, a zatim međusobno povezani preko takozvanog međujezičkog indeksa (eng. Inter-Lingual-Index — ILI). *BalkaNet* je projekat koji je od septembra 2001. do avgusta 2004. finansirala Evropska komisija [11]. Cilj ovog projekta je razvoj poravnatih semantičkih mreža tipa *Wordnet* za balkanske jezike, i to bugarski, grčki, rumunski, srpski i turski, kao i proširenje mreže za češki koja je početno bila razvijana u okviru projekta EWN. Osnovni cilj *BalkanNet* projekta je razvoj savremenih jezičkih resursa za balkanske jezike koji bi omogućili nov način pristupa informacijama koje potiču iz balkanskih jezika. Osim toga, cilj ovog projekta bio je i proširenje semantičke mreže koja je uspostavljena u okviru projekta EWN balkanskim jezicima. Svrha ovakvog proširenja bi bila da se ojača saradnja balkanskih zemalja sa članicama Evropske unije. Kao glavne aktivnosti u odviru *BalkaNet* projekta treba istaći, pre svega, razvoj mreža *Wordnet* za balkanske jezike pojedinačno (bugarski, grčki, rumunski, srpski, turski i češki) i njihovo povezivanje sa postojećom leksičkom bazom EWN. Ove glavne aktivnosti su planirane i sprovedene sinhronizovano, što znači da su jednojezičke mreže izgrađene nad zajednički dogovorenim osnovnim skupovima od 8516 koncepata već prisutnim u PWN-u. To su takozvani "bazični koncepti" (pogledati poglavlje "Analiza *Wordnet*-a"). Izvan ovih osnovnih skupova "bazičnih koncepata", za svaki pojedinačni jezik mreža se razvijala nezavisno, ali u okvirima koje je postavio PWN. Ovakav prisput razvoju mreže je postavio specifične probleme. Naime, tokom rada na razvoju mreže često su se postavljala sledeća pitanja: da li su koncepti jezički zavisni ili ne, da li su obrasci za leksikalizaciju koncepata univerzalni,

da li je struktura prinstonske mreže valjana i za druge jezike i da li je skup semantičkih relacija koje su u njega ugrađene dovoljan za sve jezike [14]. Premda je rad na razvoju zasebnih mreža za balkanske jezike često davao potvrde za negativan odgovor na ova pitanja, nije se odustalo od predhodno utvrđenog postupka. U odsustvu srpskog rečnika i dvojezičnog englesko/srpskog rečnika u elektronskoj formi, prevod koncepata iz PWN u *Srpski Wordnet* (SWN) je rađen ručno. Iz tog razloga, postavilo se pitanje validnosti SWN-a. Korišćenje jednojezičnih i višejezičnih korpusa u cilju provere validnosti sinsetova u SWN-u dovelo je do dodavanja novih i uklanjanja postojećih literala iz nekih sinsetova [17][18].

Kako se mreže tipa *Wordnet* danas razvijaju pre svega za informatičke potrebe, tako se i osnovna primena ovih mreža za balkanske jezike vidi u njihovoj ugradnji u informatičke primene koje su zasnovane na prirodno-jezičkoj obradi. Mogu se koristiti za klasifikaciju dokumenata ili višejezičko pretraživanje, uključivanje u pretraživačke mašine (obeležja domena) - poboljšanje usluga mašina za pretraživanje za većinu balkanskih jezika, konceptualno indeksiranje veb stranica, Alexandria (Memodata, Lingvistička podrška i usluge za veb korisnike) [42]. Postojanje višejezične baze sa međusobno poravnatim konceptima u ovim slučajevima je od suštinskog značaja.

Ipak, da bi se prevazišli uočeni problemi, svi partneri na projektu su se dogovorili da se kao jedan od rezultata rada na ovom projektu ugradi i skup koncepata koji su specifični za balkanske jezike [23].

4.3.1 Srpski Wordnet

Srpski Wordnet (SWN) je leksičko-semantička mreža srpskog jezika [19]. Struktura SWN je u osnovi ista kao struktura PWN. Baziran je na konceptima poznatim kao skupovi sinonima ili sinsetovi. Sinset je skup reči sa istim značenjem. U načelu, jedina gramatička informacija koja se dodeljuje sinsetu je vrsta reči: PoS (Part of Speech) koja mora biti ista za sve reči u jednom sinsetu. Samo reči iste vrste (na primer imenice, glagoli, pridevi) mogu pripadati istom sinsetu. Svaki sinset ima jedinstveni identifikacioni broj (ID). Svaka reč u sinsetu predstavljena je niskom karaktera ili literalom, za kojom sledi značenje tog konkretnog literala u konkretnom sinsetu. Ovo rešenje se zasniva na pristupu koji se koristi u klasičnim rečnicima govornog jezika, gde jednoj reči odgovara više mogućih značenja, koja se na poseban način obeležavaju. U *Wordnet*-u, kako neka reč može imati više značenja, to može biti član više različitih sinsetova.

SWN sadrži imenice, glagole, prideve i priloge. Trenutno sadrži približno 14.000 skupova sinonima [20]. Posle relacije koja povezuje sinonime u sinset, najznačajnija relacija među konceptima je nadređeni/podređeni (eng. hypernym/hyponym) relacija. Kako najčešće postoji samo jedan hipernim, ova semantička relacija vrši organizaciju imenica u hijararhijsku strukturu. Postoji 9 najopštijih (korenih) sinsetova u SWN-u koji odgovaraju relativno različitim semantičkim poljima: "entitet", "apstrakcija", "grupa", "akt", "psihičko svojstvo", "stanje", "događaj", "fenomen" i "svojina".

Svi nacionalni *Wordnet*-ovi predstavljeni su XML dokumentom iste strukture. Primer sinseta koji odgovara konceptu "božanstvo" u SWN i PWN dat je na slici 4.2.

Značenja nekih od elemenata (XML etiketa) na ovoj slici su sledeća: ID je identifikator sinseta (koncepta) i on je jedinstven u svim jezicima, POS je vrsta

```
<SYNSET>
 <ID> ENG20-08904620-n </ID>
 <SYNONYM>
   <LITERAL>bozxanstvo<SENSE>1</SENSE></LITERAL>
 </SYNONYM>
 <DEF>
 Natprirodno bicxe koje se obozxava zbog verovanxa da upravlxa
 nekim delom sveta ili nekim aspektima zxivota ili zato sxto
 personifikuje silu.
 </DEF>
 <POS>n</POS>
 <ILR>ENG20-08903509-n<TYPE>hypernym</TYPE></ILR>
 <ILR>ENG20-07660421-
n<TYPE>holo_member</TYPE></ILR>
</SYNSET>

<SYNSET>
 <ID> ENG20-08904620-n </ID>
 <SYNONYM>
   <LITERAL>deity <SENSE>1</SENSE></LITERAL>
   <LITERAL>divinity <SENSE>1</SENSE></LITERAL>
   <LITERAL>god <SENSE>2</SENSE></LITERAL>
   <LITERAL>immortal <SENSE>2</SENSE></LITERAL>
 </SYNONYM>
 <DEF>
 any supernatural being worshipped as controlling some part of the
 world or some aspect of life or who is the personification of a
force.
 </DEF>
 <POS>n</POS>
 <ILR>ENG20-08903509-n<TYPE>hypernym</TYPE></ILR>
 <ILR>ENG20-07660421-
n<TYPE>holo_member</TYPE></ILR>
 <ILR>ENG20-00670299-v
<TYPE>eng_derivative</TYPE></ILR>
</SYNSET>
```

Slika 4.2: Sinset u SWN-u i njemu odgovarajući sinset u PWN-u

reči (eng. Part Of Speach), ILR je ID koncepta koji je u relaciji sa predstavljenim konceptom. Tip te relacije dat je u okviru TYPE etikete.

Srpski Wordnet je u fazi izgradnje. Po završetku BalkaNet projekta, njegov dalji razvoj organizovan je kroz kooperativni rad [20]. Prema stanju SWN-a iz decembra 2007. godine, u tabeli 4.1 predstavljen je broj sinsetova u SWN-u za odgovarajuću vrstu reči (PoS), raspodela sinsetova u SWN-u prema vrsti reči kao i broj sinsetova i njihova raspodela prema vrsti reči u PWN-u. U poslednjoj koloni predstavljen je odnos broja sinsetova u PWN-u i SWN-u.

PoS	SWN	procenti(SWN)	PWN	procenti(PWN)	PWN/SWN
imenice	11155	80.1%	79689	69%	7.1
glagoli	1945	14%	13508	12%	6.9
pridevi	793	5.7%	18563	16%	23.4
prilozi	27	0.2%	3664	3%	135.7
ukupno	13920	100%	115424	100%	8.3

Tabela 4.1: Raspodela sinsetova (koncepata) u SWN-u i PWN-u prema vrsti reči (PoS)

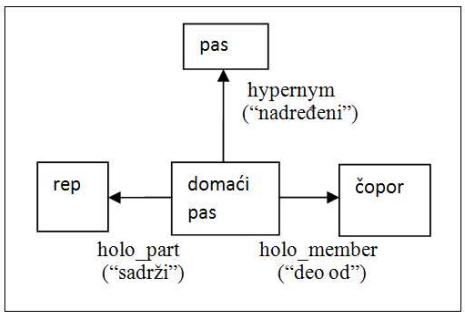
Koncepti u *Wordnet*-u su međusobno povezani semantičkim i leksičkim relacijama čineći tako semantičku mrežu. Neke od relacija su: sinonim, antonim, hiponim, hipernim i tri tipa relacija meronim i holonim. Ovo su relacije koje se koriste pri organizaciji mentalnog leksikona. Mogu biti predstavljene pokazivačima ili označenim strelicama od jednog do drugog sinseta. Ove relacije između sinsetova u SWN-u i PWN-u sumirane su u tabeli 4.2. Relacije iz PWN-a automatski su prevedene u SWN a zatim je ručno proveravana njihova ispravnost.

relacija	SWN	PWN	PWN/SWN
hypernym	13034	94844	7.28
near_antonym	680	7642	11.2
holo_part	701	8636	12.3
verb_group	163	1748	10.72
holo_member	3309	12205	3.69
be_in_state	268	1296	4.84
subevent	76	409	5.38
causes	58	439	7.57
derived	326	6809	20.89
particle	10	401	40.1

Tabela 4.2: Raspodela relacija među sinsetovima u SWN-u i PWN-u

Wordnet uključuje sledeće semantičke relacije:

- *Synonymy* je osnovna relacija u *Wordnet*-u zato što *Wordnet* koristi skup sinonima za predstavljanje značenja reči. Ova relacija služi za spajanje reči sa istim ili sličnim značenjem na primer "beznačajan" i "nebitan". Ovo je simetrična relacija.



Slika 4.3: Deo *Wordnet*-a (idealizovani model)

- *Hyponymy* ("je podređen") i njegov inverz *hypernymy* ("je nadređen") predstavljaju tranzitivnu i antisimetričnu relaciju među sinsetovima. Može se koristiti za uopštavanje značenja imenica i glagola i njihovo predstavljanje na višem nivou apstrakcije. Hiponim nasleđuje sve osobine opštijeg koncepta i poseduje još bar jednu dodatnu osobinu koja ga razlikuje od drugih koncepata. Slika 4.3 prikazuje idealizovani model dela *Wordnet*-a a slika 4.4 njegovu XML reprezentaciju.

- *Antonymy* ("je suprotan") je simetrična relacija, posebno značajna u organizovanju prideva. Antonim za reč x je ponekad ne-x ali ne uvek. Na primer, "bogat" i "siromašan" su antonimi ali reći da neko nije bogat ne znači da je siromašan.

- *Meronymy* ("je deo") i njegov inverz *holonymy* ("je celina") su kompleksne semantičke relacije. Na primer, "rep" je meronim od "pas" a "pas" je meronim od "čopor" jer je rep deo psa a pas je deo čopora (pogledati sliku 4.3).

- *Troponymy* je za glagole isto što je i *hyponymy* za imenice. Na primer, "hodanje" je troponim od "pokret".

- *Entailment* ("dovodi do") je relacija među glagolima. x dovodi do y ako tačno za y sledi iz tačnog za x. Na primer, "ubiti" dovodi do "umreti".

```
<SYNSET>
    <ID>ENG20-02000516-n</ID>
    <SYNONYM>
        <LITERAL>pas<SENSE>C1</SENSE></LITERAL>
    </SYNONYM>
    <DEF>Bilo koji od raznovrsnih sisara koji obicyno
        imaju dugu nxusxku i kandye.
    </DEF>
    <POS>n</POS>
    <ILR>ENG20-01992546-n
        <TYPE>hypernym</TYPE>
    </ILR>
</SYNSET>

<SYNSET>
    <ID>ENG20-02001223-n</ID>
    <SYNONYM>
        <LITERAL>domacxi pas</LITERAL>
        <LITERAL>pas<SENSE>C1x</SENSE></LITERAL>
        <LITERAL>pseto<SENSE>1</SENSE></LITERAL>
    </SYNONYM>
    <DEF>Pripadnik Canis familiaris, srodan vuku, pripitomlxen
     od preistorijskog doba; postoje mnoge rase.</DEF>
    <POS>n</POS>
    <ILR>ENG20-02000516-n
        <TYPE>hypernym</TYPE></ILR>
    <ILR>ENG20-07511852-n
        <TYPE>holo_member</TYPE></ILR>
</SYNSET>
```

```
<SYNSET>
    <ID>ENG20-07511852-n</ID>
    <SYNONYM>
        <LITERAL>cyopor<SENSE>1</SENSE></LITERAL>
    </SYNONYM>
    <ILR>ENG20-07510906-n
        <TYPE>hypernym</TYPE></ILR>
    <DEF>Grupa zxivotinxa koje love.</DEF>
    <POS>n</POS>
</SYNSET>

<SYNSET>
    <ID>ENG20-02074201-n</ID>
    <POS>n</POS>
    <SYNONYM>
        <LITERAL>rep<SENSE>2a</SENSE></LITERAL>
    </SYNONYM>
    <ILR>ENG20-02073042-n
        <TYPE>hypernym</TYPE></ILR>
    <ILR>ENG20-02001223-n
        <TYPE>holo_part</TYPE></ILR>
    <ILR>ENG20-02342528-n
        <TYPE>holo_part</TYPE></ILR>
    <DEF>Upadlxivo oznacyen ili oblikovan zadnxi deo.</DEF>
</SYNSET>
```

Slika 4.4: Deo *Wordnet*-a (XML reprezentacija)

5

Upravljanje leksičkim resursima

Leksički resursi pomenuti u prethodnom poglavlju predstavljaju rečnike različitog tipa, razvijene na osnovu sasvim različitih modela, pa samim tim sadrže i različite vrste leksičkih informacija. Brojne su mogućnosti njihove primene kao što su definisanje i povezivanje leksičkih podataka na način koji će omogućiti njihovo efikasnije pretraživanje, integrisanje i ponovno korišćenje u aplikacijama usmerenim ka vebu, kao i primene vezane za obradu prirodnojezičkih dokumenata. Postoji i mogućnost primene integrisanih heterogenih resursa za proširenje upita, kao i pretraživanje tekstova uopšte [30].

Integracijom ovih resursa, informacije koje sadrži jedan resurs mogu se ugraditi u onaj drugi ili se mogu koristiti za njegov razvoj [30]. Tako na primer, prenošenjem informacija iz morfološkog rečnika u sinsetove SWN-a, poboljšava se efikasnost korišćenja SWN-a u pretraživanjima s obzirom na to da je vrsta reči (PoS-Part of Speach) jedina gramatička informacija koju poseduje sinset u *Wordnet*-u. U velikom broju slučajeva ove dodatne informacije mogu da se koriste za uklanjanje dvoznačnosti, odnosno rešavanje problema homonimije. Morfološke, sintaktičke i semantičke informacije mogu se preuzeti iz srpskog morfološkog rečnika prostih reči i pridružiti rečima odgovarajućeg sinseta u SWN. Na primer [30], u sinsetovima:

```
obaviti:A1x, uraditi:4
okruzxiti:4, obaviti:B1v
```

pojavljuje se glagol `obaviti`. Međutim, u pitanju su dve različite flektivne klase (prvo lice jednine prezenta za glagol u prvom sinsetu je `obavim`, a u drugom sinsetu `obavijem`). Informacija o flektivnoj klasi ne postoji u *Wordnet*-u ali postoji u rečniku DELAS u obliku odgovarajućeg koda. Ta informacija se može preuzeti iz rečnika DELAS i pridružiti odgovarajućoj reči u SWN, u obliku XML elementa uključenog u element `<LITERAL>` kojim se definiše reč u sinsetu. U ovom primeru, u prvom slučaju pridružena informacija bila bi `V157+Perf+Tr+Iref`, dok bi istoj reči u drugom sinsetu bila pridužena informacija `V135+Perf+Tr+Iref`. Iz ovih dodatnih morfosintaktičkih informacija vidi se da se u oba slučaja radi o svršenim, prelaznim i nepovratnim glagolima, ali različite flektivne klase.

Takođe, u cilju prevazilaženja ograničenja da se samo iste vrste reči (sa istim PoS) mogu nalaziti u jednom sinsetu, u PWN-u i ostalim *Wordnet*-ovima mogu se dodati takozvane XPoS veze kojima se povezuju sinsetovi sačinjeni od različitih vrsta reči, preko relacija kao što su već pomenute relacije CAUSE i BE_IN_STATE kao i relacija DERIVED (izveden) i slično. Derivacione informacije koje postoje u srpskom morfološkom rečniku prostih reči mogu se iskoristiti za dodavanje XPoS veza u SWN, ali i za formiranje novih sinsetova od izvedenih reči.

Informacije iz SWN mogu se uspešno koristiti za obogaćivnje srpskog morfološkog rečnika, odnosno hijerarhijska struktura SWN može se upotrebiti za dodavanje semantike lemama u morfološkom rečniku. Neke osnovne semantičke informacije su već pridružene lemama, ali korišćenje *Wordnet*-a omogućava sistematičnije i detaljnije pridruživanje oznaka semantičkih informacija.

Veličina srpskog morfološkog rečnika i SWN nije uporediva. Razvoj srpskog morfološkog rečnika je započet više godina pre *Wordnet*-a, tako da on potpunije pokriva jezik. Zbog toga, *Wordnet* može ovim povezivanjem više da dobije od srpskog morfološkog rečnika nego obrnuto.

Pretraživanje informacija, kao jedan od osnovnih procesa obrade prirodnog jezika, predstavlja proces pronalaženja informacija relevantnih korisnikovoj potrebi za informacijom."Fundamentalni paradoks pretrage informacija" po Roland Hjerrpe-u, ogleda se u "potrebi za opisom nečega što nam nije poznato s namerom njegovog efikasnog pronalaženja". U ovakvim procesima obrade teksta napisanog na srpskom jeziku kao i u okviru ostalih procesa obrade, kao na primer klasifikacije tekstova i drugo, treba uzeti u obzir sledeće karakteristike srpskog jezika [25]:

- *Koriste se dve azbuke.* Tekst na srpskom jeziku može biti napisan korišćenjem službene ćirilice ili latinice koja je široko rasprostranjena;
- *Fonološki bazirana ortografija.* To znači da se reč piše onako kako se izgovara odnosno svakoj segmentalnoj fonemi odgovara posebno slovo - monografema. Tako se mogu razlikovati reči koje imaju ekavsko i ijekavsko narečje. Na primer, imenica u jednini može biti dete (ekavski) ili dijete (ijekavski);
- *Bogat morfološki sistem*;
- *Slobodan redosled reči* kao što su subjekat, predikat, objekat i drugo.

Sve ove karakteristike imaju direktan uticaj na proces obrade tekstova na srpskom jeziku, i problem višeznačnosti čine jako teškim. Leksički resursi za srpski jezik tu imaju presudnu ulogu.

Najjednostavniji upiti za pretraživanje tekstualnih sadržaja sastoje se od jedne ili više reči, koje su eventualno povezane logičkim operatorima i/ili. Kada je u pitanju sadržaj na Internetu, ovakvo postavljanje upita je najčešće i jedino raspoloživo. Ranije je veb pretraživač AltaVista imao grafički editor za definisanje upita, međutim takav način pretraživanja je napušten, a trenutno WebCorp daje mogućnost zadavanja upita već gotovim grafovima, ali rezultati nisu uvek oni koji se očekuju. To se posebno odnosi na srpski jezik, gde morfološko proširenje nije moguće bez odgovarajućih rečnika. Kada je u pitanju pretraživanje korpusa, sem najjednostavnijih upita, po pravilu je moguće formulisanje i složenijih upita regularnim izrazima. Međutim, i kada je u pitanju

tekstualni sadržaj na Internetu, i kada se pretražuju korpusi, postoje znatno veće mogućnosti za proširenjem upita.

Naime, kombinovanje resursa omogućava pretraživanje tekstova po sledećim kriterijumima [30]:

- jednostavna niska karaktera (eng. string matching);
- lema sa svim flektivnim oblicima, tj. morfološko proširenje zadate leme;
- oblik reči iz koga treba naći lemu;
- koncept, gde na osnovu zadate leme svi ili samo odabrani literali izdvojenih sinsetova (sa hipernimima ili bez njih) predstavljaju semantičko proširenje;
- pretraživanje po konceptima uz morfološko proširenje;
- pretraživanje po konceptima, prošireno na drugi jezik i slično.

Kako je većina leksičkih resursa predstavljena u XML formatu, upravljanje ovim resursima može biti znatno efikasnije korišćenjem izvornih XML baza podataka.

5.1 Biblioteka funkcija "biogfun"

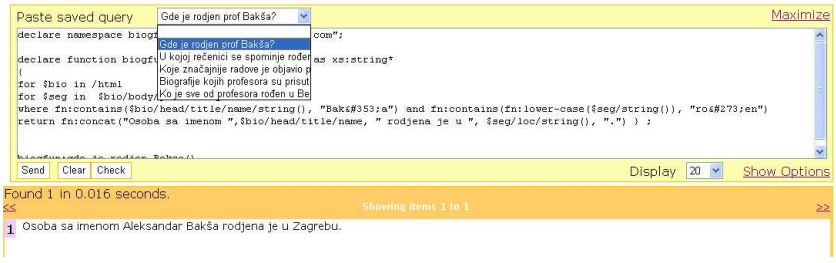
U okviru sprovedenog eksperimenta razmatrana je kolekcija od oko 80 XML obeleženih tekstova, biografija matematičara iz Spomenice Matematičkog fakulteta. Ovi tekstovi su izvorno bili u HTML datotekama ali su uz pomoć alata "Stylus Studio" konvertovani u XML dokumente. Nakon toga, tekstovi su ručno obeleženi na sledeći način: svaka rečenica u tekstu obeležena je etiketom `<seg>`, svaki datum etiketom `<dat>`, svako ime osobe etiketom `<name>`, mesto etiketom `<loc>`. Primer jednog takvog XML dokumenta za biografiju prof. Parezanovića, prvog profesora računarstva na Matematičkom fakultetu u Beogradu, dat je na slici 5.1.

U cilju izvođenja eksperimenta, razvijen je rečnik vlastitih imena osoba, po ugledu na spomenuti rečnik vlastitih imena [27]. U eksperimentu je korišćen i SWN kao još jedna vrsta leksičkog resursa. Pre smeštanja u bazu, SWN koji je izvorno "ravni" XML dokument sa sinsetovima, transformisan je u odgovarajuci "hijerarhijski" dokument, u kome su sinset elementi ugnježdeni u one sinset elemente koji su njihovi hipernimi. Ova transformacija obavljena je pomoću Java programa čiji je autor Ognjen Marić (PMF Banja Luka). Na ovaj način, omogućeno je efikasnije korišćenje XQuery funkcija u okviru kojih se na vrlo jednostavan način pristupa svim potomcima nekog elementa što u našem slučaju predstavlja sve hiponime nekog koncepta. Zbog toga je "ravni" XML dokument potrebno prvo transformisati u "hijararhijski". Ideja je da se ovako različiti leksički resursi smeste u izvornu XML bazu podataka i da se na smislen način povežu upitima. Prilikom izrade ovih eksperimenata korišćena je eXist baza. Izgled interfejsa može se videti na slici 5.2.

Za početak, razvijene su sledeće XQuery funkcije u okviru biblioteke sa imenom "biogfun" nad kolekcijom tekstova biografija matematičara iz Spomenice Matematičkog fakulteta:

```
<html>
  <head>
    <title>Dr <name>Nedeljko Parezanović</name>, redovni profesor u penziji</title>
  </head>
  <body>
    <p><name>Dr Nedeljko Parezanović</name> redovni profesor u penziji</p>
    <p><seg> Rođen je u <loc>Ivanjici</loc> <dat>25. avgusta 1932. Godine</dat>.</seg>
      <seg>Osnovnu školu je završio u <loc>Ivanjici</loc>, a srednju školu u <loc>Beogradu</loc>.</seg>
      <seg>Diplomirao je na Prirodno-matematičkom fakultetu u <loc>Beogradu</loc>
        <dat>1957. Godine</dat> na grupi za mehaniku.</seg>
      <seg>Na istom fakultetu je odbranio i doktorsku tezu <dat>1961. godine</dat>.</seg>
      <seg>Od <dat>1950.</dat> do <dat>1957. godine</dat> je radio u studiju Radio Beograda, a po diplomiranju na
        fakultetu prelazi u Vojno-tehnički institut u <loc>Beogradu</loc>, gde je radio na poslovima konstrukcije
        nomograma i primene analognih računara.</seg>
      <seg><dat>Godine 1959.</dat> prelazi u Institut za nuklearne nauke "Boris Kidrič" u <loc>Vinči</loc> u
        laboratoriju za Primenjenu matematiku.</seg>
      <seg>Tada započinje i svoj istraživački rad na poslovima postavke i analize matematičkih modela.</seg>
      <seg>U to vreme koristi mašinu za rešavanje linearnih algebarskih jednačina, repetitivni diferencijalni analizator i
        prvi veliki digitalni računar IBM-705 u Saveznom zavodu za statistiku.</seg>
      <seg>Prelaskom u Institut "M. Pupin" u <loc>Beogradu</loc> (<dat>1961. godine</dat>) počinje da radi na
        projektovanju digitalnih elektronskih računara i razvoju sistemskog i aplikativnog softvera.</seg>
      <seg>Radio je na razvoju više domaćih računara i računara "Kosmos" u saradnji sa Institutom za automatiku i
        telemehaniku iz <loc>Moskve</loc>.</seg>
      <seg>Tom prilikom je više puta boravio u <loc>Moskvi</loc>.</seg>
      <seg>U oblasti projektovanja elektronskih digitalnih računara posebnu pažnju zaslužuju <i>Idejni projekat</i>
        CER-11 i <i>Stoni elektronski kalkulator</i> CER-30 rađeni u Institutu "M. Pupin".</seg>
      <seg>CER-11 je više godina uspešno korišćen u JNA, a CER-30 je zasnovan na originalnoj ideji kalkulatora sa
        usvajanjem programa (<dat>1963. Godina</dat>).</seg>
      <seg>Za potrebe Tvornice računskih strojeva u <loc>Zagrebu</loc> radio je projekat po kome je proizvedena
        veća serija kalkulatora TRS 501 i TRS 511.</seg>
      <seg>Rad u nastavi je počeo <dat>1963. Godine</dat>, a univerzitetsku karijeru započinje
        <dat>1967. godine</dat> u zvanju vanrednog profesora na Elektronskom fakultetu u <loc>Nišu</loc>.</seg>
      <seg> Od tada je držao više kurseva na redovnim i poslediplomskim studijama u više univerzitetskih centara
        (<loc>Beogradu</loc>, <loc>Zagrebu</loc>, <loc>Novom Sadu</loc> i <loc>Nišu</loc>).</seg>
      <seg>To su najčešće bili prvi kursevi iz oblasti računarstva, kao što su: <i>Osnovi računarskih sistema</i>,
        <i>Programski sistemi</i>, <i>Programski jezici</i>, <i>Računske mašine i programiranje</i>,
        <i>Primena računara</i> i <i>Uvod u kibernetiku</i>.</seg>
      <seg>Od <dat>1969. Godine</dat> je rukovodilac Računskog centra Matematičkog instituta u
        <loc>Beogradu</loc> i drži nastavu na Matematičkom fakultetu u <loc>Beogradu</loc>.</seg>
    <p>Radovi</p>
    <p><a target="_top" href="r1s1m.htm">
        <i>REŠAVANJE FREDHOLMOVE INTEGRALNE JEDNAČINE
        PRIMENOM DIFERENCIJALNIH JEDNAČINA</i></a> </p>
     <p><a target="_top" href="r2s1m.htm">
        <i>ONE VIEW OF THE SOFTWARE QUALITY</i></a> </p>
  </body>
</html>
```

Slika 5.1: Primer XML dokumenta za biografiju prof Parezanovića



Slika 5.2: Interfejs eXist baze

```
declare namespace biogfun = "http://www.biogfun.com";

declare function biogfun:gde-je-rodjen-Parezanovic()  as xs:string*
{
  (:Za biografiju profesora Parezanovica...:)
  for $bio in /html[fn:contains($bio/head/title/name/string(), "Parezanovi&#263;")]

  (:...izdvoj sve recenice...:)
  for $seg in  $bio/body/p/seg

  (:...u kojima se javlja rec 'rodjen':)
  where fn:contains(fn:lower-case($seg/string()), "ro&#273;en")

  return fn:concat("Osoba sa imenom ",$bio/head/title/name, " rodjena je u ", $seg/loc/string(), ".")
} ;

biogfun:gde-je-rodjen-Parezanovic()
```

Slika 5.3: XQuery funkcija "Gde je rođen prof. Parezanović?"

- Gde je rođen prof. Parezanović? (slika 5.3)
  Rezultat:
  Osoba sa imenom Nedeljko Parezanović rodjena je u Ivanjici.

- U kojoj rečenici se spominje rođenje prof. Parezanovića?
  Rezultat:
  Rođen je u Ivanjici 25. avgusta 1932. godine.

- Koje značajnije radove je objavio prof. Parezanović?
  Rezultat:
  REŠAVANJE FREDHOLMOVE INTEGRALNE JEDNAČINE PRIMENOM DIFERENCIJALNIH JEDNAČINA
  ONE VIEW OF THE SOFTWARE QUALITY

- Biografije kojih profesora su prisutne u kolekciji?
  Rezultat:
  Dušan Adamović
  Branka Alimpić
  Milan Andonović
  ...

- Ko je sve od profesora rođen u Beogradu?
  Rezultat:
  U Beogradu su rođeni:
  Mijalko Ciric
  Ilija Lukacevic
  Jelena Markovic-Nikolic
  ...

Uključivanjem u razmatranje rečnika vlastitih imena osoba, biblioteka funkcija "biogfun" obogaćena je funkcijama tipa:

"Gde su rođene osobe čija se imena pojavljaju u rečniku vlasitih imena?" (slika 5.4)

```
declare namespace biogfun = "http://www.biogfun.com";

declare function biogfun:gde-su-rodjene-osobe-iz-recnika() as xs:string*
{
    (:Za sva imena iz recnika vlastitih imena…:)
    for $ime in //root/languages/language/prolexemes/prolexeme/@name/string()

    (:… za sve biografije matematicara cija se imena nalaze u recniku…:)
    for $bio in //html [fn:contains(head/title/name/string(), $ime)]
    (:…i sve recenice iz svake od tih biografija…:)
    for $seg in  $bio/body/p/seg

    (:…izdvoj one recenice koje sadrze rec 'rodjen'…:)
    where fn:contains(fn:lower-case($seg/string()), "ro&#273;en")

     (:…i iz takvih recenica izdvoj ime grada gde je osoba rodjena.:)
     return
        fn:concat("Osoba sa imenom ",$bio/head/title/name, " ro&#273;ena je u ", $seg/loc/string(), ".")
} ;


biogfun:gde-su-rodjene-osobe-iz-recnika()
```

Slika 5.4: XQuery funkcija "Gde su rođene osobe čija se imena pojavljaju u rečniku vlasitih imena?"

Na kraju je u razmatranje uključen i SWN. XQuery funkcije koje su razvijene u okviru ove biblioteke a koje povezuju kolekciju XML obeleženih tekstova biografija sa rečnikom vlastitih imena i SWN-om su tipa:

"Koji matematičari, čija su imena data u rečniku vlastitih imena, su rođeni u nekoj nacionalnoj prestonici?" (slika 5.5)

"Nacionalna prestonica" se u ovom primeru javlja kao koncept u SWN-u. Posmatraju se svi hiponimi ovog koncepta, posmatraju se sva imena iz rečnika vlastitih imena, pretražuju se sve biografije, i dobijaju se oni matematičari koji su rođeni u nekoj nacionalnoj prestonici i čije ime se javlja u rečniku vlastitih imena.

Namera je da se u daljem radu uključi u razmatranje i morfološki rečnik srpskog jezika.

Može se primetiti da se korišćenjem izvornih XML baza podataka na vrlo jednostavan način mogu izdvajati različite vrste informacija i leksički resursi se mogu na smislen način povezivati u cilju dobijanja što kvalitetnijih informacija.

Izvorne XML baze podataka mogu se primeniti i nad samo jednim od ovih leksičkih resursa kao na primer nad *Wordnet*-om čime se mogu otkriti neke nove karakteristike ovog resursa koje mogu imati važnu ulogu u procesima obrade prirodnih jezika.

```
declare namespace biogfun = "http://www.biogfun.com";

declare function biogfun:osobe-iz-recnika-rodjene-u-nacionalnoj-prestonici() as xs:string*
{
   (:Za sva imena iz recnika vlastitih imena…:)
   for $ime in //root/languages/language/prolexemes/prolexeme/@name/string()

    (:… za sve biografije matematicara cija se imena nalaze u recniku…:)
    for $bio in //html [fn:contains(head/title/name/string(), $ime)]
    (:…i sve recenice iz svake od tih biografija…:)
    for $seg in  $bio/body/p/seg

   (:… za sve gradove koji su hiponimi od 'nacionalna prestonica'… :)
   for $np in //ROOTSRPH//SYNSET[SYNONYM/LITERAL/text() = 'nacionalna prestonica']
   for $grad in $np /descendant-or-self::*[name()='SYNSET']
   for $gradliteral in $grad/SYNONYM/LITERAL

   (:…izdvoj one recenice koje sadrze rec 'rodjen' i ime neke nacionalne prestonice…:)
   where fn:contains(fn:lower-case($seg/string()), "ro&#273;en") and
         fn:contains(fn:lower-case($seg/string()), fn:lower-case($gradliteral/text()))
   return
       fn:concat("Osoba sa imenom ",$bio/head/title/name, " ro&#273;ena je u ", $seg/loc/string(), ".")
} ;

biogfun:osobe-iz-recnika-rodjene-u-nacionalnoj-prestonici()
```

Slika 5.5: XQuery funkcija "Koji matematičari, čija su imena data u rečniku vlastitih imena, su rođeni u nekoj nacionalnoj prestonici?"

6

Wordnet

Wordnet je specijalno struktuirani rečnik koji umesto reči uključuje leksičke koncepte tj. značenja predstavljena skupovima sinonima ili sinsetovima (eng. synsets). Nad leksičkim konceptima uspostavljaju se leksičke i semantičke relacije (homonimija, celina/deo, nadređenost/podređenost, i tako dalje), koje odražavaju psiholingvističke teorije leksičke memorije [31]. Kada psiholozi razmišljaju o organizaciji leksičke memorije, gotovo uvek imaju u vidu organizaciju imenica koje čovek ima na umu. Imenice se mogu organizovati hijerarhijski u nivoe, počev od opštih ka specifičnim. U *Wordnet*-u, ta organizacija je sprovedena putem semantičke relacije nadređenost/podređenost. Ima ukupno 9 koncepata koji su najopštiji, koji se nalaze na vrhovima hijerarhija (koreni koncepti) i pripadaju relativno različitim semantičkim poljima. U SWN-u to su: "entitet", "apstrakcija", "grupa", "akt", "psihičko svojstvo", "stanje", "događaj", "fenomen" i "svojina". Njihovo značenje je toliko opšte da gotovo nema nikakvog smisla. Takođe, ako se suviše spustimo niz hijerarhiju, dobijaju se koncepti koji su toliko specifični da se retko koriste u svakodnevnom govoru. Negde na sredini ovih hijerarhija nalaze se koncepti koji nisu ni suviše opšti, ni suviše specifični i takvi koncepti se zovu "bazični koncepti". Za ove koncepte može se nabrojati puno različitih karakteristika. Iznad ovog nivoa, opisi koncepata su kratki i suviše opšti, a ispod ovog nivoa, malo novih karakteristika je dodato onim karakteristikama koje međusobno razlikuju "bazične koncepte".

U cilju formalizacije pojma "bazični koncept", definisana je mera "produktivnosti" koncepta koja određuje koliko neki koncept efektivno predstavlja hijerarhiju kojoj pripada. U tom smislu, "bazični koncepti" se mogu definisati kao svi koncepti čija vrednost pridružene im mere produktivnosti, pripada nekom opsegu oko maksimalne vrednosti. Takvi koncepti se mogu smatrati kao najreprezentativniji za hijerarhiju kojoj pripadaju.

6.1 Mere produktivnosti koncepta

Mere produktivnosti koncepta mogu biti definisane na više različitih načina. Na primer, za izabrani koncept, ova mera može biti definisana kao proizvod broja svih skupova sinonima naslednika (ili samo skupova sinonima listova) tog koncepta i rastojanja tog koncepta od korenog koncepta (broj koncepata koji se nalaze na putu od korena do tog koncepta). Formalno, za izabrani koncept k

koji pripada hijerarhiji sa korenom u *koren*, mera se može predstaviti kao:

$$Mera1 : ProduktivnostKoncepta(k) = BrojNaslednika(k)^{\alpha} * Rastojanje(k)^{\beta}$$

$$Mera2 : ProduktivnostKoncepta(k) = BrojListova(k)^{\alpha} * Rastojanje(k)^{\beta}$$

pri čemu je *BrojNaslednika(k)* broj čvorova u drvetu sa korenom u konceptu k, *BrojListova(k)* je broj listova pomenutog drveta a *Rastojanje(k)* je rastojanje od koncepta k do korenog koncepta *koren*. α i β su parametri koji određuju sa kojom težinom broj naslednika, broj listova i rastojanje učestvuju u celokupnoj meri.

S druge strane, mera produktivnosti koncepta može se definisati kao odnos broja naslednika koncepta k i broja naslednika braće koncepta k. Koncept koji je brat koncepta k odnosno koji je na istom nivou sa k i ima istog roditelja kao k, biće označen sa *brat(k)*. Formalno, to se može zapisati kao:

$$Mera3 : ProduktivnostKoncepta(k) = \frac{BrojNaslednika(k)}{\sum_{brat(k)} BrojNaslednika(brat(k))}$$

Moguće je definisati meru produktivnosti i na neki drugi način. Izbor mere zavisi od svrhe u koju se ova mera koristi.

Na primer, najproduktivniji koncept za hijarhiju sa korenom u konceptu "entitet" (koja je najveća po broju koncepata u okviru SWN-a i PWN-a), ako se koristi *Mera1* sa vrednostima za $\alpha = 1$ i $\beta = 2$, je koncept "organizam" ili "živo biće". Drvo sa korenom u ovom konceptu sadrži `55%` svih koncepta u celom drvetu (sa korenom u "entitet") u SWN-u i `44%` u PWN-u i nalazi se na trećem nivou odnosno za tri koncepta je udaljen od korena. Za hijerarhiju sa korenom "svojina" (koja je najmanja po broju koncepata), ako se posmatra ista mera, najproduktivniji koncept je "potrošnja". Drvo sa korenom u tom konceptu sadrži `10%` svih koncepata u SWN-u i `30%` u PWN-u i nalazi se na šestom nivou u hijerarhiji. U tabeli 6.1 predstavljen je najproduktivniji koncept za drvo sa korenom "svojina" ako se koriste sve tri prethodno definisane mere.

"svojina" "possession"	najproduktivniji koncept (SWN)	procenti, nivo (SWN)	najproduktivniji koncept (PWN)	procenti, nivo (PWN)
Mera1	"potrošnja" "cost"	10%, 6	"potrošnja" "cost"	30%, 6
Mera2	"potrošnja" "cost"	10%, 6	"potrošnja" "cost"	30%, 6
Mera3	"imanje" "estate"	6.25%, 4	"finansijski gubitak" "financial loss"	33%, 4

Tabela 6.1: Najproduktivniji koncept za hijerarhiju sa korenom "svojina" u SWN i PWN korišćenjem različitih mera

6.2 Biblioteka XQuery funkcija "wnfun"

U cilju određivanja mera produktivnosti koncepta, razvijena je biblioteka XQuery funkcija nazvana "wnfun". Radi dobijanja čvora sa maksimalnom vrednošću

mere *Mera1*, u okviru ove biblioteke razvijene su funkcije nad *Wordnet*-om koje daju odgovor na sledeća pitanja:

- Koliki je broj naslednika nekog čvora odnosno koliki je broj čvorova (koncepata) u hijerarhiji sa korenom u tom čvoru?
- Kolika je udaljenost proizvoljnog čvora od korena (broj čvorova koji su na putu od tog čvora do korena)?
- Za proizvoljan čvor i za dato α i β, kolika je vrednost mere *Mera1*?
- Koji čvor u drvetu ima maksimum mere *Mera1* (najproduktivniji čvor)?

U cilju određivanja čvora sa maksimalnom vrednošću mere *Mera2*, osim već pomenutih funkcija, u okviru ove biblioteke razvijene su i funkcije nad *Wordnet*-om koje daju odgovore na pitanja:

- Za dati čvor, koliki je broj koncepata listova, u okviru hijerarhije sa korenom u tom čvoru?
- Za proizvoljan čvor i za dato α i β, kolika je vrednost mere *Mera2*?
- Koji čvor u drvetu ima maksimum mere *Mera2* (najproduktivniji čvor)?

Na kraju, u cilju određivanja najproduktivnijeg koncepta koristeći meru *Mera3*, razvijene su funkcije koje daju odgovore na pitanja:

- Koliki je broj naslednika sve braće posmatranog čvora odnosno koliki je broj naslednika svih čvorova koji su na istom nivou sa tim čvorom i imaju istog roditelja?
- Za proizvoljan čvor, kolika je vrednost mere *Mera3*?
- Koji čvor u drvetu ima maksimum mere *Mera3* (najproduktivniji čvor)?

6.3 Poređenje najproduktivnijih koncepata SWN i PWN

Posmatranjem SWN-a i PWN-a, posebno posmatranjem njihovih najproduktivnijih koncepata korišćenjem različitih mera produktivnosti, mogu se doneti korisni zaključci o nepotpunosti SWN-a i o pravcu u kome treba dalje razvijati SWN. U okviru ove analize uvešćemo sledeće mere: *Mera2* koja je prethodno već definisana, *Mera1Alfa1Beta1* koja će biti vrednost mere *Mera1* za $\alpha = 1$ i $\beta = 1$, *Mera1Alfa2Beta1* biće vrednost mere *Mera1* za $\alpha = 2$ i $\beta = 1$ i *Mera1Alfa1Beta2* biće vrednost mere *Mera1* za $\alpha = 1$ i $\beta = 2$. Tako se, za svaku od hijarahija u SWN-u i PWN-u, može zaključiti sledeće:

- Hijerarhija sa korenom u konceptu "entitet":

  U SWN-u nema većeg odstupanja u odnosu na istu hijerarhiju u PWN-u. Maksimumi svih prethodno opisanih mera se podudaraju i to su koncept "fizički objekat" (koji je na drugom nivou u hijerarhiji) za meru *Mera1Alfa2Beta1* i koncept "organizam" (na četvrtom nivou hijerarhije) za sve ostale mere;

- Hijerarhija sa korenom u konceptu "apstrakcija":

  Maksimumi mera *Mera1Alfa1Beta1* i *Mera2* se podudaraju u oba *Wordnet*-a i to je koncept "komunikacija" (na četvrtom nivou hijerarhije).

  U PWN-u, maksimum mere *Mera1Alfa2Beta1* je koncept "apstrakcija" i ovaj maksimum se "popeo" više uz hijerarhiju (do korena) u odnosu na SWN gde je maksimum iste mere koncept "komunikacija". Razlog tome je što u SWN-u hijerarhija sa korenom u konceptu "komunikacija" ima ukupno 1015 koncepata od 1774 koncepata koliko ima hijerarhija sa korenom u njegovom direktnom pretku "društveni odnos" (što je 57%) a u PWN-u ima 4638 od 11053 (42%). Kako mera *Mera1Alfa2Beta1* više uzima u obzir broj koncepata nego udaljenost od korena (ima težnju da se penje na gore u hijerarhiji u odnosu na meru *Mera1Alfa1Beta1*), to su dobijeni najproduktivniji koncepti logično rešenje.

  Maksimum mere *Mera1Alfa1Beta2* u SWN-u se "spustio" niže niz hijerarhiju u SWN-u do koncepta "prirodni jezik" (koji je na šestom nivou) u odnosu na PWN kod koga je maksimum iste mere koncept "komunikacija". Razlog tome je odstupanje kod SWN-a u odnosu na PWN za koncepte koji su deca koncepta "komunikacija" (najznačajniji su "jezik", "pismena komunikacija" i "poruka"), odnosno za njihove hijerarhije. U SWN-u broj koncepata u hijerarhiji sa korenom u konceptu "jezik" je 639 od 1015 koliko ima u hijerarhiji njegovog direktnog pretka (što je 63%) dok je u PWN-u taj broj 753 od 4638 (što je 16%). Zbog toga koncept "jezik" dobija na značaju u SWN-u pa koncept koji je njegovo dete ("prirodni jezik") izbija na površinu. Za koncepte "pismena komunikacija" (150 od 1015 (15%) u SWN i 1210 od 4638 (26%) u PWN) i "poruka" (61 od 1015 (6%) u SWN i 1009 od 4638 (22%) u PWN) koji su takođe deca od koncepta "komunikacija", takođe postoji velika razlika u okviru ova dva *Wordnet*-a.

  Ova analiza ukazuje na to da bi u okviru SWN-a za ovu hijerarhiju, kako bi se što više uskladio sa PWN-om, proces dodavanja novih koncepata trebalo da ide u smeru dodavanja koncepata braći koncepta "komunikacija" i "jezik" a posebno hijerarhijama sa korenom u konceptima "pismena komunikacija" i "poruka";

- Hijerarhija sa korenom u konceptu "grupa":

  Maksimumi mera *Mera1Alfa1Beta1*, *Mera2* i *Mera1Alfa2Beta1* se poklapaju u okviru oba *Wordnet*-a i to je koncept "taksonomska grupa" (na trećem nivou hijerarhije). Maksimum mere *Mera1Alfa1Beta2* u SWN-u je takođe koncept "taksonomska grupa" dok se u PWN maksimum ove mere "spustio" niže za jedan nivo niz hijerarhiju do koncepta "rod".

  Može se zaključiti da nema većih odstupanja u ovoj hijerarhiji u okviru ova dva *Wordnet*-a;

- Hijerarhija sa korenom u konceptu "akt":

  Maksimumi mera *Mera1Alfa1Beta2* (koncept "aktivnost" na drugom nivou hijerarhije) i *Mera1Alfa2Beta1* (koncept "akt" na prvom nivou - koren) u okviru SWN i PWN se poklapaju.

Maksimumi mera *Mera1Alfa1Beta1* i *Mera2* u SWN-u su u konceptu "aktivnost" dok se u PWN-u penju za jedan nivo više do korena "akt". Razlog je što hijerarhija sa korenom u "aktivnost" u PWN-u iznosi 3164 koncepata od ukupno 6704 koliko ima njegov neposredni predak "akt" (što je 47%) a u SWN-u taj odnos je 405 prema 735 (što je 55%).

Ova hijerarhija u SWN-u nema većih odsutpanja u odnosu na PWN;

- Hijerarhija sa korenom u konceptu "psihičko svojstvo":

  Maksimumi mera *Mera1Alfa1Beta1*, *Mera1Alfa2Beta1* i *Mera2* se poklapaju u okviru oba *Wordnet*-a i dostiže se u konceptu "saznanje" (na drugom nivou hijerarhije).

  Analizom mere *Mera1Alfa1Beta2* uočava se veće odstupanje SWN-a u odnosu na PWN. Maksimum ove mere u PWN-u dosiže se u konceptu "saznajni sadržaj" (na trećem nivou) dok se u okviru SWN-a "spušta" niže niz hijerarhiju do koncepta "biologija" (na devetom nivou). Razlog ovome je što u SWN-u, u okviru dece koncepta "saznajni sadržaj", znatno više koncepata je prisutno za koncept "baza znanja" (64 od 162 koncepata njegovog neposrednog pretka (40%) dok je u PWN-u prisutno 679 od 2214 (što je 30%)) nego za "verovanje" (27 od 162 (16%) dok je kod engleskog 679 od 2214 (30%)) ili koncept "ideja"(48 od 162 (30%) dok je kod engleskog 549 od 2214 (25%)). Zbog toga "baza znanja" i koncept u okviru njenog poddrveta ("biologija") izbijaju u prvi plan u SWN-u dok se u PWN-u zadržavaju kod koncepta "saznajni sadržaj".

  Zaključak je da u SWN treba uneti više koncepata u okviru hijerarhija sa korenom u konceptima "verovanje" i "ideja";

- Hijerarhija sa korenom u konceptu "stanje":

  Maksimum mere *Mera1Alfa2Beta1* se poklapa u okviru oba *Wordnet*-a i dostiže se u korenom konceptu "stanje".

  Maksimumi mera *Mera1Alfa1Beta1* i *Mera2* u SWN-u dostižu se u konceptu "stanje" a mere *Mera1Alfa1Beta2* u konceptu "postojanje" (na drugom nivou). U PWN-u maksimumi ovih mera se "spuštaju" niže niz hijerarhiju do koncepta "zdravstveni problem" (na četvrtom nivou) za mere *Mera1Alfa1Beta1* i *Mera2* odnosno do koncepta "oboljenje" (na sedmom nivou) za meru *Mera1Alfa1Beta2*. Razlog tome je što je u SWN-u relativno malo koncepata unešeno za koncept "postojanje" (99 od ukupno 256 (39%) dok je u PWN-u za isti koncept unešeno 2000 od 3135(64%)) pa se zbog toga maksimumi mera *Mera1Alfa1Beta1* i *Mera2* zadržavaju na njegovom neposrednom pretku "stanxe". U okviru koncepta "postojanje" u SWN unešeno je malo koncepata za "patološko stanje" (25 od 99 (25%) a u PWN-u je 1097 od 2000 (55%)) pa se najproduktivniji koncept (po meri *Mera1Alfa1Beta2* koja teži da ide niže niz hijerarhiju jer više značaja daje udaljenosti od korena nego brojnosti čvorova) u SWN-u zadržava na konceptu "postojanje" s obzirom da se od njegove dece niko svojom brojnošću posebno ne ističe. U PWN-u ističe se po brojnosti koncept "patološko stanje" pa koncepti u njegovom poddrvetu dolaze do izražaja.

Ovo ukazuje na to da u okviru SWN-a više koncepata treba uneti u hijerarhiju sa korenom u "postojanje" odnosno njegovom detetu "patološko stanje";

- Hijerarhija sa korenom u konceptu "događaj":

  Maksimumi mera *Mera1Alfa1Beta1*, *Mera2* i *Mera1Alfa2Beta1* se poklapaju u SWN-u i PWN-u i to je sam koren drveta odnosno koncept "događaj". Na drugom nivou, u okviru oba *Wordnet*-a ističu se dva konepta ("grupna akcija" i "dešavanje") s tim da je "grupna akcija" malo brojnija u odnosu na "dešavanje" u SWN-u i predstavlja maksimum mere *Mera1Alfa1Beta2*, dok je u PWN-u brojnija hijerarhija sa korenom u "dešavanje" i predstavlja maksimum mere *Mera1Alfa1Beta2*;

- Hijerarhija sa korenom u konceptu "pojava":

  Maksimalna vrednost mere *Mera1Alfa2Beta1* se poklapa u okviru oba *Wordnet*-a i dostiže se u korenom konceptu "pojava".

  Kod oba *Wordnet*-a u okviru drugog nivoa (dece korena) ističu se dva koncepta: "prirodna pojava" (43 od 99 (43%) u SWN-u i 599 od 1622 (37%)u PWN-u) i "proces" (49 od 99 (49%) u SWN-u i 980 od 1622 (60%) u PWN-u). Zanimljivo je da kod oba *Wordnet*-a, za meru *Mera1Alfa1Beta2*, dolazi do izražaja manje brojan koncept "prirodna pojava" jer se u okviru njegovih potomaka jedan od njih (koncept "fizička pojava") više ističe po brojnosti dok je brojnost potomaka kod koncepta "proces" više ujednačena. U PWN, maksimumi mera *Mera1Alfa1Beta1* i *Mera2* dostižu se u konceptu "proces" a maksimum mere *Mera1Alfa1Beta2* u konceptu "fizička pojava". Kod SWN-a, maksimumi ovih mera se spuštaju malo niže niz hijerarhiju. Tako je maksimum mera *Mera1Alfa1Beta1* i *Mera2* koncept "fizička pojava" a maksimum mere *Mera1Alfa1Beta2* je koncept "svetlost" (na osmom nivou hijerarhije). Ovo ne proizilazi iz neujednačenosti SWN-a i PWN-a već zbog veće brojnosti čvorova PWN-a koja nije uključena u okviru posmatranih mera.

  Što se ove hijerarhije tiče, oba *Wordnet*-a su prilično ujednačena;

- Hijerarhija sa korenom u konceptu "svojina":

  U okviru ove hijerarhije, maksimumi mera *Mera1Alfa1Beta2* (koncept "potrošnja") i *Mera1Alfa2Beta1* (koncept "svojina") u okviru oba *Wordnet*-a se poklapaju.

  U okviru drugog nivoa hijerarhije, odnosno u okviru dece korenog koncepta, po brojnosti se najviše ističu dva koncepta: "imovina" i "prenosno vlasništvo". Maksimum mera *Mera1Alfa1Beta1* i *Mera2* u SWN-u dostiže se u konceptu "svojina". U PWN-u, maksimum ovih mera se spušta niže niz hijerarhiju do koncepta "potrošnja" (na šestom nivou) koji je potomak koncepta "prenosno vlasništvo". Razlog tome je što je u okviru SWN-a relativno mali broj koncepata prisutan za hijerarhiju sa korenom u konceptu "preneseno vlasništvo" (24 od 79 (30%) a u PWN-u 331 od 753 (44%)) pa se najproduktivniji koncept zadržava na korenom elementu. U SWN-u se po brojnosti više od koncepta "prenosno vlasništvo" ističe koncept "imovina" (30 od 79 (38%) a u PWN-u je 269 od 753 (36%)) međutim njegovi potomci su po brojnosti ujednačeniji pa se više ističu koncepti u

koreni koncepti	najproduktivniji koncpet (PWN)	najproduktivniji koncpet (SWN)
"entity" ("entitet")	"organism" ("organizam") (18997 – 44%)	"organism" ("organizam") (2884 – 55%)
"abstraction" ("apstrakcija")	"communication" ("komunikacija") (4638 – 42%)	"natural language" ("prirodni jezik") (610 – 34%)
"group" ("grupa")	"genus" ("rod") (3652 – 44%)	"taxon" ("takson") (1826 – 81%)
"human action" ("ljudska aktivnost")	"activity" ("aktivnost") (3164 – 47%)	"activity" ("aktivnost") (405 – 55%)
"psychological feature" ("psihičko svojstvo")	"content" ("saznajni sadržaj") (2214 – 46%)	"biology" ("biologija") (31 – 8%)
"state" ("stanje")	"disease" ("oboljenje") (514 – 16%)	"condition" ("postojanje") (99 – 40%)
"event" ("događaj")	"happening" ("dešavanje") (941 – 44%)	"group acton" ("grupna akcija") (115 – 50%)
"phenomenon" ("fenomen")	"physical phenomenon" ("fizička pojava") (513 – 32%)	"light" ("svetlost") (5 – 5%)
"possession" ("svojina")	"cost" ("potrošnja") (233 – 31%)	"cost" ("potrošnja") (8 – 10%)

Slika 6.1: Najproduktivniji koncepti u okviru 9 hijerarhija, korišćenjem mere produktivnosti *Mera1Alfa1Beta2* u SWN i PWN

okviru poddrveta sa korenom "preneseno vlasništvo" (što se i odrazilo na maksimum mere *Mera1Alfa1Beta2* u SWN-u).

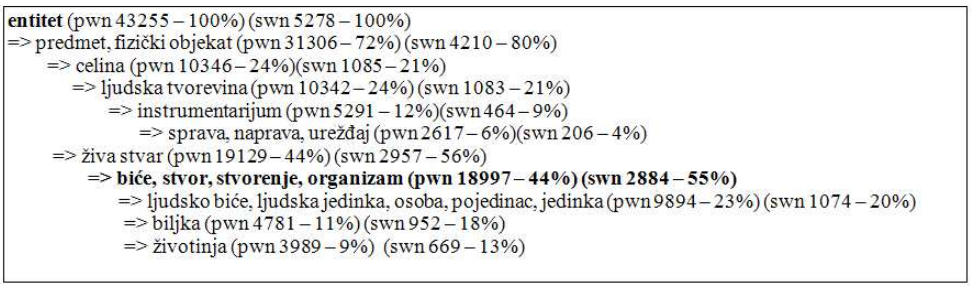
Zaključak je da bi veći broj koncepata mogao da se doda u hijerarhiju sa korenom u konceptu "preneseno vlasništvo".

U tabeli 6.1 dati su najproduktivniji koncepti u SWN-u i PWN-u za hijerarhije svih devet korenih koncepata uzimajući u obzir meru *Mera1Alfa1Beta2*, kao i procenat koliko svako poddrvo sa korenom najproduktivnijem konceptu svojim brojem koncepata učestvuje u okviru celog drveta.

6.4 Jedan primer najproduktivnijeg koncepta

Na slici 6.2 može se videti jedan primer određivanja najproduktivnijeg koncepta korišćenjem mere *Mera1* za $\alpha = 1$ i $\beta = 2$. Na slici je uz svaki koncept dat prikaz brojnosti hijerarhije sa korenom u posmatranom konceptu za SWN i PWN kao i procenat koliko to poddrvo svojim brojem koncepata učestvuje u okviru celog drveta. Na slici su prikazani samo koncepti koji se izdvajaju po svojoj brojnosti.

Kao što se može primetiti na slici 6.2 najproduktivniji koncept za hijerarhiju



Slika 6.2: Najproduktivniji koncept za hijerarhiju sa korenom u konceptu "entitet"

sa korenom u konceptu "entitet" je "organizam". Hijerarhija sa korenom u tom konceptu sadrži 44% svih koncepata polazne hijararihije u PWN-u i 55% u SWN-u. Ovaj koncept kao što se može videti pripada četvrtom nivou cele hijerarhije.

6.5 Primena najproduktivnijeg koncepta

Kako se smislene rečenice sastoje od smislenih reči, bilo koji sistem koji ima nameru da obrađuje prirodne jezike na način na koji to čine ljudi, mora imati informacije o tim rečima i njihovim značenjima. Ove informacije su tradicionalno podržane kroz rečnike. Mašinski čitljivi rečnici su danas široko rasprostranjeni, međutim ovi rečnici su namenjeni ljudskoj upotrebi pre nego upotrebi od strane računara. *Wordnet* obezbeđuje dosta efikasniju kombinaciju tradicionalnih leksikografskih informacija i modernih računara. To je leksička baza podataka projektovana za korišćenje od strane računara.

Wordnet je korišćen u okviru različitih vrsta obrade prirodnih jezika kao što su klasifikacija tekstova [34], izdvajanje informacija [36], označavanje vrsta reči u tekstu [33], rešavanje problema višeznačnosti reči [32] i tako dalje.

Klasifikacija tekstova ima za zadatak pridruživanje tekstualnom dokumentu jednu ili više prethodno definisanih kategorija, na osnovu njegovog sadržaja. Obično su tekstualni dokumenti predstavljeni kao skupovi reči koje se pojavljuju u tom dokumentu, i takav prikaz poznat je pod imenom "Model vreće reči" (eng. bag of words model).

Klasifikacija tekstova može biti izvršena na različite načine korišćenjem mera produktivnosti. Tako na primer, neki odabrani najproduktivniji koncepti mogu biti pridruženi odabranim klasama koje će predstavljati. Tekstualni dokument može biti klasifikovan posmatranjem njegove frekvencije literala (ili nekih odabranih literala kao što su imenovani entiteti) hijerarhije sa korenom u konceptu koji je pridružen klasama.

Ako se klasifikacija izvrši na ovaj način onda i izdvajanje informacija može biti obavljeno efikasnije. Ovaj zadatak se svodi na izdvajanje onog dokumenta iz kolekcije dokumenata koji najviše odgovara korisnikovim potrebama. Korisnik svoju potrebu opisuje upitom koji se sastoji od izvesnog broja reči. Sistem za izdvajanje informacija vrši poređenje upita sa dokumentima iz kolekcije i vraća kao rezultat onaj dokument koji najviše odgovara korisnikovim potrebama. Može biti jako korisno ako bi se upit proširio svim literalima iz hijerarhije

nekog dobro izabranog produktivnog koncepta.

Rad zasnovan na rezultatima opisanim u ovom poglavlju, prihvaćen je za objavljivanje na konferenciji "Sixth Language Technologies Conference" u Ljubljani, oktobra 2008 [13].

7

Klasifikacija tekstova

7.1 Uvod u klasifikaciju

Eksponencijalnim rastom Interneta javlja se sve veća potreba za relevantim informacijama. Relevantnost informacija se najviše može povećati klasifikacijom ili kategorizacijom.

Klasifikacija predstavlja preslikavanje podataka u predefinisani skup klasa koje su unapred poznate. Ulazni podatak u klasifikaciju je skup slogova (podaci za trening, eng. training set) pri čemu je svaki slog oblika (x, y) gde je x skup atributa a y specijalni atribut određen za oznaku klase. U okviru ovog postupka, potrebno je pronaći klasifikacioni model (funkciju) koji preslikava svaki skup atributa x u jednu od predefinisanih oznaka klasa y.

Koraci u primeni klasifikacionog postupka su:

1. Konstruisanje klasifikacionog modela ili funkcije pripadnosti klasi na osnovu podataka za trening;

2. Primena tog modela na novim podacima.

Dakle, cilj klasifikacije je dodeliti slogove koji nisu prethodno poznati što je moguće preciznije jednoj od klasa.

Ulazni podaci se obično dele u dva dela:

- *Podatke za trening* pomoću kojih se formira model i koji se koriste radi određivanja tačnosti modela;

- *Podatke za testiranje* koji se koriste za proveru ispravnosti modela.

Klasifikacija može biti binarna, kada su definisane samo dve klase i može biti višeklasna, kada je definisano više klasa .

Kada se obavi klasifikacija, važno je dobro proceniti kakvog je kvaliteta ta klasifikaicija [43].

7.1.1 Procena kvaliteta klasifikacije

Veoma je važno proceniti koliko je dobro klasifikacioni model uspeo da generalizuje problem na osnovu podataka za trening. Jedan od problema koji se mogu javiti jeste problem "ukalupljivanja" (eng. overfitting). To je kada se generisani

		stvarna klasa	
		da	ne
predložena klasa	da	stvarno pozitivni	lažno pozitivni
	ne	lažno negativni	stvarno negativni

Slika 7.1: Mogući ishodi kod binarne klasifikacije.

Slika 7.2: Mogući ishodi kod binarne klasifikacije.

klasifikacioni model ponaša dobro na podacima za trening, ali podbacuje na novim podacima za testiranje. Dakle, problem nije dobro generalizovan. To se obično javlja kada model pokušava da predstavi šum koji postoji u ulaznim podacima kao bitan element.

Za testiranje se mora koristiti nezavisan skup podataka (neupotrebljavan za treniranje). Proces evaluacije se sastoji u poređenju poznate klase sa onom koju je predložio klasifikator. Time dobijamo ispravno i neispravno klasifikovane jedinke [43].

Mogući ishodi kad je u pitanju binarna klasifikacija su:

- Stvarno pozitivni (true positives, TP);
- Stvarno negativni (true negatives, TN);
- Lažno pozitivni (false positives, FP);
- Lažno negativni (false negatives, FN) (slike 7.1 i 7.2).

Razlikuju se dva tipa evaluacije:

- Na nivou jedne klase;
- Na nivou više klasa.

Kada se posmatra evaluacija na nivou jedne klase, mogu se definisati sledeće mere za procenu kvaliteta klasifikacije:

- *Preciznost* koja ocenjuje tačnost klasifikacije odnosno koliki procenat test primera je ispravno klasifikovan. Izračunava se po formuli:

$$Preciznost = \frac{stvarno\ pozitivni}{stvarno\ pozitivni + lazno\ pozitivni}$$

pri čemu $stvarno\ pozitivni + lazno\ pozitivni$ predstavljaju u stvari ukupan broj primera koji pripadaju predloženoj klasi;

- *Odziv* (pokrivanje, eng. recall) koji ocenjuje koliko je model uspešan u pokrivanju klase odnosno koliko test primera iz date klase (ili klasa) model može da prepozna. Izračunava se po formuli:

$$Odziv = \frac{stvarno\ pozitivni}{stvarno\ pozitivni + lazno\ negativni}$$

pri čemu *stvarno pozitivni* + *lazno negativni* predstavljaju ukupan broj primera koji pripadaju stvarnoj klasi;

- *F-mera* koja predstavlja kombinaciju preciznosti i pokrivanja u jednoj meri kao na primer harmonijska sredina preciznosti i pokrivanja. Izračunava se po formuli:

$$F - mera = \frac{2 * Preciznost * Odziv}{Preciznost + Odziv}$$

Kvalitet klasifikacije može da se posmatra ne samo na jednoj datoj klasi već na nivou više klasa (globalni kvalitet). U tom slučaju do izražaja dolaze dve mere:

- *Mikro-prosek* gde se posmatra prosečna vrednost na skupu svih jedinki (iz svih klasa);
- *Makro-prosek* gde se posmatra prosečna vrednost na skupu klasa tako što se izračuna preciznost i odziv za svaku klasu posebno, a onda se izračuna njihova srednja vrednost.

U procesu validacije, može se primeniti i postupak n-unakrsnih validacija (n-cross validation). On se sastoji iz sledećih koraka:

- Raspoloživi skup klasifikovanih podataka se na slučajan način podeli na delove za trening i delove za testiranje (npr. u odnosu 9:1 ili 4:1, ređe 1:1);
- Primeni se algoritam za učenje na delu za trening i proceni se kvalitet naučenog klasifikatora na test delu;
- Ovaj postupak (koraci 1 i 2) se ponove n puta (obično 10), i potom se izračuna srednja vrednost posmatrane ocene kvaliteta (npr. srednja vrednost preciznosti ili srednja vrednost makro-proseka) [43].

7.1.2 Postojeće tehnike klasifikacije

Postoje različite tehnike klasifikacije, a najznačajnije su:

- Metode zasnovane na drvetima odlučivanja (eng. decision trees);
- Metode zasnovane na pravilima (eng. rule based);
- Metode zasnovane na neuronskim mrežama;
- Statistički zasnovane metode (regresija i Bajesovska klasifikacija);
- Metode zasnovane na podržavajucim vektorima (eng. support vector machines, SVM);
- Metode zasnovane na rastojanju (najbliži sused, eng. nearest neighbour);
- Ukalupljivanje modela (eng. model overfitting) [43].

7.1.3 Primena klasifikacije

Osim klasifikacije dokumenata (sport, vreme, ...) postoje i drugi primeri primene klasifikacije:

- Klasifikacija ćelija tumora kao benignih ili malignih;
- Klasifikacija ispravnosti kreditnih kartica;
- Klasifikacija sekundarne strukture proteina.

7.2 Klasifikacija tekstova

U okviru ove klasifikacije, cilj je dodeliti po jednu ili više klasa svakom dokumentu po nekom kriterijumu (u zavisnosti od primene ili domena). Dokument se pri procesu klasifikacije može predstaviti kao vektor svih reči (ili osnova reči) koje se u njemu pojavljuju. Ovakav pristup se često naziva "bag of words" ili vreća reči. Obično se ne uzima u obzir redosled pojavljivanja reči u dokumentu. Moguće je da se dokument predstavi ne kao vektor svih već kao vektor nekih odabranih reči ("bag of terms", "bag of names", "bag of ...") a mogu se i izostaviti "funkcijske" reči (npr. pomoćni glagoli, predlozi, zamenice, itd.). Svakom atributu (tj. reči) vrši se zatim dodela težine. Ova dodela može biti binarna (atribut prisutan/odsutan) a može se svakom atributu dodeliti specifična težina (npr. apsolutna/relativna frekvencija) [43].

7.2.1 Problemi i izazovi

U procesu klasifikacije, bitno je dobro izabrati odgovarajuće atribute i karateristike za predstavljanje problema jer ne moraju svi atributi biti podjednako "korisni" ("feature selection"). Problem može da predstavlja i to što su neki atributi teški za generisanje (npr. eksperimentalni podaci). Potrebno je takođe izabrati odgovarajuće težine za atribute i karateristike: binarni (0,1) ili specifične težine, zatim napraviti dobar skup podataka za trening što obično zahteva "ručnu" izradu. To je vrlo često jako skupo, sporo i podložno greškama. Posebno je potrebno povesti računa o šumovima/greškama u podacima za trening.

Ono što takođe predstavlja izazov je:

- *Brzina* kojom se gradi model i kojom se primenjuje model na nove primerke;
- *Velike i male klase.* Klasifikacija koja dodeljuje svima većinsku klasu u principu pravi najmanje greški. Izazov je napraviti klasifikator za prepoznavanje "manjinskih" klasa;
- *Zamućena (eng. fuzzy) klasifikacija* ili klasifikacija sa određenim stepenom značajnosti [43].

7.2.2 Primeri postojećih klasifikacija tekstova

Neki od primera postojećih, dobro poznatih klasifikacija tekstova su:

- *EBART*: Predstavlja najveću digitalnu medijsku dokumentaciju u Srbiji, sa gotovo milion novinskih tekstova iz dnevne i nedeljne štampe arhiviranih od početka 2003. godine naovamo. U okviru nje su pohranjena

kompletna izdanja petnaestak dnevnih i nedeljnih novina koje izlaze na teritoriji cele Srbije i odabrani tekstovi iz najvećih lokalnih nedeljnika. Arhivi se pristupa preko Ebartovog sajta www.arhiv.co.yu. U arhivi se danas nalazi preko 1.200.000 potpuno indeksiranih tekstova koje je moguće pretraživati u punom tekstu (engl. full text).

Aktuelna Arhiva je klasifikovana na tematske celine po ugledu na uobičajene novinske rubrike:

- Unutrašnja politika;
- Spoljna politika;
- Društvo;
- Ekonomija;
- Hronika i kriminal;
- Kultura i zabava;
- Sport;
- Mediji;
- Feljtoni;
- Pisma čitalaca.

Korisnicima su dostupne raznovrsne predefinisane pretrage tekstova, pa se svi paketi mogu pretraživati po:

- Temama - oko 900 tema podeljenih na 3 nivoa (primer: Industrija - Prehrambena industrija - Konditorska industrija);
- Ličnostima - oko 4400 ličnosti koje se mogu pretraživati uz različita ukrštanja sa temama i drugim ličnostima;
- Institucijama - oko 1800 institucija (ekonomske, društvene, kulturne...);
- Političkim strankama (sve parlamentarne i važnije vanparlamentarne);
- Geografskim odrednicama (Beograd, Beč, Bečej, Beočin, Balkan, Banat...i tako do 550 različitih gradova, sela, regiona, država, planina...);
- Manifestacijama (sajmovi, festivali, nagrade...);
- Dokumentima (važni zakoni, odnosno sve sto su novine pisale o hipotekama, koncesijama, porezima, radiodifuziji, stečaju, telekomunikacijama...);
- Događajima (svi izbori, afere, veliki međunarodni skupovi, vanredno stanje...).

- *Reuters novinska kolekcija*: Ima ukupno 118 kategorija ili klasa. Svaki dokument može biti u više od jedne klase (kategorije) a kolekcija za treniranje/testiranje obuhvata 21578 dokumenata;
- *Klasifikacija e-mailova*: Svaki e-mail se binarno klasifikuje kao spam ili ne-spam [43];
- *Klasifikacija e-mailova ili SMS poruka*: Svaki e-mail se klasifikuje u zavisnosti od "teme" npr. klasifikovati e-mailove/SMS koji stignu na korisnički servis kao "pohvala", "problem sa servisom", "tehnički problem", "problem sa plaćanjem", "spam", itd. [37]

7.3 Klasifikacija tekstova zasnovana na Wordnet-u

Pitanje koje se postavlje jeste da li je nekako informacije sadržane u *Wordnet*-u moguće iskoristiti u cilju bolje klasifikacije dokumenata. Na ovu temu objavljeno je svega nekoliko radova.

Jedan od najznačajnijih radova na tu temu svakako je [38].

Sam postupak klasifikacije opisane u tom radu sastoji se iz tri prolaza kroz korpus:

1. U prvom prolazu, svim rečima u dokumentima iz posmatranog korpusa pridružuje se oznaka vrste reči (imenice, pridevi, glagoli...).

2. U drugom prolazu, za svaku imenicu ili glagol, vrši se pregled *Wordnet*-a i pravi se globalna lista svih sinonima i hipernima svake od tih reči. Oni sinsetovi koji se retko javljaju u korpusu isključuju se iz posmatranja a oni preostali formiraju skup osobina.

3. Tokom trećeg prolaza, izračunava se gustina svakog sinseta (definisana kao odnos broja pojave literala iz tog sinseta i ukupnog broja reči u dokumentu).

Definiše se i parametar h koji predstavlja meru generalizacije odnosno koliko nivoa naviše treba posmatrati hipernime za dati sinset.

Pripadnost nekoj klasi se definiše korišćenjem dobijenih gustina za sinonime. Tako je u ovom radu dat primer sa dve klase, Istorija i Porez. Pravilo pripadnosti klasi se definiše preko gustine sinseta "vlasništvo". Ako je njegova gustina mala u nekom dokumentu, onda taj dokument pripada klasi Istorija a u suprotnom pripada klasi Porez. Smatra se da se reči koje se odnose na reč "vlasništvo", uglavnom koriste u tekstovima koji se bave temom poreza i vrlo retko su to neki istorijski dokumenti. Pokazalo se da je ovaj pristup dobar kod tekstova koji koriste nestandardni i prošireni vokabular.

Poznati rad iz ove oblasti je i [40]. U okviru ovog rada koristi se *Wordnet* za unapređivanje metoda zasnovanih na neuronskim mrežama i primenjuje se nad Reuters-21578 novinskom kolekcijom. Rad novijeg datuma je [39]. U ovom radu istraživano je kako uključivanje semantičkih informacija može unaprediti zadatke klasifikacije tekstova i istraživanja podataka. U okviru ova dva rada, *Wordnet* se koristi samo u cilju dobijanja sinonima neke reči.

7.3.1 Klasifikacija zasnovana na odabranim konceptima

Ideja je da se *Wordnet* iskoristi u klasifikaciji tekstova na jedan nov i originalan način.

Ukratko, postupak izvođenja eksperimenta je sledeći:

- Svi članci iz lista Politika, smeštaju se u izvornu XML bazu podataka;
- Korišćenjem XQuery upita, iz kolekcije članaka izdvajaju se oni članci sa pridruženim odgovarajućim rubrikama;
- Definišu se klase određene odabranim rubrikama;
- Izdvajaju se ključne reči koje karakterišu te rubrike odnosno klase;

- U Wordnet-u se pronalaze koncepti koji u okviru svoje hijerarhije sadrže tako odabrane ključne reči, i takvi koncepti se pridružuju klasama;
- Definišu se funkcije pripadnosti klasama na osnovu pridruženih im koncepata, kao maksimalni broj pojave literala iz hijerarhija sa korenima u tim konceptima, eventualno filtrirani po nekom domenu.

Kao što je već pomenuto, za test podatke uzeti su u razmatranje članci iz Politike za 2003, 2004, 2005 i 2006 godinu (2.74 GB). Ceo korpus sastoji se iz 48 xml datoteka (po 12 datoteka za svaku godinu odnosno po jedna za svaki mesec, na primer Januar2003, Februar2003 i tako dalje). Svaka XML datoteka je veličine oko 60 MB i u okviru nje se nalaze tekstovi iz Politike u formatu ilustrovanom sledećim primerom:

```
<Clanak Unid='0D81A42F78FF3479C1256CFB001FC5DF'>
<Novina>Blic</Novina>
<Datum>1.4.2003</Datum>
<Rubrika>Sport</Rubrika>
<Strana>29</Strana>
<Naslov>PLAVI SESTI</Naslov>
<Tekst>stoni tenis

KURMAJER - Stonoteniska reprezentacija Srbije i Crne Gore osvojila
je šesto mesto na Evropskom prvenstvu u italijanskom Kurmajeru,
pošto je sinoć u duelu za petu poziciju poražena od Hrvatske 1:3.
Plave devojke su izgubile susret sa Nemačkom (2:3), završivši
tako Šampionat kao osme. Reprezentativke Italije osvojile su
titulu prvaka Evrope pobedivši u finalu Hrvatsku 3:1.

</Tekst>
<Autor>K.B.</Autor>
<Antrfile></Antrfile>
</Clanak>
```

Svi ovi članci nalaze se u u okviru korenog elementa koji daje informaciju o mesecu i godini u kojoj su ti članci objavljeni, na primer
`<Unos Mesec='April2003'>`
i svi su smešteni u okviru izvorne XML baze.

U cilju klasifikacije ovih tekstova, odabrane su tri klase: *Sport*, *Ekonomija* i *Politika*. Za neki članak smatra se da pripada klasi *Sport*, *Ekonomija* ili *Politika* ukoliko je njegova rubrika označena sa Sport, Ekonomija ili Politika, redom. Korišćenjem XQuery upita (videti sliku 7.3) iz kolekcije svih članaka izdvojeni su samo oni članci relevantni za proces klasifikacije, odnosno oni sa pridruženim rubrikama Sport, Ekonomija i Politika.

U cilju formiranja skupa za trening, slučajnim izborom odabrano je po nekoliko članaka iz ovih rubrika. Cilj je svakoj od klasa pridružiti po jedan ili više koncepata (sinsetova) iz *Wordnet*-a koji najbolje određuje tu klasu. Za svaku od rubrika uočene su ključne reči koje tu rubriku najbolje opisuju. Ove reči su izabrane ručno, posmatranjem tekstova tih članaka. Jedna od mogućih provera koliko je neka reč ključna za neku rubriku jeste određivanje koliko članaka sa tom rubrikom sadrži tu reč. Ovakva jedna provera može se izvršiti XQuery upitom prikazanim na slici 7.4. Nakon adekvatno izabranih ključnih reči, u *Wordnet*-u

```
(: Izdvoj sve clanke iz Politike... :)
for $c in //Unos/Clanak

(:...za koje vazi da im je rubrika Sport, Ekonomija ili Politika.:)
where $c/Rubrika/text() eq 'Sport' or  $c/Rubrika/text() eq 'Ekonomija' or  $c/Rubrika/text() eq 'Politika'

return $c
```

Slika 7.3: XQuery upit za izdvajanje članaka iz Politike sa rubrikama Sport, Ekonomija i Politika

```
(: Izdvoj sve clanke iz Politike... :)
for $c in //Unos/Clanak

(:...za koje vazi da im je rubrika Sport i da u sebi sadrze rec “igra”.:)
where $c/Rubrika/text() eq 'Sport' and fn:contains($c/Tekst/string(), "igra")
return $c
```

Slika 7.4: XQuery upit za izdvajanje članaka iz Politike sa rubrikom Sport u čijem tekstu se pojavljuje reč ”igra”

su pronađeni koncepti koji u okviru svoje hijerarhije sadrže te ključne reči. U tu svrhu korišćen je VisDic [44] alat za jednostavan pristup podacima iz *Wordnet*-a.

Ovim pristupom, pomenute klase su definisane na sledeći način:

- SPORT - ovoj klasi pridruženi su koncepti:
  ”takmicyenxe” (dogadxaj->drusxtveni_dogadxaj->takmicyenxe),
  ”igra” (cyin->aktivnost->igra) i
  ”sport” (cyin->aktivnost->rekreacija->sport);
- EKONOMIJA - ovoj klasi pridružen je koncept ”svojina” iz *Wordnet*-a. Uzimaju se u razmatranje svi koncepti iz drveta sa korenom ”svojina” filtrirani po domenima ”economy”, ”banking”, ”money”, ”commerce”;
- POLITIKA - ovoj klasi pridružen je koncept ”drusxtvena grupa” (grupa->drusxtvena_grupa). Uzimaju se u razmatranje svi koncepti filtrirani po domenu ”politics”.

Svakoj od ovih klasa pridružen je vektor odgovarajućih literala (osnova literala) iz *Wordnet*-a kao što je opisano u okviru ove tri stavke.

Pozivom programa convertor.py, autora Gorana Rakića sa Matemaičkog fakulteta u Beogradu, svi članci odnosno svi tekstovi iz Politike, prebačeni su u AURORA kodni zapis (š u sx, ć u cx i tako dalje) kako bi bili u skladu sa kodnim zapisom literala iz *Wordnet*-a.

Funkcija pripadnosti klasi

Funkcija pripadnosti klasi ili klasifikacioni model je funkcija koja vrši preslikavanje dokumenta u neku od unapred definisanih klasa. Kriterijum po kome se

vrši ovo preslikavanje, definiše se na osnovu podataka za trening, odnosno dokumenata za koje je unapred poznato kojoj klasi pripadaju. Ova funkcija treba biti definisana tako da se na osnovu nje može što efikasnije, odnosno sa što manjom greškom, izvršiti klasifikacija podataka za testiranje, odnosno dokumenata za koje je nepoznata klasa kojoj pripadaju.

Pri izvođenju eksperimenta posmatrani su samo članci sa rubrikama Ekonomija, Sport i Politika. U okviru svakog članka izvršeno je prebrojavanje pojava svih literala pridruženih posmatranim klasama kao i ukupan broj reči u članku odnosno njegovom tekstu. Pronađen je maksimum vrednosti odnosa broja pojava literala iz klase i ukupnog broja reči u tekstu (što se svodi na traženje maksimuma pojave literala iz klase). Tekst je zatim pridružen klasi za koju ima maksimum ove vrednosti. Dakle, funkcija pripadnosti klasi definisana je kao maksimum broja pojava literala pridruženih svakoj od klasa.

Formalno, to se može prikazati na sledeći način:

Neka je dat dokument D koji je potrebno dodeliti nekoj od definisanih klasa. Neka je

$$klSport(D) \;=\; brojLiterala(D,\ "takmicxenje",\ "igra",\ "sport")$$

odnosno broj koliko se puta literali iz drveta sa korenima u konceptima "takmicxenje", "igra" i "sport" javljaju u okviru dokumenta D,

$$klEkonomija(D) \;= brojLiterala(D,\ "svojina"$$

$$domeni\ "economy",\ "banking",\ "money",\ "commerce")$$

odnosno broj koliko se puta literali iz drveta sa korenom u konceptu "svojina" filtrirani po domenima "economy", "banking", "money", "commerce" javljaju u okviru dokumenta D i

$$klPolitika(D) \;=\; brojLiterala(D,\ "drusxtvena\ grupa"\ domen\ "politics")$$

odnosno broj koliko se puta literali iz drveta sa korenom u konceptu "drusxtvena grupa" filtrirani po domenu "politics" javljaju u okviru dokumenta D.

Funkcija pripadnosti se u tom slučaju može definisati na sledeći način:

$$max(klSport(D), klEkonomija(D), klPolitika(D)) \;=$$

$$= \begin{cases} klSport(D) & \text{D} \in \text{Sport} \\ klEkonomija(D) & \text{D} \in \text{Ekonomija} \\ klPolitika(D) & \text{D} \in \text{Politika} \end{cases}$$

Dobijeni rezultati

Rezultati opisane metode klasifikacije su prikazani u tabelama 7.1, 7.2 i 7.3.

U tabeli 7.4 date su vrednosti mera Preciznost, Odziv i F-mere za svaku od posmatranih klasa. Ove mere ukazuju na veoma dobar kvalitet dobijene klasifikacije.

Ono što se iz ovih tabela može videti jeste da se najbolji rezultati dobijaju za klasu *Sport*, zatim za klasu *Politika* a najlošiji za klasu *Ekonomija*. Razlog tome je manji broj literala pridruženih klasi *Sport* koji su specifični za oblast sporta i nemaju tendenciju mešanja sa oblastima druge dve klase. Zbog toga je

SPORT	2003	2004	2005	2006	Ukupno
Stvarno pozitivni	29145 (84.3%)	29645 (85.5%)	31074 (84.5%)	34350 (85%)	124214 (84.8%)
Lažno pozitivni	291	373	541	397	1602
Lažno negativni	5440	5042	5685	6063	22230
Stvarno negativni	31510	32529	39718	38335	142092
Ukupan broj članaka koji pripadaju klasi Sport	34585	34687	36759	40413	146444
Ukupan broj svih članaka (test podaci)	66386	67589	77018	79145	290138

Tabela 7.1: Rezultat klasifikacije za klasu Sport.

EKONOMIJA	2003	2004	2005	2006	Ukupno
Stvarno pozitivni	6679 (73.3%)	8951 (73.3%)	10127 (74.8%)	11407 (76.3%)	37164 (74.6%)
Lažno pozitivni	4913	4840	6687	6128	22568
Lažno negativni	2427	3266	3403	3545	12641
Stvarno negativni	52367	50532	56801	58065	217765
Ukupan broj članaka koji pripadaju klasi Ekonomija	9106	12217	13530	14952	49805
Ukupan broj svih članaka (test podaci)	66386	67589	77018	79145	290138

Tabela 7.2: Rezultat klasifikacije za klasu Ekonomija.

za klasu *Sport* procenat stvarno pozitivnih dokumenata najveći ali je najveći i procenat lažno negativnih. Ako bi se klasi *Sport* proširio skup pridruženih mu literala, ovaj procenat lažno negativnih bi se popravio ali opšti rezultat bi se pogoršao. Klasi *Politika* je takođe pridružen manji broj literala usko vezanih za oblast politike dok je klasi *Ekonomija* pridružen veći broj literala. Ove dve oblasti su po terminima koje koriste dosta sličnije u odnosu na oblast sporta pa su zbog toga i rezultati nešto lošiji.

Svi literali potrebni za realizaciju opisane metode, dobijeni su korišćenjem XQuery funkcija definisanih u tu svrhu.

7.3.2 Klasifikacija zasnovana na najproduktivnijim konceptima

Umesto da se za klasifikaciju uzimaju predefinisane ključne reči pridružene klasi, čini se pokušaj da se klasa okarakteriše literalima iz hijerarhije sinsetova sa korenom u najproduktivnijem konceptu. Eksperiment vezan za ovu vrstu klasifikacije uključio je dve klase: *Sport* i *Društvo*. Njima su pridruženi najproduktivniji koncepti hijerarhija sa korenom "akt" i "dogadxaj" a to su "aktivnost" i "grupna akcija", redom. Klasi *Sport* pridruženi su literali iz drveta sa korenom u konceptu "aktivnost" filtrirani po domenima "sport", "play", "basketball" i "boxing" a klasi *Društvo* svi literali iz drveta sa korenom u konceptu "grupna

POLITIKA	2003	2004	2005	2006	Ukupno
Stvarno pozitivni	19400 (85.5%)	17364 (84%)	21678 (81.1%)	19838 (83.4%)	78280 (83.4%)
Lažno pozitivni	5455	5930	6396	6465	24246
Lažno negativni	3295	3321	5051	3942	15609
Stvarno negativni	38236	40974	43893	48900	172003
Ukupan broj članaka koji pripadaju klasi Politika	22695	20685	26729	23780	93889
Ukupan broj svih članaka (test podaci)	66386	67589	77018	79145	290138

Tabela 7.3: Rezultat klasifikacije za klasu Politika.

	Preciznost	Odziv	F-mera
Sport	98.73%	84.82%	91.25%
Ekonomija	62.22%	74.62%	67.86%
Politika	76.35%	83.38%	79.71%

Tabela 7.4: Mere kvaliteta klasifikacije za klase Sprot, Ekonomija i Politika.

akcija".

Formalno, neka je za dati dokument D

$$klSport(D) \;=\; brojLiterala(D,\ "aktivnost"$$

$$domeni\ "sport",\ "play",\ "basketball"\ i\ "boxing")$$

broj koliko se puta literali iz drveta sa korenom u konceptu "aktivnost" filtrirani po domenima "sport", "play", "basketball" i "boxing" javljaju u okviru dokumenta D i

$$klDrustvo(D) \;=\; brojLiterala(D,\ "grupna\ akcija")$$

odnosno broj koliko se puta literali iz drveta sa korenom u konceptu "grupna akcija" javljaju u okviru dokumenta D.

Funkcija pripadnosti se može definisati na sledeći način:

$$max(klSport(D), klDrustvo(D)) = \begin{cases} klSport(D) & \text{D} \in \text{Sport} \\ klDrustvo(D) & \text{D} \in \text{Drustvo} \end{cases}$$

Dobijeni rezultati prikazani su u tabelama 7.5 i 7.6:

U tabeli 7.7 date su vrednosti mera Preciznost, Odziv i F-mere koje odražavaju kvaltitet obavljene klasifikacije.

Iz dosadašnje analize, može se zapaziti da se klase koje su odabrane u cilju klasifikacije dosta razlikuju po tematici što je povoljna okolnost koja povećava kvalitet klasifikacije. Rezultati dobijeni za klasu *Sport* bolji su od onih za klasu *Društvo*. Razlog je u terminologiji koja je dosta specifičnija za oblast sporta u odnosu na društvo.

Ono što ne ide u prilog ovom pristupu, jeste odabir tekstova za klasifikaciju. To su rubrike iz dnevnog lista Politika, koje odlikuje prilično jasno i koncizno

DRUŠTVO	2003	2004	2005	2006	Ukupno
Stvarno pozitivni	7904 (72%)	8374 (71.23%)	8857 (70.49%)	9153 (71.9%)	34288 (71.4%)
Lažno pozitivni	5233	5269	5513	6015	22030
Lažno negativni	3069	3382	3708	3575	13734
Stvarno negativni	29352	29418	31246	34398	124414
Ukupan broj članaka koji pripadaju klasi Društvo	10973	11756	12565	12728	48022
Ukupan broj svih članaka (test podaci)	45558	46443	49324	53141	194466

Tabela 7.5: Rezultat klasifikacije za klasu Društvo (zasnovane na najproduktivnijem konceptu).

SPORT	2003	2004	2005	2006	Ukupno
Stvarno pozitivni	29352 (84.9%)	29418 (84.8%)	31246 (85%)	34398 (85.1%)	124414 (71.4%)
Lažno pozitivni	3069	3382	3708	3575	13734
Lažno negativni	5233	5269	5513	6015	22030
Stvarno negativni	7904	8374	8857	9153	34288
Ukupan broj članaka koji pripadaju klasi Sport	34585	34687	36759	40413	146444
Ukupan broj svih članaka (test podaci)	45558	46443	49324	53141	194466

Tabela 7.6: Rezultat klasifikacije za klasu Sport (zasnovane na najproduktivnijem konceptu).

izražavanje, stavljanje akcenta na činjenice i korišćenje ne tako bogate terminologije. Ovaj pristup dao bi bolje rezultate nad tekstovima bogatijeg vokabulara, na primer nad nekim književnim delima ili pesmama ili tekstovima u okviru kojih se koriste manje standardni termini.

Klasifikacija se može obaviti ne samo opisanim metodama već i na neki drugi način. Na primer, klasama se mogu pridružiti ne svi literali koji pripadaju drvetu sa korenom u zadatom konceptu već samo odabrani po nekom kriterijumu, na primer samo vlastita imena. Takođe može se izvršiti takozvana zamućena (engl. fuzzy) klasifikacija ili klasifikacija sa određenim stepenom značajnosti odnosno može se definisati funkcija pripadnosti tako da jedan tekst može pripadati različitim klasama sa različitim težinama (verodostojnostima). Mera pripadnosti može u okviru sebe da uključi i odnos broja različitih literala i ukupnog broja literala u drvetu i tako dalje.

Sve ovo su teme koje će biti predmet daljeg rada na problemu klasifikacije dokumenata.

	Preciznost	Odziv	F-mera
Društvo	60.88%	71.4%	65.72%
Sport	90.06%	84.96%	87.44%

Tabela 7.7: Mere kvaliteta klasifikacije za klase Društvo i Sport (zasnovane na najproduktivnijem konceptu).

8

Zaključak i dalji rad

Buran razvoj računarstva, pre svega računarske tehnologije, na početku 21. veka, dao je veliki doprinos razvoju sopstvenih oblasti, pre svega oblasti baza podataka — istraživanju novih domena primene i prilagođavanje tim domenima. Razvijene su prostorne, vremenske i tekstualne a pre svega XML baze podataka. U domenu obrade prirodnog jezika, razvoj računarske tehnologije dao je izuzetan doprinos nizu disciplina kao što su korpusna lingvistika, elektronska leksikografija, razvoj alata za obradu jezičkih resursa, automatsko prevođenje, semantički veb, prepoznavanje i razumevanje teksta i govora, pretraživanje velikih kolekcija dokumenata, ekstrakcija informacija i tako dalje [31].

Navedimo osnovne rezultate do kojih je dovelo istraživanje prikazano u ovom radu:

- Pokazano je kako korišćenje XML tehnologija i XML baza podataka može da doprinese efikasnosti realizacije jednostavnih operacija nad raznorodnim leksičkim resursima;
- Na primeru *Wordnet*-a, pokazano je kako se korišćenjem izvornih XML baza podataka, na jednostavan i efikasan način mogu dobiti različite vrste informacija iz leksičkih resursa;
- Definisane su mere "produktivnosti" koncepta u cilju formalizacije pojma "bazni koncepti";
- Razvijena je biblioteka XQuery funkcija koja na elegantan način omogućava izračunavanje "najproduktivnijih koncepata" u *Wordnet*-u;
- Izvršena je klasifikacija članaka iz dnevnog lista Politika, koji su deo EBART korpusa, na jedan nov i originalan način:
  - pridruživanjem odabranih koncepata klasama
  - pridruživanjem odabranih najproduktivnijih koncepata klasama

  i merenjem frekvencije literala njihovih hijararhija u dokumentima koji se klasifikuju. Dokument se pridružuje onoj klasi za koju dostigne maksimum učestalosti pojave literala iz hijararhije pridruženog joj koncepta.

Ovo istraživanje otvorilo je mnoga druga pitanja. Posebna pažnja će se posvetiti sledećim zadacima:

- Dovršavanje procesa analize *Wordnet*-a. Umesto najproduktivnijeg koncepta možemo uzeti u razmatranje sve koncepte čija se vrednost definisane mere produktivnosti nalazi u nekom opsegu oko maksimalne vrednosti;
- Planiramo poređenje dobijene klasifikacione šeme sa postojećim klasifikacionim sistemima kao na primer EAGLES [45];
- Primena klasifikacionog postupka nad nekim korpusom literalnih tekstova;
- Proces izdvajanja informacija zasnovan na ovako definisanoj klasifikaciji.

Literatura

[1] Ronald Bourret, 1999. XML and Databases. Available at: http://www.rpbourret.com/xml/XMLAndDatabases.htm

[2] E. Harold, 2005. Managing XML data: Native XML databases. Available at: http://www.ibm.com/developerworks/xml/library/x-mxd4.html.

[3] Dare Obasanjo, 2001. An Exploration of XML in Database Management System. Available at: http://www.25hoursaday.com/StoringAndQueryingXML.html.

[4] Gordana Pavlović-Lažetić, 2007. Native XML Databases vs. Relational Databases in dealing with XML documents. Kragujevac J. Math. 30 181—199.

[5] Gordana Pavlović-Lažetić. Web tehnologije: XML, XHTML. Available at: http://www.matf.bg.ac.yu/ gordana/webtehn/xhtmlXML.pdf.

[6] XML - Veći od HTML-a, manji od SGML-a Available at: http://tehnika.krstarica.com/l/programiranje/xml-veci-od-html-a-manji-od-sgml-a

[7] Scott W. Ambler, 1997. Mapping Objects To Relational Databases. Available at: http://www.ambysoft.com/mappingObjects.pdf.

[8] Aleksandar Stanimirović, Zoran Stanković, 2002. XML baze podataka. YUINFO 2002.

[9] Michael Champion, 2001. Storing XML in Databases. eAI Journal. Available at: http://www.eaijournal.com/PDF/StoringXMLChampion.pdf.

[10] Ronald Bourret, 2001. Tutorial: Mapping DTDs to Databases. Available at: http://www.xml.com/pub/a/2001/05/09/dtdtodbs.html.

[11] Tufis, D., Cristea, D., Stamou, S., 2004. Balkanet: Aims, Methods, Results and Perspectives. A general Overview. In (Tufis 2004a), pages 9-43.

[12] Vossen, P. (ed.), 1998. EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Dordrecht: Kluwer Academic Publishers.

[13] Jelena Tomašević, Gordana Pavlović-Lažetić, 2008. Productivity of concepts in Serbian Wordnet. Proceedings of the Sixth Language Technologies Conference, 2008.

[14] Vossen, P., 2004. Introduction to the Special Issue on the BalkaNet Project. In (Tufis 2004a).

[15] Fellbaum, C. (ed.), 1998. WordNet: An Electronic Lexical Database. The MIT Press.

[16] Courtois, B.; Silberztein, M. (eds), 1990. Dictionnaires electroniques, Langue francaise, 87, Paris, Larousse.

[17] Obradović Ivan, 2003. Application of Intex in Refinement and Validation of Serbian Wordnet. 6th Intex Workshop, 28.30th May, Sofia (2003).

[18] Krstev, C., Pavlović-Lažetić, G., Obradović, I., Vitas, D., 2003. Corpora Issues in Validation of Serbian Wordnet. Matouek, V., Mautner, P. (eds.), Text, Speech and Dialogue, LNAI 2807, Springer, 132-137.

[19] G. Pavlović-Lažetić, 2006. Electronic Resources of Serbian: Serbian Wordnet. 36th International Slavic Conference, MSC, Belgrade, Serbia, september 2006.

[20] Krstev , C., May 2008. Cooperative work in further development of Serbian Wordnet. Infotheca, ISSN 1450-9687, No 1-2, Vol IX, pp 59a-78a.

[21] Courtois, B.; Silberztein, M. (eds), 1990. Dictionnaires electroniques. Langue francaise, 87, Paris, Larousse.

[22] C. Belleil O. Piton, D. Maurel, 1999. The prolex data base : Toponyms and gentiles for nlp. In NLDB'99, pages 233-237, 1999.

[23] Cvetana Krstev, decembar 2004. Specifični koncepti Balkana u semantičkoj mreži Wordnet. Zbornik radova "Susreti kultura", Novi Sad.

[24] Ranka Stanković, Cvetana Krstev, Duško Vitas, Ivan Obradović, Gordana Pavlović-Lažetić, 2004. Integrisanje heterogenih leksičkih resursa. Infofest, 2004.

[25] Duško Vitas, Cvetana Krstev, Ivan Obradović, Ljubomir Popović, Gordana Pavlović-Lažetić, 2003. An Overview of Resources and Basic Tools for the Processing of Serbian Written Texts. Available at: http://www.matf.bg.ac.yu/ cvetana/biblio/Solun03MATF.pdf.

[26] Vitas Duško, Cvetana Krstev, Ivan Obradović, Ljubomir Popović, Gordana Pavlović-Lažetić. Resources and Basic Tools for the Processing of Serbian Written Texts. Proc. of the Workshop on Balkan Language Resources, 1st Balkan Conference in Informatics.

[27] Beatrice Bouchou, Mickael Tran and Denis Maurel, 2005. Towards an XML Representation of Proper Names and Their Relationships. Springer. Available at: http://www.springerlink.com/content/n1bpebxc3rhrp5vh.

[28] Vitas, D. et al., 2003. Processing Serbian Written Texts: An Overview of Resources and Basic Tools. In: Piperidis, S./Karakaletsis, V.(Hg): Proceedings of the International Workshop on Balkan Language Resources and Tools. Thessaloniki, S. 97104.

[29] Krstev C. et al., 2006. WS4LR: A Workstation for Lexical Resources. In: Proceedings of the 5th Internationa Resources and Evaluation, LREC 2006. Genoa, May 2006. S. 16921697.

[30] Ranka Stanković, Ivan Obradović, 2007. Integracija heterogenih tekstualnih resursa. Available at: http://www.rgf.bg.ac.yu/LicnePrezentacije/ivan_obradovic/Radovi/GRAZ_2007b.pdf.

[31] Dr Gordana Pavlović-Lažetić, 2004. Računarstvo u nauci i obrazovanju na početku 21. veka. časopis Nastava, 2004.

[32] P. Resnik, 1995. Disambiguating noun grouping with respect to wordnet senses. In Proceedings of 3rd Workshop on Very Large Corpora.

[33] F. Segond, A. Schiller, G. Grefenstette, and J. Chanod., 1997. An experiment in semantic tagging using hidden markov model tagging. In Proceedings of the Workshop in Automatic Information Extraction and Building of Lexical Semantic Resources, pages 78-81.

[34] J.M. Gomez-Hidalgo and M.B. Rodriguez., 1997. Integrating a lexical database and a training collection for text categorization. In Proceedings of the Workshop in Automatic Information Extraction and Building of Lexical Semantic Resources, pages 39-44.

[35] J.Y. Chai and A. Biermann., 1997. The use of lexical semantics in information extraction. In Proceedings of the Workshop in Automatic Information Extraction and Building of Lexical Semantic Resources, pages 61-70.

[36] Richardson, R. and A.F. Smeaton, 1995. Using Wordnet in a Knowledge-Based Approach to Information Retrieval. Working Paper, CA-0395, School of Computer Applications, Dublin City University, Ireland.

[37] Istraživanje podataka, Matematički fakultet, Beograd. dostupno na sajtu www.matf.bg.ac.yu/ nenad/ip.

[38] Scott, Sam and Matwin, Stan., 1998. Text Classifcation Using WordNet Hypernyms. Usage of WordNet in Natural Language Processing Systems: Proceedings of the Workshop. 45-51.

[39] Rosso, P., Ferretti, E., Jimenez, D., Vidal, V., 2004. Text Categorization and Information Retrieval Using WordNet Senses. CICLing 2004, Lecture Notes in Computer Science, Vol. 2945. Springer-Verlag, 2004.

[40] Manuel de Buenaga Rodriguez, Jose Maria Gomez-Hidalgo and Belen Diaz-Agudo, 1996. Using WordNet to Complement Training Information in Text Categorization. Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, Bulgaria, September, 1996.

[41] W. Peters, 2001. Lexical Resources. In: NLP group Department of Computer Science, University of Sheffield, http://phobos.cs.unibuc.ro/roric/lex_introduction.html, 2001.

[42] http://www.memodata.com/2004/en/alexandria/

[43] P.N. Tan, M. Steinbach, V. Kumar, 2006. Introduction to Data Mining. Perason Education, 2006.

[44] Aleš Horak, Pavel Smrž, 2004. VisDic – WordNet Browsing and Editing Tool. Proceedings of GWC 2004 (http://www.fi.muni.cz/gwc2004/proc/94.pdf)

[45] Sinclair J, Ball J, 1996. EAGLES Preliminary Recommendations onText Typology. Available at: (http://www.ilc.pi.cnr.it/EAGLES96/texttyp/texttyp.html))

[46] Jelena Nikolić - citira dr Duška Vitasa, 2006. Informatički pogled na jezik. Politika, 30.01.2006., Available at: http://www.knjizara.com