Ekstrakcija gramatičkih podataka na primeru Wiktionary projekta

Andrej Zurovac

***Sadržaj* — Wiktionary je bogat izvor lingvističkog znanja i primer uspešne primene crowdsourcing modela. Znanje u Wiktionary-ju je slabo strukturirano, i da bi se omogućila dalja upotreba tog znanja potrebno je da se predstavi u strukturiranom obliku koji će moći automatski da se procesuira i pretražuje. Strukture semantičkog weba su posebno pogodne zbog razvijenih standarda namenjenih za povezivanje sa drugim semantičkim bazama znanja. Osnovna ekstrakcija Wiktionary-ja je već urađena u okviru DBpedia projekta. U ovom radu biće predstavljena ekstrakcija detaljnih gramatičkih podataka koji se dobijaju spajanjem nestrukturiranog sadržaja koji je smešten unutar različitih MediaWiki stranica u XML dump fajlu. Za primer će se uzeti konjugacije francuskih glagola, što je ujedno za sada jedan od malobrojnih gramatičkih sistema sa dovoljnom složenošću koji je obrađen na Wiktionary-ju. Glavni problem koji će biti rešen je analiza i obrada podskupa MediaWiki sistema šablona. Na osnovu tog rešenja biće generisane RDF trojke koje će u potpunosti pokrivati sve podatke iz domena koji se trenutno nalaze na Wiktionary-ju.**

***Ključne reči* — Crowdsourcing, semantički web, Wiktionary.**

I. Uvod

TEMA ovog rada je ekstrahovanje delimično strukturiranih podataka sa Wiktionary-ja u strukture bazirane na otvorenim standardima.

Wiktionary je jedan od projekata Wikimedia Foundation-a (neki od ostalih su Wikipedia, Wikisource, Wikivoyage, ...). Projekti Wikimedia Foundation-a su primeri projekata koji su bazirani na crowdsourcing-u. Crowdsourcing znači da znanje koje se prikuplja u ovim projektima dolazi od velike neformalne grupe ljudi, u ovom slučaju od velike online zajednice.

Andrej Zurovac, Računarski fakultet, Srbija (telefon: 381-11-2627613; faks: 381-11-2623287).

To je u suprotnosti sa načinima dobavljanja znanja koji se koriste u tradicionalnijim proizvodima koji bi bili ekvivalentni ovim projektima (npr. enciklopedije ili rečnici koje izdaje određena izdavačka kuća / kompanija).

Konkretan projekat iz koga će biti ekstrahovano znanje, Wiktionary, je ekvivalent rečniku sa bogatim obimom informacija (značenje, izgovor, etimologija, primeri upotrebe, prevodi na mnoštvo jezika, gramatički podaci poput tabela deklinacija i konjugacija...).

Znanje koje se nalazi u Wiktionary-ju je pretežno nestrukturirane prirode, a u ovom radu biće prikazana ekstrakcija takvog znanja i njegovo predstavljanje u otvorenim standardima semantičkog weba. Semantički web je upravo ideja da se resursima na webu dâ semantičko značenje i jasna struktura, kao i definisanje i promovisanje zajedničkih otvorenih standarda na kojima bi se zasnivao sâm semantički web, i što je još značajnije, koji će omogućiti primenu semantičkog weba, koja će počivati na osnovama tih definisanih otvorenih standarda.

Kako je ekstrahovanje osnovnih podataka sa Wiktionary-ja već obrađeno u drugim radovima i projektima (najpre na rudimentarnije načine, a kasnije i korišćenjem tehnologija semantičkog weba), ovaj rad će se baviti ekstrahovanjem detaljnijih gramatičkih podataka (koji su trenutno već dostupni u značajnom obimu za pojedine jezike) i kreiranjem semantičke baze znanja na osnovu tako ekstrahovanih podataka.

Podacima sa MediaWiki servera se ne može pristupiti putem web crawling-a, jer osim što bi oduzelo previše vremena / resursa, uz to je i izričito zabranjeno. Za potrebe projekata poput DBpedije i svih ostalih kojima je potreban sadržaj svih članaka koji se nalaze na MediaWiki serverima, dostupni su odgovarajući XML fajlovi, tzv. „dump“ fajlovi.

Unutar tih XML fajlova se nalaze članci koji su pisani u posebnom markup jeziku koji koristi MediaWiki server, tzv. Wikitext-u. Wikitext kao proširenje sadrži i funkcionalnost pod nazivom šabloni (templates), a koja služi da se deo teksta koji treba da se uključi na više strana definiše samo jednom i potom poziva. Šabloni se pozivaju sa opcionim parametrima i podržavaju složene kontrolne strukture, što znači da se mogu posmatrati kao funkcije i programski jezik specifičan za domen.

Rešenje koje će biti predstavljeno u radu je realizovano korišćenjem DBpedia Extraction Framework-a, projekta otvorenog koda pisanog u Scala programskom jeziku.

DBpedia Extraction Framework kao svoj sastavni deo ima ugrađeno

učitavanje članaka iz MediaWiki XML fajlova (mnogi od takvih fajlova su veličine i po više desetina gigabajta), kao i paralelizovanu obradu članaka i serijalizaciju dobijenih semantičkih struktura. Obrada članaka se vrši putem posebnih modula, tzv. ekstraktora, koji mogu biti kratki ili veoma komplikovani. Ovaj projekat će biti urađen implementacijom jednog takvog ekstraktora, kao i dodatnih rutina potrebnih za obradu struktura koje je potrebno parsirati / obraditi.

Ono što razlikuje ovaj projekat od tipičnog projekta baziranog na DBpedia Extraction Framework-u je to što se podaci koji su potrebni za ekstrakciju nalaze duboko u više ugnježđenih poziva šablonâ. To znači da će biti potrebno uraditi detaljnu analizu korišćenih šablonâ, kao i razviti rutine za njihovo parsiranje.

Od kada je prvi put lansiran, 2002, pa do danas, količina podataka u Wiktionary-ju stalno raste. Ovaj rast je izvanredan primer rezultata koje otvoreni crowdsourcing može da postigne.

Međutim, kao što je već rečeno, znanje u Wiktionary-ju je slabo strukturirano i sledeći korak koji mora da se napravi je transformacija takvog znanja u nešto što ima strukturu. Kada je struktura dobijena, može se krenuti u dalju primenu tog znanja. Na osnovu podataka iz Wiktionary-ja su pravljeni rečnici, alati za proveru pravopisa, otvoreni API-ji za druge programe...

Glavni razlog zašto su korisne baš semantičke strukture podataka je njihova mogućnost da se povezuju sa drugim (otvorenim) semantičkim bazama znanja koje su dostupne i time se višestruko uvećava vrednost i primenljivost znanja koje je prikupljeno.

Rezultat projekta koji je tema ovog rada nije samo ekstrahovana baza znanja koja se može odmah pretraživati i primenjivati. Značaj je i u tome što će pratiti konvencije već urađenih projekata iz domena, time biti povezana sa njima i automatski pružiti još veći obim i mogućnosti pretrage i iskorišćenja znanja, kako ovog do koga se došlo u ovom projektu, tako i onog znanja koje je već dostupno zahvaljujući ranijim projektima. Osim toga, zbog crowdsourced prirode Wiktionary-ja i njegovog daljeg rasta, urađeni kôd će moći da se izvršava nad budućim verzijama XML dump-ova i dobija uvećani set podataka. Takođe će urađene rutine moći da se prošire uz manji trud na druge slične podatke (ili domene) iz Wiktionary-ja (ili možda nekog drugog Wikimedia Foundation projekta) za ekstrakciju daljeg znanja.

Najpre će biti reči o postojećim pristupima problemu. Nakon toga biće dat kraći opis problema, dok će se posle toga obaviti analiza postavljenog problema, kao i prikazati rešenje i implementacija. Na kraju biće razmatrana trenutna pokrivenost gramatike srpskog jezika na Wiktionary-ju.

II. Poređenje postojećih pristupa za izvlačenje podataka iz Wiktionary-ja

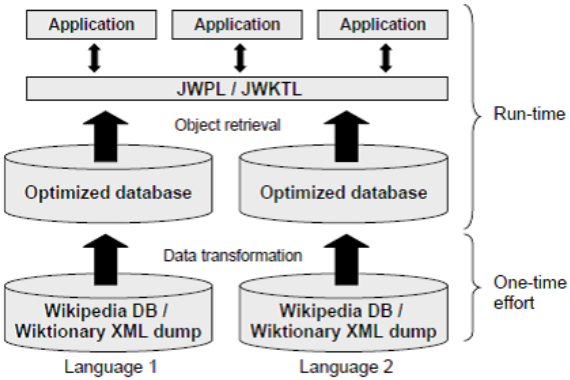
Značaj Wiktionary-ja kao izvora lingvističkih podataka je prepoznat i u prethodnim godinama je bilo više ideja i projekata koji su pokušavali da iskoriste Wiktionary kao izvor znanja i da to znanje učine dostupnim u obliku koji bi omogućio dalju upotrebu.

Uočeno je da iako je strana Wiktionary-ja (sa aspekta programskog jezika) samo jedan veliki string, u tom stringu postoji struktura koju običan čovek koji gleda Wiktionary stranu vidi, a koja se može programski detektovati i izvući iz tog stringa i potom predstaviti u strukturi na osnovu koje se mogu graditi drugi projekti koji će koristiti to konkretno znanje.

Važna stvar koja je primećena je i to da se Wiktionary stranica može posmatrati sa objektno-orijentisane tačke gledišta. Npr. članak, njegove sekcije i podsekcije mogu odgovarati klasama, postojanje podsekcije u sekciji označava postojanje relacije između odgovarajućih klasa koje predstavljaju dotičnu sekciju i podsekciju. To znači da se tekst Wiktionary stranice može predstaviti u objektno-orijentisanom obliku, i da se mogu kreirati instance objekata i popunjavati polja tih objekata na osnovu podataka iz Wiktionary-ja.[21]

Jedan od najznačajnijih projekata koji se bavio ovom problematikom je svakako JWKTL.

Nastao je u istraživačkoj laboratoriji Ubiquitous Knowledge Processing Lab (Technische Universität Darmstadt).



Sl. 1. Sistemska arhitektura JWPL i JWKTL projekata (preuzeto iz [23])

JWPL (Java Wikipedia Library) i JWKTL (Java-based Wiktionary Library) su biblioteke pisane u Java programskom jeziku koje izlažu API kojim se može pristupiti podacima koji se nalaze na Wikipedia i Wiktionary stranicama. Kao i ostala rešenja koja se bave ovom problematikom, kao izvor podataka koriste MediaWiki XML dump fajlove. Kao rešenje za problem performansi, XML dump fajlovi se inicijalno parsiraju i uvoze u lokalnu bazu podataka čija struktura je optimizovana za brzo izvršavanje JWLP/JWKTL API poziva. To znači da će svaki poziv JWLP/JWKTL API-ja zahtevati skoro konstantno vreme za izvršavanje upita ka bazi podataka.

API pozivi su objektno strukturirani.

Zbog problema različitih struktura stranica za različite jezičke verzije Wiktionary-ja, trenutno je JWKTL softver dostupan samo za englesku, nemačku i rusku verziju Wiktionary-ja.

JWKTL je korišćen i za neke druge projekte koji su nastali iz Ubiquitous Knowledge Processing Lab-a, npr. UBY (leksičko-semantički resurs za NLP, koji kombinuje podatke iz više izvora, kao što su WordNet, Wiktionary, Wikipedia...; trenutno dostupan samo za engleski i nemački), OntoWiktionary (ontologija bazirana na Wiktionary-ju).

Osim izrade novog programskog API-a, postoje i rešenja u kojima se podaci sa Wiktionary-ja ekstrahuju i smeštaju u neki od postojećih sistema za skladištenje.

Rađeni su primeri u kojima se struktura Wiktionary entry-ja transformiše u tabele i relacije u shemi relacione baze podataka. Time se znanje iz

Wiktionary-ja smešta u relacionu bazu podataka koju je moguće brzo pretraživati odgovarajućim SQL upitima. Drugim rečima, implicitna struktura (implicitna u smislu da je vidljiva ljudskom korisniku) se pretvara u eksplicitnu strukturu relacione baze podataka (eksplicitna u smislu da je struktura odmah izložena u mašinski razumljivom obliku).[21]

Shema relacione baze podataka je korisna struktura, ali mana je ograničeno povezivanje sa podacima koji postoje izvan nje. Ta mana se može prevazići korišćenjem semantičkih struktura podataka i Linked Open Data standardâ. Projekat koji ovo postiže je DBpedia Wiktionary projekat, čija je ideja da na osnovama već postojećeg DBpedia projekta napravi proširenje koje bi izložilo znanje iz Wiktionary-ja na sličan način kao što je DBpedia već uradila sa Wikipedia-om.

Svi ovi pristupi polaze od prethodno izložene ideje da se u stringu koji predstavlja sadržaj stranice može pronaći struktura koja se potom može analizirati kao sistem sa objektima i relacijama i nakon toga transformisati u neku drugu strukturu koja je korisnija (novi API, shema relacione baze podataka, strukture semantičkog weba).

S druge strane, predstavljene ideje se završavaju na analizi sadržaja stringa konkretne date stranice. Kao što će biti predstavljeno u nastavku rada, uvodi se ideja da se sadržaju stranice kod ekstrahovanja u semantičkom smislu mogu pripisati (i da prirodno pripadaju) i podaci koji se nalaze u stringovima na potpuno drugim stranicama, koje su međusobno povezane na odgovarajući način MediaWiki programskom sintaksom. Biće razrađen način ekstrahovanja takvih podataka i dopunjavanja postojećeg znanja novim podacima.

III. OPIS PROBLEMA

Cilj je od datog MediaWiki XML dump fajla projekta Wiktionary dobiti kao izlaz RDF trojke koje predstavljaju gramatičke podatke nekog jezika. Da bi bilo dovoljno podataka za ekstrahovanje, taj jezik ne sme biti previše analitičke tipologije (poput engleskog). Takođe mora biti dobro obrađen na Wiktionary-ju. Kao adekvatan izbor će se uzeti francuski glagoli, koji imaju obiman sistem konjugacije, kao i sistematsku obradu na francuskoj verziji Wiktionary-ja.

Francuski glagol tipično ima zasebnu stranicu koja prikazuje njegovu detaljnu konjugaciju. Takve stranice se nalaze u *Main* imenskom prostoru (namespace) i naslov im počinje sa *Annexe:Conjugaison en français/*.

Iako postoji opcija da se pokrene sopstvena instanca MediaWiki softvera i parsira HTML izlaz lokalnog servera (što se možda čini jednostavnijim za parsiranje), ta opcija bi bila suviše hardverski i vremenski zahtevna.

Što se resursa tiče, jeftinije je direktno uzeti podatke iz XML dump fajla i ekstrahovati iz njih potrebne podatke. Iako će se koristiti DBpedia Extraction Framework, biće od koristi samo za učitavanje i serijalizaciju.

Pošto stranice sa konjugacijama uglavnom ne sadrže običan Wikitext, već veliki broj ugnježđenih poziva šablonâ, DBpedia parser neće biti od koristi u ovom projektu, već biće neophodno napisati rutine koje će parsirati sve što bude bilo potrebno.

Tipična upotreba RDF trojki će se realizovati kroz njihovo pretraživanje korišćenjem SPARQL upita. Kako ekstrahovane RDF trojke nadograđuju postojeću DBpedia Wiktionary ontologiju, to znači da znanje koje može tako da se pretražuje nije ograničeno samo na trojke koje su ekstrahovane u ovom projektu, već obuhvata i celokupno znanje koje se nalazi u DBpedia Wiktionary-ju, kao i ostalim projektima koji bi se eventualno nadovezivali na DBpedia Wiktionary (a s obzirom na to da je tek zagrebana površina u odnosu na trenutne mogućnosti Wiktionary-ja, može se očekivati da će takvih projekata biti dosta).

Upotreba može biti u proizvodima koji bi koristili takvu bazu znanja u gotovim kreacijama (web sajtovi koji se bave temom iz domena, rečnici, drugi softverski alati i proizvodi...), naravno uz poštovanje Creative Commons Attribution + ShareAlike licence. Upotreba može biti i u istraživačke svrhe, prevashodno iz domena obrade prirodnih jezika (Natural Language Processing).

IV. Opis rešenja

Usled izuzetne složenosti MediaWiki sistema šablonâ, kao i dodatka Scribunto ekstenzije koja se koristi na stranicama koje će biti obrađene, skoro je nemoguće parsirati sve šablone na isti način na koji to radi MediWiki engine.

Ključna stvar koju treba uočiti da bi se uspešno rešio problem je to da se na stranicama koje treba analizirati koristi samo podskup ukupnih mogućnosti MediaWiki sistema šablonâ. Takođe će pomoći i ako se uoči da se izvestan deo tog podskupa koristi samo da se unapredi formatiranje strane ili da se prikaže odgovarajući link (npr. u zavisnosti od kategorije glagola), što nisu podaci u smislu ekstrakcije koja se radi. Interesantni su samo

semantički podaci u smislu konkretnog oblika glagola u određenom načinu, vremenu, licu i broju, kao i njihov izgovor u međunarodnom fonetskom alfabetu. Sve to će pomoći da se parser svede na minimum koji je dovoljan za precizno izvlačenje svih potrebnih podataka, a bez komplikovanja stvari iznad nivoa neophodnog za rešavanje problema.

Za određivanje tog minimuma biće potrebna detaljna analiza svih stranica, šablonâ i Lua skriptova koji se pozivaju za prikazivanje tih stranica. Zbog raznolikosti i nepravilnosti francuske konjugacije, ti šabloni i pozivi su komplikovani, ali ipak *konačni* i *strukturirani*.

Stranice koje će se obrađivati su tipično veoma kratke, i sastoje se najčešće od jednog poziva šablona. Taj šablon potom poziva druge šablone, itd....

Postoje šabloni koji su specijalizovani za određene konjugacije ili tipove glagola, potom šabloni specijalizovani za jednostavna ili složena vremena. Šabloni takođe uzimaju veliki broj parametara (npr. da li je glagol povratan, da li glagol počinje fonetskim vokalom, da li je bezličan, koji se pomoćni glagol koristi za građenje složenih vremena, da li je defektivan, ...).

Primera radi, stranica za glagol *finir* sastoji se samo od sledećeg poziva šablona:

{{fr-conj-2|fin|pron=fi|pc=n}}

Kao što se vidi, iako se stranica za glagol finir sastoji od samo jednog poziva šablona, parser će morati da ga zameni glomaznim kodom tempate-a *Modèle:fr-conj-2*, da izvrši supstituciju svih parametara koji su mu prosleđeni, potom da obradi kontrolne strukture poput *#if* (u nekim drugim šablonima će morati da se obrade i druge kontrolne strukture). Takođe će morati rekurzivno da se pozovu svi ugnježđeni šabloni. Konkretno, kao što se može videti inspekcijom koda ovog šablona, najvažniji ugnježđeni šablon koji *Modèle:fr-conj-2* poziva je *fr-conj*.

Osim ovih navedenih, postoji još šablona koji se pozivaju i koji će morati da budu obrađeni da bi iz njih mogli da se izvuku precizni semantički podaci koji su potrebni.

Posebna kategorija šablonâ su šabloni koji pozivaju Lua skriptove. Analizom njihovog koda i upotrebe se vidi da se u slučajevima koji su od interesa za zadato ekstrahovanje, pozivaju samo kod generisanja IPA transkripcije, i to isključivo da bi se generisao link koji vodi ka objašnjenju simbola za relevantan jezik i da bi se pružilo bolje formatiranje.

Analizom koda i upotrebe tih šablonâ vidi se da rutine za formatiranje i

generisanje hiperlinkova nisu potrebne za ekstrahovanje semantičkih podataka i da se ceo šablon može zameniti jednostavnijim kodom (koji se dobija na osnovu analize).

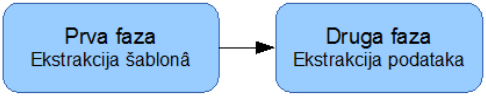
Na nekoliko mesta gde se uočava ovakav obrazac, može se ovo uraditi i daleko uprostiti dalje parsiranje, npr. zamenom šablona *pron-brut* sa *{{{1|—}}}*.

A. *Opis implementacije*

Rad je implementiran kao projekat WiktionaryGrammar u Scala programskom jeziku, korišćenjem DBpedia Extraction Framework-a, kompajliran tako da se izvršava na Java virtuelnoj mašini.

Glavni deo samog WiktionaryGrammar projekta je implementacija *WiktionaryGrammarExtractor* klase, koja je jedna konkretna implementacija karakteristike (trait) *Extractor*, kao i dodatnih rutina koje će ova klasa pozivati.

WiktionaryGrammar se sastoji od dve faze izvršavanja. Prva faza je ekstrakcija svih šablonâ, a druga faza je ekstrakcija gramatičkih podataka. To znači da će Framework proći kroz XML dump dva puta. To je neophodno zato što druga faza (čiji je glavni zadatak, s čisto tehničke strane, obrada šablonâ) mora imati sve šablone ekstrahovane i spremne pre nego što započne sa radom.



Sl. 2. Faze projekta

Drugi motiv za omogućavanje striktne razdvojenosti faza je i to što je druga faza namenjena da se izvršava svaki put kada izađe novi XML dump, dok je prvu fazu potrebno izvršavati retko (zato što se šabloni koji su potrebni projektu retko menjaju). Stoga će prva faza svoj rezultat serijalizovati za naknadnu upotrebu, čime se štede resursi.

Cilj prve faze je prikupljanje i ekstrakcija svih šablonâ, radi upotrebe u drugoj fazi. Za ovo nije potrebno koristiti instancu *Extractor* karakteristike. Prema MediaWiki dokumentaciji, postoji mogućnost da se i obične stranice tretiraju kao šabloni, ako se ispred njihovog naziva stave dve tačke (:). Analizom upotrebe šablonâ na stranicama koje će WiktionaryGrammar procesuirati, vidi se da se ta mogućnost koristi, tako da je neophodno i u

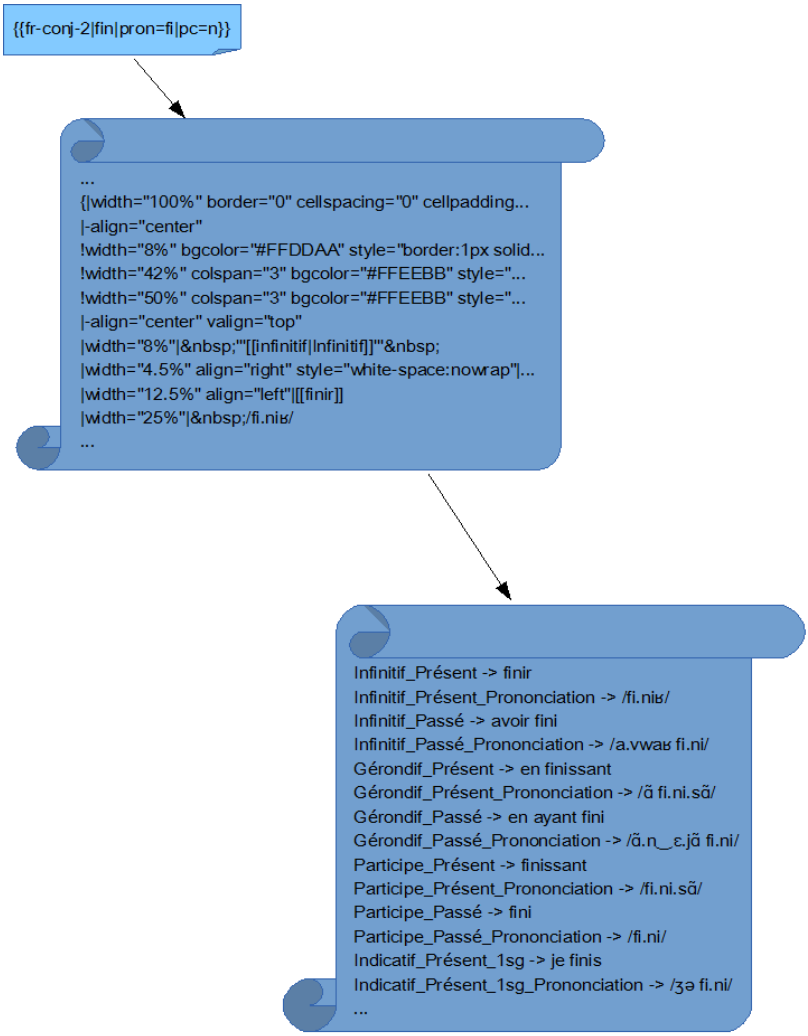
ovoj fazi procesuirati stranice iz *Main* imenskog prostora.

Na kraju prve faze dobijene su serijalizovane mape sa sadržajem svih šablonâ. Prvu fazu nije potrebno ponovo izvršavati, sve dok se struktura i način obrade šablonâ u gramatičkom delu Wiktionary-ja ne izmeni (a ovo se dešava jako retko).

Cilj druge faze je ekstrakcija podataka iz MediaWiki strana. Započinje se učitavanjem prethodno serijalizovanih struktura šablonâ iz prve faze. Procesuiraju se sve strane iz *Main* imenskog prostora koje počinju sa *Annexe:Conjugaison en français/*.

Svaka od procesuiranih strana će se proslediti ekstraktoru. Kao što je već utvrđeno, sadržaj svih tih strana će biti relativno jednostavni pozivi šablonâ. Stoga će prvi korak biti prosleđivanje tih stringova rutinama za procesuiranje šablonâ koje su razvijene. Te rutine će morati da identifikuju delove šablonâ, odgovarajuće parametre, potom da dobave potrebne šablone, da izvrše odgovarajuću supstituciju parametara, identifikaciju i procesuiranje svih kontrolnih struktura, kao i rekurzivnu obradu ugnježđenih šablonâ.

Nakon obrade šablonâ, sledeći korak je konačno ekstrahovanje svih gramatičkih podataka iz stringa i čuvanje u odgovarajućoj strukturi podataka. Pošto je konačni izlaz generisan iz malog broja šablonâ, poznata je unapred tačna struktura prečišćenog stringa, u smislu kako tačno treba napisati rutine koje će ekstrahovati sve željene podatke. Ovde se mora voditi računa i da nešto drugačije strukture važe za pojedine glagole, npr. bezlični glagoli neće imati sva lica, defektni glagoli neće imati sva vremena, itd... Nakon ovog koraka, podaci su ekstrahovani i nalaze se u mapi, koja je spremna za dalju upotrebu.



Sl. 3. Primer obrade glagola *finir*

Budući da je cilj skladištenje u strukture semantičkog weba, kao i činjenicu da se radi unutar DBpedia Extraction Framework-a, dobijena mapa treba da se pretvori u sekvencu *Quad*-ova, koja će potom biti serijalizovana kao RDF.

V. GRAMATIČKI PODACI SRPSKOG JEZIKA I WIKTIONARY

Zanimljivo bi bilo razmotriti kakvo je stanje sa gramatičkim podacima srpskog jezika na Wiktionary-ju i da li bi se sličan metod mogao primeniti za ekstrahovanje gramatičkih podataka srpskog jezika.

Srpska verzija Wiktionary-ja je još uvek izuzetno slabo razvijena, sa svega oko 16.000 unosa na svim jezicima ukupno, i bez razrađenog sistema šablonâ za unošenje gramatičkih podataka. Hrvatska verzija je tek nešto bolja sa ukupno 25.000 unosa na svim jezicima ukupno i sistemom šablonâ za gramatičke podatke koji je tek u povoju i nije dovoljno iskorišćen.

Kako svaka jezička verzija Wiktionary-ja sadrži podatke za puno različitih jezika, mogu se potražiti izvori podataka o srpskog gramatici i u drugim jezičkim verzijama. Trenutno najbolji izvor gramatičkih podataka za srpski jezik je, verovatno očekivano, engleska verzija Wiktonary-ja (koja inače ima preko 3.500.000 miliona unosa za sve jezike zajedno).

Engleska verzija ima razrađen sistem šablonâ za srpskohrvatske konjugacije i deklinacije. Sistem šablonâ je veoma jednostavan (u jakom kontrastu sa francuskim sistemom koji je obrađen u ovom radu) i sadrži relativno malo podataka. Ukupno se mogu naći konjugacije za oko 3.000 glagola i deklinacije za oko 10.000 imenica.

Šablon *sh-decl-noun*, koji se koristi za deklinaciju imenica je izuzetno jednostavan. Obrada ovog stringa ne bi zahtevala sistem razrađen u ovom projektu, već bi verovatno mogla da se uradi i unutar već postojećeg DBpedia Wiktionary projekta.

Šablon za konjugaciju glagola, *sh-conj*, je malo (ali minimalno) komplikovaniji, no ipak u istom rangu. Nema duboko ugnježđenih poziva šablonâ, i praktično ni traga razvijenoj sistematizaciji morfoloških oblika koja bi omogućila pozivanje šablonâ sa malim brojem parametara poput rešenja koje je na francuskom Wiktionary-ju napravljeno za francuski jezik, već se sve oslanja na dugačke pozive šablonâ, koji su jednostavni u strukturi, sa velikom listom parametara.

Za sada se čini da podataka o srpskoj gramatici nema dovoljno da bi Wiktionary bio vredan izvor znanja o srpskom jeziku. Potrebno je uneti veliku količinu podataka, ali i razraditi sistem koji bi omogućio jednostavno unošenje i održavanje tih podataka. Na primeru francuskog Wiktionary-ja se vidi da je takav sistem komplikovan za implementaciju, kao i za naknadnu

ekstrakciju; ali kao što ovaj rad pokazuje, ekstrakcija je moguća.

Francuski Wiktionary bi mogao biti dobar uzor u traženju ideje za razvijanje neke buduće verzije Wiktionary-ja koja bi adekvatno pokrila srpski jezik.

VI. ZAKLJUČAK

Ekstrakcijom iz Wiktionary-ja su dobijeni detaljni gramatički podaci za preko 23.000 francuskih glagola opisanih u preko 4.600.000 RDF iskaza, sa relativno jednostavnom ontologijom iz domena. Dobijena baza znanja koristi subjekte iz javno dostupnog DBpedia Wiktionary projekta, čime je povezana sa LOD oblakom.

Kako su korišćeni otvoreni standardi semantičkog weba, baza je u RDF-u i korišćenjem SPARQL-a se može pretraživati na bilo koji željeni način i snabdevati aplikacije znanjem iz baze. Trenutno se na Internetu ne mogu naći besplatne baze te razmere, a s obzirom da je baza izvučena u ovom projektu poduprta Wiktionary-jem, značaj je time veći.

Već sada je moguće na isti način napraviti baze znanja i za još neke druge jezike koji su pokriveni na Wiktionary-ju. Budućim evoluiranjem Wiktionary-ja može se očekivati da će se broj pokrivenih jezika povećavati, a takođe i dostupna količina informacija po jeziku.

Iako je često citirani problem kod ekstrakcije podataka iz Wikipedije i Wiktionary-ja upravo njihova nestrukturiranost, ovo je dobar primer gde je u relativnom nedostatku strukture pronađena izvesna strukturiranost u domenu koji je obrađivan. To je ograničilo problem na obradu samo podskupa mogućnosti MediaWiki sintakse, a bez tog pojednostavljenja problem ne bi bilo moguće rešiti na ovaj način, koji ima jasnu prednost u smislu brzine obrade.

ZAHVALNICA

Zahvaljujem se profesorki dr Snežani Popović na podršci i svim sugestijama, koje su bile veoma značajne i pomogle da ovaj rad bude mnogo bolji.

LITERATURA

[1] Grigoris Antoniou and Frank van Harmelen, A Semantic Web Primer, The MIT Press, Cambridge, Massachusetts, London, England, 2004

[2] K.K. Breitman, M.A. Casanova and W. Truszkowski, Semantic Web: Concepts, Technologies and Applications, Springer-Verlag London Limited 2007
[3] Jeffrey T. Pollock, Semantic Web for Dummies, Wiley Publishing, 2009
[4] Jorge Cardoso, Semantic Web Services: Theory, Tools, and Applications, IGI Global, 2007
[5] John G. Breslin • Alexandre Passant • Stefan Decker, The Social Semantic Web, Springer-Verlag Berlin Heidelberg, 2009
[6] W3C, RDF Primer. [Online]. http://www.w3.org/TR/rdf-primer/
[7] W3C, Terse RDF Triple Language. [Online]. http://www.w3.org/TeamSubmission/turtle/
[8] W3C, OWL 2 Primer. [Online]. http://www.w3.org/TR/owl2-primer/
[9] Wikipedia dokumentacija. [Online]. http://en.wikipedia.org/wiki/Help:Cheatsheet
[10] Wikipedia. [Online]. http://en.wikipedia.org/wiki/Wikipedia%3ANamespace
[11] Wikipedia dokumentacija. [Online]. http://en.wikipedia.org/wiki/Help:Template
[12] MediaWiki. [Online]. http://www.mediawiki.org/wiki/Help:Extension:ParserFunctions
[13] MediaWiki. [Online]. http://www.mediawiki.org/wiki/Extension:Scribunto
[14] Apache Jena. [Online]. http://jena.apache.org/tutorials/sparql.html
[15] W3C, SPARQL 1.1 Update. [Online]. http://www.w3.org/TR/sparql11-update/
[16] Wikipedia. [Online]. http://en.wikipedia.org/wiki/Semantic_web
[17] Tim Berners-Lee, W3C, Linked Data. [Online]. http://www.w3.org/DesignIssues/LinkedData.html
[18] Florian Bauer, Martin Kaltenböck, Linked Open Data: The Essentials, edition mono/monochrom, Vienna, Austria 2012
[19] Tom Heath and Christian Bizer, Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool, 2011
[20] Jonas Brekle, "Flexible RDF data extraction from Wiktionary", master teza, Universität Leipzig, Fakultät für Mathematik und Informatik, 2012
[21] Andrew Krizhanovsky, "Transformation of Wiktionary entry structure into tables and relations in a relational database schema", CORR – Computing Research Repository, vol. abs/1011.1, 2010
[22] Christof Müller, Iryna Gurevych, "Using Wikipedia and Wiktionary in Domain-Specific Information Retrieval", Proceedings of CLEF'08 Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access, pp. 219-226, 2008
[23] Torsten Zesch , Christof Müller , Iryna Gurevych, "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary", Proceedings of the Conference on Language Resources and Evaluation LREC, electronic proceedings, Ubiquitious Knowledge Processing, Universität Darmstadt, 2008
[24] The Linking Open Data cloud diagram. [Online]. http://lod-cloud.net/
[25] Wikimedia Foundation. [Online]. http://meta.wikimedia.org/wiki/Data_dumps
[26] DBpedia projekat. [Online]. http://dbpedia.org/About

ABSTRACT

Wiktionary is a rich source of linguistic knowledge and an example of a successful application of the crowdsourcing model. Knowledge in Wiktionary is only weakly structured, so in order to enable the use of that knowledge, it is necessary to represent it in a structured form which can be automatically searched and processed. Semantic web structures are especially suitable for this task because of the developed standards for interlinking different semantic web knowledge bases. Basic Wiktionary extraction has already been done as a part of DBpedia project. We present the extraction of detailed grammatical data which is obtained by merging unstructured content contained within different pages of the MediaWiki XML dump file. As an example, we'll process French verb conjugations, which is currently one of the few such examples of sufficient complexity found on Wiktionary. The main problem we will solve is analyzing and parsing a subset of the MediaWiki template system and its control structures. Based on that, we will generate RDF triples which will completely cover all domain data that is currently included in Wiktionary.

GRAMMATICAL DATA EXTRACTION FROM WIKTIONARY

Andrej Zurovac