UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Uroš Jurglič

SEMANTIČNO OZNAČEVANJE BESEDILA S POMOČJO POVEZANIH PODATKOV IN ODPRTOKODNE PROGRAMSKE OPREME

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Marko Bajec

Somentorica: doc. dr. Dunja Mladenić

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01569/2009
Datum: 05.04.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **UROŠ JURGLIČ**

Naslov: **SEMANTIČNO OZNAČEVANJE BESEDILA S POMOČJO POVEZANIH PODATKOV IN ODPRTOKODNE PROGRAMSKE OPREME**

SEMANTIC TEXT ANNOTATION WITH HELP OF LINKED OPEN DATA AND OPEN SOURCE SOFTWARE

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Storitve semantičnega označevanja omogočajo, da iz besedila preko omenjenih entitet odkrijemo pomen njegove vsebine. Zaradi tega postajajo vse bolj nepogrešljiv del implementacije naprednejših funkcionalnosti spletnih portalov ali sistemov za upravljanje z vsebinami, kot so npr. priporočilni sistemi ali semantični iskalniki.

V okviru diplomske naloge predstavite področje semantičnega označevanja besedila in ga umestite v širše področje semantičnega spleta. Nato preučite možnost lastne implementacije takšne storitve in predlagajte njeno arhitekturo. Izvedite testno implementacijo in jo primerjajte s podobnimi storitvami. Predstavite rezultate testiranja in njihovo interpretacijo. Proučite možnosti nadaljnega razvoja storitve.

Mentor:
prof. dr. Marko Bajec

Somentor:
doc. dr. Dunja Mladenić

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:
prof. dr. Franc Solina

Izjava o avtorstvu diplomskega dela

Spodaj podpisani **Uroš Jurglič**,

z vpisno številko **63010053**,

sem avtor diplomskega dela z naslovom:

Semantično označevanje besedila s pomočjo povezanih podatkov in odprtokodne programske opreme

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Bajca in somentorstvom doc. dr. Dunje Mladenić
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 1.7.2009 Podpis avtorja:

Zahvala

Za pomoč, podporo in potrpežljivost pri izdelavi diplomskega dela se zahvaljujem mentorju prof. dr. Marku Bajcu in somentorici doc. dr. Dunji Mladenić. Hkrati bi se za podporo zahvalil tudi moji družini, sodelavcem in sošolcem.

Vsebina

Seznam uporabljenih kratic in simbolov

ANNIE	*A Nearly New Information Extraction System,* komponenta orodja GATE, ki skrbi za prepoznanjavanje imenovanih entitet v besedilu.
API	*Application Programming Interface,* aplikacijski programski vmesnik.
GATE	*General Architecture for Text Engineering*, orodje za obdelavo naravnega jezika.
HTTP	*Hypertext Transfer Protocol,* aplikacijski protokol za prenos hiperteksta.
IE	*Information Extraction,* področje obdelave naravnega jezika, ki se ukvarja z avtomatskim izločanjem strukturiranih informacij iz besedila.
JSP	*Java Server Pages,* temeljna javanska tehnologija za dinamične spletne strani.
LGPL	*GNU Lesser General Public License*, licenca za uporabo odprtokodne programske opreme, opredeljena s strani organizacije Free Software Foundation (FSF).
LOD	*Linked Open Data*, prosti povezani podatki, projekt pod okriljem semantičnega spleta.
MUC	*Message Understanding Conference*, konferenca o razumevanju besedil, ki spada pod domeno izločanja informacij (IE).
NER	*Named Entity Recognition*, prepoznavanje imenovanih entitet.
NLP	*Natural Language Processing*, obdelava naravnega jezika.
POS	Part of Speech, *stavčni člen oz. vloga besede v stavku.*
RDF	*Resource Description Framework*, jezik za opis razmerij med viri glede na vnaprej opredeljeno ontologijo s poudarkom na ponovni rabi ontologije.
REST	*Representational State Transfer*, programski arhitekturni slog za distribuirane hipermedijske sisteme, npr. svetovni splet.

SPARQL	*SPARQL Protocol and RDF Query Language*, jezik in protokol za izvajanje poizvedb nad podatkovno bazo RDF.
TF-IDF	*Term Frequency – Inverse Document Frequency*, statistična mera, ki se uporablja v domeni izločanja informacij za ugotavljanje pomembnosti besede v dokumentu ali korpusu dokumentov.
URI	*Unified Resource Identifier*, enotni označevalnik vira, ki enolično določa nek vir na omrežju.
VSM	*Vector Space Model*, model vektorskega prostora.
YAGO	*Yet Another Great Ontology*, velika semantična podatkovna baza, zgrajena na podlagi informacij pridobljenih z Wikipedie.

Povzetek

Storitve semantičnega označevanja omogočajo, da iz besedila preko omenjenih entitet odkrijemo pomen njegove vsebine. Svoje delo opravljajo dobro, kvaliteta in uporabnost pa se neprestano izboljšuje. Zaradi navedenega postajajo vse bolj nepogrešljiv del implementacije naprednejših funkcionalnosti spletnih portalov ali sistemov za upravljanje z vsebinami, kot so npr. priporočilni sistemi ali iskalniki.

Namen diplomskega dela je implementacija storitve semantičnega označevanja angleškega besedila s pomočjo prostih povezanih podatkov in odprtokodne programske opreme, ki vrača zadovoljive rezultate in ob nadaljnem razvoju predstavlja realno alternativo komercialnim storitvam.

Diplomsko delo kratko predstavi področja semantičnega označevanja besedila, semantičnega spleta ter uporabe prostih povezanih podatkov. Opisani so uporabljeni programski in podatkovni gradniki lastne implementacije storitve semantičnega označevanja. Predstavljene so predlagana arhitektura in implementacijske podrobnosti realizacije. Opravljena je preprosta primerjava lastne implementacije storitve z dvema komercialnima storitvama, njeni rezultati pa so grafično predstavljeni in interpretirani. Delo zaokroža poglavje o možnostih nadaljnega razvoja storitve in sklepne ugotovitve o izkušnjah razvoja lastne storitve ter o prihodnosti storitev semantičnega označevanja.

Ključne besede

semantično označevanje besedila, prepoznavanje imenovanih entitet, semantični splet, povezani prosti podatki, obdelava naravnega jezika, tekstovno rudarjenje

Abstract

Services for semantic text annotation enable developers to extract meaning through mentioned named entities. They provide good results and their quality and usability is constantly improving. Therefore these services are becoming a frequently used component for implementations of advanced features in web sites or content management systems, e.g. recommendation systems or search engines.

The goal of this thesis is implementation of semantic annotation service of english text with help of linked open data and open source software. Service should provide good enough results and, in the case of further development, represent a viable alternative to commercial services.

This thesis briefly introduces domains of semantic text annotation, semantic web and linked open data. It describes software and data components of proprietary implementation of semantic text annotation service. It presents suggested text processing architecture and service implementation details. There is a simple comparison of proprietary service to two commercial services and their results are graphically shown and discussed. Further development options are presented. The thesis concludes with proprietary development experiences and a brief forecast about semantic text annotation services.

Keywords

semantic text annotation, named entity recognition, semantic web, linked open data, natural language processing, text mining

1 Uvod

1.1 Semantično označevanje besedila

Pod pojmom označevanje besedila si lahko predstavljamo kakršnokoli bogatenje besedila z oznakami in v splošnem metodo za posredovanje podatkov skupaj z besedilom. Semantično označevanje pa je označevanje z metapodatki, ki skušajo opredeliti pomen besedila, in se lahko avtomatizira s pomočjo metod analize besedila in obdelave naravnega jezika (NLP, Natural Language Processing). V okviru semantičnega označevanja besedila se skuša prepoznati imenovane ljudi, kraje, pojme – v splošnem entitete. Temu postopku pravimo prepoznavanje imenovanih entitet (NER, Named Entity Recognition). Entitete je potrebno v nadaljevanju še enolično identificirati. Slednje je pomembno, saj omogoča, da lahko oznake enega besedila primerjamo z oznakami drugega besedila. Tako lahko ugotavljamo, kako blizu sta si besedili po pomenu. Naslednja raven identifikacije pa je povezovanje identificiranih imenovanih entitet iz besedila z entitetami iz ontologije ali baze znanja, kot so npr. povezani odprti podatki (LOD, Linked Open Data).

Ker so označevanje besedila in njegovi rezultati za prevladujoče računalniške uporabnike razmeroma abstraktni, je na tej podlagi težko zgraditi zaokrožen končni izdelek. Zato je semantično označevanje besedila najuporabnejše za implementacije naprednih funkcionalnosti obstoječih izdelkov, ki zahtevajo vsaj delno razumevanje besedila. Nekateri primeri takih funkcionalnosti so naslednji:

- bogatenje vsebine člankov na spletu s pomočjo hiperpovezav
- priporočanje povezanih ali podobnih vsebin na spletnih portalih
- indeksiranje semantičnih oznak besedila za boljše iskanje po vsebinah
- pomoč pri prevajanju besedil z avtomatskim razpoznavanjem entitet
- hitre ali vnaprejšnje poizvedbe in prikaz podatkov o entitetah, ki so omenjene v člankih
- priprava semantično obogatenih vsebin v skladu s priporočili svetovnega spleta
- (delna) avtomatizacija procesa ustvarjanja in objavljana vsebin

1.2 Semantični splet in povezani podatki (Linked Data)

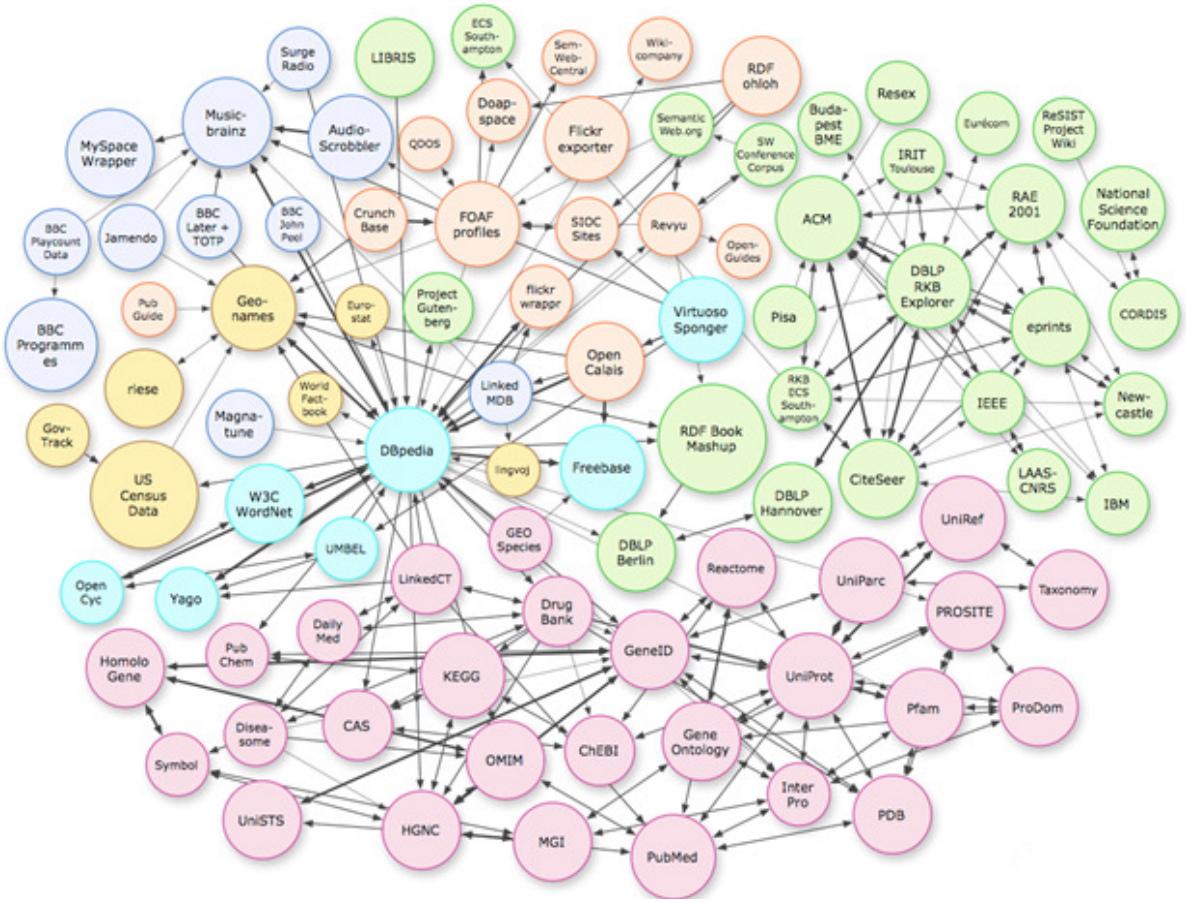
Semantični splet predstavlja evolucijo in razširitev svetovnega spleta, ki je prvotno namenjen ljudem. Ljudje običajno nimamo posebnih težav z razumevanjem besedila in multimedijskih

vsebin, ki se pojavljajo na spletu. Težava se pojavi šele pri vključevanju računalniških programskih agentov v svetovni splet. Takrat želimo, da nek program uporablja podatke s svetovnega spleta na podoben način, kot to počnemo ljudje. Ker podatki v obliki prostega besedila in multimedijskih vsebin niso dovolj strukturirani in informacijsko razčlenjeni, jih programski agenti največkrat ne morejo razumeti in koristno uporabiti. Zaradi navedenega si je pobudnik svetovnega spleta Tim Berners Lee zamislil rešitev - semantični splet, ki ga je predstavil v svojem enako imenovanem prispevku [1].

V semantičnem spletu so podatki in storitve opredeljeni precej bogatejše kot v navadnem svetovnem spletu. Pravimo, da so opredeljeni s semantiko ali pomenom. To omogoča, da lahko podatke na semantičnem spletu koristno uporabljajo ne samo ljudje, temveč tudi računalniški programi. Avtor te zamisli si zato predstavlja semantični splet kot uporabniku prijaznejšo različico svetovnega spleta, kjer lahko manj zaželena rutinska opravila namesto uporabnika opravijo kar računalniški programski agenti. Med tovrstna opravila spadajo iskanje po spletu in povezovanje informacij ali dokumentov, lociranje storitev, ipd. [1].

Kljub temu, da je vizija semantičnega spleta sedaj že precej stara, pa je njegova uresničitev še vedno omejena na razmeroma redke in izolirane predele svetovnega spleta, ki idejo o semantično obogatenih dokumentih implementirajo bolj ali manj celovito. Zaradi počasnega širjenja idej semantičnega spleta se je rodilo spoznanje, da se le-ta ne bo zgodil naenkrat, ampak se bo udejanjil postopoma. Prvi korak predstavlja povezovanje podatkov med različnimi odprtimi podatkovnimi bazami, kar nas pripelje do povezanih prostih podatkov (LOD).

Povezani podatki predstavljajo posebno in hkrati eno najuspešnejših področij semantičnega spleta. To področje spodbuja in opredeljuje metode izpostavljanja, deljenja in povezovanja podatkov v strukturirani obliki kot jo predvideva semantični splet. Za povezovanje entitet iz drugih baz se uporabljajo t.i. dereferenciabilni enolični identifikatorji URI. Slednje omogoča, da s pomočjo enoličnega identifikatorja entitete preko osnovnega spletnega protokola HTTP pridemo do strukturiranega opisa entitete, ki ga razumejo računalniški programski agenti [2][3].



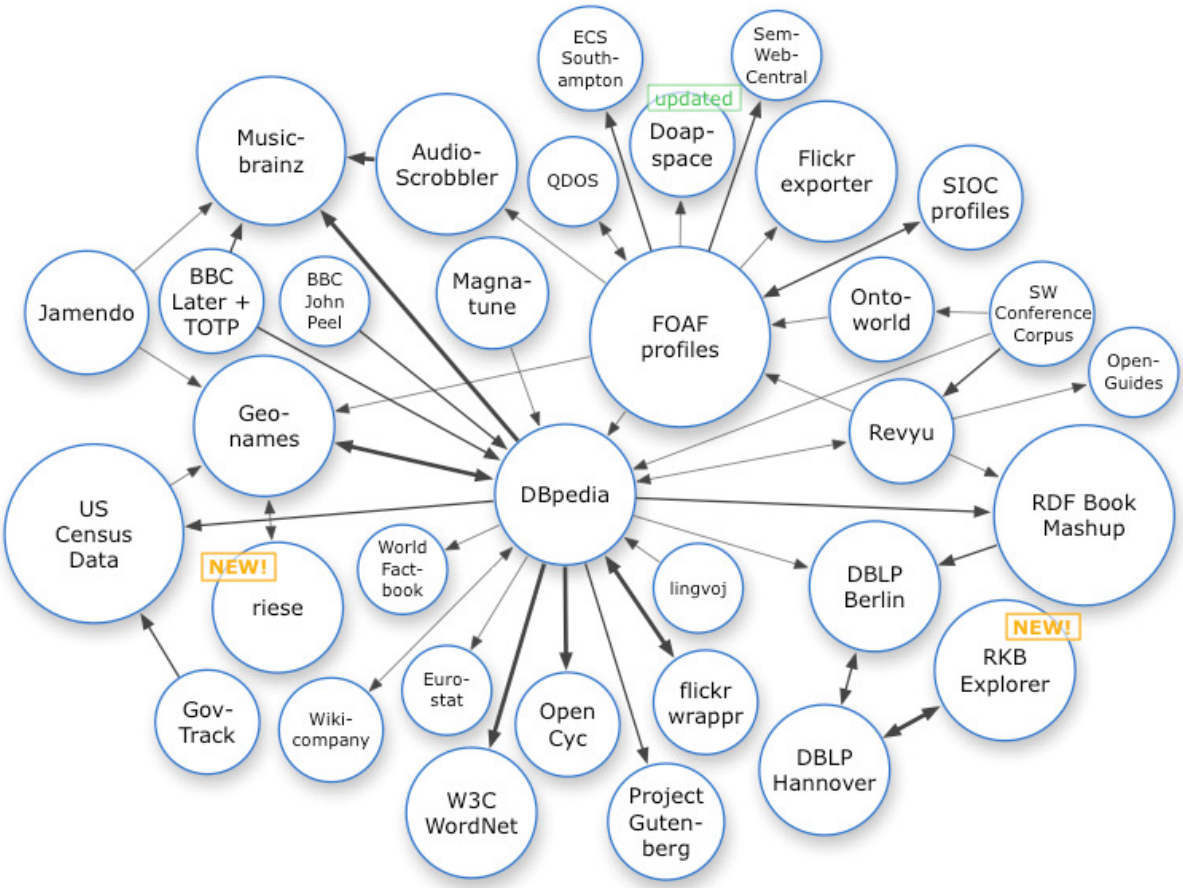
Slika 1: Stanje spleta povezanih podatkov marca 2009 [3].
Barve označujejo različne domene podatkov.

Rezultat povezovanja podatkov je t.i. splet povezanih podatkov (Linked Open Data Cloud), ki predstavlja informacijske temelje za delovanje semantičnega spleta v obliki, kot si jo je zamislil avtor. Slika 1 prikazuje stanje spleta povezanih podatkov. V splet povezanih podatkov se vključujejo raznovrstne podatkovne baze, zaradi česar se velikost spleta neprestano veča. Ena izmed prvih podatkovnih baz spleta povezanih podatkov je DBPedia – strukturirana različica 'bolj človeške' Wikipedie. Le-ta je hkrati tudi osrednji del spleta povezanih podatkov, saj služi kot stičišče večih podatkovnih baz.

1.3 Pomen semantičnega označevanja s pomočjo povezanih podatkov

Rezultat semantičnega označevanja besedila s pomočjo povezanih podatkov so oznake, ki kažejo na entitete odprtih podatkovnih baz iz spleta povezanih podatkov. V praksi to pomeni, da se za vsako entiteto skriva dereferenciabilen enoličen identifikator, preko katerega se lahko

prikopljemo do dodatnih strukturiranih informacij o eniteti. Ker smo s tem identifikatorjev prišli v splet povezanih podatkov, je rudarjenje za globljimi informacijami neomejeno oziroma odvisno od potreb aplikacije. Omejeno je le z velikostjo spleta povezanih podatkov, kateri trenutno raste s povečano hitrostjo. Za ponazoritev si na sliki 2 oglejmo stanje spleta povezanih podatkov izpred dobrega leta. Če jo primerjamo z novejšo sliko 1, je jasno razvidno, da se je število vključenih baz v splet povezanih podatkov v letu dni več kot podvojilo.



Slika 2: Stanje spleta podatkov marca 2008 [3]

1.4 Obstoječe rešitve in lastna implementacija storitve

Obstoječe programske rešitve za obdelavo naravnega jezika, izločanje informacij (IE, Information Extraction) in semantično označevanje se medseboj zelo razlikujejo. To velja tako za obseg funkcionalnosti kot tudi za način uporabe. V grobem jih lahko razdelimo v dve skupini:

a) odprtokodne rešitve, kamor sodijo GATE, NLTK, OpenNLP
b) komercialne rešitve, kamor sodijo Zemanta, OpenCalais, Hakia

Rešitve iz prve skupine so običajno na voljo v obliki knjižnice in izvirajo iz raziskovalnih akademskih laboratorijev. Za drugo skupino pa je značilna preprostost uporabe in integracije, saj so na voljo v obliki storitve in dostopni preko različnih spletnih protokolov (REST/XML, REST/JSON,...).

Primaren cilj diplomske naloge je lastna implementacija storitve semantičnega označevanja angleškega besedila z uporabo obstoječe odprtokodne programske opreme in prosto dostopnih podatkovnih baz. Predlagana programska rešitev se pri načinu uporabe zgleduje po komercialnih različicah označevanja besedil. Pri tem si pomaga z uporabo povezanih podatkov. To je trenutno zelo aktualen pristop, ki ga zaenkrat uporabljajo nekatere komercialne rešitve (npr. Zemanta) in raziskovalci s področja semantičnega spleta (npr. [4]). Podobno kot pri drugih rešitvah, za vhodne podatke služi besedilo in baza znanja iz odprtih povezanih podatkov. Rezultat je v obliki označenega izvornega besedila in seznama prepoznanih entitet. Želimo si, da bo predlagana rešitev uporabna do te mere, da ob nadaljnem razvoju lahko predstavlja alternativno izbiro komercialnim različicam.

2 Uporabljena orodja in podatki

2.1 DBPedia, strukturirana oblika Wikipedie

DBPedia [http://dbpedia.org] je projekt odprte skupnosti, katerega cilj je izluščiti informacije s prosto dostopne enciklopedije Wikipedia in jih ponuditi v strukturirani obliki. DBPedia omogoča uporabnikom, da nad podatkovno bazo Wikipedie izvajajo napredne poizvedbe, ter da si pri uporabi podatkov lahko pomagajo tudi z ostalimi povezanimi podatki [5].

Wikipedia je v večji meri sestavljena iz tekstovnih člankov, a obenem vsebuje tudi različne tipe strukturiranih informacij, npr.: povzetek glavnih značilnosti v obliki »infobox okvira« (glej primer na sliki 3), kategorije člankov, priložene slike, geo-koordinate, povezave na zunanje spletne strani, ipd. Takšne strukturirane informacije je moč enostavno izluščiti in shraniti v strukturirani obliki, ki omogoča izvajanje naprednejših poizvedb [6].

```
{{Infobox Town AT |
  name = Innsbruck |
  image_coa =  InnsbruckWappen.png |
  image_map = Karte-tirol-I.png |
  state = [[Tyrol]] |
  regbzk = [[Statutory city]] |
  population = 117,342 |
  population_as_of = 2006 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 574 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_hem = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_hem = E |
  postal_code = 6010-6080 |
  area_code = 0512 |
  licence = I |
  mayor = Hilde Zach |
  website = [http://innsbruck.at] |
}}
```

Slika 3: Primer infobox okvira s strukturiranimi podatki

Na začetku leta 2009 je DBPediina baza vključevala več kot 2,6 milijona stvari, vključno 213.000 ljudi, 328.000 krajev, 57.000 glasbenih albumov, 36.000 filmov in 20.000 podjetij. Podatkovni nabor vsebuje vsaj nazive in kratke povzetke za vse naštete pojme v 30 različnih jezikih. Vsebuje tudi preko 610.000 povezav do slikovnega gradiva, 3,15 milijona povezav do zunanjih strani, 4,9 milijona povezav do zunanjih RDF virov, 415.000 Wikipedia kategorij in 75.000 YAGO kategorij. DBPedia uporablja standard Resource Description Framework (RDF). Le-ta predstavlja temelje za prilagodljiv podatkovni model, ki ga neprestano rastoča baza podatkov kot je DBPedia potrebuje. Na začetku 2009 je DBPedia vsebovaja več kot 275 milijonov RDF trojčkov, ki izvirajo iz angleške, nemške, francoske, španske, italijanske, portugalske, poljske, švedske, nizozemske, japonske, kitajske, ruske, finske in norveške verzije Wikipedie. Celoten podatkovni nabor DBPedie je na voljo po licenčnimi pogoji licence GNU Free Documentation License [5].

DBPediini podatki so tesno povezani z ostalimi bazami iz spleta povezanih podatkov - vse povezave so vnešene na nivoju RDF. To omogoča uporabnikom, da podatke iz DBPedie po potrebi obogatijo še s podatki iz podatkovnih baz. DBPedia je tesno povezana vsaj s sledečimi podatkovnimi bazami iz spleta povezanih podatkov: Freebase, OpenCyc, UMBEL, GeoNames, Musicbrainz, CIA World Fact Book, DBLP, Project Gutenberg, DBTune Jamendo, Eurostat in US Census [5].

DBPediina podatkovna baza sestoji iz modularnih sklopov podatkov o stvareh, takoimenovanih podatkovnih naborov (datasets). Taki podatkovni nabori so npr: nazivi, opisi, sinonimi, kategorije, ipd. Posamezni nabori se delijo še po jezikih. Uporabniki si lahko s pomočjo naborov sestavijo svojim potrebam prilagojeno podatkovno bazo o stvareh in pojmih z Wikipedie, ki jo sestavljajo poljubni atributi v poljubnih jezikih [6].

Uporaba DBPedie v procesu semantične anotacije se zdi kot naravna odločitev. DBPedia predstavlja največjo odprto podatkovno bazo o stvareh v najbolj splošnem pomenu besede. Hkrati predstavlja središčno točko spleta povezanih podatkov, saj je tesno povezana z ostalimi odprtimi podatkovnimi bazami. Na voljo je tudi podatkovni nabor, ki vsebuje množice za razločanje pomenov, kar je še posebej dobrodošlo v procesu razdvoumljanja.

2.2 OpenRDF Sesame, podatkovna baza RDF

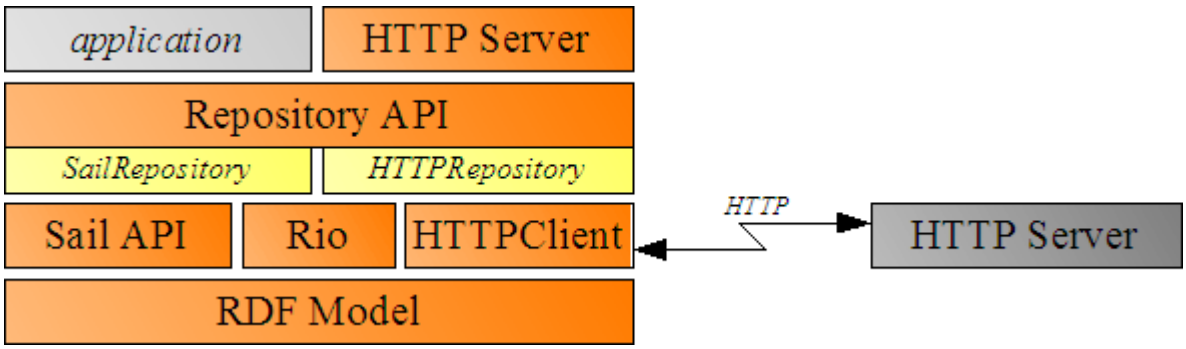
OpenRDF Sesame [http://www.openrdf.org] je javansko odprtokodno ogrodje za shranjevanje in poizvedovanje podatkov nad RDF podatkovno bazo. Podpira tudi sklepanje s pomočjo RDF

shem ter poizvedbe z različnimi poizvedbenimi jeziki. Razvoj Sesame se je začel z razvojem prototipa v okviru raziskovalnega projekta Evropske Unije, imenovanega On-To-Knowledge. Sedaj ga vzdržujejo, razvijajo in nadgrajujejo pri podjetju Aduna skupaj z organizacijo NLnet ter ostalimi razvijalci iz odprtokodne skupnosti [7].

Sesame je bil zasnovan s fleksibilnostjo in skalabilnostjo v mislih. Lahko se priključi na raznovrstne sisteme za shranjevanje podatkov: relacijske baze, datotečne sisteme, neposredno na indekser ali preprosto na hitri pomnilnik. Za čimvečjo prilagodljivost vsebuje dodatna orodja za obdelovanje podatkov. Uporabljamo ga preko neposredne integracije s programskega vmesnika (API), ki omogoča tako lokalno kot tudi oddaljeno postavitev. Na voljo pa je tudi SPARQL vozlišče. Sesame je zgrajen iz komponent, ki so razdeljene na več plasti (slika 4). Na prvem se nahaja t.i. RDF model, ki predstavlja temelje ogrodja. Ker gre za RDF-orientirano ogrodje, so vse komponente razmeroma vezane na RDF model. Le-ta definira vmesnike in vsebuje implementacije vseh glavnih RDF entitet: URI, prazno vozlišče, prazno vozlišče, literal in stavek [8].

Rio, (RDF Input/Output) je sestavljen iz razčlenjevalnikov in pisalnikov različnih formatov RDF datotek. Razčlenjevalniki zmorejo brati RDF datoteke in iz njih ustvariti stavke – trojčke. Pisalniki opravljajo obratno operacijo; trojčke pretvarjajo v ustrezne datotečne formate. Rio je moč uporabiti tudi zunaj Sesame ogrodja. Programski vmesnik Sail predstavlja plast shrambe in sklepanja (Storage and Inference Layer), ki nudi nizkonivojski dostop do RDF shramb in različnih implementacij sklepanja. Namen Saila je, da abstrahira podrobnosti shrambe in sklepanja, tako da se lahko uporabi poljubne implementacije. Teh je več vrst, npr. pomnilniška (MemoryStore) ali domača (NativeStore), ki uporablja lasten zapis podatkov na datotečni sistem [8].

Programski vmesnik za dostop do repozitorija (Repository API) pa predstavlja visokonivojski vmesnik, ki razvijalcu nudi metode za upravljanje RDF podatkov. Sem spadajo metode za nalaganje, poizvedovanje, za izvažanje in obdelovanje podatkov. Obstaja več implementacij programskega vmesnika, npr. SailRepository in HTTPRepository. Prva se uporablja za dostop do lokalnih shramb, druga pa za oddaljene. Najvišjo plast predstavlja HTTP strežnik. Ta sestoji iz kopice javanskih servletov, ki skupaj implementirajo protokol za dostop do Sesame repozitorijev preko HTTP protokola. Navadno se za uporabo HTTP protokola uporablja knjižnica na odjemalčevi strani, ki skrije podrobnosti spodaj ležečega protokola [8].



Slika 4: Pregled komponent Sesame

OpenRDF Sesame je ena izmed najbolj razširjenih odprtokodnih RDF podatkovnih baz. Zanj je značilna velika prilagodljivost, dobra dokumentacija in visoka aktivnost razvoja, kar je pri odločitvah o uporabi vzhajajočih tehnologij zelo pomembno. Sesame spada med hitrejše RDF podatkovne baze, kar potrjujejo tudi neodvisni primerjalni testi zmogljivosti [9]. V svojih projektih ga uporabljajo in podpirajo tudi vodilni deležniki v razvoju semantičnih tehnologij, kot so MIT za projekt Simile, Stanford v priljubljenem urejevalniku ontologij Protege ali Univerza v Innsbrucku.

2.3 Apache Lucene, knjižnica za iskanje

Apache Lucene [http://lucene.apache.org] je odprtokodna implementacija visoko zmogljivega in polno opremljenega iskalnika, napisana v čisti Javi. Avtor osnovne verzije je razvijalec Doug Cutting. Iskalnik Lucene je bil uspešno preveden v druga okolja, kot so Delphi, Perl, C#, C++, Python, Ruby in PHP. Nadaljni razvoj podpira organizacija Apache Software Foundation. Knjižnica je na voljo pod odprto licenco Apache Software License.

Lucene je tehnologija, ki je primerna za vsako aplikacijo, ki potrebuje indeksiranje in iskanje po polnem besedilu. Še posebej pa je Lucene priljubljen med implementacijami spletnih iskalnikov/pajkov ali lokalnih iskalnikov znotraj posameznih spletnih mest. Lucenov indeks konceptualno sestoji iz nabora dokumentov, ki so sestavljeni iz množice polj. To Lucenu omogoča veliko prilagodljivost, saj je neodvisen od posameznih datotečnih formatov. Besedila v formatih TXT, HTML, PDF, MS Word, ipd. je enostavno indeksirati, če le lahko dostopamo do zapisanega besedila. Lucene omogoča dostop do funkcionalnosti indeksiranja in iskanja preko preprostega javanskega programskega vmesnika. Je hiter in skalabilen. Ker ima nizke

performančne zahteve, ga lahko uporabljamo tudi na šibkejših konfiguracijah z omejenim pomnilnikom ali omejenim prostorom na trdem disku. Preko programskega vmesnika ponuja dostop do naprednejših funkcionalnosti, kot so sofisticirane poizvedbe, rangiranje rezultatov, iskanje po izbranih atributih, urejanje in iskanje po datumskih intervalih, ipd. Med drugim omogoča tudi inkrementalno indeksiranje ter vzporedno branje in pisanje v indeks.

Glavna karakteristika, po kateri si je Lucene prislužil današnji sloves, je vrednotenje oz. rangiranje rezultatov (scoring). Lucenovo rangiranje rezultatov je izjemno hitro, hkrati pa skrije veliko kompleksnosti pod pokrovom implementacije indekserja. Za rangiranje uporablja model vektorskega prostora (VSM, vector space model) skupaj z boolovim modelom. Ideja, ki stoji za VSM je sledeča: dokumenti so predstavljeni v obliki vektorjev terminov, kjer vsaka dimenzija predstavlja frekvenco pojavitve termina v dokumentu. Če je frekvenca pojavitve termina višja glede na vse pojavitve termina v vseh dokumentih, potem je dokument bolj relevanten zadetek z višjim rangom; sicer obratno. Temu algoritmu pravimo TF-IDF algoritem in velja za enega najbolj učinkovitih algoritmov za rangiranje rezultatov. Poleg priloženega je možno uporabiti tudi lastne implementacije rangiranja. Ob iskanju se najprej uporabi Boolov model, da se omeji na množico veljavnih dokumentov iz nabora vseh dokumentov, ki ustrezajo logičnemu delu poizvedbe. Doda še nekaj optimizacij in dodatkov (npr.: fuzzy iskanje), vendar v osnovi ostaja čistokrvna implementacija VSM [10].

2.4 GATE, orodje za obdelavo naravnega jezika

General Architecture for Text Engineering ali kratko GATE [http://gate.ac.uk] je odprtokodno javansko orodje za obdelavo besedila in izločanje informacij (IE). Uporabljajo ga priznani znanstveniki, velika podjetja, učitelji in študenti po vsem svetu za namene obdelave besedila v različnih jezikih. V osnovi GATE sestoji iz treh elementov [13]:

- arhitekture, ki sestoji iz temeljnih komponent sistemov za obdelavo besedila
- javanskega ogrodja oz. programske knjižnice
- grafičnega razvojnega okolja, ki predstavlja uporabniku prijazen grafičen vmesnik za uporabo ogrodja

GATE je prostodostopen in odprtokoden projekt, katerega rezultati so na voljo pod LGPL licenco. Napisan je v čisti Javi, zato preverjeno teče na platformah Linux, Windows in Mac OS X. Je zrel in aktiven projekt z približno 20 razvijalci. GATE je celovito orodje, ki nudi ali omogoča

ročno označevanje, okolje za preverjanje uspešnosti obdelav besedila, izločanje informacij, (pol)-avtomatsko semantično označevanje in še mnogo drugega. Zanj je na voljo več kot 50 različnih dodatkov, ki so že vključeni v standardno distribucijo. GATE nima težav z uporabo različnih formatov besedil. S priloženimi vhodnimi filtri lahko bere besedila v formatih XML, HTML, PDF, MS Word, e-pošte ali v navadni pusti obliki. Podatki med delovanjem bivajo skupnem pomnilniškem repozitoriju, ki je namenjen dokumentom, korpusom in označevanju. Za obstojne podatke ima podporo za XML ter podatkovne baze Oracle in PostgreSQL. Podatke lahko shranjuje tudi v obliki javine serializacije.

GATE že vključuje znane standardne algoritme za pogosta opravila obdelave besedila, kot so členjenje na besede, označevanje stavčnih členov (POS tagging), deljenje povedi in stavkov, prepoznavanje imenovanih entitet, razčlenjanje zaimkov, strojno učenje, ipd. GATE je tudi globoko integriran z ostalimi odprtokodnimi projekti. Za pridobivanje informacij uporablja Apache Lucene (tudi Nutch in Solr), iskalnike Google in Yahoo ter MG4J. Za strojno učenje uporablja Weko, MaxEnt, SVMLight, ipd. Za delo z ontologijami ima integriran Sesame in OWLIM, razčlenjanje informacij pa uporablja RASP, Minipar in SUPPLE. Uporablja tudi projekte UIMA, implementacije korenjenja Snowball in podatkovno bazo besed Wordnet.

Prva verzija orodja GATE je bila izdana leta 1996. Zatem je bil na novo zasnovan, implementiran ter izdan leta 2002. Danes je to eden najbolj razširjenih tovrstnih sistemov, saj predstavlja celovito in robustno infrastrukturo za obdelavo besedil in z njo povezana opravila. Tudi IBM-ov projekt UIMA je dobil inspiracijo v GATE arhitekturi; IBM pa je celo sponzoriral razvoj interoperabilnostne ravni med obema sistemoma.

Glavne značilnosti GATE so naslednje:

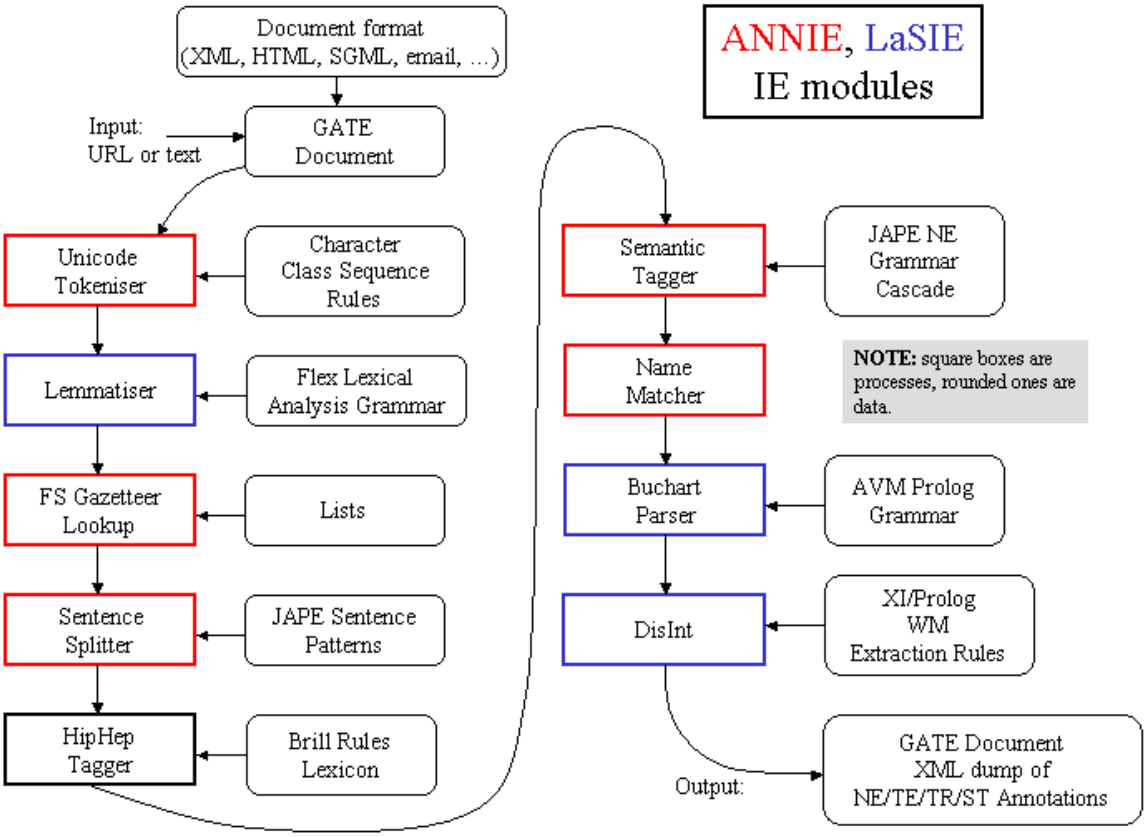
- komponentno orientiran razvoj, ki lajša režijsko breme integracije in razvoja pri raziskovalnih projektih
- avtomatsko merjenje zmogljivosti obdelav besedila, kar spodbuja kvanititativno primerjalno ocenjevanje
- ločevanje med nizkonivojskimi (shranjevanje podatkov, vizualizacija podatkov, nalaganje komponent) in visokonivojskimi opravili za obdelavo besedila
- jasna razmejitev med podatkovnimi strukturami in algoritmi za obdelavo besedila
- konsistentna uporaba standardnih mehanizmov za komponente
- uporaba odprtih standardov (Unicode, XML)

- osnoven cevovod za obdelavo besedila sestavljen iz posameznih komponent, ki jih uporabnik lahko poljubno nastavlja ali zamenjuje s svojimi implementacijami

Analiza besedila je proces, ki sprejme podatke v obliki naravnega jezika, rezultate pa vrne v fiksnem formatu jasnih in nedvoumnih podatkov. Taki podatki se lahko uporabijo za neposreden prikaz uporabnikom. Lahko pa se jih shrani v podatkovno bazo ali preglednico za kasnejšo analizo ali za namene indeksiranja v aplikacijah za izločanje informacij iz besedila. Analiza besedila pokriva družino aplikacij, ki obsega tudi prepoznavanje imenovanih entitet, prepoznavanja relacij in razpoznavanje dogodkov. GATE je bil za tovrstne naloge uspešno uporabljen v domenah bioinformatike, zdravstva in pri obdelavi preteklih zapisnikov sodišč.

GATE vključuje komponento za izločanje informacij ANNIE (A Nearly New Information Extraction System). Ta sloni na algoritmih končnih avtomatov in JAPE jeziku. Njegove temelje predstavlja cevovod za izločanje informacij iz besedila prikazan na sliki 5, ki ga sestavljajo sledeče prednastavljene komponente [14]:

- **Tokenizator**, ki besedilo razdeli na atomarne elemente: besede, ločila, števila, simbole, presledke, ipd. Cilj je pripraviti besedilo za nadaljno obdelavo, in hkrati omogočiti najvišjo mero fleksibilnosti in obvladljivosti atomov besedila.
- **Gazetteer seznami**. Ti predstavljajo sezname pogosto uporabljenih imenovanih entitet, kot so mesta, organizacije, ljudje, valute... Služijo lažjemu prepoznavanju entitet, predvsem pa klasifikaciji entitet.
- **Več razdelilnikov povedi**, ki ugotavljajo, kateri atomarni elementi skupaj tvorijo poved. Tudi tu so rezultati le delni in so namenjeni nadaljni obdelavi.
- **Oblikoskladenjski označevalnik** ali Part Of Speech Tagger (POS). Osnovni cevovod uporablja dopolnjeno verzijo Brill označevalca, ki vsaki besedi pripiše vlogo v vsebujočem stavku. Le-ta uporablja priložen leksikon in nabor pravil, ugotovljenih s pomočjo strojnega učenja na velikem korpusu dokumentov iz Wall Street Journala. Obadva je možno zamenjati s svojimi podatki.
- **Semantični označevalnik**, ki je osnovan na jeziku pravil JAPE. Deluje nad delnimi rezultati predhodnih komponent v osnovnem cevovodu.
- **Razpoznavanje zaimkov**, ki se ga lahko vključi po potrebi.

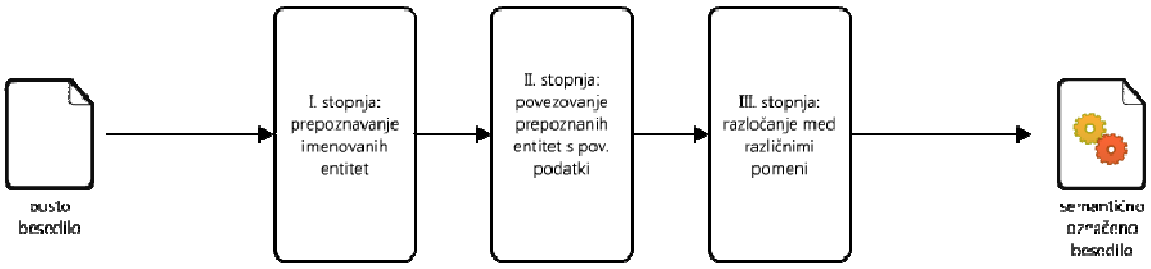


Slika 5: Shema cevovoda sistema ANNIE (rdeče obarvane komponente)

3 Implementacija označevanja besedila s pomočjo povezanih podatkov

3.1 Pregled arhitekture

Arhitekturo obdelave besedila (cevovod) lastne implementacije semantičnega označevanja sestavljajo tri zaporedne stopnje, kot je prikazano na sliki 6. Prva skrbi za prepoznavanje imenovanih entitet – ljudi, krajev, pojmov, ipd. Druga stopnja skrbi za povezovanje prepoznanih imenovanih entitet z entitami iz spleta povezanih podatkov. Včasih prvi dve stopnji zadoščata za učinkovito povezovanje entitet s povezanimi podatki. Običajno pa imamo situacijo, ko neka beseda ali besedna zveza lahko predstavlja dve ali več entitet, zaradi česar potrebujemo še tretjo stopnjo. Naloga te je, da poskrbi za razdvoumljanje entitet, t.j. razločanje med različnimi pomeni besede. Končni rezultat je tako prava oz. po pomenu najbližja entiteta s spleta povezanih podatkov.



Slika 6: Arhitektura obdelave besedila s tremi zaporednimi stopnjami

Vsaka od navedenih treh stopenj uporablja specifične podatkovne nabore, knjižnice in metode obdelave besedila za opravljanje svoje naloge. V nadaljevanju sledi podroben opis realizacije vseh stopenj predlagane arhitekture.

3.2 Prepoznavanje imenovanih entitet besedila

Za prepoznavanje imenovanih entitet je bila integrirana knjižnica GATE in njena komponenta za izločanje informacij iz besedila ANNIE. Knjižnica je napisana v Javi, zato s samo integracijo ni bilo težav. Poleg prepoznavanja entitet zna ANNIE tudi klasificirati entitete po tipih (človek, kraj, organizacija). Za namene obdelave je bilo nastavljeno prepoznavanje ljudi, krajev in organizacij brez neznanih entitet. S slednjim se označuje nepoznane pojme, ki jih ANNIE ne uspe

klasificirati. Izkušnje so pokazale, da podatek o tipu spada med manj zanesljive; velik delež prepoznanih entitet se namreč klasificira s tipom neznano. Zaradi navedenega ta podatek pri nadaljni obdelavi ni bil upoštevan. Rezultat prve stopnje so prepoznane imenovane entitete v obliki besed ali besednih zvez. Vsaka entiteta ima priloženo prilegajočo poved, ki jo potrebujemo v zadnji stopnji za ugotavljanje semantične bližine.

3.3 Povezovanje prepoznanih entitet z entitetami spleta povezanih podatkov

Iskanje entitet iz spleta povezanih podatkov in povezovanje s prepoznanimi imenovanimi entitetami predstavlja zajeten del lastne implementacije. Tu predstavlja glavni izziv obvladljivost podatkov in skalabilnost iskanja. Ker gre za splošno namensko semantično označevanje, je potrebno zajeti zelo širok nabor entitet. Zato je bila prva odločitev v zvezi z uporabo povezanih podatkov ta, da se zaenkrat uporabi le entitete z DBPedie. Splet povezanih podatkov namreč neprestano raste in integracija vseh dostopnih podatkovnih naborov že na začetku bi pomenila veliko oviro. Na drugi strani pa je DBPediina baza najbolj široka in splošna baza entitet, saj izvira iz »zakladnice človeškega znanja« – Wikipedie. To pa povsem odgovarja splošno namenski usmerjenosti aplikacije. Uporabniki se lahko še vedno dokopljejo do domensko specifičnih podatkov (t.i. drill-down), saj je DBPediin podatkovni nabor globoko zasidran v spletu povezanih podatkov. Za prepoznavanje domensko specifičnih entitet to žal ne velja. Uporabniki sami ne morejo dodajati podatkovnih zbirk, ki služijo prepoznavanju. Vendar pa se lahko za specifične namene osnovni podatkovni nabor entitet v prihodnosti preprosto dopolni tudi z dodatnimi nabori.

Ker je DBPedia ogromna podatkovna baza, je v prid obvladljivosti in prilagodljivosti razdeljena na več delov, t.i. podatkovnih naborov. Le-te sestavljajo trojčki oz. stavki RDF. Stavki največkrat opredeljujejo podatkovni atribut neke entitete, manj pogosto pa gre za povezavo med dvema entitetama. Za iskanje in prepoznavanje entitet so bili uporabljeni sledeči podatkovni nabori DBPedie:

- nazivi entitet (Titles)
- preusmeritve (Redirects)

Nazivi entitet predstavljajo najbolj osnoven podatkovni nabor, sestavljen iz entitet na eni strani ter njihovimi nazivi v angleškem jeziku na drugi strani. Entitete so pri tem in vseh ostalih naborih opredeljene z enoličnimi identifikatorji URI. Drugi podatkovni nabor pa je izvleček

podatkov, ki jih Wikipedia uporablja pri preusmeritvah uporabnika ob iskanju terminov. Pomen teh podatkov je v dodatnih besedah, sinonimih ali akronimih, ki se uporabljajo za naslavljanje neke entitete. Oba podatkovna nabora sta bila združena v enoten repozitorij Sesame baze trojčkov. Žal se je ob prvi implementaciji izkazalo, da je že osnovni nabor nazivov DBPedie prevelik zalogaj za bazo trojčkov Sesame. Na običajnem prenosnem računalniku (enojedrni procesor s taktom 1.8 MHz, 1Gb RAM) je iskanje in sprehajanje po bazi trojčkov zelo počasno opravilo. Tudi izgradnja vseh indeksov, ki jih Sesame podpira, te težave ni odpravila. Ker so se časi povsem običajnih poizvedb gibali v predelu nekaj minut, je bila sprejeta odločitev, da se za iskanje in povezovanje entitet uporabi preizkušeno rešitev - indekser in iskalnik Apache Lucene.

V sledeči implementaciji je potek priprave in integracije podatkovnih naborov DBPedie potekal v dveh dolgotrajnih korakih. V prvem se podatki iz surovega datotečnega zapisa trojčkov uvozijo v repozitorij baze trojčkov Sesame. Združita se oba uporabljena podatkovna nabora. Nato se s pomočjo Apache Lucena zgradi indeks, ki indeksira vse nazive in sinonime entitet, pri čemer se korak tokenizacije izpusti. To je potrebno zato, ker se za iskanje uporablja samo točno in ne le delno ujemanje nazivov. V stopnji iskanja entitet se uporablja zgolj omenjeni indeks, repozitorij baze trojčkov pa služi le v procesu priprave podatkov.

Omenjena stopnja prejme kot vhodne podatke prepoznane imenovane entitete od predhodne stopnje. Nato s pomočjo hitrega indeksa poišče tako imenovane entitete iz spleta povezanih podatkov (DBPedie). Pri tem upošteva vse nazive, sinonime ter akronime, ki jih pridobimo iz podatkov o preusmeritvah. Rezultat stopnje so entitete z DBPedie, opredeljene z enoličnim identifikatorjem URI. V primeru, da se pri povezovanju in iskanju pojavi več zadetkov za eno prepoznano entiteto, sestavlja rezultat več entitet. Takrat pride do izraza naslednja stopnja, ki skrbi za razločanje med različnimi pomeni.

3.4 Razločevanje med pomeni entitete

Zadnja stopnja v cevovodu obdelave besedila skrbi za razločevanje med prepoznanimi pomeni besede ali besedne zveze. Razločevanje med pomeni je lahko zelo široka tema. Tudi tu sem se dosledno držal načela pragmatičnosti in uporabil različico znanega pristopa s posamičnim razločevanjem entitite. Zadovoljivo rešitev sem implementiral s pomočjo podatkov z DBPedie in lastne hevristike, ki temelji na uporabi Lucenove komponente za rangiranje rezultatov.

Iz predhodnih stopenj se za vsako prepoznano imenovano entiteto uporabijo podatki o besedi ali besedni zvezi, poved v kateri se nahaja, ter možne entitete iz DBPedie, ki predstavljajo vse mogoče pomene. Izhodiščna ideja je sledeča: uredimo vse možne pomene po semantični bližini do omenjene entitete ter nato izberimo najbližjo, ki označuje pomen te entitete. Poglejmo podrobneje, kako je hevristika implementirana in kateri podatki se uporabljajo.

DBPedia ima po zaslugi Wikipedie na voljo t.i. množice za razdvoumljanje (disambiguation set), ki so opredeljene za večino entitet. Tako za neko entiteto obstaja množica drugih entitet, ki imajo enak ali podoben naziv, a različen pomen. Takim besedam pravimo tudi homonimi. Sprva se združi množice za razdvoumljanje vseh različnih pomenov v eno množico. Nato se za vse entitete v množici pridobi še njihove kratke opise. Sedaj imamo na eni strani imenovano entiteto iz besedila, ter na drugi strani množico entitet iz povezanih podatkov DBPedie in njihove opise. Naslednja naloga je izbira prave entitete z DBPedie, ki predstavlja ustrezen pomen besede oziroma prepoznane imenovane entitete iz besedila. Tukaj se uporabi Lucenov model vektorskega prostora VSM, ki temelji na pogosto uporabljeni predstavitvi besedila z vektorjem besed [11]. Tam je vsak dokument predstavljen z vektorjem, kjer vsaka dimenzija predstavlja besedo iz besedila oziroma njeno frekvenco pojavljanja v besedilu. Za ta namen se izgradi hitri pomnilniški Lucene indeks, za dokumente pa se uporabijo entitete, ki predstavljajo različne pomene. Opise in polne nazive se indeksira s standardno analizo, ki vključuje tokenizacijo in nekatere dodatne filtre (odstranitev neinformativnih besed, ipd.). Nato se nad indeksom izvede poizvedbo, kjer se nastavi posebno težo nazivu prepoznane entitete iz besedila. Le ta mora biti prisotna v originalni obliki. Zraven pa se opcijsko upoštevajo še besede iz povedi izvornega besedila, ki vsebuje prepoznano imenovano entiteto. Rezultat te poizvedbe so vsi pomeni besede, urejeni po semantični bližini do omenjene imenovane entitete.

Delovanje opisane hevristike si oglejmo še na naslednjem primeru. Označimo besedilo, ki vsebuje imenovano entiteto Java: *»I like programming in Java.«* Prepozna se entiteta Java, v ozadju pa se sedaj izgradi pomnilniški indeks vseh možnih pomenov. Nato se po indeksu izvede naslednja poizvedba, ki vsebuje tako naziv kot vsebujoči stavek: *»"Java"^12 Java i like programming java«*. Naziv entitete ima glede na vsebujoči stavek dodatno utež (12).

Rezultat poizvedbe so možni pomeni entitete skupaj s pripadajočimi enoličnimi identifikatorji URI, urejeni po semantični bližini (prvih 5):

1. Java (programming language): http://dbpedia.org/resource/Java_%28programming_language%29

2. Java: http://dbpedia.org/resource/Java
3. Java (band): http://dbpedia.org/resource/Java_%28band%29
4. Java (chicken): http://dbpedia.org/resource/Java_%28chicken%29
5. Java (dance): http://dbpedia.org/resource/Java_%28dance%29

Prvi rezultat in končno izbrana entiteta:

Java (programming language):, http://dbpedia.org/resource/Java_%28programming_language%29

Opis končne entitete, ki je pripomogel pri pravilni izbiri, saj so si nazivi različnih pomenov zelo podobni, glavne razlike pa se kažejo v opisih:

»Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode that can run on any Java virtual machine regardless of computer architecture.«

3.5 Implementacija vmesnika storitve in aplikacijsko okolje

Rešitev je implementirana kot spletna storitev, ki se lahko uporablja programsko preko vmesnika REST/HTTP iz kateregakoli programskega okolja. Druga možnost pa ročna uporaba preko grafičnega vmesnika v obliki spletne strani. Uporaba je enostavna: vnesemo pusto besedilo, ki ga želimo semantično označiti, ter sprožimo proces označevanja. Ko se zahteva izvede, se prikaže preko hiperpovezav označeno izvorno besedilo ter pod njim seznam vseh prepoznanih imenovanih entitet. Le-ta vključuje tudi poljubne metapodatke, kot npr. število ponovitev, URI na povezane podatke in zaupanje. Poleg lastne implementacije omogoča še označevanje s komercialnimi storitvami OpenCalaisa in Zemante, kar se izkaže posebno koristno pri primerjalnih testih.

Za implementacijo se je zaradi razširjenosti in razpoložljivosti knjižnic za obdelovanje besedila uporabilo javansko okolje. Cela aplikacija je napisana kot spletna aplikacija J2EE, z nekaj servleti in JSP stranmi. Za strežniško okolje služi Apache Tomcat. Na njem poleg aplikacije semantičnega označevanja teče tudi strežnik podatkovne baze trojčkov Sesame. Od ostalih knjižnic se uporabljajo že omenjeni Apache Lucene za indeksiranje in določanje semantične bližine, ter GATE ANNIE za prepoznavanje imenovanih entitet iz angleškega besedila.

4 Primerjava z obstoječimi rešitvami

4.1 Metodologija testiranja

Za testiranje sem se poslužil načina, kjer sprva pripravimo testno množico dokumentov, ki jih ročno semantično označimo. Nato dokumente semantično označimo še z vsemi storitvami, rezultate pa ovrednotimo na podlagi ročnih oznak. Rezultate posameznih storitev zberemo, izračunamo evalvacijske metrike in jih medsebojno primerjamo. Za evalvacijske metrike uporabimo uveljavljene mere natančnosti, priklica ter metrike F_1. Vse navedene metrike imajo dolgo tradicijo v domeni izločanja informacij iz besedila (IE), uporabljajo pa jih tudi na priznanih tekmovanjih IE MUC (Message Understanding Conference) [17].

Rezultate lastne implementacije semantičnega označevanja primerjamo s storitvijo OpenCalaisa [http://www.opencalais.com] in Zemante [http://www.zemanta.com/api]. Vse tri storitve omogočajo semantično označevanje, vendar vsaka to počne na svoj način s svojimi podrobnostmi. Zemanta označi prve pojavitve entitete, a zgolj tiste, ki so po nekem internem kriteriju dovolj pomembne [16]. Nasprotno OpenCalais označi vse entitete ne glede na velikost besedila oziroma pomembnost entitete v besedilu. Za razliko od Zemante ne uporablja DBPediinih identifikatorjev, temveč lastne enolične identifikatorje. Storitev sicer že nekaj mesecev omogoča, da se dokopljemo do ustreznih DBPediinih identifikatorjev, vendar to zahteva občutno več dela z integracijo, primarno pa še vedno uporabljajo lastne enolične identifikatorje [15]. Tudi zaradi omenjenih razlik pri delovanju je potrebno upoštevati, da rezultati nudijo le osnovno primerjavo med storitvami.

Testiranje je potekalo v sledečih korakih:

1. **Priprava nabora testnih dokumentov**, ki vsebujejo načrtno izbrana besedila. Tukaj sem izbral 5 angleških člankov s spletnega portala CNN [http://www.cnn.com], ki pokrivajo različne domene in vsebujejo različne imenovane entitete. Zaradi divergentnih odzivov posameznih storitev na različne dolžine člankov sem se osredotočil na krajše članke – povprečna dolžina znaša 440 besed. S tem se želel čimbolj izenačiti delovanje lastne implementacije z delovanjem storitev OpenCalaisa in predvsem Zemante, ki pri večjih dokumentih vrača manj entitet.
2. **Ročno semantično označevanje.** Dokumente sem nato ročno semantično označil, tako da sem v besedilu identificiral vse imenovane entitete. Zaradi podrobnosti označevanja,

ki so vezane na posamezne storitve, sem se osredotočil na primerjavo seznama prepoznanih entitet. Tu sem zajel vse imenovane entitete, ki se pojavijo v besedilu. Ker storitve običajno označujejo le lokacijo prve pojavitve imenovane entitete, podatki o lokaciji oznake niso del nadaljne obravnave. Povprečno število pojavitev imenovanih entitet v testnih dokumentih znaša 15.

3. **Označevanje besedila s storitvami.** Vsak dokument iz nabora testnih podatkov se semantično označi z vsemi tremi storitvami. Rezultati se analizirajo in podatki o pravilno prepoznanih, nepravilno prepoznanih in neprepoznanih imenovanih entitetah se shranijo za nadaljnjo obravnavo.
4. **Izračun natančnosti, priklica in F_1 metrike (Precision, Recall, F_1).** Preštel sem pravilno prepoznane, nepravilno prepoznane in neprepoznane entitete. Iz teh podatkov sem za vsako storitev za vsak dokument izračunal t.i. kazalce natančnosti in priklica po sledečih formulah:
   - $natančnost = \frac{pravilno\_prepoznane}{pravilno\_prepoznane + nepravilno\_prepoznane}$
   - $priklic = \frac{pravilno\_prepoznane}{pravilno\_prepoznane + neprepoznane}$

   Število *pravilno_prepoznane* označuje število vseh pravilno prepoznanih entitet. Število *nepravilno_prepoznane* označuje število vseh prepoznanih entitete, ki niso prave imenovane entitete in zato ne sodijo med rezultate. *Neprepoznane* pa označuje število entitet, ki jih storitev ni prepoznala kljub temu, da so prave imenovane entitete in zato sodijo med rezultate.

   Poleg priklica in natančnosti sem izračunal tudi uveljavljeno metriko F_1, ki s pomočjo utežene harmonične sredine združi oba podatka. Pri metriki F_1 imata tako priklic kot natančnost enako utež, saj je pomembnost enega napram drugemu v realnosti odvisna od primera uporabe. Npr. pri implementaciji iskalnika ali priporočilnega sistema je bolj pomembna natančnost, medtem ko bi bil za orodje za polavtomatsko prevajanje bolj pomemben priklic. Metriko F_1 izračunam po sledeči formuli:

   - $F_1 = 2 * \frac{natančnost * priklic}{natančnost + priklic}$
5. **Izračun agregiranih podatkov, priprava diagramov in primerjava.** Za vsako storitev sem izračunal povprečne kazalnike za vse dokumente in iz njih pripravil nekaj diagramov. Slednji so temelj primerjave vseh storitev.

Spodaj se nahaja primer označenega testnega dokumenta. Označene so zgolj prve pojavitve imenovanih entitet (članek CNN, Nasa):

NASA *will attempt to land space shuttle* ***Atlantis*** *at* ***California's Edwards Air Force Base*** *on Sunday, after rainy* ***Florida*** *weather precluded a* ***Kennedy Space Center*** *landing for a third day, officials said.*

Rain at Kennedy Space Center in Florida canceled plans to land the space shuttle Atlantis on Saturday. The first attempt will be made at 11:39 a.m. ET at Edwards, north of ***Los Angeles****. Another opportunity will come at 1:17 p.m. ET.*

Rainy weather postponed the shuttle landing on Friday and Saturday. While Atlantis could conceivably remain in space until Monday, NASA has said it wants to land Sunday. Officials said Sunday's Florida weather was better than conditions Saturday, but atmospheric conditions in Florida remained too unstable for landing. The landing would be the 53rd at Edwards, NASA officials said. In the early days of the space shuttle program, Edwards was its primary landing site.

Atlantis launched May 11 for NASA's final repair visit to the ***Hubble Space Telescope****. Shuttle astronauts conducted space walks during the mission to perform routine repairs and replace key instruments, in what has been called one of the most ambitious space repair efforts ever attempted.*

Hubble was released back into orbit Tuesday morning. Hubble, which has been in space for nearly two decades, can capture clear images that telescopes on ***Earth*** *cannot, partly because it does not have to gaze through murky atmospheres.*

Seznam imenovanih entitet v dokumentu obsega naslednje entitete: *NASA, Atlantis, California, Edwards Air Force Base, Florida, Kennedy Space Center, Los Angeles, Hubble Space Telescope, Earth*

Ostali testni dokumenti so na voljo v prilogi.

4.2 Rezultati

V nadaljevanju sledijo rezultati o prepoznanih imenovanih entitetah za vse stestirane dokumente po posameznih storitvah:

lastna impl. / dokument	1	2	3	4	5
pravilno_prepoznane	9	9	5	11	8
nepravilno_prepoznane	0	1	2	2	3
neprepoznane	11	8	4	3	2

Zemanta / dokument	1	2	3	4	5
pravilno_prepoznane	10	10	8	9	7
nepravilno_prepoznane	0	0	0	1	2
neprepoznane	12	7	1	5	3

OpenCalais / dokument	1	2	3	4	5
pravilno_prepoznane	18	12	6	10	8
nepravilno_prepoznane	2	2	0	0	0
neprepoznane	4	5	3	4	2

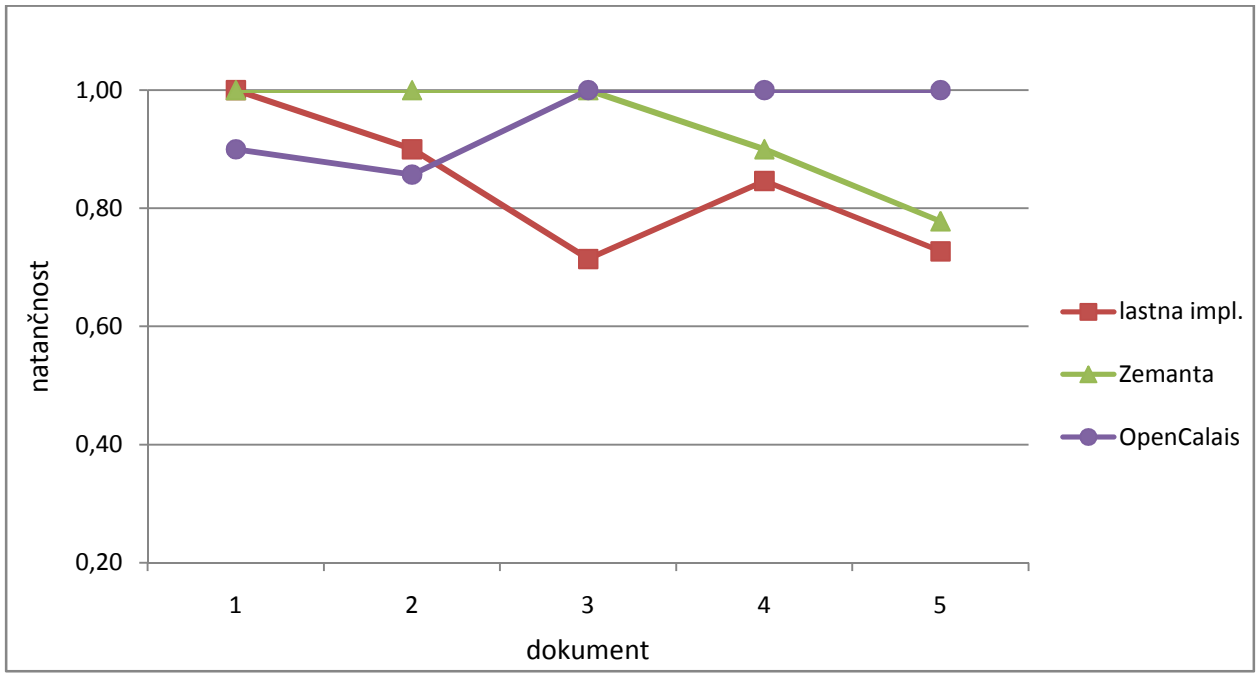
Za primer poglejmo napake posameznih rešitev. Medtem ko lastna implementacija pravilno prepozna 9 entit iz prvega dokumenta, jih Zemanta prepozna 10, od tega pa je skupnih natanko 6. Zemanta se bolje obnese pri prepoznavanju aktualnejših ali manj pogostih entitet, kar bi lahko pri lastni implementaciji izboljšali z integracijo DBPedie in knjižnice NLP (opisuje poglavje 5.1). Primera takih entitet iz prvega dokumenta sta »Facebook« in »Farsi«. Če rezultate prvega dokumenta primerjamo še z OpenCalaisovo storitvijo, vidimo, da slednja prepozna kar 18

entitet, od tega vse razen ene (»Farsi«), katere prepoznata tudi ostali storitvi. Številčno sta lastna implementacija in Zemanta vrnila zelo podobne rezultate: 9 oz. 10 pravilno prepoznanih entitet, 0 nepravilno prepoznanih, ter 11 oz. 12 neprepoznanih entitet, kar kaže na podobno realizacijo.

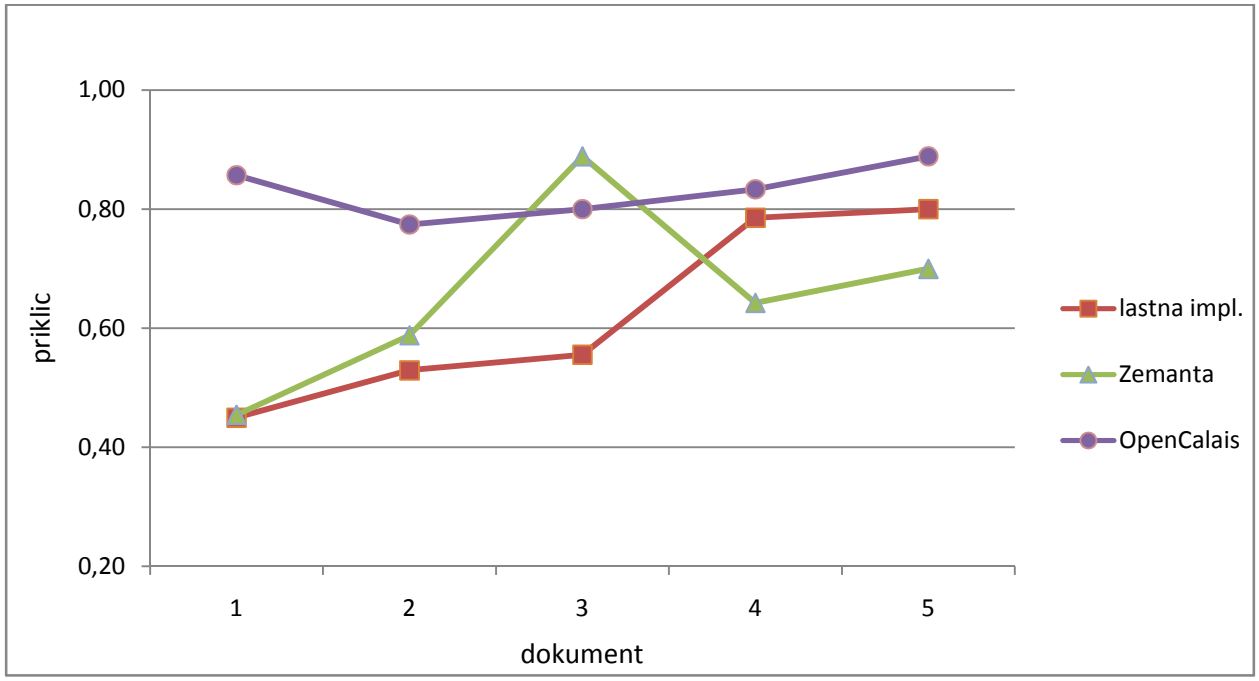
Podobna zgodba se ponovi tudi pri ostalih dokumentih, še največji odklon pa se pojavi pri tretjem dokumentu, kjer se Zemanta odreže bolje kot pri ostalih. Uspe ji označiti vseh 8 razen ene entitete (»California«), pri tem pa ne vrne nepravilno prepoznanih entitet. Manjkajočo entiteto »California« je precej trivialno prepoznati in označiti, zato se zastavlja vprašanje, zakaj je Zemanta v tem primeru ni prepoznala. Najverjetneje je temu tako, ker je »California« del pridevnika druge entitete (»California's Edwards Air Force Base«), saj v nasprotnem primeru označi tudi entiteto California. Drugi razlog pa je ta, da ima Zemanta interne kriterije za ugotavljanje pomembnosti entitet, ki določajo prag, pod katerim entitete ne vrača. V drugačnih okoliščinah namreč tudi Zemanta prepozna navedeno entiteto.

Lastna implementacija je pri tretjem dokumentu dosegla najslabšo natančnost na testu. Pravilno je prepoznala 5 od 9-ih entit in ob tem tudi 2 nepravilno (»Rain«, »Air force«). Pri prvi gre za običajno nepravilno prepoznano entiteto, ki se nato še (napačno) poveže z entiteto z DBPedie (http://dbpedia.org/resource/Rain). Pri entiteti »Air force« pa je krivda na strani komponente ANNIE, ki skrbi za prepoznavnaje entitet. Težave se pojavijo, če entiteto označuje več besed. Pravilno prepoznana entiteta v našem primeru pa bi bila »Edwards Air Force Base«, medtem ko ANNIE zazna le »Air Force«. OpenCalais je v tretjem dokumentu prepoznal le 6 entitet brez nepravilnih, manjkale pa so mu 3 entitete. Ker je tretji dokument najkrajši in ima tudi najmanj vsebovanih imenovanih entitet, bi rezultati lahko nakazovali, da je Zemanta bolj optimizirana za kratka besedila, OpenCalais pa za daljša.

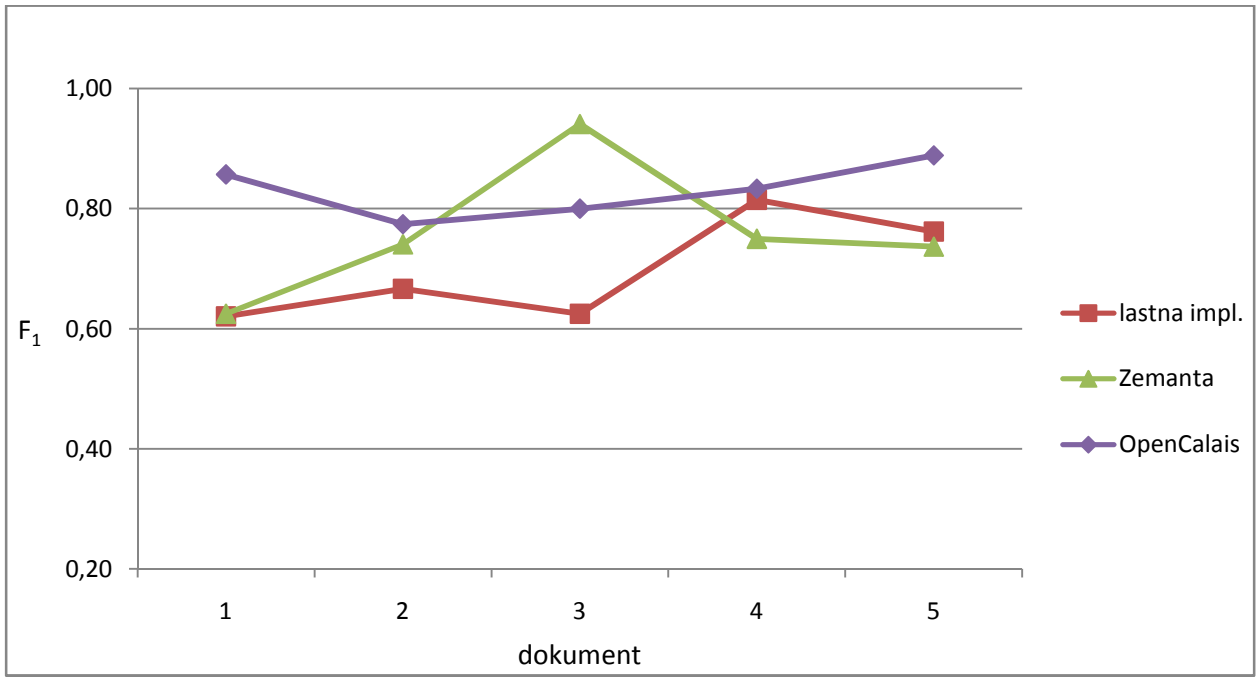
Navedene rezultate prikazujejo diagrami natančnosti, priklica in metrike F_1:



Slika 7: Natančnost po storitvah in dokumentih



Slika 8: Priklic po storitvah in dokumentih

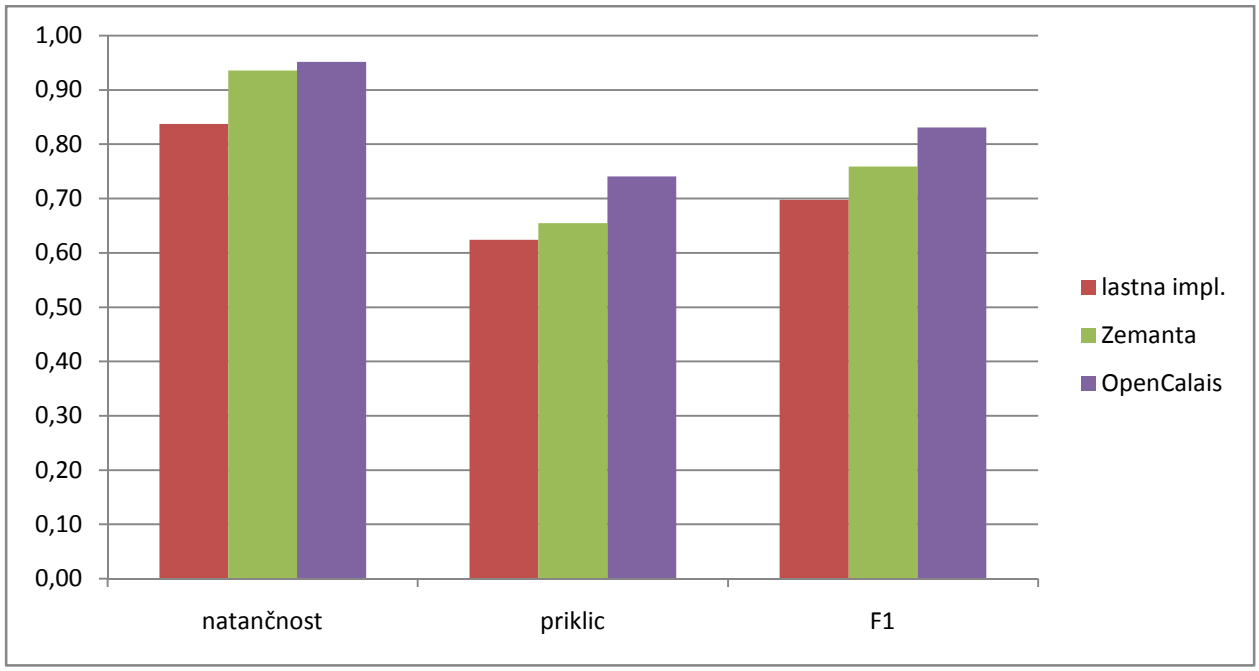


Slika 9: Metrika F_1 po storitvah in dokumentih

Agregirani povprečni kazalci za posamezne storitve pa so sledeči:

	Lastna impl.	Zemanta	OpenCalais
Natančnost	0,84	0,94	0,95
Priklic	0,62	0,65	0,74
F_1	0,70	0,76	0,83

Iz tabele kot tudi slike 10 je jasno razvidno, da ima lastna implementacijo nekaj slabšo natančnost (0,84) od ostalih dveh servisov (Zemanta 0,94 oz. OpenCalais 0,95). Pri priklicu je razlika z Zemanto manjša, saj znaša priklic za lastno implementacijo 0,62, za Zemanto pa 0,65. Občutno boljši od obeh pa je priklic pri OpenCalaisu, ki znaša kar 0,74. Metrika F_1 ne pokaže ničesar novega, česar ne bi razkrila že natančnost in priklic.



Slika 10: Povprečni kazalniki za primerjane storitve

4.3 Interpretacija rezultatov

Kot že zapisano, je zaradi razlik v načinih delovanja težko postaviti vse tri storitve v isti koš in opraviti kvalitetno primerjavo. Tu je zaradi potrebnega ročnega dela za preverjanje kvalitete rezultatov testiranje potekalo na zelo majhnem naboru razmeroma kratkih dokumentov. Zato je potrebna previdnost pri interpretaciji rezultatov, slednji pa se morajo razumeti zgolj kot približni. Kljub temu rezultati jasno potrjujejo praktične izkušnje, ki kažejo, da se najbolje odreže OpenCalais. Pri uporabi le-tega le redko naletimo na nepravilno prepoznano entiteto. To potrjuje tudi najvišja natančnost na testu. Malo pogosteje se pripeti, da katera izmed imenovanih entitet manjka. Vendar je pri priklicu testiranje pokazalo najvišjo vrednost izmed vseh primerjanih storitev.

Za primerjavo z lastno implementacijo semantičnega označevanja nam najbolj ustreza Zemanta. Čeprav je tudi tu opazna kvalitetna razlika med rezultati lastne implementacije in komercialno Zemanto, pa je ta veliko manjša kot v primeru OpenCalais-a. Na podlagi povprečja metrike F_1 lahko povzamemo, da je razkorak med lastno implementacijo in Zemanto podoben razkoraku med Zemanto in OpenCalaisom. Povprečni kazalniki kažejo, da lastna implementacija in

Zemanta vračata podobne rezultate tako po natančnosti kot tudi priklicu. Tu je Zemanta v rahli prednosti, ki se odraža predvsem pri natančnosti (0,94 napram 0,84 pri lastni implementaciji). Podobnost pa potrjujeta tudi diagrama kazalnikov (sliki 7 in 8) po dokumentih, saj se krivulji natančnosti in priklica obeh storitev deloma prilegata. Če primerjamo Zemanto z OpenCalaisom, pa vidimo, da je le-ta podobno natančen, a ima boljši priklic in konsistentnejše rezultate. Zemanta je torej natančna, a vrača manj rezultatov, posledično pa je več tudi neprepoznanih imenovanih entit, ki manjkajo med rezultati.

Kljub začetnim predpostavkam o težavnosti testiranja smo dobili jasne in razmeroma konsistentne rezultate, ki potrjujejo praktične izkušnje. Pričakovati je bilo, da se bosta komercialni rešitvi, v katere je bilo vloženih nekaj 100-krat več ur razvoja kot v lastno implementacijo, odrezale bistveno bolje in za kvalitetno raven ali dve višje. Vendar pa so rezultati testiranja vzpodbudni tudi za lastno implementacijo, saj razkorak v kvaliteti ni dramatičen. Poleg tega je pri tej še precej prostora za izboljšave. S to tematiko pa se ukvarja naslednje poglavje.

5 Možne izboljšave in nadaljni razvoj

Med razvojem lastne implementacije semantičnega označevanja sem opazil veliko priložnosti za izboljšave, ki jih bodisi zaradi omejenega časa bodisi zaradi drugih omejitev nisem uspel realizirati. Najbolj smiselne izmed teh izboljšav navajam v spodnjih razdelkih.

5.1 Boljše prepoznavanje imenovanih entitet

Ker je testiranje pokazalo, da je ena glavnih težav lastne implementacije v neprepoznanih imenovanih entitetah, je izboljšanje prepoznavanja teh entitet še kako zaželjeno. Tudi natančnost prepoznanih entitet bi lahko bila za odtenek boljša, kar kaže primerjava z ostalimi storitvami. Iz navedenega lahko sklepamo, da je prostora za izboljšave precej in da bodo te izvirale ravno iz tega področja.

Prva izboljšava prepoznavanja entitet je integracija DBPedie v komponento ANNIE, ki sedaj služi prepoznavanju imenovanih entitet. Trenutno je uporaba DBPediinih podatkov omejena na drugo in tretjo stopnjo v verigi označevanja besedila. Ideja je ta, da se DBPediini podatki uporabijo tudi za pomoč pri prepoznavanju imenovanih entitet. ANNIE med drugim uporablja tudi veliko število t.i. gazetter seznamov, ki so v standardni distribuciji GATE-a že napolnjeni s podatki. Tu bi namesto priloženih seznamov pripravili lastne sezname (ali presek obeh) s pomočjo DBPediinih podatkovnih naborov. Le-ti so veliko večji in ažurnejši kot priloženi GATE gazetteer seznami, kar se opazi tudi med testiranjem (npr.ANNIE ne prepozna entitete »Facebook«). S tem bi se izboljšala tudi klasifikacija imenovanih entitet (ljudje, kraji, organizacije...), kar bi lahko v nadaljevanju koristno uporabili v koraku razločanja med pomeni.

Če pa z rezultati ANNIE še vedno ne bi bili zadovoljni, bi lahko eksperimentirali tudi s katero drugo knjižnico NLP. Pri uporabi komponente ANNIE se pojavlja razmeroma veliko napačnih zadetkov, še posebno, kadar imamo nastavljeno prepoznavanje »neznanih« entitet, ki jih ANNIE ne uspe klasificirati. Zato se zastavlja vprašanje, ali je lahko pri tem početju katera izmed alternativnih knjižnic učinkovitejša. Ker trenutna integracija komponente za prepoznavanje entitet ni tesna, tudi zamenjava ne bi smela biti preveč boleča. Namesto ANNIE se lahko uporabi druge knjižnice za prepoznavanje imenovanih entitet, kot so npr. odprtokodni OpenNLP, Stanford CRF NER, Apache UIMA ali Balie.

5.2 Boljše razločanje med pomeni in identifikacija entitet

Pri razločanju med pomeni je bilo opaziti znatno manj težav kot pri prepoznavanju imenovanih entitet. Kljub temu se je porodilo nekaj idej, kako bi lahko proces razločanja in povezovanja z DBPedia entitetami še izboljšali.

5.2.1 Uporaba krnilnika pri izračunavanju semantične bližine

V procesu razločanja med pomeni se v pomnilniku izgradi hitri indeks Lucene, ki obsega vse možne pomene entitete in njihove oznake in opise. Opisi so sestavljeni iz ene do nekaj povedi. Nato se nad indeksom izvede poizvedba, katere rezultati so vsi možni pomeni, urejeni po semantični bližini do referenčne entitete. Tu bi se najverjetneje uporaba krnilnika (stemmer) izkazala za učinkovitejšo metodo kot je standardna analiza, ki se sedaj uporablja pri izgradnji indeksa. Ta je v Lucenu prednastavljena in koraka krnjenja ne vključuje. Krnjenje besedam odstrani vse pripone in predpone, ki so posledica različnih pregibanj ali pretvorb besede. V splošnem to pomeni, da se besede zreducirajo na najmanjši skupni imenovalec – koren besede. Le-ta predstavlja idealen temelj za ugotavljanje semantične bližine. Ker v našem primeru ne potrebujemo iskanja po dobesednih terminih, ampak uporabljamo iskalnik in indekser z namenom ugotavljanja semantične bližine, lahko pričakujemo, da s pomočjo krnilnika ob izgradnji indeksa izboljšamo rezultate možnih pomenov, ki so posledično pravilneje urejeni po semantični bližini do referenčne entitete.

5.2.2 Čiščenje podatkovnega nabora DBPedie

Druga izboljšava za razločanje pomenov je povezana z vnaprejšnjo pripravo podatkov. Trenutno se za razločanje uporablja celoten podatkovni nabor DBPedie, ki vsebuje vse prepoznane entitete. Težava je v tem, da vse entitete z DBPedie ne spadajo med možne imenovane entitete, saj DBPedia vsebuje tudi bolj splošne koncepte. Eden takih primerov so dnevi v tednu, ki so v DBPedii zavedeni enakovredno z ostalimi entitetami. Ti se v besedilu ne morejo samostojno pojaviti kot imenovana entiteta, razen seveda če gre za del ali celoten naziv knjige, organizacije, ipd - kar seveda pomeni tudi drugo entiteto. Če bi bilo potrebno povsem ročno čiščenje, bi taka izboljšava lahko vzela zelo veliko časa. Ker pa so DBPediine entitete klasificirane v več shem kategorij (YAGO, OpenCyc, lastna ontologija DBPedie), obstaja alternativna pot. Klasifikacija omogoča, da ročno določimo kategorije, ki vsebujejo potencialne imenovane entitete. Nato preko skripte prečistimo podatkovni nabor entitet DBPedie ter izločimo vse entitete, ki niso uvrščene v ustrezne kategorije, ter ponovimo uvoz in izgradnjo indekserja DBPedie.

5.2.3 Upoštevanje podatka o razredu entitete

Uporabljena komponenta za prepoznavanje imenovanih entitet ANNIE omogoča tudi klasifikacijo prepoznanih entitet po preprostih razredih: ljudje, kraji, organizacije, dogodki, ipd. Ta podatek je zaenkrat neizkoriščen, ker praktične izkušnje kažejo, da znaten delež prepoznanih entitet ANNIE uvrsti kar v splošen razred »neznano«. Vendar pa je za pričakovati, da se bo učinkovitost z zgoraj omenjeno izboljšavo integracije DBPediinih podatkovnih naborov bistveno izboljšala. V tem primeru bi bilo smiselno te podatke uporabiti tudi v koraku razločanja med pomeni. Eden od kriterijev za razvrščanje možnih pomenov entitete po semantični bližini bi tedaj lahko bila tudi oddaljenost razreda entitete možnega pomena od referenčne entitete, kar bi gotovo prispevalo k izboljšanju kakovosti razločanja med možnimi pomeni.

5.3 Dodatne funkcionalnosti

Med dodatne funkcionalnosti spadajo tiste izboljšave, ki neposredno ne vplivajo na kvaliteto semantičnega označevanja, temveč zgolj povečajo uporabnost oznak in znižajo zahtevnost uporabe. Ena od teh je tudi klasifikacija oznak. V tem trenutku semantične oznake, ki so rezultati obdelave storitve, ne vsebujejo podatkov o razredu entitete in kategorijah, med katere se le-ta uvršča. Seveda so ti podatki na voljo posredno, skozi uporabo prosto dostopne DBPedie, s katero so povezane semantične oznake. To pa tudi pomeni en korak več za uporabnika naše storitve, da pride omenjenih podatkov. Za uporabnika bi bilo enostavneje, če bi ta dva podatka vsebovale že kar semantične oznake same. Implementacija take funkcionalnosti je preprosta. Potrebno je le pripraviti podatkovno bazo in zgraditi indeks, ki vsebuje tudi podatkovne nabore DBPedie, kateri vsebujejo podatke o kategorijah in razredu entitete. Semantične oznake se nato tik pred vračanjem rezultatov obogatijo s temi podatki.

6 Sklepne ugotovitve

Semantično označevanje predstavlja enega izmed prvih korakov k razumevanju besedila za širši krog razvijalcev in uporabnikov na svetovnem spletu. Pojav lahko dostopnih in prijaznih komercialnih storitev omogoča, da se razvijalci programske opreme in spletnih strani osredotočijo na razvoj naprednih funkcionalnosti. Ni jim potrebno razvijati zahtevnih orodij in komponent NLP, ki se trudijo razvozlati primarno orodje človeške komunikacije – naraven jezik. Sodobne storitev za semantično označevanje so preproste za uporabo in svojo nalogo zadovoljivo opravijo. Skupaj z uporabo povezanih podatkov LOD postanejo semantične oznake resnično uporabne in zmogljive. Ideja povezanih podatkov, kjer so podatki na voljo v strukturirani programsko dostopni in računalniško razumljivi obliki, medsebojno poljubno prepleteni in povezani, je zelo dobrodošla. Omogoča namreč, da se uporabniki prikopljejo do poljubno podrobnih podatkov o entitetah iz besedila. To narekujejo njihove potrebe, ki določajo, kako globoko po povezanih naj rudarijo. Splet povezanih podatkov zelo hitro raste in lahko zgolj predvidevamo, kako globoke bodo ter kje se bodo ustavile meje dostopnosti podatkov.

Če pa imajo uporabniki teh storitev specifične potrebe po označevanju v določeni domeni in nekaj razvojnih virov, pa se lahko lotijo implementacije semantičnega označevanja tudi sami. Moja implementacija je vrste »preverba koncepta« in kot taka dokazuje, da je mogoče v razmeroma kratkem času implementirati zadovoljivo storitev semantičnega označevanja. Odprte komponente in orodja za obdelavo in razumevanje jezika zadovoljivo opravljajo svojo nalogo. Tudi rast spleta povezanih podatkov je in bo v prihodnosti zaslužna za boljšo učinkovitost in prepoznavanje entitet. Še posebej smiselna je lastna implementacija v primerih, kjer se cilja na označevanje besedil v neki določeni domeni, kjer se lahko splošnonamenske storitve za semantično označevanje obneseje slabše.

Pričakujemo lahko, da se bodo s časom pojavljale nove storitve semantičnega označevanja z dodatnimi funkcionalnostmi in še višjo kvaliteto rezultatov. Najverjetneje se bo na tem področju oblikovala kopica najzmogljivejših vodilnih storitev, ki bodo postale pogosta izbira razvijalcev. Stestirani OpenCalais je zaradi nadpovprečnih rezultatov že eden od možnih kandidatov. Prav tako lahko pričakujemo, da se bo zaradi pojava odprtih storitev in povezanih podatkov cena uporabe takih storitev znižala ali postala brezpredmetna. S tem pa bo tudi semantični splet korak bližje realnosti.

7 Priloge

7.1 Testni dokumenti

7.1.1 Thousands gather to hear, cheer Iran's Michelle Obama (dokument 1)

Dancing in public in not allowed in **Iran**, but thousands could hardly contain themselves at a recent presidential campaign rally in the capital city, **Tehran**.

Supporters hope **Zahra Rahnavard** will become Iran's future first lady.

On this day, the deafening cheers were not for presidential hopeful **Mir Hossein Mousavi**, but rather for his wife -- a woman some are calling Iran's **Michelle Obama**.

The comparisons to the first lady of the **United States** stem from the role Zahra Rahnavard is playing in her husband's quest for the presidency.

Never in the history of **Iranian presidential elections** has a candidate put his wife in the forefront of his campaign.

Wherever Mousavi -- a centrist candidate -- goes, Rahnavard is usually nearby. "We look at her and we say, 'we want to be like her in the future, ' " said **Shakiba Shakerhosseie**, one of 12,000 people who packed into Tehran's indoor **Azadi (Freedom) sports stadium** to hear Rahnavard speak.

Iran became an **Islamic republic** in 1979 after the ruling monarchy was overthrown and **Shah Mohammad Reza Pahlavi** was forced into exile.

The revolution also ended the ceremonial role of first lady that the last queen, **Farah**, enjoyed.

At this rally, Rahnavard -- a writer and art professor -- spoke for her husband, who was campaigning elsewhere.

Wearing a floral headscarf and a traditional black chador -- a full-length loose robe that women in Iran wear like a cloak -- Rahnavard called for freedoms she says were lost during **President Mahmoud Ahmadijenad**'s term.

"I hope freedom of speech, freedom of the pen and freedom of thought will not be forgotten," she said.

The crowd, which was clad in Mousavi's trademark color green, cheered wildly. It waved placards with his picture and swayed from side to side, chanting and beating drums.

The women sat on one side; the men on the other.

The overwhelming majority were young voters, many of whom said they attended because of Mousavi's wife, a mother of three.

Iran's population -- estimated at more than 66 million -- has a median age of 27.

"I am really angry here in Iran with the position of women," said **Saghar Kouhestani**, adding that she supports Mousavi because of his wife.

Mousavi, a former prime minister, is considered a threat to **Ahmadinejad**, a hard-liner, in the June 12 elections. He is credited for successfully navigating the Iranian economy during a bloody eight-year war with Iraq in the 1980s.

Over the weekend, the **Iranian government** blocked access to the social networking site **Facebook**, where Mousavi has a page with more than 5,000 supporters, the semi-official **Iranian Labor News Agency (ILNA)** said.

Those attempting to visit Facebook received a message in Farsi saying, "Access to this site is not possible."

Political science professor **Mohammad Marandi** downplays Rahnavard's impact. She may win over reformists and women, he says, but what will win the election is a solution to the floundering economy and a strong performance in the debates.

"If Ahmadinejad does well in the debates, I don't think anyone will be able to defeat him," Marandi said.

But try selling that to Rahnavard's enthusiastic supporters.

"This is the first time after the **Revolution** we see a lady behind the president," said **Farhad Mahmoudi**. "And this is why we're so happy because we can have a first lady."

7.1.2 Royal chauffeur suspended after alleged palace security breach (dokument 2)

A royal chauffeur was suspended Sunday after he allegedly allowed undercover reporters from a **British tabloid** to enter **Buckingham Palace** in exchange for cash.Buckingham Palace has been the subject of high-profile security breaches before.

Mazher Mahmood, of the London-based **News of the World**, claimed he was allowed to enter the **London** residence of **Queen Elizabeth II** without security checks after paying a man identified as a **Buckingham chauffeur** £1,000 ($1,591).

Footage of the incident filmed undercover showed the chauffeur giving Mahmood, whose face was blurred, a tour of the royal garage and, at one point, allowing him to sit in one of the vehicles.

A palace spokesperson told **CNN** that the chauffeur "has been suspended pending further investigation."

Britain's Press Association named the chauffeur as **Brian Sirjusingh** and added that he had been suspended following reports that two reporters, posing as wealthy **Middle-Eastern** businessmen, were allowed into the palace

Meanwhile, Mahmood's editor, **Robert Jobson**, told **ITN**: "Nobody stopped him, nobody actually challenged him. It actually exposes a serious lapse in security at Buckingham Palace."

Jobson added that even senior members of the royal family and longtime staffers are required to present photo identification cards upon entry to the palace.

"Our investigator is sitting where the queen sits in the royal limo," Jobson said, referring to the video. "And the fact is, we've been told that security has been tightened up, that these things wouldn't happen again, new rules and regulations were brought in -- they simply haven't worked."

Buckingham Palace has experienced a number of high-profile security lapses in the past. In 2003 an investigation was launched after "comedy terrorist" **Aaron Barschak** gatecrashed Prince William's 21st birthday party at **Windsor Castle**, **PA** reported.

Wearing a dress, beard and sunglasses, Barschak climbed on stage as the prince addressed the crowd, and kissed him on both cheeks.

That same year, a journalist with the **Daily Mirror** newspaper spent two months "working undercover" as a palace footman.

7.1.3 NASA to try California shuttle landing (dokument 3)

NASA will attempt to land space shuttle **Atlantis** at **California's Edwards Air Force Base** on Sunday, after rainy **Florida** weather precluded a **Kennedy Space Center** landing for a third day, officials said.

Rain at Kennedy Space Center in Florida canceled plans to land the space shuttle Atlantis on Saturday.

The first attempt will be made at 11:39 a.m. ET at Edwards, north of **Los Angeles**. Another opportunity will come at 1:17 p.m. ET.

Rainy weather postponed the shuttle landing on Friday and Saturday. While Atlantis could conceivably remain in space until Monday, NASA has said it wants to land Sunday.

Officials said Sunday's Florida weather was better than conditions Saturday, but atmospheric conditions in Florida remained too unstable for landing.

The landing would be the 53rd at Edwards, NASA officials said. In the early days of the space shuttle program, Edwards was its primary landing site.

Atlantis launched May 11 for NASA's final repair visit to the **Hubble Space Telescope**. Shuttle astronauts conducted space walks during the mission to perform routine repairs and replace key instruments, in what has been called one of the most ambitious space repair efforts ever attempted.

Hubble was released back into orbit Tuesday morning.

Hubble, which has been in space for nearly two decades, can capture clear images that telescopes on **Earth** cannot, partly because it does not have to gaze through murky atmospheres.

7.1.4 Boxer 'worried' about transferring Gitmo detainees (dokument 4)

California Sen. **Barbara Boxer** told **CNN** Sunday she and her **Senate** colleagues are "worried" about the possibility of transferring current **Guantanamo Bay** detainees into the United States, adding she is awaiting a more comprehensive plan on the matter from **President Obama**.

"We only have one max security prison in California and it's, right now, overbooked, that's the case," Boxer told CNN's **John King** on **State of The Union**. "In all, we are worried and we want to to see what the plan is."

Boxer's comments come days after Senate **Democrats** voted to withhold funding to close the Guantanamo Bay facility until the president lays out a more detailed plan on where the current 240 detainees will go.

Alabama Sen. **Richard Shelby**, also appearing on State of The Union, cautioned the president against relocating the most "incorrigible" detainees in the **United States**.

"Nobody wants them. We got all kinds of places in the world we can house these people," Shelby said. "If we have to move them from **Cuba**, from **Gitmo**, we have other territory that can bring them in, but don't bring them to the United States of America.

In a high profile speech Thursday, Obama pledged no detainees would be brought to the United States "if it would endanger our national security."

7.1.5 King: Luck running low in Las Vegas (dokument 5)

A search online turns up rooms at **Fitzgeralds Casino and Hotel** for as low as $26 -- a sure sign the recession is taking a toll on **Las Vegas** and rates are being slashed to attract visitors.

Judy Bagley, a casino cashier who traded chips for cash and vice versa, found out about recession's toll another way.

"My supervisor came and said I had to close the booth and she was going to count me out and I was to go and meet with the manager and director," Bagley told **CNN**. "When I went up there they said my services were no longer needed and my job was being eliminated."

So three months ago, after more than 28 years at Fitzgeralds, she was out of work.

"It felt like I was more or less stabbed in the back after all the years I had been there," Bagley said. "I'd been very loyal to the company and never called in sick and I have very little discipline [issues] and it felt like betrayal."

The sting became a little less personal as she watched more and more friends and colleagues lose their jobs.

"It was probably the economy with the banks in bad trouble and things like that," she said. "You know this is not any fault of the casinos. ... It's never been this bad before, with all the foreclosures that are going on in the city and all over this state, with people being laid off and jobs eliminated."

At Fitzgeralds the motto is "**Luck of the Irish**." But, by the numbers, this is hardly a lucky time in Vegas:

- **Nevada's** unemployment rate is 10.4 percent.

- The number of visitors arriving in Vegas by plane in March 2009 was down nearly 12 percent from March 2008, according to the **Las Vegas Convention and Visitors Authority**.

- Attendance at Las Vegas conventions was off 30 percent in that same March to March comparison.

- And the average daily hotel room rate -- they drop when the vacancy rate is up -- was down more than 31 percent in the city's March 2008-March 2009 comparison.

"These are times completely different than anything I have experienced in my lifetime," **Mayor Oscar Goodman** told CNN in his City Hall office. "I didn't see this coming, and when it hit it hit virtually overnight."

The mayor believes the troubles were exacerbated by a remark **President Obama** made back in early February during a dustup about spending by financial institutions taking federal bailout money.

"You can't get corporate jets, you can't go take a trip to Las Vegas or go down to the **Super Bowl** on the taxpayer's dime," Obama had said.

Mayor Goodman believes Obama made a mistake in mentioning Las Vegas, but sees a chance to rectify the problem. Obama is scheduled to travel to Las Vegas in the coming week, and Goodman is hoping to make a point.

"If I ask him the question, 'Mr. President, isn't Vegas a great place to do business?' as he is standing here in Las Vegas, I hope he says yes."

Regardless, Goodman sees signs that the worst is over.

"Now a lot of people will always be pessimistic -- and for them the glass is always going to be half empty," Goodman said. "I am a half-full guy."

Bagley has a harder time being optimistic.

Out of work for three months means lifestyle changes.

"We used to go out to eat a couple times a week. We don't do that anymore. I clip a lot of coupons to try to try to save money that way," said Bagley, whose husband is retired.

Just about every day she visits the **Culinary Workers Union** hall to search a jobs database.

Cashier jobs like the one she held at Fitzgeralds are hard to find -- in part because of the recession and in part because more and more of those kinds of jobs are automated. So she is expanding her search.

"I've put in applications at some of the grocery stories here in town for cashier work," she said. "You put in an application and you have 150 to 200 people applying for the same job. It is very hard to get the work."

She does not, however, assign any of the blame to Obama and the remark Goodman believes cost the city some business.

"Absolutely not," she said. "I think President Obama is doing the very best that he can and I am impressed with him and I hope that he continues to do what he feels is right for the country because I know he has our best interests at heart."

8 Literatura

[1] T. Berners-Lee, J. Hendler in O. Lassila, “The Semantic Web,” Scientific American Magazine, maj 2001. Dostopno na: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[2] Wikipedia, Semantic Web. Dostopno na: http://wikipedia.org/wiki/Semantic_web

[3] I. Herman, “Introduction to the Semantic Web,” predstavljeno na Semantic Technology Conference, San Jose, California, USA, 2009.

[4] T. Štajner, “Razločevanje entitet v besedilih s strojnim učenjem in predznanjem,” diplomsko delo na univerzitetnem študiju, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija, 2009.

[5] DBPedia, “About DBPedia,” dec. 2008. Dostopno na: http://dbpedia.org/About

[6] DBPedia, “The DBPedia Data Set”, nov. 2008. Dostopno na: http://wiki.dbpedia.org/Datasets

[7] OpenRDF, About OpenRDF. Dostopno na: http://www.openrdf.org/about.jsp

[8] OpenRDF, Sesame System Documentation 2.0. Dostopno na: http://www.openrdf.org/doc/sesame2/system/

[9] R. Lee, “Scalability Report on Triple Store Applications,” Simile Project, jul. 2004. Dostopno na: http://simile.mit.edu/reports/stores/stores.pdf

[10] Apache, “Apache Lucene – Scoring,” jun. 2009. Dostopno na: http://lucene.apache.org/java/2_4_1/scoring.html

[11] D. Mladenić in M. Grobelnik. Text and web mining. Data mining and decision support : integration and collaboration, (The Kluwer international series in engineering and computer science, SECS 745). Boston; Dordrecht; London: Kluwer Academic Publishers, 2003, str. 15-22.

[12] Wikipedia, Vector space model. Dostopno na: http://en.wikipedia.org/wiki/Vector_space_model

[13] GATE. Dostopno na: http://gate.ac.uk/index.html

[14] GATE, ANNIE : a Nearly-New Information Extraction System. Dostopno na: http://gate.ac.uk/sale/tao/splitch8.html#x10-2040008

[15] OpenCalais, OpenCalais Web Service Overview. Dostopno na: http://www.opencalais.com/documentation/calais-web-service-api

[16] Zemanta, Zemanta API Companion. Dostopno na: http://developer.zemanta.com/docs/Zemanta_API_companion

[17] C. van Rijsbergen, *Information Retrieval*. London: Butterworths, 1979.

9 Seznam tabel in slik