# Analiziranje sadržaja tekstova korišćenjem ontologije

Dejan Prodanović

**Sadržaj** — ***U ovom radu predstavljen je metod za analiziranje sadržaja tekstova korišćenjem ontologije. Pomoću opisanog metoda se u tekstu izvršava pretraživanje svakog koncepta iz ranije napravljene ontologije. Zatim se izvršava ocenjivanje pronađenih koncepata. Ovim ocenjivanjem se dobija mera zastupljenosti određene teme kojoj koncept pripada Prilikom realizacije sistema korišćena je i činjenica da za jedan koncept može biti definisano više sinonima, a analizu reči u tekstu treba prilagoditi jeziku gde su izražene njihove morfološke promene.***

***Ključne reči*** **— koncept, ontologija, pretraživanje.**

## I. Uvod

Savremeni čovek je okružen velikim brojem informacija u različitim oblicima, a najčešće u elektronskim. Na taj način danas postaje najveći problem kako među tolikim brojem elektronskih dokumenata pronaći upravo ono što je potrebno. Ovako velika mogućnost primene automatskog pretraživanja i ocenjivanja određenog teksta, ova istraživanja definišu kao jedna od trenutno najinteresantnijih.

Sa druge strane duži niz godina u okviru, kako veštačke inteligencije, tako i drugih oblasti računarske nauke, koncept ontologije je često korišćen. Pojavom semantičkog Weba i Web servisa, ontologija je dobila primenu i u okviru servisa pretraživanja elektronskih dokumenata.

Upravo na osnovu osobina koncepta ontologije, realizovan je algoritam za analizu i ocenjivanje sadržaja tekstova koji je opisan u ovom radu. Opisani algoritam može imati veliku primenu u automatizaciji pronalaženja odgovarajućih koncepata iz dokumenata.

U samom radu nakon prikaza koncepta ontologije, opisuje se postupak formiranja rečnika iz analiziranih tekstova. Zatim se objašnjava sam postupak analize primenom ontologije, i verifikacija algoritma pomoću dobijenih rezultata.

## II. ONTOLOGIJA

Semantičke mreže i tehnologije semantičkih mreža nude mogućnost obrade i kontrole podataka i procesa, čime se stvara osnova za kreiranje i korišćenje semantičkih metapodataka. Navedeni metapodaci mogu postojati na dva nivoa. Sa jedne strane mogu opisati web stranu ili deo nekog dokumenta, a sa druge oni mogu opisivati entitete unutar dokumenta kao i relacije koje postoje između njih. Definicija ontologije [7] naglašava dve centralne tačke: da je konceptualizacija formalna i dakle zahteva izvođenje zaključaka od strane računara; i da se konkretna ontologija kreira za određeni domen od interesa.

U praksi, razvoj ontologije uključuje:

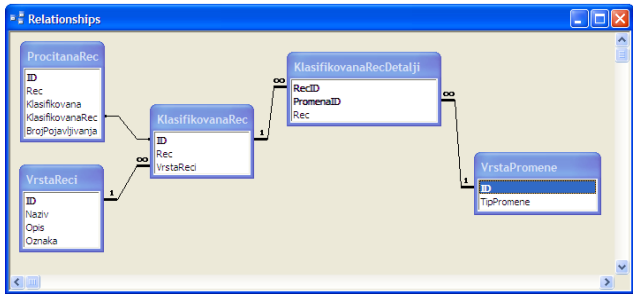
- definisanje klasa u ontologiji
- raspoređivanje klasa u hijerarhiji
- definisanje svojstava instanci i skupa dozvoljenih vrednosti za instance
- unošenje vrednosti za skup instanci

Prilikom kreiranja ontologije treba se pridržavati osnovnih pravila:

- ne postoji najispravniji način (šablon) kako se kreira ontologija u domenu
- uvek treba razmišljati o mogućnosti proširenja ontologije, odnosno da je rad na određenoj ontologiji iterativni proces
- koncepti u ontologiji treba da budu bliski objektima (fizičkim ili logičkim) i relacijama u domenu od interesa. Kao primer mogu poslužiti pojmovi poput imenica ili glagola u okviru rečenica

## III. FORMIRANJE REČNIKA

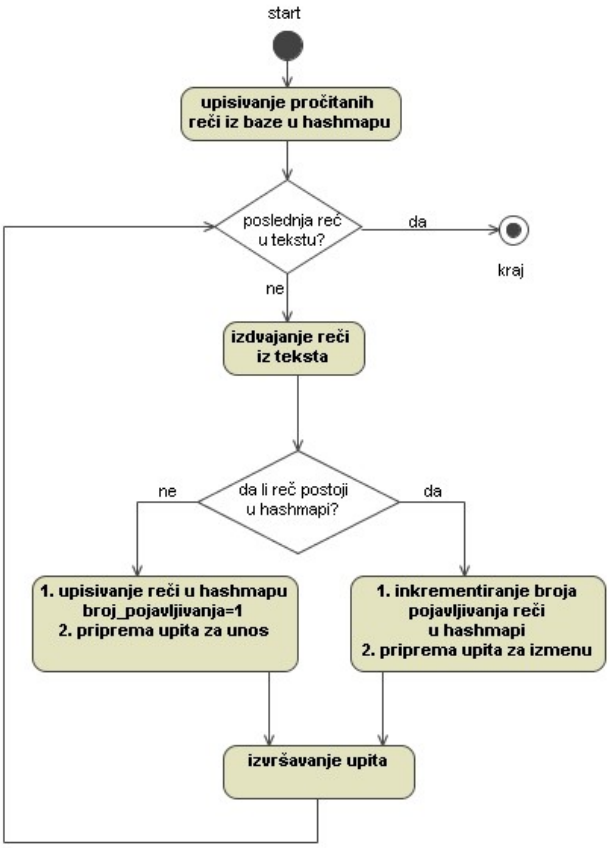
U procesu analize sadržaja elektronskih dokumenata, prvi korak je formiranje rečnika. U ovom koraku se sve reči iz analiziranih tekstova smeštaju u bazu podataka, čiji je relacioni model prikazan na slici 1. U datom modelu najbitniji entiteti su: *tabela pročitanih reči* i *tabela klasifikovanih reči*.




Sl 1. Relacioni model baze podataka

U prvoj fazi kreiranja rečnika, analizirani su reprezentativni tekstovi iz knjiga i skripti koje su korisćene u realizaciji nastave iz predmeta oblasti

arhitekture i organizacije računara. Na slici 2, prikazan je dijagram aktivnosti u ovoj fazi kada se analizira reprezentativni tekst i reči pročitane iz teksta upisuju u bazu podataka.

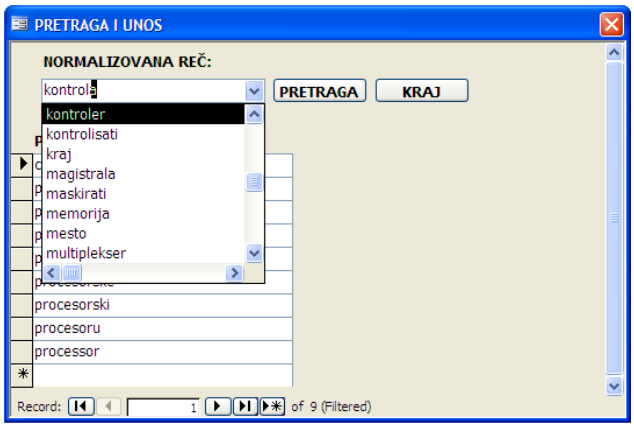



Sl 2. Dijagram aktivnosti u fazi analize teksta i upisa pročitane reči u bazu podataka

Svi zapisi iz tabele pročitanih reči upisuju se u strukturu podataka, koja je tipa hashmap-a, i gde je ključ pročitana reč. Za svaku pročitanu reč iz teksta proverava se da li ona postoji u mapi pročitanih reči. Ako postoji, inkrementira se ukupan broj pojavljivanja reči i kreira upit za izmenu zapisa u bazi, inače kreira se nov zapis u mapi i kreira upit za unos zapisa u bazi. Nakon toga, izvršava se upit i sve promene u mapi reflektuju se i u bazi.

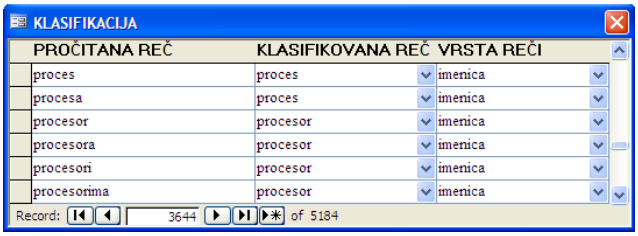
Prilikom obrade reprezentativnog teksta svaka pročitana reč se upisuje u tabelu pročitanih reči i beleži se njena ukupna frekvencija. Naknadno, u drugoj fazi formiranja rečnika vrši se klasifikacija reči od strane eksperta u domenu. Svaki zapis u tabeli pročitanih reči ima referencu (strani ključ) ka primarnom ključu (ID) u tabeli klasifikovanih reči. U tabeli pročitanih reči trenutno se nalazi 5194 reči. Prilikom implementacije korišćena je MySql baza podataka koje je pokazala i više nego zadovoljavajuće perfomanse. Na slici 3 prikazana je forma za pretragu po svim rečima koje su klasifikovane (normalizovane). Kao primer mogu poslužiti reči kao što su: *processor, CPU, microprocessor*.

Nakon kreiranja rečnika, izdvojene su reči koje se po broju pojavljivanja mogu svrstati u skup jako frekventnih reči. Rečnik se koristi u fazi pronalaženja reči izdvojenih u procesu obrade tekst i služi da bi se dobila bolja ocena samog sadržaja dokumenta.




Sl 3. Forma za pretragu po svim rečima koje su klasifikovane

U fazi pretraživanja proverava se da li pročitana reč ima referencu ka klasifikovanoj reči i ukoliko nema, da li postoji reč iz skupa klasifikovanih reči gde je težinsko rastojanje [1] manje od definisanog praga.



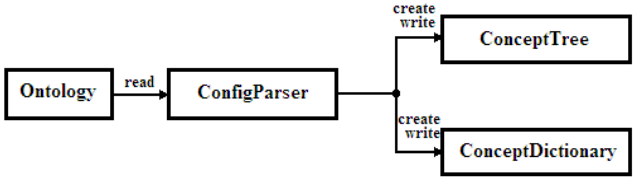

Sl 4. Forma za klasifikovanje pronađenih reči

## IV. PRIMENA ONTOLOGIJE

U daljem postupku korišćenjem Java framework-a Jena šđ, kreiran je metod za parsiranje pomoću kojeg se prolaskom kroz hijerarhiju ontologije formiraju odgovarajuće strukture podataka, koje će se koristiti u fazi pronalaženja i ocenjivanja koncepata, kao što je prikazano na slici 5.

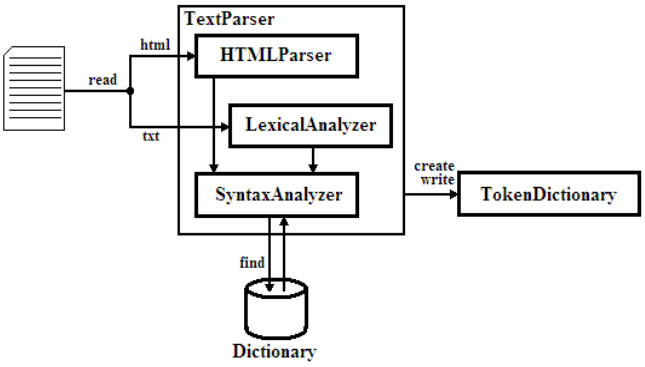
U stablu koncepata **(ConceptTree)** nalazi se hijerarhija klasa iz ontologije, gde svaka klasa odgovara konceptu koji reprezentuje određeni entitet ili proces. Ovakva struktura je pogodna pri pronalaženju svih direktnih potomaka za određeni koncept u fazi ocenjivanja.

Rečnik koncepata **(ConceptDictionary)**, koji ima osobine heš mape, koristi se u fazi pronalaženja koncepata u dokumentu gde se smešta lista sa pozicijama koncepta u tekstu i za upis ocene koncepta.




Sl 5. Entiteti i strukture podataka u fazi parsiranja ontologije

Početni korak u analizi jeste čitanje datoteka. Trenutno su implementirani parseri za html i tekstualne datoteke. Za svaku pročitanu reč iz datoteke traži se njen koren u rečniku **(SyntaxAnalzyer)**, a nakon toga vrši upis u rečnik pročitanih reči **(TokenDictionary)**. Rečnik pročitanih reči ima osobine heš mape i za svaku pročitanu reč pored njenog osnovnog oblika sadrži i listu sa pozicijama reči u tekstu.




Sl 5. Parsiranje dokumenata i kreiranje rečnika sa pročitanim rečima

Pod pozicijom reči podrazumevaju se sledeći atributi: redni broj reči u tekstu, indeks reči (redni broj znaka u odnosu na početak teksta) i dužina reči.

Dalje postoje dve bitne faze: *faza identifikacije* koncepta i *faza registracije* identifikovanog koncepta.

Faza identifikacije koncepta se obavlja tako što se iz rečnika sa konceptima, za svaki koncept vrši sažimanje reči. Iz rečnika sa sažetim rečima koji je formiran prilikom leksičke analize teksta, traže se liste sa pozicijama za svaku saženu reč iz koncepta. Utvrđuje se da li u listama postoje kombinacije sa rednim brojevima reči kod kojih je razlika 1. Rad ove faze, identifikacija koncepta, biće ilustrovan na primeru koncepta: two's complement. Tekst za analizu dat je [8]. Sažimanjem ovog koncepta i pronalaženjem listi sa pozicijama za sažete reči, dobija se rezultat koji je prikazan u tabeli 1.

U fazi registracije, kada je koncept identifikovan, potrebno je utvrditi da li postoji preklapanje u rečima između identifikovanog koncepta i postojećih koncepata iz skupa registrovanih koncepata. Za svaki redni broj reči u konceptu ubacuje se referenca na koncept kome ta reč pripada. Ukoliko za određeni redni broj reči već postoji koncept u skupu registrovanih koncepata, vrši se poređenje dužine postojećeg i identifikovanog koncepta i upisuje se onaj koncept čija je broj reči veći. Ako je dužina identifikovanog koncepta veća, vrši se njegovo prepisivanje preko postojećeg koncepta i inkrementira broj njegovog pojavljivanja u tekstu. Broj pojavljivanja postojećeg koncepta se dekrementira.

Ocena koncepta predstavlja meru zastupljenosti određene oblasti u dokumentu gde je koncept pronađen. Posmatrajući međusobne relacije između koncepata, njihove različite nivoe u semantičkom podstablu i pojavljivanje u dokumentu, mogu se izdvojiti sledeći relevantni faktori, koji se koriste u fazi ocenjivanja:

- frekvencija koncepta u dokumentu,
- koncentacija,
- relevantnost koncepta u semantičkom podstablu,
- relevantnost koncepta u dokumentu.

U nastavku biće opisan svaki od navedenih parametara, dobijanje ukupne ocene, kao i sam postupak pronalaženja i ocenjivanja koncepta.

**Frekvencija koncepta** je broj koliko se puta određeni koncept pojavio u dokumentu i koristi se kod određivanja statističke zastupljenosti koncepta u dokumentu. Pod pojmom pojavljivanje koncepta u dokumentu misli se na pojavljivanje u tekstu bilo koje od sinonima vezanih za taj koncept.

**Koncentracija** koncepta predstavlja meru prisutnosti koncepta i njegovih potomaka u semantičkom podstablu. Težinski koeficijent za koncept određuje se na osnovu složenosti sinonima za taj koncept koja je pronađena u tekstu po sledećem kriterijumu:

$$w_{c_i} = \begin{cases} 0, ncount(c_i) = 0 \\ 15, ncount(c_i) > 0 \wedge ntokens(c_i) = 1 \\ 20, ncount(c_i) > 0 \wedge ntokens(c_i) = 2 \\ 30, ncount(c_i) > 0 \wedge ntokens(c_i) \geq 3 \end{cases} \quad (1)$$

, gde je:

- $ncount(c_i)$ - broj pojavljivanja sinonima u tekstu,
- $ntokens(c_i)$ - broj reči u pronađenom sinonimu.

Koncentracija određenog koncepta određuje se na osnovu njegovog težinskog koeficijenta i koncentracija svih direktnih podkoncepata po sledećem obrascu:

$$score(c_i) = w_{c_i} + \frac{\sum_{j=1}^{child(c_i)} score(c_j)}{child(c_i)} \quad (2)$$

, gde je:

- $w_{c_i}$ - težinski koeficijent za koncept $c_i$,
- $child(c_i)$ - broj direktnih potomaka koncepta $c_i$.

Kao što je navedeno u prethodnim sekcijama, koncept može biti potomak koncepta ili može imati podređene koncepte.

**Relevantnost koncepta u semantičkom podstablu** predstavlja meru zastupljenosti svih njegovih potomaka u dokumentu i određuje se po sledećem obrascu:

$$cr(c_i) = \frac{nsc(c_i)}{nsub(c_i)} \quad (3)$$

, gde je:

- $nsc(c_i)$ - broj podkoncepata za koncept ci računajući i koncept ci, sa koncentracijom većom od definisanog praga,

- $nsub(c_i)$ - ukupan broj podkoncepata za koncept $c_i$ računajući i koncept $c_i$.

**Relevantnost koncepta u dokumentu** predstavlja meru statističke zastupljenosti koncepta u dokumentu i određuje se po sledećem obrascu:

$$dr(c_i) = \frac{n(c_i)}{nntokens} \quad (4)$$

, gde je:

- $n(c_i)$ - broj pojavljivanja koncepta $c_i$ u dokumentu,
- *nntokens* - ukupan broj reči u tekstu koje ne pripadaju skupu jako frekventnih reči

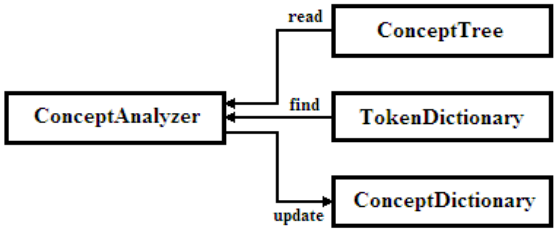
Ukupna ocena za koncept formirana je na osnovu parametara koji su opisani u prethodnim delu rada i računa se po sledećem obrascu:

$$totalscore_{c_i} = w_{sc} \cdot score(c_i) + w_{cr} \cdot cr(c_i) + w_{dr} \cdot dr(c_i) \quad (5)$$

, gde su:

- $w_{sc}$, $w_{cr}$, $w_{dr}$ - koeficijenti srazmere za koncentraciju, relevantnost koncepta i relevantnost dokumenta respektivno.

Na slici 6 prikazani su svi entiteti koji su relevantni u postupku pronalaženja i ocenjivanja koncepta.




Sl 6. Korišćeni entiteti u fazi pronalaženja i ocenjivanja koncepta

Da bi koncept bio ocenjen, moraju biti ocenjeni njegovi direktni potomci. Inicijalno, ocenjuju se koncepti koji su listovi i nemaju potomke u semantičkom podstablu, a zatim svi koncepti iznad njih.

**ConceptAnalyzer** čita iz stabla (hijerarhije) koncept koji se nalazi na vrhu semantičkog podstabla, vrši pretraživanje za svaki od sinonima za zadati koncept u rečniku sa pročitanim rečima **(TokenDictionary)** i kreira listu sa njihovim pozicijama u tekstu. Broj elemenata u listi predstavlja frekvenciju koncepta. Ako koncept ima direktne potomke, računaju se njihove ocene, odnosno postupak ocenjivanja se ponavljati rekurentno sve dok se ne izračuna ocena za koncepte-listove u semantičkom podstablu. Nakon toga, inverzno, računaju se ocene svih nadređenih koncepata.

Kada je izračunata ocena, vrši se izmena njegovih atributa (parametara za ocenu kao i sama ocena) u rečniku sa konceptima **(ConceptDictionary)**, koji ima osobine hash-mape.

Nakon završetka pronalaženja i ocenjivanja svih definisane koncepte u ontologiji, iz rečnika sa konceptima kreira se sortirana lista, koja se koristi kod prezentacije rezultata krajnjim korisnicima.

## V. ZAKLJUČAK

Postupak automatske analize elektronskih dokumenata je veoma interesantna i zanimljiva tema savremenih računarskih nauka. U radu je dat predlog jednog algoritma za ovaj postupak koji se zasniva na primeni koncepta ontologije.

Sistem je primenjen na analizu tekstova iz oblasti arhitekture i organizacije računara. Dobijeni rezultati su verifikovali ustanovljene pretpostavke.

Primena opisanog algoritam je velika. Trenutno, autor ga koristi u okviru sistema za analizu studentskog foruma Elektrotehničkog fakulteta u Beogradu na predmetima iz arhitekture i organizacije računara. Rad sistema omogućuje predavačima kvalitetne informacije o prihvaćenom gradivu od strane studenata.

## LITERATURA


[1] Lingpipe - Java libraries for the linguistic analysis of human language http://alias-i.com/lingpipe/demos/tutorial/stringCompare/read-me.html
[2] Jena Semantic Web Framework, http://jena.sourceforge.net
[3] Stuart J. Russell, Peter Norvig, Artificial inteligence: A Modern Approach
[4] Computational Linguistics and Intelligent Text Processing, Second International Conference, CICLing 2001, Mexico City, Mexico, February 18-24, 2001 Proceedings
[5] Mark Watson, Practical Artificial Intlligence Programming With Java
[6] Silvio Peroni, Enrico Motta and Mathieu d'Aquin, Identifying key concepts in an ontology through the integration of cognitive principles with statistical and topological measures; Knowledge Media Institute;The Open University
[7] Gruber T. 1993. A translation approach to portable ontologies. Knowledge Acquisition 5(2): 199–220, http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html


## ABSTRACT


This paper presents a method of analyzing text content using ontology. Using this method a text search is performed for every concept from previously made ontology. Found concepts are then evaluated. This evaluation gives us the measure of appearing of the specific topic to which the concept belongs. During the implementation of this system the fact was taken into consideration that several synonims can be defined for one concept, and word analysis should be adjusted to the language in which morphological changes of the words are present.


## ANALYZING TEXT CONTENT USING ONTOLOGY


Dejan Prodanović