Universitatea Politehnica Bucuresti
Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

Web Semantic

Masterand: COCORU Vlad-Cosmin
(IISC)

Retele de Calculatoare si Internet
iunie 2010

Curpins

1. Scurt istoric:

1989 – Tim Berners-Lee propune si realizeaza prima comunicare HTTP (Hypertext Transfer Protocol) server-client. Astfel se naste WWW-ul (World Wide Webul) in 1991, care este un sistem global de documente hipertext interconectate logic (prin hiperlegaturi, sau hiperlinks). Tot acum apare si familia de protocoale TCP/IP, ce faciliteaza comunicarea dintre sistemele de calcul din internet.

1991 – 2003 – WWW este referit retroactiv drept Web 1.0. Acest nume desemneaza starea de fapt a lucrurilor dinaintea aparitiei Web 2.0. Web 1.0 se caracterizeaza prin pagini statice – HTML – folosirea de frameset-uri, guestbookuri online, message boards si altele.

2003 – prezent – maturizarea Web 2.0, aducand in plus aplicatii web ce faciliteaza raspandirea de informatii in mod interactiv, interoperabilitate, design avand ca tinta nevoile utilizatorului si colaborarea intre oameni. Utilizatorii unui web-site pot interactiona in sensul ca pot coopera la intretinerea informatiilor, diferit fata de simpla consultare a textelor specifica pana atunci. Servicii specifice Web 2.0: comunitati bazate pe web (Twitter / Facebook, Hi5), aplicatii web (e-applications), retele sociale, siteuri de sharing video sau foto, pagini wiki, bloguri etc.

Iterativ, utilizatorii de internet asteapta aparitia Web 3.0, care ar presupune pe langa dinamizarea totala a paginilor web, si implementarea webului semantic, ca unealta a calculatoarelor de a observa adevaruri in text si de a genera noi adevaruri (informatii). Acelasi Tim Berners-Lee descrie webul semantic drept o componenta a Webului 3.0: „Oamenii ma intreaba ce este Webul 3.0. Cred ca daca adaugi grafica vectorizata – tot continutul sa fie dinamic, interactiv si atragator – peste Web 2.0 si oferi acces la Web Semantic peste un spatiu enorm de date, obtii acces la o resursa de date incredibila” (Tim Berners-Lee, 2006).

2. Introducere

„Cuvântul “semantic” reprezinta ”semnificatia” sau ”studiul semnificatiilor si schimbarilor acestora”. Webul semantic implica faptul ca semnificatiile datelor pot fi descoperite nu numai de oameni, dar si de calculatoare. În prezent, cele mai multe semnificatii de pe web sunt deduse de oameni care citesc pagini web si de alti oameni care scriu software specializat pentru lucrul cu date. Webul semantic implica o viziune în care calculatoarele (prin software) pot gasi, citi, intelege si utiliza date din World Wide Web precum oamenii. Scopul acestui uz este de a realiza obiective ce satisfac necesitatile sau dorintele utilizatorului uman”[1].

Desigur, deja utilizam software pentru a realiza unele lucruri pe web: navigam pe web, cumparam lucruri pe site-uri web, parcurgem pagini în diverse cautari, citim etichete si hiperlegaturi si decidem care legaturi sa fie activate (urmate). Dar ar fi mult mai eficient daca o persoana ar putea lansa un proces, care sa continue de sine statator, probabil verificat din când în când pe masura ce lucrul progreseaza.

Scopul webului semantic este sa puna la dispozitia tuturor utilizatorilor astfel de facilitati. Pe scurt, webul semantic ar trebui sa permita ca datele, localizate oriunde pe web, sa fie accesibile si întelese atât de oameni, cât si de calculatoare.

„Primul pas în acest sens a fost realizat prin XML care facilitează modul de formulare a informațiilor şi de partajare a acestora (XML este un limbaj de caracterizare simpla a datelor). Următorul pas se doreşte a fi utilizarea tehnicilor de reprezentare a cunoaşterii pentru exprimarea semanticii Web-ului şi realizarea inferențelor. În acest sens, abordările utilizate în inteligența artificială, bazele de date şi limbajele de programare trebuie reexaminate, adaptate şi extinse pentru a realiza integrarea bazei de cunoştiințe in format XML, inter-operabilitate şi inferențierea ca şi componentă a Web-ului semantic. Toate acestea implică nu numai reprezentarea cunoaşterii, dar şi structurarea informației şi automatizarea raționamentului.”[1]

Pe scurt, trebuie adaugata logica actualului Web. Termenul utilizat este de fapt „ontologie”, adica o reprezentare formala a cunostintelor cu ajutorul unui set de concepte si relationarea conceptelor existente. Pe scurt, o ontologie este o specificare explicita a unui concept.

3. Webul hipertext si limitarile sale

Multe fisiere de pe un computer tipic pot fi impartite in doar 2 categorii: documente si date. Documentele, cum ar fi emailurile, rapoartele si brosurile sunt citite de catre oameni. Datele, precum calendarele, playlisturile si spreadsheeturile sunt prezentate cu ajutorul unor aplicatii care le permite vizualizarea, cautarea si modificarea, dar sunt in principal destinate interpretarii computerului.

Momentan, WWWul este bazat in principal pe documente scrise in HTML (HyperText Markup Language), o conventie de marcare ce este folosita pentru a coda continutul unui text intretesut cu obiecte multimedia, cum ar fi imagini sau formulare interactive.

Tagurile metadata, de exemplu:

<**meta** name = „keywords” content = „computing, computer studies, computer”>
<**meta** name=”description” content=”catalog de soliutii software pentru

recunoastere de forme”>
<**meta** name=”author” content=”Vlad Cocoru”>

reprezinta o metoda prin care calculatoarele pot categorisi continutul paginilor web.

Cu HTML si un instrument care il poate interpreta (browser web sau editor html), se poate crea si prezenta o pagina care ofera informatii. Pe baza tagurilor de mai sus, un interpretor poate face rationamente simple, la nivel de document, de genul ca „acest document este un curs despre IT&C”, dar nu se poate intra in continutul efectiv al cursului pentru a folosi informatiile. Nu exista capabilitati specifice HTML-ului pentru a afirma cu certitudine ca, de exemplu, la pozitia 5 se gaseste cel mai popular pachet software, dar al carei producator a dat faliment, sau nu se poate face legatura intre acest item si pretul lui. Nici macar nu se poate face distinctia intre software si pret, cum nu se poate deduce ca documentul este de fapt un catalog.

Webul semantic implica obiceiul de a marca intentia (etichetare asemanatoare cu cea specifica HTML) in favoarea prezentarii informatiei in mod direct.

4.Premisele webului semantic

Pentru a putea vorbi de un Web Semantic este nevoie de 3 elemente:

4.1. Stocarea informatiilor semantice.

Aceasta se poate face fie la nivel local al paginii fie in alte locatii web. Un exemplu ar fi ***taggingul*** si ***labellingul***, ce reprezinta procedee de evidentiere a unor elemente din continutul unei pagini web. Taggingul se refera la etichetarea unei informatii scurte, simple,concise (cum sunt tagurile HTML, de exemplu), pe cand labelling presupune etichetarea unui ansamblu de informatie (structuri de metadate). Aceste procedee se pot realiza manual sau automat si reprezinta o tehnica elementara, dar care are avantajul puterii prin simplitate, pregatind informatia in vederea unei clasificari.

Stocarea informatiilor semantice se poate face si sub forma de ***Uniform Resource Identifier (URI)***, adica un string de identificare ce poate fi de natura unei locatii (***URL***) sau unui nume (***URN***). URN-ul este asemanator in conceptie cu numele unei persoane, pe cand URL-ul este asemanator cu adresa sa. Un exemplu elocvent este cazul cartilor, cu catalogarea in functie de ISBN. Accesul catre identificatorul **urn:isbn:0-486-27557-4** conduce catre o editie a cartii lui Shakespeare, „Romeo si Julieta”, iar accesul de genul file:///home/username/RomeoAndJuliet.pdf acceseaza cartea propriu-zisa, in format electronic. In esenta, URI-ul este tot o forma de etichetare a unor elemente de web content, dar in plus permit articularea unui limbaj comun intre utilizatori (prin referire la un element identificabil printr-un ID).

De exemplu, daca momentan o pagina web non-semantica descrie un item dintr-o lista in maniera: **<item>**calculator**</item>**, codarea aceleiasi informatii intr-o pagina cu continut semantic poate fi de genul **<item rdf:about**="http://database.org/resource/computer">calculator**</item>**, aratand unui browser semantic unde sa caute informatii despre obiectul pe care il reprezinta.

4.2. Aparitia de formate de stocare a datelor semantice.

Este nevoie de crearea unor limbaje de programare si publicare a continutului care sa aiba in vedere relatiile dintre date. De exemplu:

4.2.1. RDF (Resource Description Framework).

Modelul de date RDF este similar clasicului concept de modelare sub forma de diagrame de clase, intrucat se bazeaza pe ideea de a construi propozitii despre resurse (in principal resurse Web) suv forma unor structuri sintactice de forma subiect-predicat-obiect. Aceste obiecte sunt referite prin terminologia RDF „*triple*”. Subiectul denota resursa, iar predicatul denota trasaturi sau aspecte ale resursei si indica relatia care exista intre subiect si obiect. De exemplu, informatia „Cerul este de culoare albastra” este pusa sub forma unei *triple* RDF: **subiect** – „cerul”, **predicat** – „are culoarea”, **obiect** – „albastru”.[4]

O colectie de propozitii in format RDF reprezinta intrinsec un graf orientat si etichetat. De aceea, un model de date in format RDF este mai potrivit pentru reprezentarea cunostintelor decat un model relational sau alte obiecte ontologice. In practica, insa, datele sub forma RDF sunt de fapt stocate sub forma de baze de date relationale sau reprezentari native numite *Triplestores* sau *Quadstores* in caul in care pentru fiecare RFD se pastreaza si contextul (de exemplu numele grafului).

Vocabularul RDF, asa cum apare in specificatiile limbajului, este format din[5]:

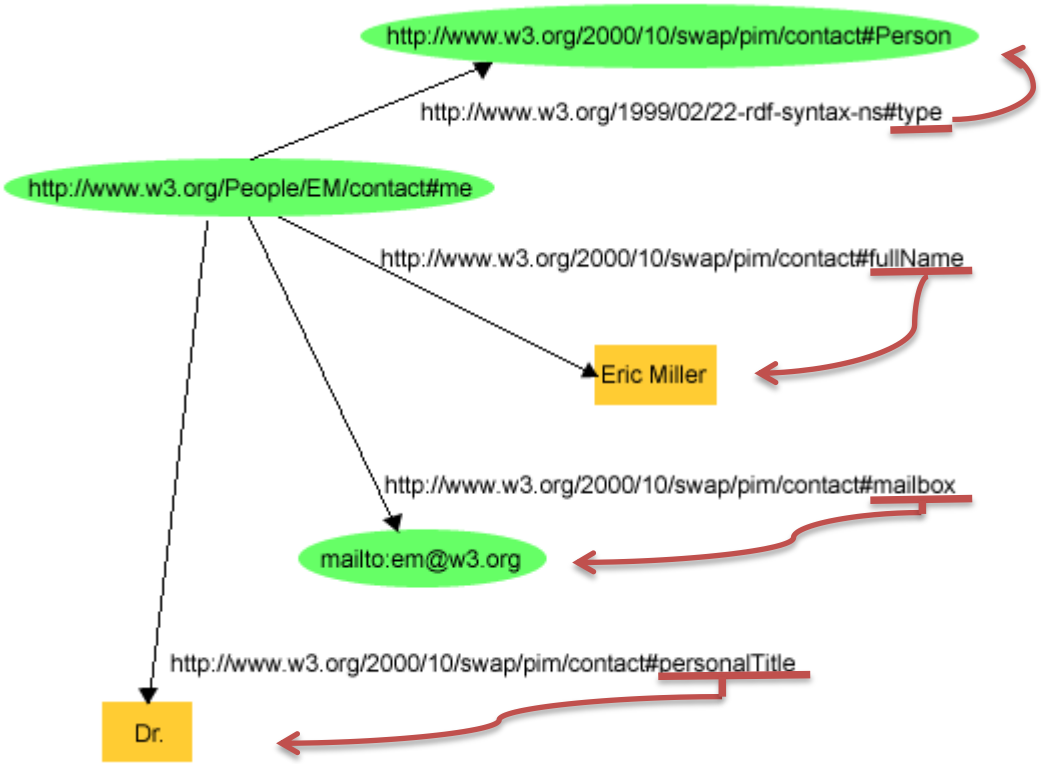
- ***rdf:type***: un predicat este folosit pentru a afirma ca o resursa este o instanta a unei clase
- ***rdf*:XMLLiteral:** clasa obiectului scris sub forma literala
- ***rdf:Property***: clasa de proprietati
- ***rdf*:*Alt, rdf:Bag, rdf:Seq:*** – sunt, in ordine, container de alternative, container de obiecte neordonate, container de obiecte ordonate (clasa ***rdfs:Container*** este superclasa acestora 3)
- ***rdf*:*List:*** – clasa pentru obiecte de tipul lista RDF
- ***rdf:nil:*** – o instanta a rdf:List ce reprezinta o lista vida.
- ***rdf*:Statement*, rdf:subject, rdf:predicate, rdf:object*** – folosite pentru ipotetizare sau reformulare.

Cunostintele, adevarurile din baza de date pot fi modelate sub forma unor propozitii ce suporta reformulari, in care fiecare propozitie (tripla subiect-predicat-obiect) primeste un URI si este tratata ca o resursa despre care se pot face alte afirmatii. De exemplu „Alina spune ca *cerul este de culoare albastra*”. Aceasta prelucrare a informatiei este importanta pentru a deduce un nivel de incredere asupra fiecarei propozitii.

Conform acestui concept, intr-o baza de date in care s-au petrecut reformulari, fiecare propozitie (ea insasi o resursa) are foarte probabil cel putin 3 alte propozitii formulate despre ea (cate una pentru subiect, predicat, obiect, fiecare fiind la randul lor cate o resursa). Exista si cazul in care exista mai multe propozitii despre afirmatia initiala, depinzand de nevoile aplicatiei.

Preluand concepte din sfera logicii, cateva implementari de model RDF propun ideea de a grupa propozitiile pe baza unor criterii, numite situatii sau contexte. De exemplu, o propozitie poate fi asociata cu un context, referit printr-un URI, cu scopul de a obtine o informatie de genul „*este adevarat in contextul X*”. O alta propunere este retinerea si sursei fiecarei propozitii, identificabila tot printr-un URI. Cand se aduc modificari sursei, afirmatiile modelului, care au fost deduse din acea sursa, se pot modifica dupa caz.

Mai jos se prezinta exemplul unui grafic ce caracterizeaza o persoana:



4.2.2. OWL (Web Ontology Language).

Reprezinta o familie de limbaje de reprezentare a cunsotintelor pentru a descrie ontologii pentru Web. Acestea se caracterizeaza prin semantica formala si se bazeaa pe limbaj RDF sau XML. Familia de limbaje a fost prezentata prima data in 2004 si din 2009 este la versiunea 2.0.

Datele descrise de o ontologie in familia OWL sunt interpretate ca un set de elemente ce au o serie de „afirmatii despre proprietati", care relationeaza aceste elemente intre ele. O ontologie reprezinta un set de axiome care instaleaza legaturi intre elemente (numite clase) si specifica tipurile de relatii dintre ele. Aceste axiome ofera inteles semantic,permitand sistemelor sa formuleze informatii aditionale pe baza datelor prezentate explicit.

De exemplu, o ontologie ce descrie familiile poate include axiome ce afirma ca proprietatea „AreMama" este prezenta intre 2 indivizi numai cand exista si proprietatea „AreParinte", si ca indivizii clasei „AreGrupaSanguina0" nu vor fi niciodata legati prin proprietatea „AreMama" de indivizi „AreGrupaSanguinaAB".

Altfel spus, daca se afirma ca Alex este legat prin proprietatea „AreMama" de Maria, si ca Alex este un membru al clasei „AreGrupaSanguina0", se poate afirma ca Maria nu este un membru al clasei „AreGrupaSanguinaAB"

OWL opereaza asupra claselor de entitati URI, stabilind relatii generice intre ele. Se pot genera automat RDF-uri, adica adevaruri noi. Emitand axiome si reguli de constructie logica, se pot genera constructii logice, de tipul teoremelor intalnite in gramaticile formale[3].

4.3. Existenta de aplicatii web capabile sa opereze si sa exploateze informatiile semantice.

De exemplu, browserele semantice, capabile sa ofere cautari inteligente, ce folosesc principiile sinonimiei si ale stemmingului, clusterizari semantice ale raspunsurilor, capacitatea de a face sugestii de cautare sau sumarizari automate.

O alta categorie ar fi aplicatiile ce nu au ca scop primar cautarea de informatii on-line, ci ghidarea unui utilizator uman in desfasurarea unor activitati, si oferirea de sfaturi pe care aplicatia le descopera prin intermediul webului semantic (aplicatia intelege contextul in care lucreaza utilizatorul uman si ofera sugestii si cauta solutii pe baza informatiilor semantice).

4.3.1. Extragerea limbii probabile

Un cuvant stop este un termen specific unei limbi, care are urmatoarele 2 calitati: este foarte frecvent folosit in acea limba si nu poarta o amprenta tematica bine definita. Altfel spus, consultarea exclusiv a acelui cuvant, cititorul nu isi poate face nicio parere

despre continutul textului. Daca un text contine un numar suficient de mare de cuvinte stop ale unei anumite limbi, este de presupus ca este scris in limba respectiva. Densitatea de utilizare a cuvintelor stop, si in general a oricaror cuvinte intr-un text, se exprima sub forma frecventei de utilizare la suta de cuvinte: **F= (Nr. Incidente)/(Nr Total Cuvinte) *100.**

Decizia asupra limbii probabil a textului este o problema care presupune existenta unei limite. Fie **T** un text si **F** frecventa de utilizare a cuvintelor stop dintr-l limba data **L**. Experimental, se determina un prag **p** pentru care

T** este scris in limba **L** daca **F>p

Blocurile decizionale pot fi monolingvistice sau polilingvistice si pot actiona la nivelul intregului text sau la nivelul paragrafelor sau chiar a propozitiilor, permitand astfel o corecta tratare a documentelor neuniforme din punct de vedere lingvistic.

Cuvintele stop pentru limba engleza sunt:

I	are	certainly	entirely	gave	how's	matter	nothing
a	aren't	chosen	equally	general	I'd	maybe	now
aboard	around	clear	especially	generally	if	me	occasionally
about	as	clearly	etc	get	I'll	mean	of
above	aside	common	even	getting	I'm	means	off
according	ask	completely	eventually	given	immediately	meant	on
across	at	constantly	ever	giving	in	merely	once
actually	away	could	every	gone	indeed	might	one
additional	back	couldn't	everybody	got	instead	mightn't	only
after	be	daily	everyone	greatly	into	mine	onto
again	became	daren't	everything	had	is	Miss	or
against	because	dawn	everywhere	hadn't	isn't	more	or
ago	become	determine	exactly	halfway	it	most	other
ahead	becoming	did	extra	happen	its	mostly	ought
alike	been	didn't	fact	has	it's	must	oughtn't
all	before	do	far	hasn't	itself	mustn't	our
almost	behind	does	farther	have	i've	my	ours
alone	being	doesn't	feature	haven't	just	myself	ourselves
along	belong	doing	few	having	kind	naturally	out
already	below	done	fewer	hay	larger	near	outer
also	beneath	don't	fifteen	he	last	nearby	outline
although	beside	double	fifth	he'd	late	nearer	outside
am	best	down	fifty	he'll	least	nearest	over
among	better	dozen	five	hello	less	nearly	own
amount	between	during	follow	her	let	necessary	partly
an	beyond	each	for	here	let's	needed	parts
and	both	earlier	forth	here's	like	needn't	perfectly
another	brief	early	forty	hers	likely	never	perhaps
any	built	easily	four	herself	lot	next	possibly
anybody	but	eight	fourth	he's	made	nine	previous
anyone	by	either	frequently	him	mainly	no	probably
anything	cannot	eleven	from	himself	make	nobody	proper
anyway	can't	else	full	his	making	none	properly
anywhere	carefully	enough	fully	how	many	nor	put
apart	certain	entire	further	however	March	not	putting

quarter	somewhere	thousand	week	year
quite	soon	through	well	yes
rather	St	throughout	we'll	yesterday
recently	still	thus	were	yet
same	such	thy	we're	you
second	sudden	till	weren't	you'd
seeing	suddenly	to	we've	you'll
seems	suppose	together	what	your
seldom	sure	tomorrow	whatever	you're
seven	surrounded	tonight	what's	yours
several	taken	too	when	yourself
shall	ten	toward	whenever	yourselves
shan't	than	twelve	when's	you've
she	that	twenty	where	
she'd	that's	twice	where's	
she'll	the	under	wherever	
she's	their	underline	whether	
should	theirs	understanding	which	
shouldn't	them	unless	while	
shown	themselves	until	who	
sides	then	unusual	whole	
similar	there	up	whom	
simply	therefore	upon	who's	
since	there's	upper	whose	
Sir	these	upward	why	
sitting	they	us	why's	
six	they'd	using	wide	
slightly	they'll	usual	wish	
so	they're	usually	with	
some	they've	various	within	
somebody	third	very	without	
somehow	thirty	was	won't	
someone	this	wasn't	worse	
something	those	we	would	
sometime	thou	we'd	wouldn't	

4.3.2. Stemming

Stemmingul este o tehnica de normalizare a multimilor cuvintelor unei limbi in clase de echivalenta, in cadrul carora termenii sa fie inruditi morfologic si semantic. Desi seamana cu procesul lexical de extragere a radacinii cuvintelor, nu este acelasi lucru, intrucat accentul nu cade pe identificarea radicinii si sensului acestuia, ci pe semnatica[6].

Mai precis, la efectuarea extragerii radacinii unui cuvant, din punct de vedere strict lexical, se efectueaza operatii de desufixizare si deprefixizare, cautandu-se o radacina cunoscuta ce are un sens de baza. De exemplu:

infrigurat -> in-frig-ur(i)-ate => frig (lat. frigus / frigoris)
frigorific -> frig-orific => frig (lat. frigus / frigoris)
frigaruie -> frig(e)-ar(e)-uie=> frige(lat. frigere)

In cazul stemmingului, deprefixizarea este absenta intrucant produce intotdeauna schimbari ale sensului: Operatia de stemming pe baza exemplelor de mai sus:

infrigurat -> infrigur-ate => infrigur
frigorific -> frigor-ific => frigor
frigaruie -> frigar-uie => frigar

Pentru algorimtii de stemming (automat) se intocmeste o lista a sufixelor si regulilor de desufixizare aferente Sufixele se elimina in ordinea descrescatoare a lungimii lor. De exemplu, algoritmul Lovins de stemming reprezinta prima propunere de algoritm de stemming valabil pentru limba engleza. Adaptarea la alte limbi presupune schimbarea listei de sufixe si de reguli lexicale in care se pot elimina sufixele.

Exista si cateva incercari de adaptare a algoritmului pentru limba romana [2].

Pas 0: eliminarea formelor de plurar (si alte simplificari). Cautarea celui mai lung sufix dintre urmatoarele si efectuarea operatiei indicate:

ul ului -> stergere
aua -> inlocuire cu ***a***
ea ele elor -> inlocuire cu ***e***
ii iua iei iile iilor ilor -> inlocuire cu ***i***
ile -> inlocuire cu ***i*** daca nu este precedat de ***ab***
atei -> inlocuire cu ***at***
aţie aţia -> inlocuire cu ***aţi***

Pas 1: reducerea sufixelor combinative. Cautarea celui mai lung dintre sufixele urmatoare si efectuarea operatiei indicate. Repetarea acestui pas pana cand nu se mai face nicio modificare:

abilitate abilitati abilităi abilităţi -> inlocuire cu ***abil***
ibilitate -> inlocuire cu ***ibil***
ivitate ivitati ivităi ivităţi -> inlocuire cu ***iv***
icitate icitati icităi icităţi icator icatori iciv iciva icive icivi icivă ical icala ic ale icali icală -> inlocuire cu ***ic***
ativ ativa ative ativi ativă aţiune atoare ator atori ătoare ător ători -> inlocuire cu ***at***
itiv itiva itive itivi itivă iţiune itoare itor itori -> inlocuire cu ***it***

Pas 2: eliminarea sufixelor standard. Cautarea celui mai lung sufix dintre urmatoarele, si efectuarea operatiei indicate:

at ata ată ati ate ut uta ută uti ute it ita ită iti ite ic ica ice ici ică ab il abila abile abili abilă ibil ibila ibile ibili ibilă oasa oasă oase os osi oşi ant anta ante anti antă ator atori itate itati ităi ităţi iv iva ive ivi ivă
-> sterge

iune iuni

-> sterge daca este precedat de un ***ţ***, si inlocuieste ***ţ*** cu ***t***.

ism isme ist ista iste isti istă işti

-> inlocuieste cu ***ist***

Pas 3: (doar daca nu s-a eliminat niciun sufix la pasii 1 si 2) Eliminarea sufixelor verbale. Cautarea celui mai lung sufix dintre urmatoarele si efectuarea actiunii indicate.

are ere ire âre indând indu ându eze ească ez ezi ează esc eşti eşte ăsc ăşti ăşte am ai au eam eai ea eaţi eau iam iai ia iaţi iau ui aşi arăm a răţi ară uşi urăm urăţi ură işi irăm irăţi ră âi âşi ârăm ârăţi âră asem a seşi ase aserăm aserăţi aseră isem iseşi ise iserăm iserăţi iseră âsem âseşi â se âserăm âserăţi âseră usem useşi use userăm userăţi useră

-> sterge daca este precedat de o consoana sau de ***u***

ăm aţi em eţi im iţi âm âţi seşi serăm serăţi seră sei se sesem seseşi sese seserăm seserăţi seseră

-> sterge

Pas 4: eliminarea vocalei finale. Cautarea celui mai lungi dintre urmatoarele sufixe si eliminarea lui: ***a e i ie ă***

4.3.3. Masuri de similaritate

In urma analizei unui text din caruia i s-a determinat limba, s-au eliminat cuvintele stop si s-a aplicat algoritmul de stemming, se poate face un calcul al frecventei utilizarii cuvintelor. Rezulta o posibila plasare vectoriala a textului in spatiul limbii respective, daca fiecare cuvant (rezultat in urma stemmingului) este considerat a fi o axa a spatiului, iar fiecare frecventa este considerata coordonata in functie de acea axa.

Majoritatea componentelor unui astfel de vector de frecvente ar fi nule, iar dimensiunea spatiului imaginat este de regula extrem de mare, de ordinul sutelor de mii sau milioanelor. Totusi, cu puterea de calcul actuala, aceasta nu reprezinta un inconvenient.

Avand 2 texte **T1** si **T2**, si reprezentarile lor vectoriale **F1** si **F2**, se poate imagina o masura de similaritate ca fiind norma euclidiana **DE(T1,T2) = || F1 – F2 ||** sau produsul scalar: **DS(T1,T2) = <F1-F2>**

4.3.4. Determinarea domeniului tematic al unui document

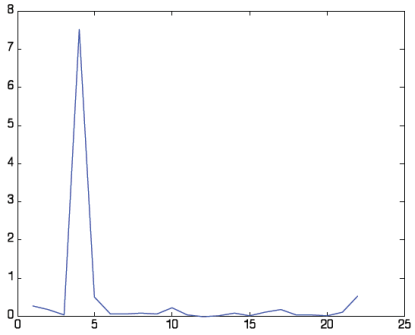
Aceasta abordare presupune efectuarea a 2 pasi premergatori:

1) alegerea unui numar de domenii care sa acopere realitaeta discursurilor in general, la o rezolutie cat mai potriivta. De exemplu, s-ar putea propune lista:
   Dom = { Administratie, Mediu, Finante, Auto, Economie, Educatie, Imobiliare-Constructii, IT&C, Media, Politica, Moda, Sanatate, Justitie, Sport, Stiinta, Turist, Vedete-Monden, Sex, Cultura, Animale, Meteo} ***cu 21 de elemente.***

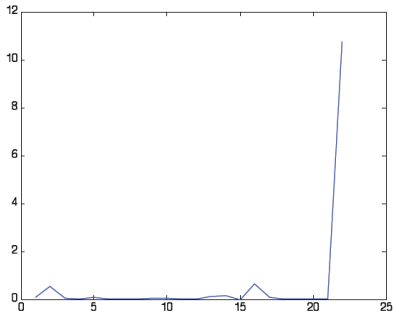
2) Adunarea de documente bine incadrate tematic, pentru fiecare dintre domeniile din lista, si intr-un numar suficient de mare. Pentru fiecare document se va face o radiografie a notiunilor comune, a denumirilor si expresiilor, adica se va intocmi un vector al frecventelor de utilizare.

Apoi, la analiza unui text se poate face analiza termenilor si plasarea fiecarui cuvant intr-unul din domeniile disponibile, cu o anumita probabilitate. In final, se va face o medie a tuturor cuvintelor si se poate spune cu o probabilitate destul de ridicata la ce domeniu se refera documentul.

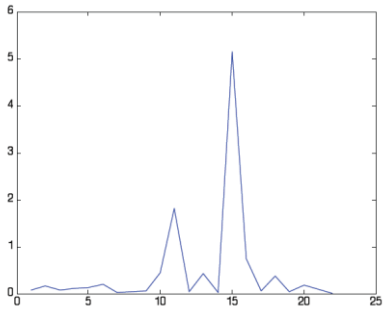
TERMEN	DOMENIU 1	DOMENIU 2	DOMENIU 3	DOMENIU M
constant	0.01	0.2	0	0.11
construct	0.3	0	0	0.3



Profil tematic al cuvantului „masina”, cu varful pe domeniul „Auto”



Profil tematic al cuvantului „grad”, cu varful pe domeniul „Meteo”



Incadrare tematica a unui text ce relateaza despre un parlamentar european ce detine o echipa de fotbal.

4.3.5. Identificarea subiectului, personajelor

Personajele, sintagmele si citatele joaca intr-un text rolul reperelor. Ele permit o rapida orientare in legatura cu sensul si incadrarea tematica a unui discurs. De aceea, pentru o abordare automata a unui document, este necesara dezvoltarea unor instrumente ce recunosc si extrag aceste componente.

Personajele se pot recunoaste:

- **autarhic** – prin logica interna a unui document si a limbii. Are avantajul ca permite descoperirea unor personaje noi si flexibilitatea si autonomia in functionare, dar are dezavantajul unei rate ridicate de confuzii si erori, induse de indisciplina sintactica si ortoepica a documentelor.
- **In mod catalog** – pe baza unor dictionare externe de personaje. Avantajul prezentat este corectitudinea extragerilor, dar dezavantajul este ca nu permite identificarea de termeni noi.

Similar, se poate construi o lista de abrevieri. Se cunoaste ca in functie de context, o abreviere inseamna lucruri diferite. Exemplu: ITI = *International Theater Institution (Unesco)* sau *Information Technology Institute* sau *Intestinal Tract Infection*. Pe baza determinarii contextului se poate inlocui un acronim cu varianta sa completa.

4.3.6. Rezumatizare automata

Rezumatul unui text coerent este un alt text coerent, mai scurt, care reuseste sa comprime informatiile primului. Nu exista un standard, dar exista o orientare „intuitiva” a rezumatizarii corecte. Elementele ce trebuiesc regasite intr-un rezumat de calitate sunt:

A. Concizia absoluta
B. Concizia relativa la un standard de lungime impus
C. Cursivitatea logica a textului
D. Pastrarea doar a personajelor centrale
E. Pastrarea doar a povestii centrale
F. Repovestirea in limbaj minimalist

Rezumatele sunt destinate lecturii umane. Niciunul din criteriile de mai sus nu poate fi apreciat decat in mod subiectiv, de aceea calitatea rezumatizarii nu poate fi masurata propriu-zis. Cum insa criteriul A este extrem de relativ, intrucat nu depinde doar de subiectivitate, cat si de scopul urmarit si chiar de aspecte legate de copyright, iar F presupune cunoasterea si recunoasterea sintaxei, pentru a putea elimina figurile de stil si

propozitiile secundare din cadrul unei fraze, gasirea formularilor alternative, se propune implementarea unui algoritm care sa respecte criteriile B, C, D si E.

Pas 1: Descompunerea textului in propozitii/fraze. In functie de algoritm, se pot obtine rezolutii diferite, de la identificarea propozitiilor simple (dictate de logica gramaticala, sintactica si de punctuatie) pana la identificarea doar a frazelor (dictata de regulile de punctuatie)

T -> { P1, P2, .. , Pn}

Pas 2: Se extrag toate cuvintele din T si se echivaleaza cu clasa lor stemmica, se traseaza chestiunea termenilor proprii (identificare, reducere la formele eliptice, sinonimizari) si a sintagmelor, cat si numararea aparitiilor la nivel de label stemm. Se acorda o valoare de importanta fiecarei notiuni, in functie de natura si frecventa de utilizare in intreg textul. Reguli de notare posibile:

- Un nume propriu este mai important ca o sintagma
- O sintagma este mai importanta ca un cuvant comun
- Un termen este mai important daca frecventa de utilizare este mai ridicata in text
- Un termen este mai important daca frecventa de utilizare este mai ridicata in limba
- Un termen este mai important daca are un profil tematic mai asimetric in limba
- Un termen de la inceputul unui text e mai important ca unul de la mijlocul textului
- Un termen de la sfarsitul textului este mai important ca unul de la mijlocul textului

In urma acestui pas rezulta un tabel al importantelor cuvintelor, de forma:

Termen	carte	Ion Creanga	Amintiri din copilarie	...
Importanta	0.01	0.12	0.7	...

Pas 3: Se realizeaza o punctare a fiecarei propozitii in parte, in care se folosesc informatiie de la pasul anterior. Propozitiile primesc un scor prin insumarea ponderata a valorilor de importanta a cuvintelor care le compun.

Propozitia	P1	P2	...	Pn
Scorul	s1	s2	...	sn

Pas 4: Se realizaeaza o matrice a distantelor de asemanare semantica intre propozitii (pe baza masurilor de similaritate). Cu ajutorul acestor distante se modifica scorul fiecarei propozitii. Daca o propozitie este legata semantic de alte propozitii, va primi un plus de scor

de la fiecare, proportional cu scorul propozitiei surori si invers proportional cu distanta semantica fata de aceasta.

Astfel, se ofera posibilitatea ca unele propozitii de legatura, ce nu aveau un scor important, sa reintre in joc, facand textul mai citibil

Pas 5: Se reindexeaza propozitiile, in ordinea descrescatoare a scorului calculat anterior. Informatia se completeaza cu informatii de gabarit (nr cuvinte, nr caractere) asupra fiecarei propozitii si se retine sub forma unui tabel de genul:

Propozitia	P1	P2	...	Pn
Scorul	S1 >	S2 >	... >	Sn
Indexul original	i1	i2	...	in
Nr. cuvinte	w1	w2	...	wn
Nr. caractere	c1	c2	...	cn

Pas 6: Se extrage rezumatul. Gabaritul rezumatului poate fi controlat la diferite rezolutii:

- Procent din numarul de propozitii originale
- Procent din numarul de cuvinte originale
- Procent din numarul de caractere initiale
- Numar aproximativ de cuvinte
- Numar aproximativ de caractere

Dupa setarea gabaritului, se colecteaza in limitele lui propozitiile cu scorurile cele mai mari. Apoi aceste propozitii se reordoneaza dupa indexul original si se asambleaza intr-un text.

5. Concluzii asupra webului semantic

In concluzie, cand vorbim despre Webul Semantic ne referim la urmatorul set de caracteristici

- **Date accesibile calculatoarelor** – webul semantic presupune ideea de a avea datele pe web definite si legate in modalitati care sa permita calculatoarelor sa le utilizeze nu numai in scopul afisarii lor, dar mai ales pentru automatizarea, integrarea si utilizarea lor in diferite aplicatii
- **Agenti inteligenti** – scopul webului semantic este sa faca webul actual mai usor de citit de calculatoare, pentru a permite agentilor inteligenti artificiali sa restabileasca si aa manipuleze informatii
- **Baze de date distribuite**: webul semantic sa faca pentru date ceea ce HTML-ul a facut pentru informatiile textuale. Sa poata sa reprezinte toate bazele de date si

regulile logice ale acestora ca un tot. Este o incercare de a face pentru datele prelucrabile de catre calculatoare ceea ce WWWul a facut pentru documentele citibile de catre oameni.

- **Infrastructura informatizata**: webul semantic va fi o infrastructura, nu o aplicatie. Problema este ca webului ii lipseste o structura de baza usor de automatizat.
- **Serviciu pentru umanitate**: aplicatiile vor putea sa elibereze utilizatorul uman de sarcina consumatoare de timp de a localizare resursele relevante pe web, sa le extraga, sa integreze si sa indexeze informatiile continute in ele. Se doreste colectarea automata de informatii din diferite surse, integrarea lor si procesarea lor pentru extragerea adevarurilor, pe care apoi aplicatiile le vor prezenta utilizatorilor drept produs finit.
- **Adnotare superioara**(cu metadate): webul semantic va asigura adnotare evoluata pentru documente, in formate accesibile procesarii automate si efectuarii legaturilor dintre ele
- **Cautare imbunatatita**. Dezideratul este accesarea informatiilor de pe web avand in vedere continutul, si mai putin cuvintele.

6. Dezvoltare

Este important de retinut de faptul ca Webul semantic este o realitate emergenta, experimentala. Drept urmare, exista numeroase zone in care lucrurile sunt in curs de dezvoltare. Se pot observa, totusi, doua directii mari de lucru:

- Transpunerea web contentului actual, din formatul clasic, in forma potrivita interogarilor de natura semantica. Volumul imens de informatie face aceasta activitate imposibila de gestionat de oameni, fiind nevoie de automatizare

- Construirea de aplicatii de extragere de informatii cu caracter semantic. Se doreste maturizarea aplicatiilor pana la nivelul de gasire a raspunsurilor la intrebari date, descoperirea de adevaruri noi, imbogatirea datelor existente cu straturi de informatie semantica.

Bibliografie

1. „Observatorul Militar”, pp. B8, nr. 44 (2-8 nov. 2005) – www.presamil.ro
2. Romanian Stemmer: http://snowball.tartarus.org/algorithms/romanian/stemmer.html
3. http://en.wikipedia.org/wiki/Web_Ontology_Language
4. http://en.wikipedia.org/wiki/Resource_Description_Framework
5. http://www.w3.org/TR/PR-rdf-syntax/ "Resource Description Framework (RDF) Model and Syntax Specification"
6. http://en.wikipedia.org/wiki/Stemming
7. Julie Beth Lovins (1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics 11:22–31.
8. http://en.wikipedia.org/wiki/Text_mining
9. K. Bretonnel Cohen & Lawrence Hunter (January 2008). "Getting Started in Text Mining"