**UNIVERSITATEA ”LUCIAN BLAGA” DIN SIBIU**

**Contract UEFISCDI nr. 27 / 04.08.2010**

**Programul: Resurse umane**
**Tipul proiectului: Proiect de cercetare post-doctorală**
**Cod proiect: PD_670 / 28.07.2010**

**SINTEZA DE CERCETARE**

# SISTEM DE CLASIFICARE AUTOMATA A DATELOR NESTRUCTURATE FOLOSIND METACLASIFICATOARE BAZATE PE METODE DE TIP SUPPORT VECTOR MACHINE ŞI NAIVE BAYES

**Etapa unica 2012**
**Anul III**

**DIRECTOR PROIECT,**

**Sef lucrări dr. Ing. Daniel MORARIU**

**iunie, 2012**

# 1 Rezumatul proiectului

Acest proiect se încadrează în domeniul clasificării automate a documentelor text şi îşi propune îmbunătăţirea rezultatelor clasificării prin realizare unor strategi de combinare a rezultatelor metodelor de clasificare folosite.

Un număr impresionant de documente se găsesc în format electronic, iar utilizatorul are nevoie de componente de clasificare automată a acestora pentru a le putea gestiona. Gestiunea lor a devenit o problema foarte importanta în ultima perioada. Devine esenţială existenta unor programe inteligente de organizare automată a documentelor în categorii pentru a facilita analiza şi prelucrarea acestor documente. Datorită domeniului foarte vas în care ar trebui să lucreze acestea, devine dificil de realizat un singur clasificator cu performanţe foarte bune. Abordarea actuala este de a utiliza mai mulţi clasificatori de diferite tipuri combinaţi într-un meta-clasificator, sau realizarea unei clasificări hibride care se bazează pe predicţia clasificatorului cel mai bun pentru o problema particulară folosind caracteristicile vectorilor de intrare ale documentelor şi istoria clasificărilor. Având mai mulţi clasificatori de baza, de tipuri diferite (SVM - Support Vector Machine, Bayes, reţele neurale, etc), idea este de a învăţa un meta-clasificator care prezice gradul de corectitudine pentru fiecare dintre clasificatorii de bază. Meta-eticheterea unei instanţe indică încrederea în clasificarea făcută de acesta, dacă instanţa este clasificata corect de către acel clasificator dintre toţi ceilalţi clasificatori utilizaţi. Regula de clasificare a meta-clasificatorului este ca fiecare clasificator de baza să atribuie o clasa la instanţa curentă şi apoi meta-clasificatorul să decidă dacă clasificarea este demnă de încredere sau nu. Pe lângă creşterea acurateţei de clasificare, prin exploatarea sinergismului mai multor clasificatoare, un alt avantaj al meta-clasificării constă în posibilitatea de exploatare a paralelismelor funcţionale (multiprocesor).

# 2 Obiectivele proiectului pentru anul 2012

| Etapa | Obiective | Activităţi | Categorii de buget (Valoare lei) |
|---|---|---|---|
| Unică | 1. Implementarea unui sistem de răspuns automat la întrebăr. | 1.1 Realizarea conexiunii la motoare de căutare web. Realizare practica. | 19000,00 |
| | | 1.2 Clasificarea rezultatelor utilizând meta-clasificatoarele anterior antrenate şi extragerea informaţiilor relevante în sensul unui răspuns adecvat. Realizare practica | 19500,00 |
| | | | **Total etapa: 38500,00** |

# 3 Sinteza activităţilor de cercetare realizate în 2012

În această etapă s-a realizat un sistem online de răspuns automat la întrebări care se găseşte disponibil la adresa http://193.226.29.26/ACUD/Default.aspx. Acest sistem permite introducerea a unei întrebări formulate în limbaj natural în limba engleză. Aplicaţi realizată recunoaşte 5 tipuri de întrebări: Person, Location, Time, Description şi Reason. Ca şi rezultat deocamdată întoarce o listă de snipet-uri. Această listă este sortată în funcţie de încrederea pe care o dă analiza lexicală efectuată asupra snipet-ului. Încrederea este calculată relativ la faptul dacă în acel snippet se găseşte sau nu răspunsul aşteptat. Aplicaţia realizată analizează din punct de vedere sintactic cuvintele din interogare pedicţionând pentru fiecare cuvânt tipul de vorbire al acestuia în funcţie de contextul în care apare acesta în întrebare. În această aplicaţie contextul considerat este de maxim 3 cuvinte. Cuvintele considerate esenţiale pentru interogare sunt transmise motorului de căutare Bing iar de la acesta se aşteaptă o lisă de maxim 20 de răspunsuri pe care motorul de căutare le consideră ca fiind cele mai relevante pentru cuvintele din interogarea specificată. Fiecare snipet este analizat din punct de vedere sintactic şi în funcţie de tipul întrebării se caută anumite părţi de vorbire în vecinătatea cuvintelor din interogare. Snipetul este etichetat cu o valoare mai mare dacă în preajm cuvintelor specificate în interogare se găsesc cuvinte care sunt etichetate cu tipul de vorbire aşteptat.

## 3.1 Motoare de căutare

### 3.1.1 Caracteristici generale

O dată cu dezvoltarea sa exponenţiala, World Wide Web-ul a devenit o sursă imensă de informaţii, fie ele text, imagini sau multimedia. Lipsa unei structuri centrale şi libertatea de a nu urma o sintaxă strictă a făcut însă ca această cantitate vastă de informaţii să fie disponibila pe Web, dar regăsirea ei să nu fie atât de uşoară. Pentru accesul eficient la aceste informaţii utile, sunt absolut necesare abordări şi interfeţe corespunzătoare pentru căutarea şi navigarea prin aceste colecţii imense, care să fie eventual capabile să întoarcă doar răspunsurile la căutările utilizatorului.

Motoarele de căutare (sau serviciile de căutare pe Web) sunt cea mai utilizată metodă folosită în zile noastre pentru a accesa informaţiile de pe Web. Utilizarea motoarelor de căutare se bazează pe o paradigma simpla: pentru o interogare (query) formată din unul sau mai multe cuvinte cheie, motorul de căutare răspunde cu o listă de rezultate ale căutării. În cazul general această listă reprezintă o serie de pagini web pe care s-au găsit cuvintele specificate în interogare. În această listă fiecare pagină este descrisă print-un scurt rezumat numit snippet. Snippet-ul este compus dintr-un titlu, o adresa a documentului (URL-ul acestuia) şi un text scurt care evidenţiază conţinutul documentului referit.

Motoarele de căutare parcurg periodic paginile web şi indexează cuvintele din descrierile acestor pagini astfel în momentul căutării vor întoarce acele pagini care conţin cuvinte comune cu interogarea. Primul motor de căutare propriu-zis a fost Archie lansat în 1990 şi era un serviciu de căutare a fişierelor FTP. Primul motor de căutare care a încercat să adune conţinutul WEB-ului într-o bază de date a fost Wander lansat în 1993. Tot atunci apare şi Aliweb care indexa paginile de pe un server astfel încât să devină mai simplă căutarea anumitor informaţii pe acel server.

Doar din 1995, putem vorbi deja de prima generaţie de motoare de căutare (1995-1997) care se bazau doar pe informaţia text din cadrul paginii (frecvenţa cuvintelor, limba). Exemple de astfel de motoare de căutare care au fost şi folosite pe scară largă sunt: Lycos şi Excite.

A doua generaţie a motoarelor de căutare apărute în anii 1998 este reprezentată de apariţia sistemelor care folosesc şi informaţii externe paginii propriu-zise. Aceste sisteme analizează link-urile utilizatorului, informaţiile despre rezultatele alese de către utilizatori (click-through data) precum şi referinţele la pagina respectivă şi referinţele de la pagina respectivă la alte pagini. Cu cât pagina avea mai multe referinţe din exterior şi mai multe referinţe către alte pagini importante era considerată mai relevantă. Cel mai reprezentativ motor de căutare a fost şi este în continuare Google-ul.

A treia generaţie de motoare de căutare este cea actuală, încă la nivel experimental şi în plină dezvoltare. Principalele îmbunătăţiri ale acestora sunt legate de analiza semantică a textelor indexate, determinarea contextului căutării (locaţia utilizatorului, profilul acestuia şi istoria căutărilor realizate de acesta) precum şi îndreptarea atenţiei spre nevoile utilizatorului nu doar pe căutarea în sine. Motoarele de căutare din această generaţie sugerează utilizatorului anumite contexte de căutare, corectează greşelile gramaticale şi prezintă utilizatorului interfeţe cât mai sugestive pentru prezentarea rezultatelor.

Un motor de căutare conţine mai multe componente dintre care cele mai importante sunt prezentate în figura 1.

*Crawler-ul* numit şi spider are rol de a descoperii automat noi pagini web pentru ca acestea să poată fi indexate şi regăsite uşor ulterior. Această descoperire de pagini se face prin urmărirea hyperlink-urilor din paginile deja vizitate.

*Indexarea documentelor* se ocupă cu memorarea anumitor informaţii utile dintr-o pagină web pentru ca aceasta să poată fi regăsită în cazul căutării după anumite cuvinte cheie.

*Cache-ul de documente* se ocupă cu păstrarea resurselor indexate în forma originală sau păstrarea link-urilor la acele resurse după indexarea acestora

*Sistem de ordonare a documentelor* – de obicei pentru o căutare sunt întoarse ca rezultate milioane de astfel de linkuri. Este nevoie de metode şi tehnici care să ordoneze link-urile întoarse de motorul de căutare astfel încât cele mai relevante şi cele mai interesante pentru utilizator să apară primele în listă. Primele motoare de căutare ca Excite sau Infoseek utilizau algoritmi care ordonau

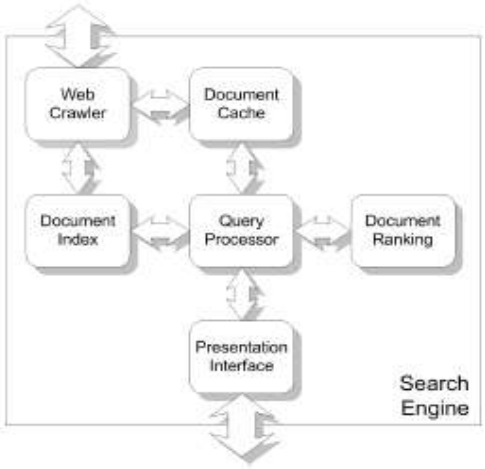


Figura 1. Componentele unui motor de căutare standard

documentele pe baza conţinutului acestora. Această abordare avea deficiente majore pentru că se putea profita uşor pentru a-ţi aduce propria pagina cât mai sus în ierarhie prin popularea excesivă a paginii cu cuvinte nu neapărat legate de conţinutul real al paginii (proces de *spamming*). Motoarele de căutare din a doua generaţie implementează algoritmi de ordonare bazaţi pe analiza hyperlink-urilor dintre documentele web. Premisa este că numărul de hyperlink-uri care duc la o pagină, este direct proporţional cu popularitatea şi calitatea acelei pagini. Prin urmare, cu cât o pagina are mai multe linkuri către ea, cu atât va apărea mai sus în ierarhie. În plus, este luată în considerare şi poziţia în ierarhie a paginilor care duc la pagina în discuţie, deci cu cât sunt mai importante paginile de proveniență a linkurilor, cu atât îi creşte poziţia în ierarhie a paginii în discuţie. Dezavantajul acestui tip de algoritm ar fi importanţa prea mare pe care o acordă paginilor cu o vechime mare (care sigur că au multe legături din exterior) în detrimentul paginilor noi care pot fi uneori mai relevante. Cel mai cunoscut algoritm de acest tip este cel al motorului Google, PageRank.

*Unitatea de procesat interogări* are rolul de a coordona execuţia căutării termenilor introduşi de către utilizator. Executantul de interogări comunică cu index-ul de documente, cu cache-ul şi cu sistemul pentru ordonarea documentelor pentru a obţine o listă a documentelor relevante cu căutarea curentă. Această listă este apoi trimisă interfeţei de prezentare a rezultatelor.

Interfaţa de prezentare a rezultatelor (Presentation Interface) se ocupă cu prezentarea într-un mod cât mai intuitiv şi simplu a rezultatelor interogării curente astfel încât utilizatorul să găsească cât mai uşor paginile căutate. Motorul de căutare google foloseşte de exemplu o listă de snipet-uri ordonate, dar sunt motoare de căutare cum ar fi Yippy [yippy] care încercă pe baza unor algoritmi de învăţare nesupervizată o grupare a paginilor întoarse de un motor de căutare standard în categorii, în funcţie de cuvintele din snippet, altele decât cele din căutare. Astfel utilizatorul poate alege o categorie mai generală care să îl conducă mai repede la informţiile căutate.

În această situaţie, utilizatorii ar putea benefica de o interfaţa de căutare în care să poată introduce nevoia informațională sub forma unei întrebări în limbaj natural, iar apoi să primească răspunsul la acea întrebare, fie sub forma unui simplu cuvânt, fie un document, site, fişier audio sau video. Spre exemplu, dacă un utilizator dorește să afle în ce an s-a născut compozitorul austriac Mozart, să fie de ajuns să scrie într-un motor de căutare întrebarea (“When was Mozart born?”) şi să primească înapoi răspunsul corect (“1756”), eventual alături de un document (URL) care conține acel răspuns pentru lămuriri suplimentar.

O motivație suplimentara pentru care este necesară aplicarea şi integrarea algoritmilor de Question Answering în motoarele de căutare este ca 15-20% din log-urile cu căutări ale motoarelor de căutare conțin de fapt căutări sub forma de întrebare. Deci utilizatorul are o nevoie informațională tot mai mare, ştie ca pe Web poate găsi orice şi este deci tot mai tentat să caute pe Web aşa cum ştie el, adică întrebând motorul de căutare în limbaj natural.

### 3.1.2 Conectarea la API-urile motoarelor de căutare

Cele mai importante motoare de căutare oferă ca facilitate conectarea directă din aplicaţie la acestea, prin intermediul API-ului propriu. Prin funcții parametrizabile specifice se apelează sistemul de căutare al motorului care va returna sub forma de fișier XML sau structuri de date rezultatele căutării pentru a fi procesate de către aplicația utilizatorului. Comunicarea se realizează prin protocoale SOAP sau REST.

Pentru această aplicație au fost accesate interfețe către următoarele servicii oferite de motoarele de căutare: Google API şi Bing API. Aceste servici necesită doar o înregistrarea prealabilă (momentan google nu mai oferă gratuit acest servici) pentru a obține o cheie (Application Key) pentru a ţine evidenţa utilizării serviciilor proprii în diverse sisteme software. Cheia permite şi descărcarea de pe Web a unei biblioteci care conține funcțiile ce trebuiesc apelate pentru accesarea serviciului de căutare. În continuare o să prezint conectarea la motorul de căutare BING.

Chiar dacă serviciul Bing API este gratuit, este nevoie de un ID de aplicaţie pentru a utiliza serviciul. Acest ID se poate crea accesând adresa http://www.bing.com/developers/createapp.aspx. De la acea adresă după înregistrare se obține un AppID. După crearea proiectului trebuie adăugată o referința web pentru adresa: http://api.search.live.net/search.wsdl?AppID=YourAppId, unde YourAppID este ID-ul obținut. În momentul achiziției ID-ului se preia şi biblioteca cu funcțiile API necesare conectării al motorul de căutare urmând ca în continuare să se utilizeze funcțiile din acele biblioteci pentru transmiterea interogării şi parametrizarea numărului de răspunsuri returnate de motorul de căutare. Astfel într-un obiect de tip *SearchRequest* se poate specifica întrebarea adresată motorului, numărul maxim de rezultate pe care le dorim de la motorul de căutare precum şi ofsetul adică de la ce index din lista proprie în continuare dorim să obținem rezultatele. La apelul metodei *Search* se realizează conectarea efectivă la motorul de căutare şi preluarea rezultatelor care vor fi întoarse utilizatorului într-un obiect de tip *SearchResponse* de unde ulterior aplicaţia le poate procesa.

## *3.2 Detectarea părţilor de vorbire*

Pentru a obţine de la motorul de căutare rezultate cât mai relevante cu interogare specificată trebuie specificat acestuia cuvintele de interogare esenţiale care se stabilesc pe baza analizei lexicale. De asemenea răspunsul este eticheta în funcţie de analiza sintactică a acestuia. Astfel problema majoră abordată în momentul de faţă este detectarea automată şi cât mai corectă a tipului întrebării (pentru a şti tipul de răspuns) şi părţilor de vorbire din cadrul unei propoziţii în limba engleză pentru a putea interpreta corect semantica propoziţiei. Domeniul care se ocupă cu „mineritul” sensurilor unui cuvânt se numeşte WSD (Word Sense Disambiguation) [Ste03] şi este un domeniu foarte apropiat de lexicografie care se ocupă de descoperirea şi descrierea sensurilor cuvintelor. În 1994 Hanks vorbeşte despre două feluri de extragere a înţelesurilor „normală şi exploratorie”. Un cuvânt are un sens comun şi în majoritatea timpului acest sens este cel utilizat de către interlocutor. Acest sens constituie sensul normal dar o limbă este întotdeauna deschisă interpretărilor, combinaţiilor şi astfel apar noi „setări” ale cuvintelor.

Metodele bazate pe cunoaştere reprezintă o categorie distinctă a WSD, alături de metodele bazate pe corpusuri. Performanţele acestora este întrecută de către cele bazate pe corpusuri, însă ele au o acoperire (în sensul de aplicabilitate) mai mare. De obicei aceste metode sunt aplicate dezambiguizării tuturor cuvintelor în texte nerestricţionate pe când cele bazate pe corpusuri sunt aplicate, în principiu, adnotării cuvintelor în corpusuri. În [Cret12] am prezentat o seria de algoritmi utilizaţi în WSD pentru descoperirea de cunoştinţe.

Pentru detectarea automată a părţilor de vorbire am folosit baza de date WordNet şi am dezvoltat o serie de algoritmi de predicţie a părţilor de vorbire. Această problemă este dificilă chiar şi pentru limba engleză deoarece o parte semnificativă din cuvintele limbii engleze au mai multe forme sintactice şi sunt cuvinte a căror sens depinde de context.

### 3.2.1 WordNet

Baza de date WordNet [WorNet] este des folosită în literatura de specialitate şi este un proiect lansat în 1985 la Princeton University. Este o bază de date lexicală iar structura sa se doreşte a fi o reţea semantică a sensurilor cuvintelor. Acest dicţionar semantic a fost proiectat pentru a stabili conexiuni între patru tipuri de părţi de vorbire: substantive, verbe, adjective şi adverbe.

Cea mai mică unitate din WordNet este synset, care reprezintă o anumită semnificaţie a unui cuvânt. Acesta include cuvântul, explicaţia acesteia, şi sinonimele sale. Înţelesul specific al unui cuvânt sub un singur tip de parte de vorbire este numit un sens. Fiecare sens al unui cuvânt este într-un synset diferit. Synsets sunt echivalente cu sensurile = structuri care conţin seturi de termeni cu sensuri sinonime. Fiecare synset are un gloss care defineşte conceptul care îl reprezintă. Cuvintele polisemantice pot aparţine mai multor synset-uri.

### 3.2.2 Metode de detectare a părţilor de vorbire

Pentru evaluarea algoritmilor realizaţi pentru detectarea părţilor de vorbire am folosit benchmark-ul Brown Corpus [BrownCorp] care reprezintă o colecţie de texte pentru cercetare în domeniul limbajelor naturale. Aceste benchmark-uri conţin după fiecare cuvânt sau semn de punctuaţie o etichetă care reprezintă tipul de vorbire iar fiecare tip de vorbire cu formele lui, existând aproximativ 80 de astfel de taguri disponibile. Am testat mai multe metode de etichetare iar pentru fiecare metodă am calculat acurateţea de etichetare ca şi raport între numărul de etichete corecte (comparativ cu etichetele deja existente pentru fiecare cuvânt din benchmark, pe care le-am considerat perfecte) şi numărul total de cuvinte de etichetat.

Folosind doar WordNet pentru a eticheta fiecare cuvânt din benchmark am obţinut o limită maximă a acurateţe de 72.93%. În cazul introducerii în algoritm şi a informaţiilor din limba engleză referitoare la sufixele unor cuvinte şi partea de vorbire am crescut această limită a acurateţei de etichetare la 83.4%. Gramatica limbii engleze are o serie de reguli întotdeauna valabile, de exemplu întotdeauna după articolul „the” urmează un substantiv sau un adjectiv. Am introdus şi aceste reguli în algoritm cu scopul de a reduce lista de părţi de vorbire posibile identificate la pasul anterior. Acurateţea de clasificare nu s-a îmbunătăţit în acest caz. Valorile maxime obtenabile pentru fiecare parte de vorbire întoarsă de către WordNet sunt prezentate în figura 2.

Ca şi metodă de etichetare utilizată pentru determinarea părţii de vorbire corecte în contextul în care se foloseşte WordNet, reguli şi sufixe este metoda care ia în considerare doar 3 cuvinte succesive din propoziţie. Metoda încearcă să predicţioneze partea de vorbire a unui cuvânt pe baza cuvântului anterior acestuia, pe baza cuvântului următor acestuia şi pe baza combinaţilor între toate sensurile respectivelor cuvinte, alegând întotdeauna combinaţia care obţine scorul cel mai mare. Acest scor se calculează în funcţie de probabilitatea de apariţie a unui tip de vorbire după şi înaintea unui alt tip de vorbire relativ la WordNet şi benchmark-ul Brown Corpus. Testat pe benchmark-ul Brown corpus această metodă obţine o acurateţe de etichetare de 68.83% în medie şi de 87.4% pentru etichetarea substantivelor.

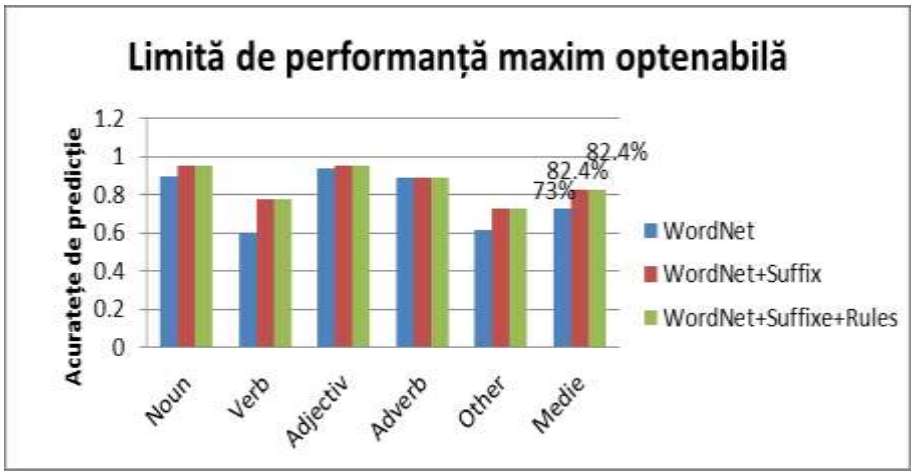


Figura 2. Limita maximă obţinută pentru fiecare parte de vorbire.

## 3.3 Modulele aplicaţiei de răspuns automat la întrebări

Aplicaţia de răspuns automat la întrebări conţine mai multe module principale: modulul de analiză a întrebări, modulul de conectare la motorul de căutare şi aducere snippet-uri, modulul de procesare a snippet-urilor şi modulul de extragere a răspunsurilor.

### 3.3.1 Modulul de analiză a întrebării

Modulul de analiză a întrebării are mai multe etape. În prima etapă se identifică tipul răspunsului( Person, Location, Time, etc) în funcţie de cuvântul cu care începe întrebarea şi astfel după anumite cuvinte cheie se obţine tipul de răspuns. O altă etapă este identificare cuvintelor esenţiale prin stabilirea părţii de vorbire a fiecărui cuvânt din propoziţie. În ultima etapă se formează interogarea din cuvintele esenţiale. Cuvintele esenţiale sunt acele cuvinte care definesc întrebarea utilizatorului şi astfel vom acorda o încredere mai mare răspunsurilor în care apar cuvintele esenţiale. Pentru identificarea cuvintelor esenţiale am folosit WordNet şi o serie de reguli şi sufixe folosite în limba engleză pentru stabilirea părţii de vorbire.

### 3.3.2 Modulele de aducere şi procesare snipet-uri

Modulul de aducere a snippet-urilor - are rolul de a realiza conectarea cu motorul de căutare, preia interogarea formată într-un query şi returnează o listă de snippet-uri.

Modulul de procesare a snippet-urilor – are rolul de a împărţii snippet-urile primite de la Modulul de Aducere a Snippet-urilor în propoziţii. Acest modul se ocupă de extragerea paragrafelor din fiecare snippet. Acest pas este necesar deoarece un snippet este alcătuit din una sau mai multe fraze, delimitate de semne de punctuaţie. Un snippet poate conţine mai multe propoziţii delimitate de anumite caractere dar poate conţine şi mai multe părţi din propoziţii diferite.

### 3.3.3 Modulul de extragere răspunsuri

Modulul de extragere răspunsuri are rolul de a analiza propoziţiile primite de la Modulul de procesare a snippet-urilor, caută răspunsurile probabile după tipul răspunsului şi calculează un punctaj pentru fiecare răspuns probabil. Fiecare propoziţie va fi analizată lexical, această analiză presupune identificarea cuvintelor din documentul de intrare. În principiu se va separa textul în cuvinte (token-uri), în funcţie de anumiţi separatori speciali. Se vor elimina cuvintele de legătură şi în funcţie de tipul de răspuns solicitat propoziţiile vor fi procesate diferit. Deocamdată se detectează răspunsuri de tip Person, Location, Time, Description şi Reason.

După ce a fost extrasă lista de răspunsuri posibile trebuie ca aceste răspunsuri să fie filtrate şi la fiecare răspuns, în funcţie de tipul de răspuns se stabileşte o anumită încredere. Încrederea este calculată în funcţie de nr. de cuvinte din interogare, nr. de propoziţii în care apare răspunsul, încrederea în site-ul de unde vine răspunsul şi o încredere bazată pe tipul de vorbire întors de WordNet pentru răspuns şi tipul de vorbire aşteptat pentru categoria respectivă de răspuns. De exemplu pentru o întrebare de tip Location ne aşteptăm ca şi cuvântul din răspuns să fie substantiv şi acordăm o valoare calculată pe baza sensurilor din cuvânt pe baza WordNet şi a regulilor.

# 4 Concluzii

Aplicaţia de răspuns automat la întrebări realizată practic se găseşte online la adresa http://193.226.29.26/ACUD/Default.aspx. Permite introducerea unei întrebări în limbaj natural, şi în această etapă deocamdată afişează o listă de snipet-uri în care se poate găsi răspunsul la întrebarea formulată. Această listă este ordonată în funcţie de încrederea pe care a primit-o fiecare snipte în parte. Această încredere pentru fiecare snippet se calculează pe baza tipurilor de vorbire ale cuvintelor din vecinătatea cuvintelor din interogare raportat la tipurile de vorbire pe care le aşteptăm în funcţie de tipul întrebării. Deocamdată aplicaţia funcţionează pentru 5 tipuri de întrebări (Person, Location, Time, Reason şi Description).

Deoarece este dificil de calculat care este acurateţea predicţiei de etichetare pentru algoritmii implementaţi în cazul utilizării direct în aplicaţia de răspuns automat am efectuat câteva experimente pe baza unui benchmark gata etichetat. Valoarea maximă obţinută de algoritmul

implementat este de 68.83% în medie pentru toate cele 4 tipuri de vorbire întoarse de WordNet şi este de 87.4% pentru predicţia etichetelor substantivelor. Pentru acest breanchmark am calculat de asemenea valoare maximă care am fi putut să o obţinem care este de 82.4% ceea ce arată că algoritmul implementat este încă departe de performanţele maxime. Încercările avute folosind algoritmi de învăţare (de exemplu Naive Bayes sau predictorul markov) pentru predicţia părţii de vorbire deocamdată nu a dat rezultatele asteptate şi a crescut timpul de răspusn pentru o astfel de aplicaţie destul de mult. De exemplu predictorul Markov pentru brenchmark-ul folosit a obţinut 68.53% dar a crescut timpul necesar aşteptării pentru răspuns având în vedere că predicţia părţii de vorbire este folosită atât în partea de analiză a întrebării cât şi în partea de analiză a snippet-urilor.

# 5 Referinţe bibliografice


[BrownCorp] Brown University Standard Corpus of Present-Day American English (Brown Corpus) http://icame.uib.no/brown/bcm.html.

[Cret12] R. Cretulescu., D. Morariu, M. Breazu, L. Vintan – Word Sense Disambiguation for Text Mining, The third international conference in Romania of Information Science and Information Literacy, ISSN 2247-0255, April 2012

[Han94] Hanks, P. “Linguistic Norms and Pragmatic Explanations, or Why Lexicographers need Prototype Theory and Vice Versa' in F. Kiefer, G. Kiss, and J.Pajzs (eds.), Papers in Computational Lexicography, 1994.

[Ste03] M. Stevenson, Word Senseisambiguation, The case for Combinations of Knowledge Sources, CSLI Publictions, 2003.

[WordNet] Princeton University WordNet-A lexical database for English /http://wordnet.princeton.edu.

[yippy] http://search.yippy.com/