Analizor semantic stocastic pentru sisteme automate de dialog vocal om-calculator

Dan Bohuş

Departamentul de Calculatoare
Facultatea de Automatică şi Calculatoare
Universitatea "Politehnica" din Timişoara
danbo@ear.utt.ro

Rezumat

Sistemele automate de dialog vocal om-calculator constituie un pas important pe drumul spre realizarea unor interfeţe perfect adaptate stilului de comunicare uman, iar un rol fundamental în aceste sisteme îl joacă analiza semantică a limbajului natural. În acest articol, după o scurtă introducere în domeniul sistemelor de dialog, este prezentat un analizor semantic stocastic proiectat în vederea includerii în asemenea sisteme. Sunt evaluate principalele altenative de proiectare, după care sunt descrise metodele şi algoritmii utilizaţi în construcţia analizorului. În încheiere sunt prezentate rezultatele obţinute până în prezent în construcţia şi evaluarea analizorului proiectat.

1. Sisteme de dialog

În paralel cu dezvoltarea sistemelor de calcul, interfeţele dintre acestea şi utilizator au cunoscut o evoluţie continuă, marcată de salturi de la o paradigmă la alta. Sistemele automate de dialog vocal om-calculator constituie un pas important pe acest drum spre o interfaţă multimodală ideală, adaptată perfect la stilul de comunicare uman.

Realizarea unor sisteme de dialog om-calculator, robuste şi cu performanţe ridicate, a devenit posibilă doar relativ recent, iar această "tinereţe" a domeniului se traduce într-o lipsă de metodologii de dezvoltare consacrate, lucru care face proiectarea lor o activitate anevoioasă, situată deseori la limita dintre ştiinţă şi artă (Munteanu, 1999). Indiferent însă de tipul şi domeniul sistemului de dialog, problemele ce trebuie rezolvate sunt în principiu aceleaşi: recunoaşterea vorbirii, analiza semantică, controlul dialogului, generarea răspunsurilor şi sinteza vorbirii. În consecinţă, majoritatea sistemelor sunt proiectate modular, putând fi identificată o arhitectură generică, dictată de aceste probleme (figura 1.)

- **Modulul de recunoaştere a vorbirii** preia semnalul vocal de la utilizator şi generează secvenţa de cuvinte cea mai probabilă.
- **Modulul de analiză semantică** primeşte textul produs de modulul de recunoaştere şi generează o reprezentare formalizată a sensului lui.
- **Modulul pentru controlul dialogului** este practic nucleul sistemului, cu rolul de gestionare a dialogului şi a interacţiunii cu utilizatorul.
- **Modulul de generare a răspunsurilor** este comandat de controlul dialogului, şi formulează răspunsul potrivit la un moment dat, folosind eventual şi informaţii dintr-o bază de date.
- **Modulul de conversie text-vorbire** preia răspunsul în formă textuală de la generatorul de răspunsuri, şi sintetizează forma sonoră a acestuia.

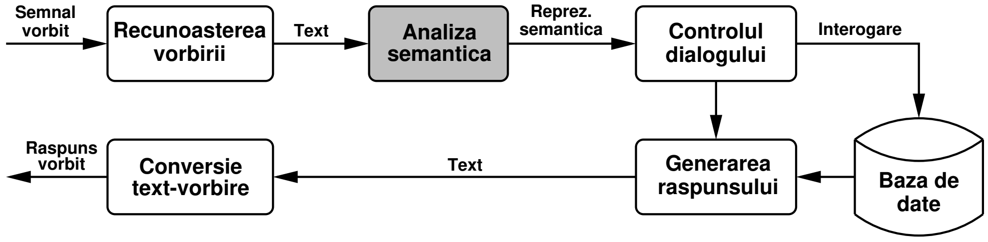


Figura 1 : Structura generala a unui sistem de dialog

Articolul tratează în continuare problema proiectării şi implementării modulului de analiză semantică. După o prezentare a principalelor alternative de proiectare, se trece la detalierea structurii unui analizor semantic realizat pe principii stocastice, ilustrând atât suportul teoretic cât şi soluţiile de implementare alese pentru fiecare componentă a acestuia. În final sunt prezentate câteva din experimentele efectuate şi rezultatele obţinute.

2. Alternative în analiza semantică

Analizorul semantic reprezintă o componentă esenţială în orice sistem de dialog, el având rolul de a extrage şi clarifica sensul formulărilor utilizatorului la nivelul independent de context, şi de a produce o reprezentare formalizată a acestuia, ce poate fi procesată în continuare de modulul de control al dialogului.

2.1. Reprezentarea cunoştinţelor

Primul pas în proiectarea unui analizor semantic îl constituie alegerea unui formalism adecvat pentru reprezentarea cunoştinţelor, a informaţiei extrase. Datorită legăturilor care se pot stabili între analiza semantică şi cea sintactică, a apărut natural ideea de a extinde gramaticile utilizate în analiza sintactică pentru a capta aspectele specifice celei semantice.

Utilizarea directă a gramaticilor sintactice Chomsky pentru extragerea şi reprezentarea conţinutului semantic presupune un anumit grad de expertiză lingvistică şi un efort foarte mare pentru elaborarea unui set de producţii care să capteze toate aspectele limbajului natural (Minker et al., 1999). În plus, chiar dacă am avea definită o astfel de gramatică, într-un sistem de dialog pot să apară formulări incorecte sintactic datorită diverselor efecte ce caracterizează vorbirea spontană (repetiţii, starturi false, disfluenţe, ezitări ş.a.m.d), dar care sunt totuşi perfect acceptabile în uzul curent. În consecinţă, gramaticile Chomsky nu reprezintă o opţiune viabilă pentru proiectarea unui analizor semantic pentru limbajul natural.

Pe lângă aceste gramatici, au fost definite o serie de alte formalisme, precum reţelele de tranziţii, gramaticile extinse, gramaticile arborescente, gramaticile semantice ş.a.m.d. Unele dintre acestea sunt ceva mai potrivite pentru modelarea limbajului natural, având şi o expresivitate crescută, însă toate suferă de o anumită rigiditate (moştenită de la gramaticile Chomsky) faţă de efectele specifice vorbirii spontane.

Mai potrivit pentru analiza semantică a limbajului natural este formalismul case-grammar, introdus de Fillmore în 1968 şi extins de Bruce în 1975. Elementul fundamental al acestuia este noţiunea de *cadru* (case-frame), incluzând un aşa-numit *concept* (fixat) şi o mulţime de *sloturi* care sunt completate cu informaţii ce reprezintă "cunoştinţele" sistemului. Activitatea de completare a sloturilor unui cadru o vom numi *instanţierea cadrului*. Figura 2 ilustrează reprezentarea semantică a unei formulări ipotetice a unui utilizator în discuţia sa cu un sistem de dialog pentru furnizarea de informaţii despre orar.

Când are grupa doi curs cu profesorul Popescu ?

<**când**>	
an:	-
grupa:	2
subgrupa:	-
tip-materie:	curs
profesor:	Popescu

Figura 2: Exemplu de cadru

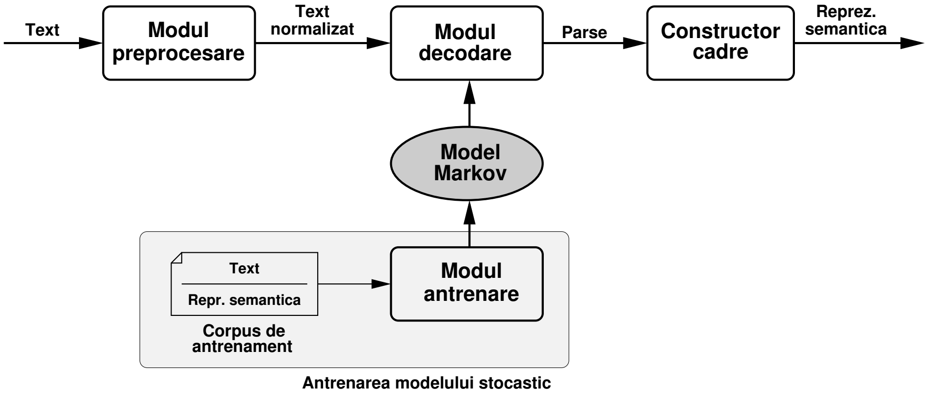
O a doua noţiune importantă în formalismul case-grammar este aceea de *marcaj* (case-marker). Un marcaj este practic un cuvânt din text care realizează o anumită constrângere în ceea ce priveşte completarea sloturilor (ex: cuvântul **grupa** este un marcaj indicând faptul că urmează probabil valoarea slotului `grupa`). Acest gen de constrângeri lexicalizate în instanţierea cadrelor modelează practic sintaxa limbajului, limitând construcţiile interpretabile prin formalismul case-grammar. Extragerea efectivă a informaţiei se bazează în mare măsură pe marcaje, ele indicând unde se află informaţia şi în acelaşi timp clarificând sensul ei.

Pornind de la cadre individuale, se poate realiza un *sistem de cadre* (case-system) care să exprime prin legături interdependenţele semantice dintre cadrele ce îl compun. Legăturile sunt materializate prin faptul că un slot al unui cadru, în loc să fie completat cu o valoare explicită, este completat cu o legătură spre un alt cadru din sistem. Prin formalism case-grammar se înţelege deci un astfel de sistem de cadre, având aspectele sintactice incluse prin constrângerile lexicalizate, dictate de marcaje, în instanţierea sloturilor (Minker et al., 1999).

În concluzie, a fost ales formalismul case-grammar, el fiind cel mai potrivit pentru reprezentarea cunoştinţelor în analizorul semantic proiectat.

2.2. Metoda de parsing

Odată stabilită modalitatea de reprezentare a cunoştinţelor, următorul pas este stabilirea metodei pe care o vom utiliza pentru extragerea efectivă a informaţiei din text. Aici metodele existente se grupează în două categorii: *metode bazate pe reguli* şi *metode stocastice*.




```
Text:             (aa) ce profesor tine laboratorul de s e la grupa (aa) doi
Text normalizat: ce-profesor laborator [MATERIA:"Sisteme expert"] grupa [NR:"2"]
Parse:            <ce-profesor> (m:laborator) (v:laborator) (m:grupa) (v:grupa)
Repr. semantica: <ce-profesor>
                    <identificare>
                      grupa = 2
                    <specif-materie>
                      laborator = "Sisteme expert"
```

Figura 3: Arhitectura analizorului semantic

Soluţiile din prima categorie utilizează un set de reguli care controlează modul în care se face identificarea conceptelor şi completarea sloturilor cu valorile corespunzătoare din text. În general, regulile sunt lexicalizate, definind familiile de cuvinte care identifică conceptele/cadrele, şi marcajele. În plus, trebuie definite reguli care descriu legăturile dintre marcaje şi valori. Pe baza acestora se pot determina cuvintele care au conţinut semantic şi dau valori sloturilor cadrului identificat.

Alternativa o reprezintă metodele de parsing stocastic. Aici ideea fundamentală este de a utiliza un model stocastic pentru decodarea semantică a formulărilor utilizatorului. Acest model se comportă practic ca un mecanism de învăţare automată care operează în două moduri: în modul de antrenare, el primeşte la intrare perechi (text, reprezentare semantică), şi îşi reajustează parametrii interni pe baza unor algoritmi specifici astfel încât să capteze statistic corespondenţa dintre text şi reprezentarea semantică. În al doilea mod, de decodare, modelul primeşte la intrare doar textul, şi furnizează reprezentarea lui semantică cea mai probabilă.

Metoda de parsing stocastic prezintă numeroase avantaje faţă de soluţiile bazate pe reguli: pe de o parte este eliminată complet necesitatea definirii unui set de reguli, acestea fiind practic învăţate automat din corpusul de antrenament, iar pe de altă parte creşte robusteţea sistemului şi gradul lui de flexibilitate. În plus, în cazul portării analizorului în alt domeniu de dialog (sau la extinderea domeniului), singurul lucru necesar este o reantrenare a modelului cu un corpus de formulări adecvat (Minker et al., 1999).

Datorită avantajelor prezentate, soluţia aleasă pentru analizorul proiectat este una stocastică.

3. Arhitectura analizorului semantic

Structura analizor semantic stocastic utilizând formalismul case-grammar este ilustrată în figura 3. În continuare vom descrie funcţionalitatea fiecăruia dintre modulele componente.

3.1. Modulul de preprocesare

Modulul de preprocesare este situat la graniţa dintre modulul de recunoaştere a vorbirii şi decodorul semantic propriu-zis. Rolul principal al acestuia este de a realiza o normalizare a textului primit de la modulul de recunoaştere. Preprocesarea constă în efectuarea unor operaţii elementare asupra textului:

- **Eliminarea evenimentelor nonlexicale** - urmăreşte eliminarea eventualelor evenimente acustice transcrise, dar fără conţinut semantic (ezitări, zgomote etc.)

- **Normalizarea numerelor** - realizează transformarea numerelor din forma scrisă furnizată de modulul de recunoaştere în reprezentarea cu cifre arabe corespunzătoare.

- **Reducerea flexiunilor** - spre deosebire de alte limbi, limba română este puternic flexionară, lucru care face oportună introducerea unui mecanism de reducere a flexiunilor. Astfel se realizează o normalizare a intrării şi o reducere a vocabularului domeniului, fără a afecta prea mult conţinutul semantic al formulării.

- **Unificarea expresiilor** - realizează unificarea sub formă de expresii a unor grupuri de cuvinte cu un corespondent semantic unitar.

- **Înlocuiri de alias-uri** - este o operaţie dependentă de domeniul sistemului, care înlocuieşte alias-uri (prescurtări, sinonime etc.) ale anumitor cuvinte sau formulări.

- **Unificarea categoriilor** - are ca scop gruparea cuvintelor în categorii semantice pentru a reduce dimensiunile vocabularului decodorului şi a facilita analiza ulterioară.

- **Eliminarea cuvintelor din afara domeniului** - implică eliminarea cuvintelor care nu sunt relevante în contextul domeniului stabilit al sistemului de dialog.

3.2. Modelul Markov: antrenare si decodare

Problema analizei semantice este în esenţă una de decodare: scopul analizorului semantic este de a obţine corespondentul semantic (într-un anumit formalism) al textului de intrare. În acest sens, analiza semantică, privită ca o decodare a textului de intrare, poate fi realizată folosind un model Markov cu stări ascunse (HMM).

Modelul acţionează ca un decodificator care generează secvenţa de etichete semantice (parse) cea mai probabilă, dată fiind o secvenţă de intrare. Cuvintele din textul normalizat vor juca rolul de simboluri observate ale modelului, iar etichetele semantice vor corespunde stărilor. Astfel, dat fiind un model $\mathcal{M}$ şi o secvenţă de intrare $\mathcal{O}$ (textul de intrare) se poate determina prin intermediul unor algoritmi specifici cea mai probabilă succesiune de stări interne $\mathcal{S}$ care generează intrarea $\mathcal{O}$, cu alte cuvinte, cea mai probabilă succesiune de etichete semantice corespondente.

Rămâne problema creării modelului $\mathcal{M}$, astfel încât decodarea să se desfăşoare corect, adică să respecte semantica domeniului şi a limbajului în discuţie. Aceasta se realizează într-o etapă premergătoare de antrenare, în care resursa esenţială este un corpus de formulări utilizator tipice pentru sistemul de dialog proiectat, etichetate semantic în prealabil. Evident, dimensiunile modelului (numărul de stări şi simboluri observate) sunt fixate de numărul de etichete semantice, respectiv dimensiunea vocabularului normalizat. Odată stabiliţi aceşti parametri, parametii modelului (distribuţiile de probabilitate) A, B şi Π sunt estimaţi pe baza corpusului de antrenament. Matricea A, reprezentând probabilităţile de tranziţie a stărilor, va capta practic modurile posibile în care etichetele semantice se pot succeda. În mod similar, B, reprezentând probabilităţile simbolurilor observate, va capta corespondenţa dintre etichetele semantice şi lexemele normalizate, cele două distribuţii modelând împreună corespondenţa dintre textul normalizat şi reprezentarea sa semantică.

Estimarea distribuţiilor de probabilitate A, B şi Π se poate face prin simpla numărare a apariţiilor evenimentelor corespunzătoare (tranziţii între stări şi corespondenţe stări-simboluri) în corpusul de antrenament. Datorită însă dimensiunilor în general reduse ale corpusurilor de antrenament, este posibil ca acestea să nu surprindă toate succesiunile valide de etichete semantice. Pentru modelarea situaţiilor de acest tip s-a utilizat reestimarea Katz (Katz, 1987) în calculul parametrilor modelului. Ideea fundamentală a acesteia este de a micşora probabilitatea unora din tranziţiilor observate prin înlocuirea lor cu estimări Turing, şi a redistribui masa de probabilitate astfel eliberată între evenimentele neobservate în corpusul de antrenament (vezi figura 4).

Odată antrenat modelul stocastic, obţinerea succesiunii de etichete semantice corespunzătoare unui text de intrare se face prin algoritmului Viterbi (Rabiner şi Juang, 1986) – un algoritm tipic de programare dinamică având complexitatea $O(Tn^2)$ (unde T este lungimea textului de intrare, iar n numărul de stări a modelului). Acesta, împreună cu algoritmii de antrenare şi reestimare, constituie o bază eficientă pentru realizarea analizei semantice stocastice în sistemele de dialog.

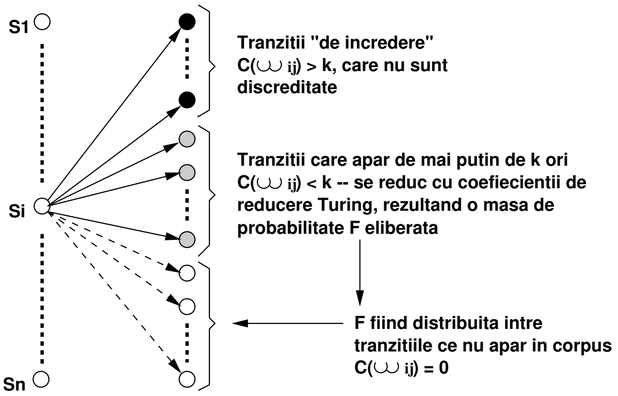


Figura 4: Principiul reestimării Katz

3.3. Constructorul de cadre

Constructorul de cadre este ultimul modul utilizat în lanţul de decodare semantică. El primeşte de la modulul de decodare succesiunea de etichete semantice şi, pornind de la aceasta, construieşte reprezentarea prin cadre a formulării utilizatorului.

Algoritmul utilizat este relativ simplu. Într-un prim pas se identifică printr-o parcurgere a succesiunii de etichete semantice toate conceptele care apar. Pe baza lor, cunoscând întregul sistem de cadre utilizat, constructorul instanţiază cadrele necesare, împreună cu eventualele subcadre ale acestora. Trebuie remarcat că, datorită modalităţii stocastice de realizare a decodării semantice, nu există nici o garanţie că succesiunea de etichete semantice va conţine un concept: într-o astfel de succesiune pot exista nici una, una, sau mai multe etichete-concept. Primul caz necesită o tratare aparte, iar soluţia utilizată este ca modulul de control al dialogului să furnizeze în lista de concepte aşteptate, împreună cu probabilitaţile asociate.

4. Soluţii de implementare

Implementarea efectivă a fost realizată în limbajul C++, ales datorită eficienţei, a capabilităţilor de programare orientată pe obiecte şi a portabilităţii codului. Dezvoltarea propriu-zisă a fost făcută pentru sistemul de operare Linux, utilizând mediul de dezvoltare integrată KDevelop (KDevelop, 2000).

Pentru implementarea modulelor menţionate au fost dezvoltate o serie de clase care pot fi împărţite în două categorii: clase ce modelează datele care apar în analiza semantică – `CSequence`, `CStringSequence`, `CCaseFrame`, `CCaseFrameSystem`, `CUtterance`, `CUtteranceCorpus` şi `CDictionary`, şi clase care implementează algoritmii utilizaţi şi funcţionalitatea modulelor implicate – `CPreprocessor`, `CHMM`, `CCaseFrameBuilder`, `CSemanticAnalyzer`.

Forma finală în care analizorul este distribuit este cea a unei biblioteci statice de clase, pentru a fi cu uşurinţă inclus într-un sistem de dialog proiectat într-un cadru mai larg. Pe baza acestei biblioteci pot fi realizate implementări ale analizorului semantic sub diverse forme: programe filtru, care lucrează în pipe, servere la nivel sockets, RPC (Remote Procedure Call) sau RMI (Remote Method Invocation) etc.

5. Experimente şi rezultate

Pentru evaluare, analizorului semantic descris anterior este în curs de integrare într-un sistem de dialog pentru furnizarea de informaţii despre orarul Departamentului de Calculatoare. Primul pas în acest sens l-a constituit o analiză preliminară a domeniului, prin care s-a realizat o definire a limitelor analizorului şi a noţiunilor cu care acesta va opera: materii, profesori, săli de clasă, informaţii de identificare (ani de studii, specializări, grupe, subgrupe), specificatori de timp (zile, intervale orare, perioade ale zilei) etc.

Al doilea pas a constat în culegerea de formulări utilizator pe baza cărora să se realizeze antrenarea modelului Markov. Folosind mediul pentru dezvoltarea sistemelor de dialog MDWOZ (Munteanu şi Boldea, 2000), până în prezent au fost culese, în trei etape succesive, trei astfel de corpusuri (totalizând 1088 formulări utilizator). Pentru o evaluare a performanţelor analizorului neafectată de eventuale erori anterioare ale modulului de recunoaştere, semnalele audio au fost transcrise manual.

Construirea analizorului semantic (crearea fişierelor de control pentru preprocesare, antrenarea modelului şi specificaţiile sistemului de cadre) se realizează incremental (metoda bootstrapping). Ideea principală este de a crea modele succesive pe baza unor porţiuni din ce în ce mai mari din corpusul de antrenament, şi de a le utiliza pentru a eticheta automat noi porţiuni din corpus. În cazul nostru, cele trei corpusuri de antrenament vor fi utilizate în trei etape succesive de antrenare a modelului.

Astfel, într-o primă etapă, de iniţializare, formulările din primul corpus de antrenament au fost preprocesate şi etichetate semantic manual, eliminând formulările care ieşeau din domeniul de dialog prestabilit. În baza acestei etape manuale a fost construită o primă varintă pentru fişierele de control ale preprocesorului. Deasemeni s-au identificat etichetele semantice şi cadrele necesare pentru domeniul specificat. Pe baza acestui corpus etichetat manual a fost creat prin antrenare un prim model, utilizând estimarea de probabilitate maximă (reestimarea Katz va fi aplicată doar la ultima fază de antrenare a modelului).

Evaluarea analizorului semantic după această fază de iniţializare se poate face la mai multe nivele: al preprocesorului, al decodorului semantic, şi global. Evaluarea preprocesării se face evident prin verificarea corectitudinii ei. O metrică importantă în acest sens o reprezintă măsura în care se stabilizează dicţionarul de intrare al decodorului (preprocesorul nu este "surprins" de cuvinte noi). Evaluarea decodificării se face prin verificarea corectitudinii etichetării semantice, iar la nivel global evaluarea se face prin verificarea corectitudinii cadrelor generate.

Analizorul semantic configurat pe baza primului corpus de antrenament a fost utilizat pentru procesarea automată a celui de-al doilea. Cu această ocazie a fost realizată şi o primă evaluare a analizorului după criteriile enunţate mai sus (tabelul 1).

Evaluare analizor		
Model-1	Număr formulări de test	317
	Erori de preprocesare (procente)	35 11.0%
	Erori de decodare (procente)	37 11.8%
	Erori de construcţie cadre (procente)	1 0.3%
	Total erori **(procente)**	**73** **23.1%**

Tabelul 1: Rezultate test Model-1

După corectarea erorilor apărute, al doilea corpus (acum etichetat semantic) a fost utilizat pentru a realiza o reantrenare a modelului Markov. În baza acestui nou model s-a realizat etichetarea semantică a celui de-al treilea corpus de antrenament. În prezent este în curs de desfăşurare activitatea de evaluare şi corectare manuală a acestor rezultate, care vor fi utilizate într-o nouă etapa de reantrenare, obtinându-se astfel formele finale ale modelului Markov, fişierelor de control al preprocesării şi sistemului de cadre. În paralel se realizează şi culegerea celui de-al patrulea corpus de formulări, care va fi utilizat pentru evaluarea finală a performanţelor analizorului.

Modelul final va fi ameliorat prin reestimare Katz, după care se va face o evaluare detaliată comparativă a performanţelor celor două analizoare semantice rezultate (cu şi fără reestimare Katz) prin teste pe ultimul corpus cules.

6. Concluzii şi continuări

În acest articol au fost prezentate structura, proiectarea, configurarea şi o primă evaluare a performanţelor unui analizor semantic stocastic pentru sisteme automate de dialog om-calculator. Analizorul utilizează un model Markov cu stări ascunse pentru a realiza decodarea semantică, şi generează reprezentări semantice în formalismul case-grammar.

Configurarea incrementală a analizorului pentru un domeniu de dialog ales în vederea testării metodologiei şi evaluării performanţelor este încă în curs de desfăşurare. După completarea acesteia şi evaluarea detaliată a performanţelor, cercetările vor continuă în două direcţii: îmbunătăţirea performanţelor prin reantrenări succesive şi optimizarea algoritmilor utilizaţi (e.g., integrarea unor prelucrări bazate pe automate finite şi dicţionare morfosintactice în faza de preprocesare (Bohuş şi Boldea, 2000)) şi integrarea analizorului într-un sistem de dialog complet, cu evaluarea performanţelor în acest context.

7. Bibliografie

Bohuş, D. şi M. Boldea, 2000. A Web-based Text Corpora Development System. In *Proceedings Second International Conference on Language Resources and Evaluation*. Atena, Grecia.

Katz, S.M., 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*.

KDevelop, 2000. http://www.kdevelop.org.

Minker, W., A. Waibel, şi J. Mariani, 1999. *Stochastically-Based Semantic Analysis*. Kluwer Academic Publishers.

Munteanu, C., 1999. Mediu pentru dezvoltarea sistemelor de dialog prin metoda Vrăjitorului din Oz. Lucrare de diplomă. Departamentul de Calculatoare, Universitatea "Politehnica" din Timişoara.

Munteanu, C. şi M. Boldea, 2000. MDWOZ: A Wizard of Oz Environment for Dialog Systems Development. In *Proceedings Second International Conference on Language Resources and Evaluation*. Atena, Grecia.

Rabiner, L.R. şi B.H. Juang, 1986. An Introduction to Hidden Markov Models. *IEEE Acoustics, Speech and Signal Processing Magazine*.