Lucrările atelierului
Resurse lingvistice şi instrumente pentru prelucrarea limbii române
Iaşi, 14-15 decembrie 2007

Editura Universităţii "Alexandru Ioan Cuza" Iaşi

Volum apărut cu sprijinul Ministerului Educației şi Cercetării,
prin Autoritatea Națională pentru Cercetarea Ştiințifică

Lucrările atelierului
Resurse lingvistice şi instrumente pentru prelucrarea limbii române
Iaşi, 14-15 decembrie 2007

Editori:
Ionuț Cristian Pistol
Dan Cristea
Dan Tufiş

Organizatori:
Facultatea de Informatică,
Universitatea „Alexandru Ioan Cuza” Iaşi

Institutul de Cercetări pentru Inteligență Artificială
Academia Română, Bucureşti

Institutul de Informatică Teoretică
Academia Română, Filiala Iaşi

COMITETUL DE PROGRAM

Corneliu Burileanu, Facultatea de Electronică, Universitatea Politehnica Bucureşti şi Institutul de Cercetări în Inteligență Artificială, A.R., Bucureşti

Monica Busuioc, Institutul de Lingvistică "Iorgu Iordan - Al. Rosetti", A.R., Bucureşti

Constantin Ciubotaru, Institutul de Matematică şi Informatică, Academia de Ştiințe a Moldovei, Chişinău, R. Moldova

Svetlana Cojocaru, Institutul de Matematică şi Informatică, Academia de Ştiințe a Moldovei, Chişinău, R. Moldova

Dan Cristea, Facultatea de Informatică, Universitatea "Al. I. Cuza" şi Institutul de Informatică Teoretică, A.R., Iaşi

Nicolae Curteanu, Institutul de Informatică Teoretică, A.R., Iaşi

Cristina Florescu, Institutul de Filologie Română "Al. Philippide", A.R., Iaşi

Corina Forăscu, Facultatea de Informatică, Universitatea "Al. I. Cuza", Iaşi şi Institutul de Cercetări în Inteligență Artificială, A.R., Bucureşti

Maria Georgescul, ISSCO / TIM, ETI, Universitatea Geneva, Elveția

Gabriela Haja, Institutul de Filologie Română "Al. Philippide", A.R., Iaşi

Cătălina Hallett, Open University, Anglia

Radu Ion, Institutul de Cercetări în Inteligență Artificială, A.R., Bucureşti

Rodica Marian, Institutul de Lingvistică şi Istorie Literară "Sextil Puşcariu", A.R., Cluj-Napoca

Rada Mihalcea, Universitatea North Texas, SUA

Vivi Năstase, EML Research, Germania

Constantin Orăsan, Universitatea Wolverhampton, Anglia

Oana Postolache, ISI - Universitatea California, SUA

Irina Prodanoff, ILC-Pisa şi Universitatea Pavia, Italia

Georgiana Puşcaşu, Universitatea Wolverhampton, Anglia

Violeta Seretan, Departamentul de lingvistică, Universitatea Geneva, Elveția

Valentin Tablan, Universitatea Sheffield, Anglia

Amalia Todiraşcu, Universitatea Marc Bloch, Strasbourg, Franța

Doina Tătar, Universitatea "Babeş-Bolyai", Cluj-Napoca

Horia-Nicolai Teodorescu, Institutul de Informatică Teoretică, A.R. şi Universitatea Tehnică, Iaşi

Dan Tufiş, Institutul de Cercetări în Inteligență Artificială, A.R., Bucureşti şi Universitatea "Al. I. Cuza", Iaşi

Ioana Vintilă-Rădulescu, Institutul de Lingvistică "Iorgu Iordan - Al. Rosetti", A.R., Bucureşti

Adriana Vlad, Facultatea de Electronică, Universitatea Politehnica Bucureşti şi Institutul de Cercetări în Inteligență Artificială, A.R., Bucureşti

COMITETUL DE ORGANIZARE

Dan Cristea, FII-UAIC şi IIT-AR (dcristea@info.uaic.ro)

Corina Forăscu, FII-UAIC şi ICIA-AR (corinfor@info.uaic.ro)

Dan Tufiş, ICIA-AR şi FII-UAIC (tufis@racai.ro)

Cuprins

CUVÂNT ÎNAINTE

Atelierul de lucru "Resurse lingvistice şi instrumente pentru prelucrarea limbii române" a ajuns la a cincea ediţie, cu o participare sporită atât numeric cât şi calitativ. Dacă primele două întâlniri, organizate de Comisia Academiei Române de informatizare pentru limba română au fost mai formale şi restrânse la membrii Comisiei, începând de la cea de a treia ediţie (noiembrie 2005, Iaşi) manifestarea a devenit deschisă şi, ca atare, de mai mare amploare. Prin organizarea ei în regim de tele-conferinţă, manifestarea a permis participarea activă, foarte apreciată, a unor specialişti români care lucrează în diferite institute şi universităţi din străinătate, dar şi audierea lucrărilor de către specialişti interesaţi care, din varii motive, nu au putut participa "in situ". Atunci, dat fiind interesul manifestat de o comunitate mai largă decât cea a Comisiei şi respectiv Consorţiului de Informatizare pentru Limba Română, a apărut ideea publicării contribuţiilor atelierului nostru. Cu sprijinul Ministerului Educaţiei şi Cercetării, lucrările celui de al IV-lea atelier de lucru (noiembrie 2006) au fost editate într-un volum publicat la Editura Universităţii "Al. I. Cuza" şi de asemenea pe situl *Consorţiului de Informatizare pentru Limba Română* (http://consilr.info.uaic.ro). Cu un număr de peste 4500 de vizitatori, foarte mulţi din străinătate, primul volum al seriei *"Resurse lingvistice şi instrumente pentru prelucrarea limbii române"* a avut un impact semnificativ în lumea ştiinţifică. În prefaţa volumului apărut anul trecut, ne exprimam speranţa că vom reuşi publicarea lucrărilor de la ediţiile viitoare. Şi iată că al doilea volum al seriei a ajuns sub ochii dumneavoastră.

Comitetul de program, format din specialişti de primă mână, din ţară şi din străinătate, a fost în acest an mai selectiv, dintre cele 26 lucrări transmise Atelierului fiind reţinute pentru prezentare şi publicare ulterioară doar 19. În acest fel am putut atribui fiecărei lucrări un spaţiu mai mare, de aprox. 10 pagini. Cele aproape 5000 de vizite ale sitului Atelierului din 2007, contorizate până la data publicării acestui volum, atestă că varianta electronică a volumului este deja aşteptată cu mare interes.

Am menţinut aceleaşi titluri de capitole ca în primul volum al seriei, corespunzătoare direcţiilor pe care le-am considerat dominante în domeniu, respectiv prelucrarea vorbirii, prelucrarea textelor, aplicaţii ale tehnologiilor lingvistice şi modelare lingvistică. Comparând numărul de lucrări din cele două volume, grupate în aceste subdomenii, se observă o relativă creştere a interesului în zona prelucrării vorbirii (16% faţă de 7%), a puternică creştere a interesului în domeniul aplicaţiilor lingvistice (de la 34,5% la 47,5%), dar şi o scădere a numărului de lucrări din zona modelărilor lingvistice (de la 20,5% la 5%). Credem că această ultimă tendinţă este una întâmplătoare şi nu reflectă scăderea interesului cercetătorilor pentru descrieri teoretice dedicate limbii române.

Ca şi anul trecut, întâlnirea a fost găzduită de Biblioteca Facultăţii de Informatică a Universităţii „Al. I. Cuza” din Iaşi şi a beneficiat de implicarea MECT în finanţare. Această carte n-ar fi putut fi tipărită fără această generoasă finanţare şi fără sprijinul Editurii Universităţii „Al.I.Cuza” Iaşi. Îi suntem recunoscători domnului Eugen Rotariu de la firma IntegraSoft pentru oferirea sistemului Hermix, care ne-a permis să îmbunătăţim condiţiile de tele-participare la lucrările Atelierului. Le mulţumim, de asemenea, participanţilor la atelier, aflaţi în sală sau conectaţi prin Internet, cât şi membrilor comitetului de program care ne-au ajutat să îmbunătăţim calitatea lucrărilor.

Editorii Iaşi, ianuarie 2008

CAPITOLUL 1

RESURSE LINGVISTICE PENTRU PRELUCRAREA VORBIRII

ATLAS LINGVISTIC ELECTRONIC

SILVIU BEJINARIU[1], VASILE APOPEI[1], RAMONA LUCA[1], LUMINIȚA BOTOŞINEANU[2], FLORIN OLARIU[2]

[1]*Institutul de Informatică Teoretică,* [2]*Institutul de Filologie Română „A. Philippide”, Academia Română, Filiala Iaşi*

silviu.bejinariu@gmail.com, vapopei@iit.tuiasi.ro, ramona.luca@gmail.com, lumi.botosineanu@gmail.com, olariuft@yahoo.com

Rezumat

Aplicațiile ALR si EditTD au stat la baza realizării prospectului celui de-al III-lea volum al *Noului Atlas Lingvistic român, pe regiuni. Moldova şi Bucovina*, apărut în anul 2005 sub formă de volum şi CD multimedia, precum şi a publicării volumului complet al atlasului, aflat în prezent în faza finală de pregătire pentru tipar. Lucrarea prezintă o parte dintre facilitățile oferite de cele două aplicații, a căror implementare este finalizată: editarea transcrierilor fonetice pentru varietățile regionale ale limbii române, sistemul pentru sinteza simbolurilor asociate, întreținerea dicționarelor asociate *Noului Atlas lingvistic român, pe regiuni. Moldova şi Bucovina*, generarea automată a planşelor cu hărți lingvistice şi material necartografiat, instrumente pentru generarea de hărți sintetice, cu gruparea punctelor de anchetă după fenomene fonetice, ocurența anumitor termeni sau în funcție de criteriul semantic, editarea planşelor generate automat şi generarea de hărți combinate, generarea automată a planşelor combinate cu material necartografiat pentru mai multe cuvinte de bază, funcții de căutare şi prelucrare a informației, generarea automată pentru indexul de cuvinte şi forme, editarea de texte dialectale.

1. Introducere

Aplicațiile ALR si EditTD reprezintă rodul colaborării colectivelor de la Institutul de Informatică Teoretică şi Institutul de Filologie Română „A. Philippide” din Iaşi, în cadrul proiectului de cercetare interdisciplinar „Proiectarea şi implementarea unui sistem integrat de aplicații software pentru editarea textelor dialectale şi realizarea *Noului Atlas lingvistic român, pe regiuni*”. Implementarea celor două aplicații este finalizată, iar sistemul dezvoltat a stat la baza realizării prospectului celui de-al III-lea volum al *Noului Atlas lingvistic român, pe regiuni. Moldova şi Bucovina* (NALR-Mold. Bucov.), apărut în anul 2005 sub formă de volum şi CD multimedia, precum şi a publicării volumului complet al atlasului, aflat în prezent în faza finală de pregătire pentru tipar.

La nivel mondial, editarea asistată de calculator a atlaselor lingvistice, care a debutat prin simpla generare pe calculator a simbolurilor necesare pentru transcrierea fonetică a răspunsurilor din anchetă, înregistrează o primă etapă notabilă prin apariția atlasului lingvistic sonor intitulat *L'Atlante linguistico del ladino centrale e dialetti limitrofi* (Hans Goebl şi Roland Bauer, 1978-1989), care se întemeiază pe asocierea bazei de date constituită din materialul de anchetă prezentat în transcriere fonetică cu fişierul audio corespunzător. Mai aproape, în timp şi în privința concepției de ansamblu, de inițiativa de informatizare a autorilor şi colaboratorilor NALR.-Mold. Bucov. este viziunea care a stat la baza elaborării *Atlasului lingvistic italian* (*Atlante linguistico italiano*…, L. Massobrio, G. Ronco et alii, vol. I, 1995; vol. II, 1996; vol. III, 1997), care izbuteşte performanța de a se menține pe linia cartografiei

lingvistice de tradiție clasică valorificând în acelaşi timp de resursele tehnoredactării asistate de calculator. În aceeaşi ordine de idei trebuie menționat şi *Atlas multimédia prosodique de l'espace roman* (AMPER), proiect inițiat de Centrul de Dialectologie al Universității Stendhal–Grenoble 3, care vizează proiectarea cartografică a variabilității intonaționale în spațiul romanic cu ajutorul mijloacelor puse la dispoziția lingviştilor de noile direcții informatice. În domeniul românesc, opțiunea pentru exploatarea acestor resurse în beneficiul geografiei lingvistice i-a mai atras şi pe alți autori ai seriei NALR (între care autorii *Noului Atlas lingvistic român, pe regiuni. Crişana*), care însă, până în momentul de față, nu au ajuns la rezultate semnificative, aşa încât aplicația ALR de editare a NALR.-Mold. Bucov. se prezintă în prezent ca singura realizare românească de acest fel a cărei aplicabilitate practică a fost deja verificată. (pentru informații mai detaliate despre istoricul atlaselor lingvistice editate electronic, vezi St. Dumistrăcel, *Prefață* la Vasile Arvinte et al., 2007).

Sistemul software care modelează atlasul lingvistic electronic conține module care realizează gestionarea următoarelor grupe de informații:

1. simboluri pentru editarea transcrierilor fonetice;
2. dicționarele atlasului lingvistic (cuvinte de bază, puncte de anchetă, transcrieri fonetice);
3. informații grafice pentru descrierea hărților;
4. planşele atlasului lingvistic, care pot fi consultate şi / sau tipărite;
5. texte dialectale.

Din punct de vedere funcțional, atlasul lingvistic electronic este structurat în două componente principale (vezi Figura 1):

6. proceduri pentru pregătirea datelor primare;
7. interfața multimedia.

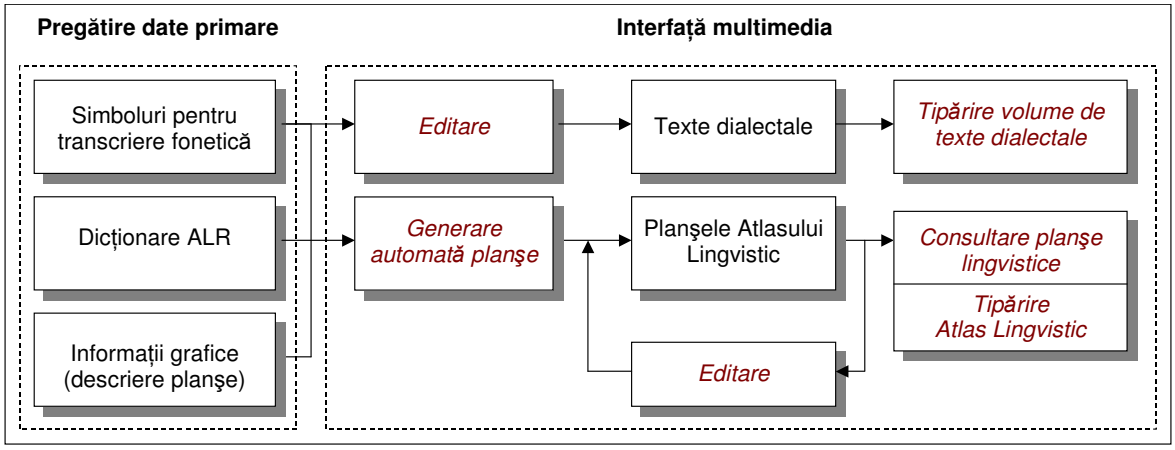


Figura 1: Componentele atlasului lingvistic electronic

În continuare vor fi detaliate cele mai importante facilități puse la dispoziție de cele două aplicații.

2. *Editarea transcrierilor fonetice pentru limba română*

Simbolurile folosite pentru editarea transcrierilor fonetice sunt clasificate după cum urmează:

– semne grafice care au drept corespondente sunete „primare":

- litere (vocale sau consoane) existente în alfabetul latin şi care se regăsesc pe tastatură;
- litere (vocale sau consoane) cu semne diacritice, care nu se regăsesc pe tastatură, dar pot fi obținute prin combinații de taste;

– semne grafice care au drept corespondente sunete marcate de unul sau mai multe fenomene fonetice.

Sunt definite un număr de 17 vocale „primare", fiecare dintre acestea având un număr de 3 variante accentuate (a - á à ấ). Fenomenele fonetice asociate vocalelor, în număr de 12, sunt clasificate în 5 grupe. Fiecărei vocale îi pot fi aplicate până la 5 fenomene fonetice, cel mult unul din fiecare grupă.

În cazul consoanelor, fenomenele fonetice sunt în număr de 9 şi sunt grupate tot în 5 categorii, dar fiecărei consoane îi pot fi aplicate simultan cel mult două fenomene fonetice. În acest caz, multe dintre combinații nu sunt posibile.

Din cele descrise mai sus rezultă că pentru editarea transcrierilor fonetice ar fi necesară proiectarea unui număr de aproximativ 400 de fonturi (vocale şi consoane) a câte 80 corpuri de literă fiecare, care să permită afişarea tuturor caracterelor, cu toate combinațiile posibile de fenomene fonetice (Bejinariu et al., 2000).

Pentru a elimina acest neajuns a fost implementat un sistem de generare on-line a imaginii simbolurilor cu fenomene fonetice (Apopei et al., 2002). În acest fel, introducerea unui caracter specific transcrierilor fonetice este realizată prin selectarea sunetului de bază de la tastatură, urmată de aplicarea fenomenelor fonetice prin selectarea acestora din bara de instrumente.

3. Aplicația ALR

În continuare prezentăm câteva dintre facilitățile puse la dispoziție de aplicația ALR, folosită pentru pregătirea planşelor atlasului lingvistic.

3.1 Întreținerea dicționarelor asociate atlasului lingvistic electronic

Stocarea informațiilor specifice atlasului lingvistic este realizată prin folosirea a trei fişiere dicționar:

2. Dicționarul „Cuvinte de bază" conține fondul de cuvinte (titlurile hărților şi ale textelor-sinteză de tip material necartografiat) din atlasul lingvistic, modalitatea (directă, dar cel mai adesea indirectă) în care a fost formulată întrebarea, precizare urmată de textul întrebării, aşa cum a fost ea formulată în momentul anchetei, corespondențele cu alte atlase lingvistice româneşti sau romanice, note, observații, şi eventual imagini.
3. Dicționarul „Puncte de anchetă" conține informații despre localitățile anchetate: numărul de ordine şi numele localității, cu precizarea comunei şi a județului de care aparține, şi, acolo unde este cazul, corespondența cu numărul atribuit punctului respectiv în anchetele pentru *Atlasul lingvistic român* (ALR I, de S. Pop, I, 1938; II, 1942; ALR II, de E. Petrovici, I, 1940) sau pentru atlasul lingvistic al lui Gustav Weigand (*Linguistischer Atlas des dacorumänischen Sprachgebietes*, 1909).
4. Dicționarul de transcrieri fonetice conține transcrierea fonetică a răspunsului la întrebarea pusă în momentul anchetei pentru fiecare cuvânt din dicționarul „Cuvinte de bază", în fiecare dintre punctele de anchetă din rețea, transcrierii fonetice fiindu-i asociată, acolo unde este

posibil, şi înregistrarea audio corespunzătoare din baza de date sonore. Abundența şi varietatea materialului înregistrat în anchetă a făcut ca, de cele mai multe ori, răspunsurile propriu-zise să însoțite (completate, contextualizate) de o serie de informații şi comentarii (ale informatorului însuşi sau ale anchetatorului), care sunt introduse într-un câmp aparte (Nota II). În momentul proiectării pe hartă, aceste informații complementare nu vor apărea alături de punctul de anchetă, ci în secțiunea de jos a paginii, iar dacă materialul este redat sub formă de liste-sinteză, datele din Nota II se vor distribui automat după numărul de ordine al punctului de anchetă în care au fost înregistrate.

Transcrierile fonetice sunt stocate în structuri de date compacte, în funcție de:

5. caracterul corespunzător sunetului primar (codificare UNICODE);
6. atribute:
   - poziționare: normal, deasupra sau „la umăr”;
   - mod de subliniere: linie sau zigzag;
   - cursiv, aldin;
- fenomene:
   - tip sunet: vocală sau consoană;
   - fenomene specifice aplicate.

În Figura 2 este prezentată interfața folosită pentru editarea conținutului celor 3 dicționare.

3.2 Instrumente pentru generarea hărților sintetice

În vederea realizării de hărți sintetice, sistemul permite gruparea punctelor de anchetă după criteriul stabilit de utilizator: distribuția anumitor fonetisme sau a unor tipuri morfologice, ocurența unor termeni etc.

În fiecare dintre aceste situații, la selectarea comenzii corespunzătoare, sistemul afişează o fereastră de dialog precum aceea din Figura 3 (ilustrând gruparea în funcție de criteriul semantic), în care poate fi editată lista de sensuri, cu ocurențele corespunzătoare.

Această fereastră de dialog conține 4 zone de lucru care sunt descrise în continuare:

- **Numele de identificare a grupării pe sensuri** – sintagma descriptivă folosită pentru identificarea sensurilor şi care funcționează ca titlu al legendei, dacă se generează o hartă sintetică pe sensuri.
- **Lista de sensuri a cuvântului –** conține sensurile definite, cu un sumar al informațiilor asociate, la care se adaugă şi comenzi pentru adăugarea / ştergerea unui sens, pentru modificarea ordinii in lista de sensuri sau pentru modificarea informațiilor grafice asociate fiecărui sens.
- **Lista punctelor de anchetă asociate sensului selectat** – conține lista punctelor de anchetă în care s-a înregistrat sensul respectiv, precum şi comenzi pentru editarea comentariului sau pentru ştergerea de puncte din listă.
- **Lista de puncte de anchetă disponibile** – cuprinde lista punctelor de anchetă, cu transcrierile fonetice aferente, definite în dicționar. Fereastra mai afişează butoane de comandă pentru adăugarea punctului de anchetă la sensul selectat în lista de sensuri.

3.3 Generarea automată a planşelor cu hărți lingvistice şi material necartografiat

Planşele atlasului lingvistic creează o conexiune între informaţia aflată în dicţionare sub formă de transcrieri fonetice, informaţia grafică folosită pentru descrierea hărţilor şi toate opţiunile pe care utilizatorul le poate alege în momentul generării hărţii (Apopei et al., 2004).

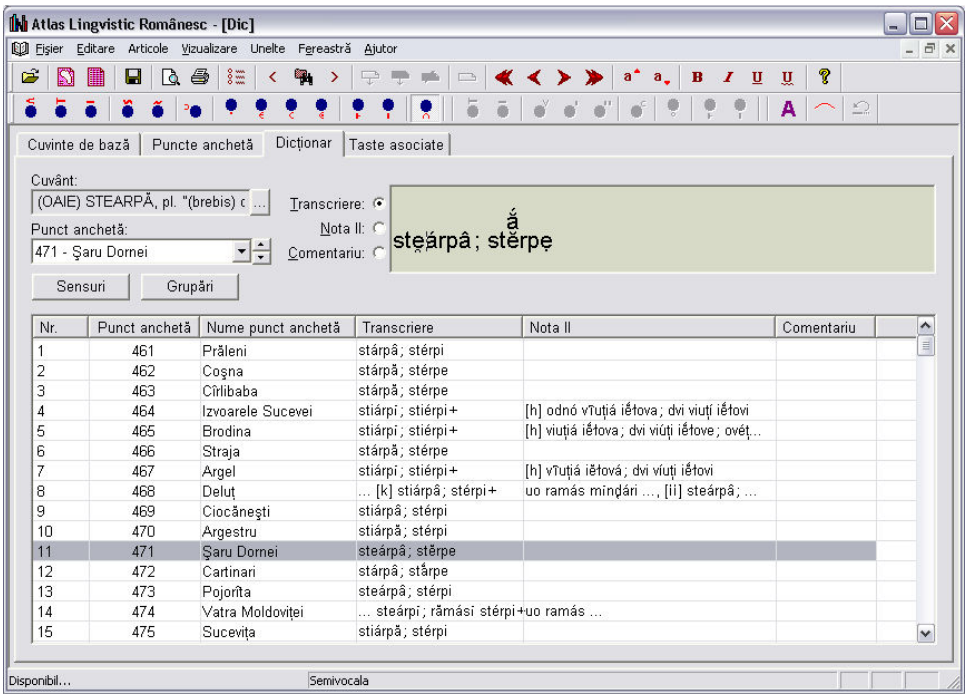


Figura 2: Fereastra de editare a dicţionarului de transcrieri fonetice

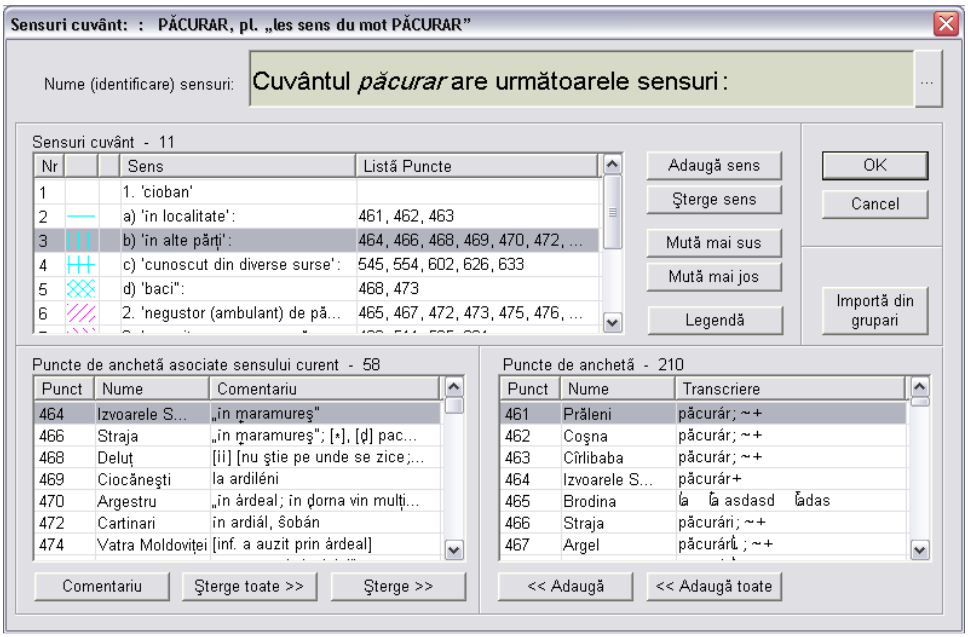


Figura 3: Fereastra de dialog folosită la editarea grupărilor după sensuri

În acest moment, sistemul este capabil să genereze **automat** trei tipuri de planşe pentru atlasul lingvistic:

- **hărți lingvistice** – transcrierile fonetice asociate unui anumit cuvânt-titlu sunt plasate pe harta regiunii respective. Planșa poate conține sau nu și o hartă sintetică;
- **planșe cuprinzând un text-sinteză de tip material necartografiat (*MN*)** – răspunsurile asociate unui cuvânt-titlu sunt organizate după criteriul frecvenței și după criteriul onomasiologic și sunt prezentate în format tabelar;
- **planșe combinate de tip material necartografiat (*MN combinat*)** – informațiile asociate mai multor cuvinte-titlu (dintr-o listă specificată în prealabil) sunt prezentate în format tabelar, pe mai multe pagini.

Toate celelalte tipuri particulare de planșe sunt realizate folosind modulul de editare. Facem observația că atlasul lingvistic conține pagini în format A3. Pentru planșele cu material necartografiat, tipărirea se face în mod natural, câte o planșă pe pagină, cu orientare de tip Portrait. În cazul planșelor care conțin hărți lingvistice, acestea sunt împărțite pe câte două pagini. Fiecare pagină conține câte o jumătate din harta lingvistică, cu orientare de tip Landscape. Sistemul realizat de noi permite tipărirea planșelor în acest mod, cu observația că cele două jumătăți ale hărții fonetice sunt considerate ca fiind planșe separate.

Planșele atlasului lingvistic sunt generate automat. Utilizatorul trebuie să selecteze:

- tipul de planșă: hartă lingvistică sau material necartografiat;
- dicționarul ce va fi folosit în procesul de generare;
- cuvântul-titlu vizat;
- modul de grupare folosit (numai pentru materialul necartografiat).

În cazul planșelor de tip hartă lingvistică (vezi Figura 4), utilizatorul poate selecta și prezentarea altor informații:

- includerea sensurilor cuvântului în Nota III;
- afișarea hărții sintetice explicative corespunzătoare sensurilor sau uneia dintre grupările care au fost definite în prealabil;
- afișarea planșei complete sau deschiderea opțională a jumătății superioare sau a celei inferioare.

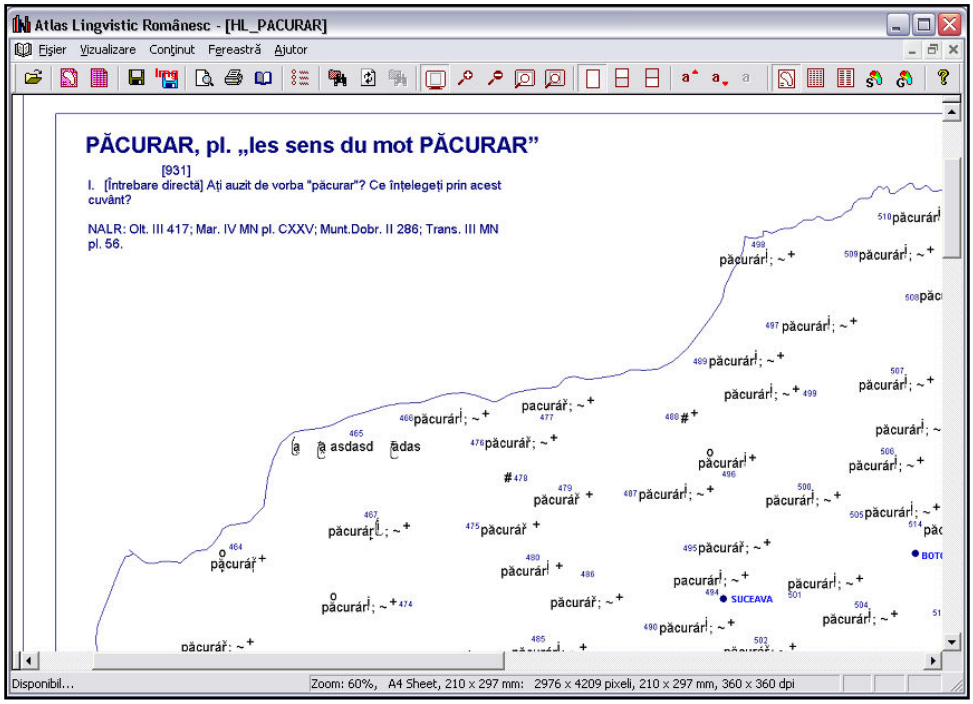


Figura 4: Hartă lingvistică

După ce au fost generate, planşele lingvistice pot fi tipărite, deoarece modul de desenare implicit rezolvă în proporţie mare toate situaţiile care apar în paginile atlasului lingvistic. Dacă se consideră necesar, se poate efectua rearanjarea obiectelor pe pagină sau, în cazul realizării de planşe ce conţin numai hărţi sintetice explicative, planşele generate pot fi salvate în fişiere, în vederea editării ulterioare. Sistemul pune la dispoziţie acest sistem de editare.

3.4 Generarea automată pentru indexul de cuvinte şi forme

Sistemul realizat permite identificarea tuturor ocurenţelor unui sunet sau grup de sunete, cu sau fără fenomene fonetice, într-un volum de date. Prima etapă a acestui proces, constând în stabilirea parametrilor de căutare, este realizată folosind interfaţa prezentată în Figura 5 (Bejinariu et al., 2006).

Parametrii funcţiei de căutare sunt următorii:

– Filtrul **Cuvânt** – permite restrângerea căutării la un anumit cuvânt de bază;
– Filtrul **Punct** (de anchetă) – permite restrângerea căutării la un singur punct de anchetă;
– **Căutare în** – permite stabilirea câmpurilor din dicţionar în care se face căutarea. Opţiunile posibile sunt „Transcriere” fonetică şi/sau „Nota II”;
– **Mod căutare** – este folosit pentru a specifica modul în care se realizează căutarea, posibilităţile disponibile fiind „Text” şi „Transcriere fonetică”;
– **Forma de căutat** – permite utilizatorului să editeze textul ale cărui apariţii doreşte să le identifice.

La selectarea comenzii „Caută!”, sistemul parcurge cuvântul (sau cuvintele) de bază selectat/-e şi identifică toate ocurenţele formei indicate. Este disponibilă o comandă de „Sincronizare”, care produce deschiderea în fereastra „Dicţionar” a cuvântului de bază selectat în index. Indexul generat automat poate fi tipărit în vederea consultării.

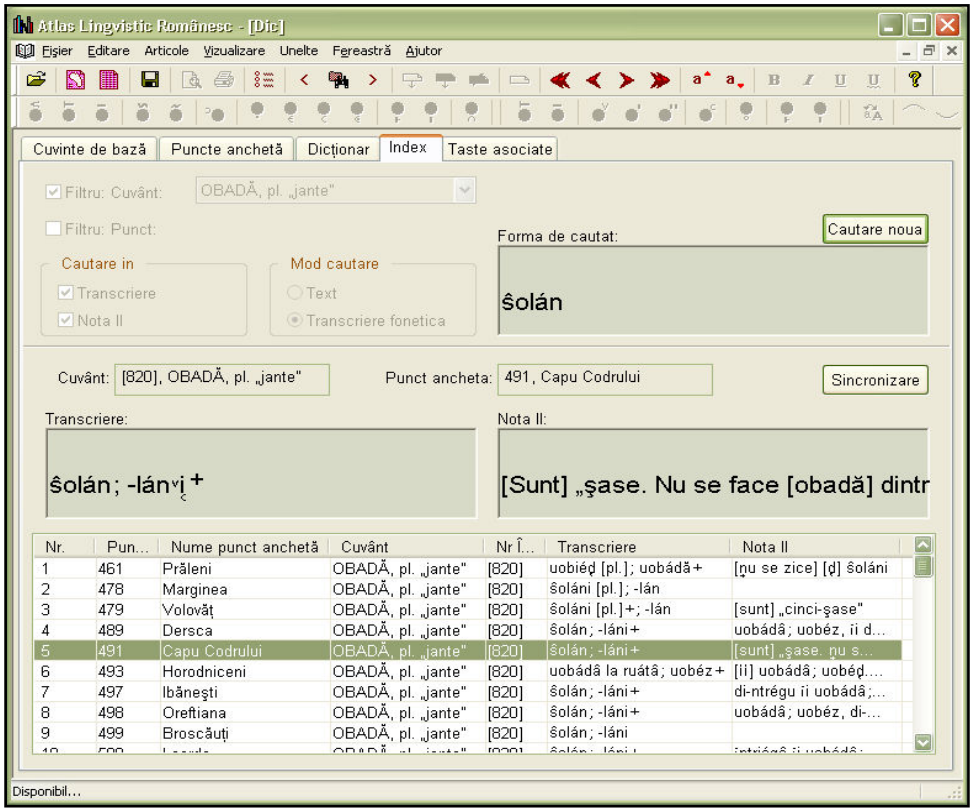


Figura 5: Indexul de forme, generat de aplicația ALR.

4. *Editorul de texte dialectale. Aplicația EditTD*

O componentă importantă care a fost realizată pentru tipărirea *Noului Atlas lingvistic român, pe regiuni. Moldova şi Bucovina* este editorul de texte dialectale.

După cum se poate vedea în Figura 6, interfața realizată este asemănătoare cu cea folosită în controlul pentru editarea dicționarelor, care a fost descrisă anterior.

În mod stand-alone, editorul pune la dispoziție un set restrâns de funcții (comparativ cu editoarele specializate), dar suficient de puternice pentru a răspunde cerințelor impuse de publicarea volumelor cu texte dialectale, şi anume:

- funcții specifice editării de texte în general:
  - paginare, stabilirea dimensiunii paginii, stabilirea marginilor;
  - numerotarea paginilor;
  - aliniere (stânga, dreapta şi justify);
  - inserarea de salturi forțate la pagină nouă;
  - adăugarea de note de subsol;
  - modificarea locală a dimensiunii fontului folosit;
  - tipărire, în întregime sau parțial.
- funcții specifice editării de texte dialectale:
  - aplicarea fenomenelor fonetice;
  - modificarea poziției caracterelor (la umăr, suprapuse),
  - numerotarea rândurilor textului cu un pas care poate fi stabilit de utilizator.

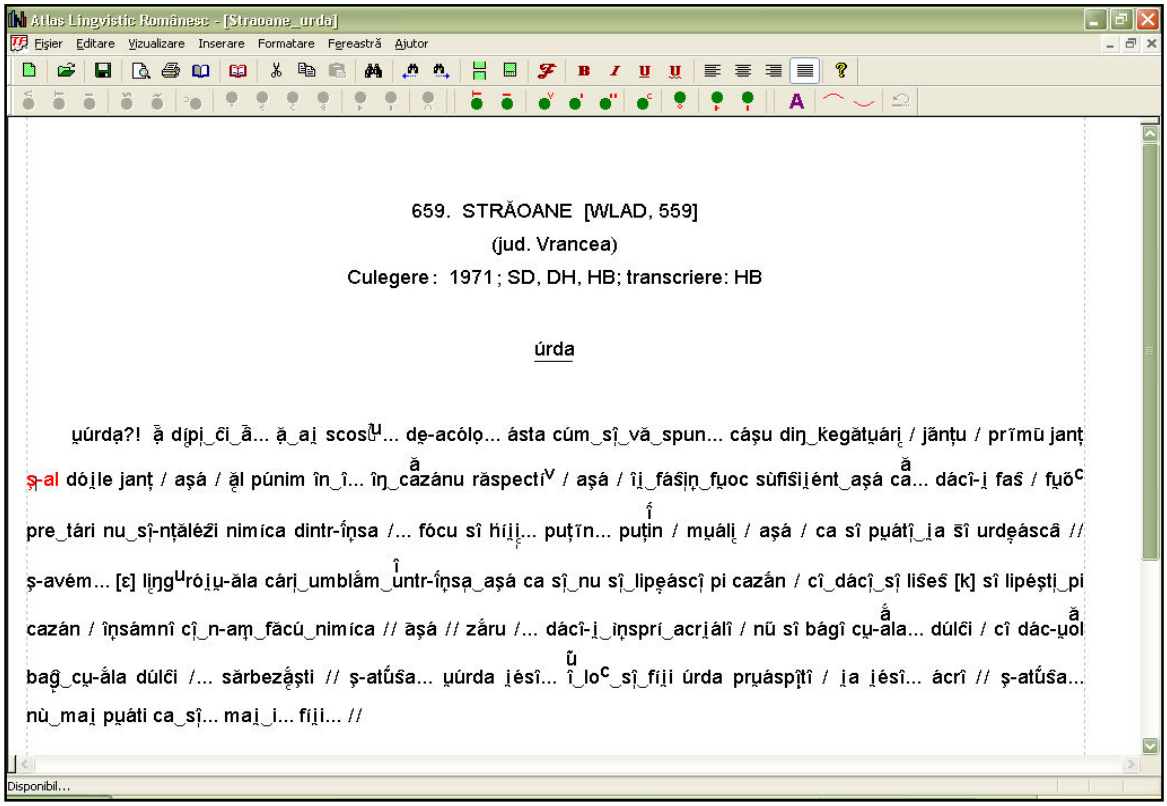


Figura 6: Editorul de texte dialectale

5. *Concluzii*

Aplicaţiile ALR si EditTD au permis realizarea planşelor Prospectului celui de-al III-lea volum al *Noului Atlas lingvistic român, pe regiuni. Moldova şi Bucovina*, apărut în anul 2005 sub formă de volum şi CD multimedia, precum şi a planşelor volumului complet al atlasului, aflat în prezent în faza finală de pregătire pentru tipar.

Au fost implementate şi alte instrumente care însă nu au fost suficient testate, motiv pentru care nu au fost detaliate în această prezentare:

- o bază de date geografice proiectată folosind conceptul de Sistem de Informaţii Geografice (Gâlea et al., 2006);
- o bază de date MySQL pentru stocarea informaţiilor lingvistice. Noul mod de reprezentare a unificat dicţionarele iniţiale ce conţineau doar informaţiile specifice unor grupuri de cuvinte de bază;
- a fost proiectată o interfaţă de acces la baza de date MySql, care asigură accesul concurent al mai multor utilizatori.

Referinţe bibliografice

Arvinte, V.; Dumistrăcel, St., Florea, I., Nuţă, I., Turculeţ, A. şi Botoşineanu, L., Hreapcă, D., Olariu, Fl. (2007). *Noul Atlas lingvistic român, pe regiuni. Moldova şi Bucovina*, vol. III, Iaşi, Editura Universităţii „Al. I. Cuza".

Apopei, V., Bejinariu, S., Roman, M. (2002) „Graphic Symbols Generator for the Phonetic Transcription in the Electronic Linguistic Atlas", ECIT 2002.

Apopei, V., Bejinariu, S., Bulancea, C. (2003) „Sistem pentru proiectarea planşelor Atlasului Lingvistic Românesc", Symposium on Intelligent Systems and Applications SIA2003, Iaşi, Romania, September 19-20.

Apopei, V., Bejinariu, S., Bulancea, C., Olariu, F. (2004) „Plates Preparation for Linguistic Atlases Publishing", European Conference of Intelligent Technologies, Iasi, Romania, July, 2004.

Bejinariu, S., Apopei, V., Luca, R., Olariu, F., Botoşineanu, L. (2006) „Electronic Linguistic Atlases. Tools for Information Analysis", Proceedings of the ECIT, 2006, September, 20-23, Iaşi, România, ISBN 978-973-730-246-5.

Bejinariu, S., Roman, M., Apopei, V., Olariu, Fl. (2000) „Sistem pentru editarea transcrierii fonetice în ALR", Zilele Academice Ieşene, Iaşi, 6 oct. 2000.

Gâlea, D., Bejinariu, S., Nită, C.D., Muscă, E., Lazăr, C., Luca, R. (2006) „Atlases Modeling using GIS", Proceedings of the ECIT 2006, September, 20-23, Iaşi, România, ISBN 978-973-730-246-5.

MICRO-CORPUS DE SUNETE GNATOSONICE ŞI GNATOFONICE

HORIA-NICOLAI TEODORESCU[1,2], MONICA FERARU[1]

[1] *Technical University of Iasi, Iasi, Romania*
[2] *Institute for Computer Science, Romanian Academy, Iaşi Branch– România*

{hteodor, mferaru}@etc.tuiasi.ro

Rezumat

În această lucrare prezentăm un mic corpus de înregistrări gnatofonice şi gnatosonice, cu comentarii şi discuţii privind utilitatea sa practică.

1. Introducere

Preocupările pentru realizarea de arhive clasice de voce vorbită, sub formă de înregistrări pe disc sau bandă magnetică, s-au materializat în ultimul secol prin numeroase arhive, în special dialectale, depozitate în instituţii de cercetare lingvistică naţională, în universităţi, sau în depozite ale unor foruri naţionale, precum Academia Română – vezi de ex. (Academia Română, Institutul de Lingvistică "Iorgu Iordan - Al. Rosetti" din Bucureşti). În ultimele decenii, grupuri de cercetare au realizat mici arhive de voce vorbită, în format electronic, cu scopuri particulare, precum realizarea de sintetizoare de voce, analiză de voce, voci patologice etc. Asemenea arhive, raportate sporadic şi mai mult local, au avut un impact, credem, minor asupra cercetării în domeniu, iar ele nu au fost, în general, accesibile altor grupuri de cercetare decât celui care a elaborat arhiva. Situaţia la nivel naţional este în mare măsură similară celei la nivel internaţional, cu diferenţa notabilă că unele firme mari, interesate de comunicaţiile vocale, precum firma Bell, au dezvoltat arhive ample de voce vorbită, dar proprietate a firmei şi puţin accesibile cercetătorilor externi firmei.

La nivel internaţional, preocuparea pentru arhive electronice ample de voce vorbită, constituite în adevărate corpusuri de limbă vorbită, au fost destul de intense după 1990 şi s-au materializat în corpusuri verbale analizate şi adnotate, uneori însoţite de instrumente specifice de adnotare. Exemple sunt numeroase şi pot fi, multe dintre acestea, găsite pe Internet.

2. Structura arhivei de sunete gantosonice şi gnatofonice

2.1 Voci normale şi voci afectate de patologii ale aparatului stomatognat

Atunci când se creează un corpus de voci specifice unei limbi se uită adesea că o limbă reprezintă *o populaţie* şi nu un obiect (proces) abstract independent. Pentru ca limba să fie reprezentată statistic relevant, în corpus trebuie să fie incluse voci care reprezintă statistic întreaga populaţie dintr-o ţară, sau dintr-o regiune. Din nefericire, acest criteriu elementar de statistică este rar luat în seamă, astfel încât multe corpusuri includ doar înregistrări de “voci alese”, adesea voci de actori, produse în condiţii cu totul artificiale. (Există excepţii notabile, precum unele corpusuri de cuvinte pronunţate telefonic, selectate aleator dintr-un mare număr de convorbiri.)

Pe baza criteriului statistic de reprezentativitate pentru o bază de date vocale reprezentând o limbă, un procent dintre voci este de persoane de vârste avansate, sau de persoane cu probleme la nivelul aparatului stomatognat. Unele dintre aceste probleme pot afecta semnificativ vorbirea, precum lipsa unor dinți, care afectează unele foneme dentale, probleme ale articulației temporo-mandibulare, care afectează dinamica mandibulei în timpul vorbirii etc.

Nu cunoaştem nici la nivelul României, nici pentru alte țări, o statistică privind incidența patologiilor aparatului stomatognat care afectează vorbirea. Unele statistici privind starea aparatului stomatognat, pentru diverse țări, sunt însă relevante indirect privind incidența unor influențe ale patologiilor stomatognatice asupra vorbirii. Astfel, (AIHW Statistics and Research Unit, 2001) prezintă informații suficient de complete şi utile pentru noi privind tratamentul stomatologic al populației din Australia. Pentru Australia, între 7,5% şi 27% din populație (pentru grupele de populație "favorizată" şi "defavorizată") suferă cel puțin o extracție dentară pe an, cu o medie de 13,7%. Ținând cont că o bună parte din populația "defavorizată" nu îşi permite imediat – şi nici măcar în cursul aceluiaşi an – un tratament recuperator şi/sau o protezare corespunzătoare, putem *estima* că între cca. 5% şi 20% din populație va prezenta probleme de alterare temporară sau definitivă a vocii sau modului de vorbire datorită disfuncționalității sistemului stomatognat. Un asemenea procent este, desigur, semnificativ şi nu poate fi neglijat, nici măcar la nivelul unei baze de date (corpus) de voci vorbite "naturale" (adică, statistic reprezentative populațional). Este opinia primului autor că nu pot fi excluse dintr-o limbă vorbită procese de vorbire pe motiv că nu sunt „standard"; la fel, nu pot fi excluse dintr-o bază de date reprezentativă pentru o limbă vorbită voci pe motiv că aparțin unor persoane a căror stare de sănătate nu este perfectă: asemenea persoane fac parte, totuşi, din populația respectivă.

O situație similară este prezentată în (London Health and Public Services Committee, 2007). Din acest raport, rezultă că în Londra, între 27% şi până la 45% din populație, funcție de grupul social, fie din motive de cost, fie datorită dificultății de a găsi un dentist cu acoperire prin asigurare medicală, fie din alte motive, întârzie efectuarea tratamentului stomatologic – şi ca urmare acest procentaj poate fi temporar afectat de disfuncționalități în vorbire.

Pe baza considerațiilor de mai sus, este perfect justificată statistic introducerea într-un corpus de limbă vorbită a unor înregistrări cu voci ale unor persoane ce prezintă probleme de disfuncționalitate – inclusiv accentuată – la nivelul aparatului stomatognat. O asemenea secțiune a corpusului este, fără îndoială, utilă şi medical.

2.2 Problematica gnatofoniei şi gnatosoniei

Gnatosonia a fost introdusă în anii 1970 de către Watt (Watt, 1967), (Watt şi Wakabayashi, 1978), (Watt şi McPhee, 1985) [ultimele două ne-au fost disponibile doar sub formă de rezumat], ca metodă de analiză a disfuncționalității ocluziei dentare şi a fost parțial acceptată ca metodă de diagnostic preclinic în medicina dentară. Fără ca sunetele ocluzale să afecteze direct vocea (aceste sunete sunt produse prin mişcări de tip masticator), ele pun în evidență disfuncții care pot afecta şi pronunția – de exemplu, disfuncții articulare, la nivelul articulațiilor temporo-mandibulare. De aici, interesul

nostru pentru includerea şi a unui mic corpus de date gnatosonice pe lângă un corpus exemplificator de date gnatofonice, pe situl Internet *Sunetele limbii române*.

Gnatofonia este o metodă de analiză a deficienţelor de pronunţie introdusă de primul autor, parţial la sugestia Prof. Leonid Teodorescu.

2.3 Metodologie

Cuvintele utilizate pentru înregistrările gnatofonice sunt alese astfel încat să se poată analiza comparativ modificările de siflante şi de consoane semi-vocalice. De exemplu, compararea pronunţiei siflantei *f* cu a consoanei (semi-vocalei) *v*, ca şi analiza fiecăreia dintre acestea, permit determinarea unor imperfecţiuni ale dentiţiei, sau, după caz, danturii (protezei dentare).

Metodologia de culegere a semnalelor gnatofonice este identică cu cea de culegere de semnal vocal. S-au utilizat protocoalele prezentate pe situl indicat. Culegerea de semnale gnatosonice s-a realizat cu acelaşi sistem, cu precizarea că microfonul a fost menţinut direct în faţa gurii, în planul sagital, iar subiecţilor li s-a indicat să menţină buzele întredeschise. S-au utilizat protocoalele prezentate la adresa: http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/ro/protocol_nou.htm.

Cuvintele utilizate pentru înregistrările gnatofonice sunt: vată/ fată; var/ far; vuiet (pronunţat vvvvvuiet)/ vuiet (pronunţat normal, scurt, vuiet)/ fui/ vaiet (pronunţat vvvvvaiet)/ vaiet (pronunţat vaiet)/ faieton/ vecin/ fecior/ vânt (pronunţat vvvvvânt)/ vânt (pronunţat vânt)/ fân/ vvvvvvine, vvvine, vvvine/ vine/ fine/ vehement/ ferment/ vierme/ fierbe/ vâjâit/ vvvvvvvâjjjjjâit/ vvvvâjjjjjâie/ ffffâşşşşşşâie/ ffffâşşşşşâit/ fâşâit/ sâsâit/ sssssâsssssâie/ gâjâit/ zâzâie/ bââzzzzâââie/ bâzâie.

Gruparea indicată de cuvinte corespunde grupurilor cu diferenţe de pronunţie la nivelul unor consoane afectate semnificativ de patologia aparatului gnatic (*f*, *v*, *s*, *ş*, ...), după experienţa primul autor. În figurile 1 şi 2 sunt prezentate exemple de înregistrări (brute, neprelucrate) cu sunete gnatofonice şi cu sunete gnatosonice.

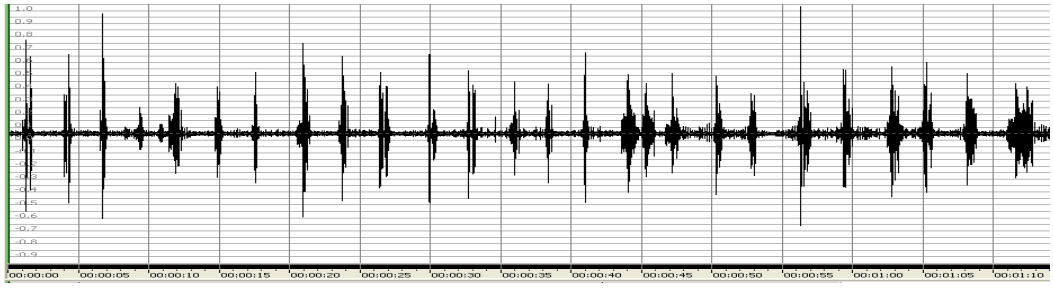


Figura 1: Exemplu de înregistrare gnatofonică

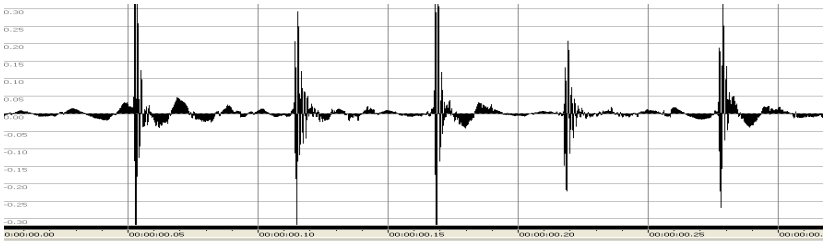
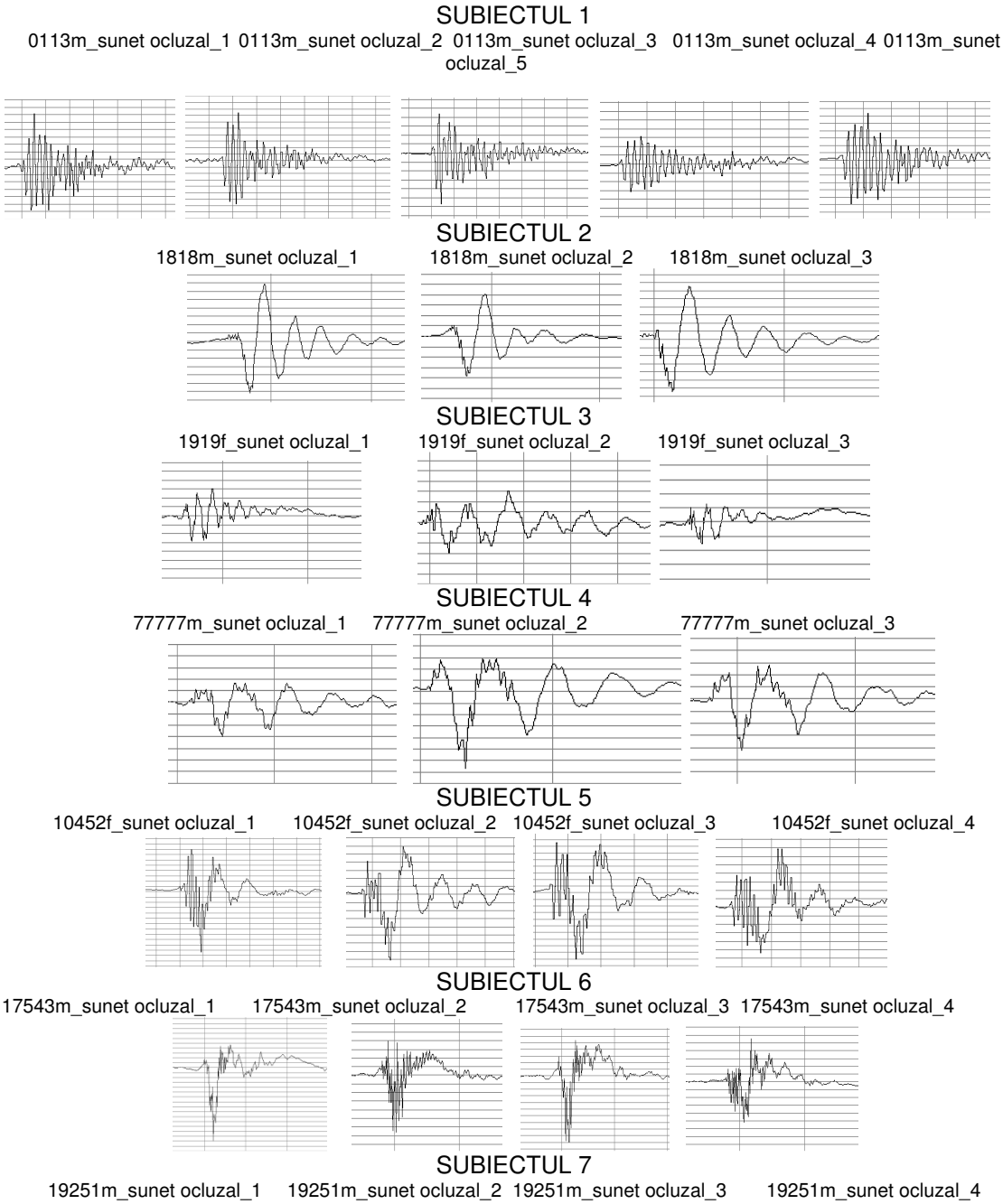


Figura 2: Exemplu de înregistrare gnatosonică cu detalii (Efect de saturație la prima şi la a treia înregistrare)

Elemente privitoare la tehnica de înregistrare folosită pot fi găsite în (Teodorescu *et al*., 2005-2007), (Teodorescu 2007 a), (Teodorescu, Feraru, 2007).

2.4 Exemple de înregistrări gnatosonice

În acest paragraf, cu titlu de exemplificare, sunt prezentate (în imaginile din figura 3) unele fragmente semnificative de înregistrări de sunete gnatosonice aflate în arhiva menționată, la adresa: http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/ro/sunete_gnatosonice.htm.



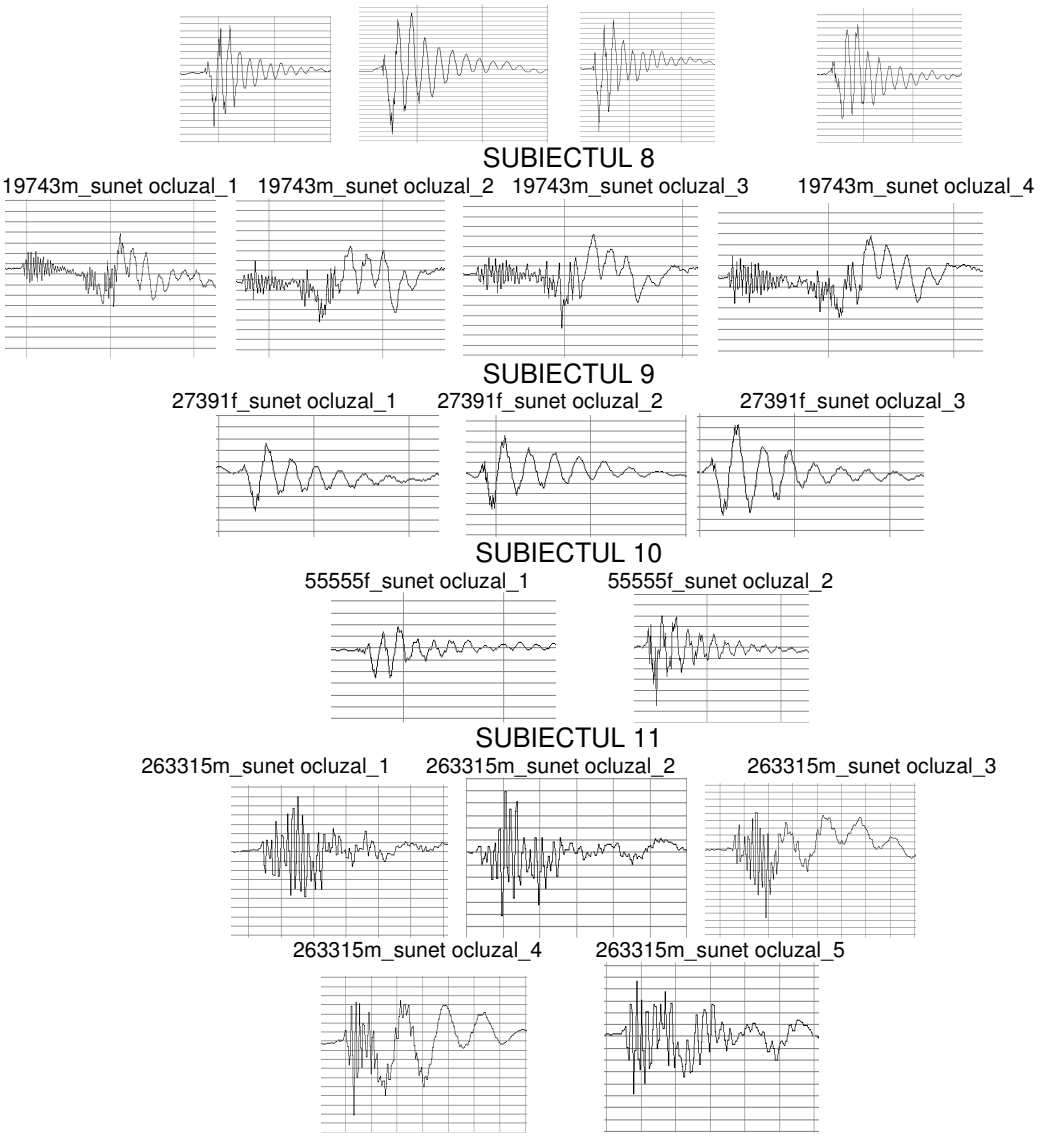


Figura 3: Exemple de semnale gnatosonice din arhivă (semnale brute, înainte de eliminarea perturbației de 50 Hz – brum de rețea)

O analiză sumară la imaginile care corespund subiectului nr. 1 relevă că toate cele 5 înregistrări corespund unui acelaşi tip de ocluzie, cu o ocluzie inițială principală, relativ fermă, cu tendința de a apare un al doilea contact ocluzal şi o alunecare.

La subiectul nr. 2, se constată un singur contact ocluzal, scurt (axa timpului mult expandată). Acelaşi lucru se poate afirma şi despre subiecții nr. 3 şi 4. La subiectul 5, durata contactului este mai mare, ceea ce arată o alunecare la ocluzie (ocluzie imperfectă).

La subiectul 8, apare evident un dublu sunet ocluzal, cele două sunete fiind net separate în timp; aceasta denotă două puncte de contact, deci ocluzie deficitară, care poate duce în timp la deficiențe de mişcare mandibulară (supunere la tensiuni asimetrice în articulația temporomandibulara, erodare a dinților etc.)

La ultimul subiect se observă o ocluzie prelungită, uneori cu apariţia unui al doilea sunet ocluzal. Analiza şi interpretarea medicală de detaliu nu fac obiectul acestei lucrări şi vor fi expuse în altă parte.

Figura 4: Imaginea ecranului la intrarea în arhivă

Toate înregistrările se afla pe situl indicat, în secţiunea „Arhiva pentru aplicaţii de gnatosonie şi gnatofonie”, pe pagina ilustrată în figura 4. Situl pe care se află arhiva a fost descris în (Teodorescu *et al*., 2005-2007), (Teodorescu *et al*., 2007), (Teodorescu, Feraru, 2007).

3. Rezultate preliminare

3.1 Reproductibilitate, consistenţă, relevanţă

Reproductibilitatea, consistenţa şi relevanţa unui test sunt esenţiale în fundamentarea oricărei metode de studiu. Deoarece domeniul gnatosoniei este relativ nou şi insuficient investigat, iar cel al gnatofoniei este nou, este necesar să determinăm gradul de reproductibilitate, de consistenţă şi de relevanţă a analizelor propuse in domeniile respective. Reamintim următoarele definiţii relativ larg acceptate. Un test este repetabil dacă prin repetarea lui în condiţii identice se obţin rezultate (cel puţin statistic) identice. Similar, un test este reproductibil dacă, folosind o procedură standard şi echipamente uzuale, rezultatele obţinute în două locaţii, de către două echipe diferite (laboratoare diferite), la momente de timp oarecare, se obţin rezultate (statistic) identice. Precizăm că repetabilitatea poate fi bună pentru un test dat, dar reproductibilitatea redusă. Standardizarea are rolul principal de a creşte reproductibilitatea. În prezent, putem vorbi de o bună repetabilitate a unora dintre rezultatele noastre, dar încă nu putem discuta

bine reproductibilitatea decât pentru gnatosonie. Relevanța este proprietatea unui test de a da informație, de a valida, sau de a prezice un fapt de interes. Relevanța este deci determinată în raport cu un obiectiv, un fapt de interes. În cele ce urmează, relevanța va fi determinată prin capacitatea analizelor gnatofonice de a prezice o anumită patologie. Precizăm că dacă, la rândul ei, patologia este relevantă în prezicerea unei anume clase de modificări (alterări) fonatorii, testul va permite clasificarea vorbirii unei persoane într-o anume clasă de modalități de vorbire, deci selecția modelului de limbă vorbită pentru vorbitorul respectiv.

În primul rând, înregistrările prezentate mai sus arată că sunetele produse de un acelaşi subiect sunt suficient de asemănătoare între ele, deci pot fundamenta un mijloc de diagnoză şi de identificare a diverselor patologii (satisfac criteriul de consistență). În al doilea rând, se constată că sunetele sunt specifice subiectului, dar în primul rând patologiei: există diferențe nete între sunetele produse cu deficiențe diferite; se satisface deci criteriul specificității. De asemenea, înregistrările indică şi unul dintre factorii principali care afectează negativ înregistrările – perturbațiile de frecvența rețelei de alimentare, care produc artefacte vizibile. Aceste artefacte trebuie eliminate înainte de utilizarea semnalelor de către medic sau de sistemul de analiză automată.

Din experiența primului autor şi conform literaturii, repetabilitatea şi reproductibilitate sunt suficient de bune în gnatosonie.

Repetabilitatea este relativ dificilă în gnatofonie, deoarece trebuie ținut cont de pronunție, de emoții (Teodorescu, Feraru, Tandabăț, 2006), (Teodorescu, Feraru, 2007), de alte posibile patologii care apar temporar etc. Pentru a asigura repetabilitatea testelor gnatofonice, aceste variabile trebuie determinate în cadrul fiecărui test, iar dacă nu sunt îndeplinite condițiile „normale" testul trebuie reluat. Cel puțin unul dintre teste s-a dovedit puțin vulnerabil la condițiile amintite mai sus (emoții etc.), anume analiza siflantelor. Generarea siflantelor este in mare măsură un proces fizic primar, ne-influențat de emoție, de starea de sănătate generală, sau de un mod particular de pronunție (de melodia frazală). Ca urmare, analiza în frecvență a siflantelor credem că este un test robust. Deoarece primul autor a efectuat teste pe siflante în trei locații diferite cu rezultate practic similare, a putut trage concluzia că cel puțin echipamentul şi locația nu joacă un rol în acest test, care se dovedeşte astfel reproductibil, robust. Deci, se poate concluziona că, în gnatofonie, siflantele sunt dintre cele mai "stabile" procese la variații circumstanțiale, fiind în acelaşi timp sensibile la unele forme de deficiențe ale dentiției.

Privitor la modul de analiză, primul autor consideră că parametrii temporari sunt cei esențiali în gnatosonie; cei frecvențiali sunt mai puțin, sau neesențiali. În schimb, parametrii frecvențiali sunt importanți în gnatofonie, iar cei temporari mai puțin importanți.

3.2 Metode de analiză automată

În scopul utilizării metodelor gnatosonice şi gnatofonice într-un sistem automat de (pre)diagnoză, se parcurg următoarele etape: completarea fişei de pacient şi introducerea datelor; culegerea în condiții standard a semnalelor; preprocesarea semnalelor (filtrare – eliminare de artefacte); extragerea de caracteristici; clasificare şi recunoaştere (de forme/ *pattern*-uri); clasificare şi prediagnostic; procesări statistice de

tip *data-mining*. Sistemul este conceput în întregime de primul autor şi se află în faza de realizare.

Privitor la metodele de prelucrare automată, cu extragere de caracteristici semnificative pentru diagnostic, a se vedea de exemplu (Teodorescu, 2006), (Teodorescu, Burlui, Leca, 1986). Procesarea presupune în primul rând o filtrare preliminară, în special pentru eliminarea zgomotelor de 50 Hz (brum de reţea) şi a frecevenţelor înalte (peste 3 kHz, care nu aparţin semnalelor gnatosonice, sau peste 10 kHz, nesemnificative în analiza gnatofonică, în stadiul actual). În cazul semnalelor gnatosonice, procesarea propriu-zisă implică detectarea numărului de „vârfuri principale", deci de contacte ocluzale, a duratei globale a sunetului ocluzal, a palierelor dintre două contacte ocluzale – dacă acestea există etc. (Teodorescu, 2006 a, b, c), (Teodorescu, Burlui, Leca, 1986). În cazul gnatofoniei, se impune compararea spectrelor siflantelor cu spectre „normale"; o analiză mai fină presupune detectarea unor caracteristici ale proceselor neliniare de curgere a aerului la producerea siflantelor. De asemenea, pentru fonemul *v* este necesară determinarea caracterului acestuia – consonantic sau semi-vocalic – şi a spectrului, pentru determinarea „alunecării" lui *v* către o siflantă, precum *f*.

4. Discuţii şi concluzii

Analiza gnatosonică şi cea gnatofonică sunt relevante atât în medicină, cât şi în recunoaşterea vorbirii şi în sistemele de răspuns telefonic automat. Privind ultimele, detectarea unei anume patologii (clase de voci) poate permite selectarea unui anume model de limbă vorbită în recunoaşterea vorbirii, sau poate permite dirijarea unui apel cu voce afectată de patologii către un operator uman, mai capabil să înţeleagă apelul.

În această lucrare am prezentat în primul rând motivaţia statistică a includerii în corpusurile de vorbire "naturală", caracteristică unei largi populaţii, a unor înregistrări de voci produse în situaţia unor disfuncţionalităţi ale aparatului stomatognat. De asemenea, am argumentat şi prin utilitatea medicală astfel de mici corpusuri de înregistrări gnatofonice şi gnatosonice. Apoi, am prezentat câteva exemple de înregistrări gnatosonice din cadrul sitului *Sunetele Limbii Române*, urmând ca în alte lucrări să prezentăm şi înregistrări gnatofonice. Înregistrările au fost comentate sumar.

Ca obiectiv pentru viitorul imediat, ne propunem realizarea unei arhive *publice* de câte cinci înregistrări pentru până la 10 patologii tipice, un număr mai mare de înregistrări urmând să fie disponibile la cerere. Precizăm că, în prezent, numărul de înregistrări de care dispunem este mult mai mare decât cel accesibil liber pe situl menţionat: majoritatea înregistrărilor sunt protejate din motive etice şi de păstrare a confidenţialităţii datelor personale ale pacienţilor.

O problemă care rămâne de studiat este modul cum funcţionarea deficitară a articulaţiei temporomandibulare influenţează vorbirea. Efecte posibile ale deficienţelor funcţionale articulatorii sunt deschidere mai mică şi mai lentă a gurei. Incidenţa disfuncţiilor articulaţiei temporomandibulare este, din câte ştim, slab cunoscută la noi în ţară, iar studii nu s-au făcut pe tema rolului acestei articulaţii în modificările de vorbire, cel puţin nu pentru limba română.

Mulţumiri. Autorii mulţumesc celorlalţi co-autori ai sitului ***Sunetele Limbii Române*** pentru cadrul favorabil creat pentru includerea pe acest sit a arhivelor menţionate în lucrare. Parţial, cercetarea la tema prezentată în lucrare a fost sprijinită de un contract CEEX, Ministerul Educaţiei şi Cercetării (Program VIASAN - Proiect: Sistem automat de diagnostic paraclinic în sindromul disfuncţional al sistemului stomatognat).

Referinţe bibliografice

Academia Română, Institutul de Lingvistică "Iorgu Iordan - Al. Rosetti" din Bucureşti. Arhiva fonogramică a limbii române, Corpus de română vorbită (CORV), http://www.dianaghido.com/corv/controller.php?page=alfr.php)

(2001), Oral health and Access to Dental Care – the gap between the 'deprived' and the 'privileged' in Australia. Research report, March 2001. AIHW catalogue No. DEN 67, ISSN 1323-8744 www.arcpoh.adelaide.edu.au/publications/report/research/pdf_files/rr15_deprived.pdf.

(2007). Health and public services committee, teething problems. A review of NHS dental care in London, November 2007, www.london.gov.uk/assembly/reports/health/dentistry.pdf.

Distributed Access Management for Language Resources. http://www.dam-lr.eu/. (Accesată 20 oct. 2007).

Teodorescu, H.N., Burlui, V., Leca, P.D. (1986). Gnathosonic analyser. Med Biol Eng Comput. 1988 Jul; 26(4):428-31.

Teodorescu, H.N., Feraru, M., Trandabăţ, D., Zbancioc, M., Luca, R., Verbuţă, A., Hnatiuc, M., Ganea, R., Voroneanu, O., Pistol, L., Şcheianu, D. (2005-2007). Situl Web Sunetele Limbii Române http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/index.htm.

Teodorescu, H.N. (2006). Occlusal Sound Analysis Revisited, Proc., 3rd International Conference on Advances in Medical, Signal and Information Processing (MEDSIP 2006), 17-20 Iulie 2006, The Institution of Engineering and Technology, Glasgow, UK

Teodorescu, H.N., Gnatophonetics. (2006). A New Discipline Analyzing Relations between Speech and the Stomato-Gnathic System. Zilele Academice Ieşene, Simp Inventica. Simpozionul naţional "Bazele performanţei şi inventică" organizat în cadrul "Zilelor Academice Ieşene" ISBN 973-730-244-3, 978-973-730-244-1, 9 Septembrie 2006.

Teodorescu, H.N. (2006). Gnatofonia şi Gnatosonia, Ed. Performantica, 2007, Iaşi, România.

Teodorescu, H.N., Feraru M., Trandabăţ, D. (2006). Nonlinear Assessment of Professional Voice 'Pleasantness', Conference BIOSIGNAL 2006, ISBN 80-214-3152-0, Brno, 28-30 Iunie 2006, 63-66.

Teodorescu, H.N., Feraru, M. (2007). A sudy on Speech with Manifest Emotions 10th International Conference on Text, Speech and Dialogue, TSD 2007, Pilsen, Czech Republic, 3-7 Septembrie, 2007, Lecture Notes in Computer Science, Springer Verlag, vol. 4629/2007, pp. 254-262, ISBN 978-3-540-74627-0.

Teodorescu, H.N., Trandabăţ, D., Feraru, M., Zbancioc, M., Luca, R. (2007). A corpus of the sounds in the Romanian spoken language for language-related education. Chapter Six, pp. 73-90. În volumul Carlos Periñán Pascual (Editor), "Revisiting Language Learning Resources", Cambridge Scholars Publishing (CSP), UK, ISBN 1-84718-156-2; ISBN 13: 9781847181565, 2007.

Watt, D.M. (1967). A gnathosonic study of tooth impact. Dent. Pract. Dent. Rec. 1967 May; 17(9): 317-24.

Watt, D.M., Wakabayashi, Y. (1978). Study of a classification of occlusion. J Oral Rehabil. 1978 Apr; 5(2):101-10.

Watt, D.M., McPhee, PM. (1985). Gnathosonic monitoring of occlusion of complete and partial dentures. J. Oral Rehabil. 1985 Mar; 12(2): 107-12.

CORPUS DE VOCE PENTRU LIMBA ROMÂNĂ ADNOTAT CU ETICHETE FUNCȚIONALE LA NIVELUL UNITĂȚILOR DE ACCENTUARE

DOINA JITCĂ, VASILE APOPEI

Institutul de Informatică Teoretică

Academia Română - Filiala Iaşi

vapopei@ iit.tuiasi.ro

Rezumat

În lucrare se propune o clasificare a unităților de accentuare după patternurile de contur F0 aşa cum au rezultat din analiza rostirii unui fragment din romanul „1984". Pe baza categoriilor de patternuri obținute s-a definit un set corespunzător de etichete cu ajutorul căruia se poate face o descriere a intonației. Etichetele astfel definite sunt introduse în structura intonațională a rostirilor ca valori ale unui atribut funcțional la nivelul tag-ului AU corespunzător unităților de accentuare. Avantajul acestei descrieri a intonației constă în evitarea detaliilor fonetice şi fonologice implicate de descrierile prin secvențe de tonuri (ex. ToBI), pentru specialiştii lingvişti, interesați în asocierea structurilor sintactico-semantice şi de discurs cu descrierile intonaționale.

1. Introducere

Pentru implementarea intonației în sinteza vocală s-au delimitat în lucrarea (Apopei V, Jitcă D (2007)) două module principale: cel de predicție al unei structuri intonaționale corespunzătoare textului de intrare şi cel de generare a conturului frecvenței F0 pornind de la ieşirea din primul modul. În unele aplicații, legătura între cele două module este concretizată într-un fişier XML cu tag-uri prozodice. În varianta de implementare pentru limba română prezentată în lucrarea - Apopei V, Jitcă D (2007)- fişierul XML este generat manual iar schema de adnotare intonațională a textului este cea propusă în lucrarea (Apopei V, Jitcă D (2006)). Concluziile rezultate în urma acestei implementări, cât şi din încercările de asociere automată a structurilor sintactice cu cele intonaționale (Curteanu N. ş.a (2007)), au pus în evidență necesitatea abordării unei descrieri intonaționale care să faciliteze atât aceste asocieri automate cât şi cele ale structurilor intonaționale cu segmentele elementare de contur F0.

În lucrarea prezentă propunem o descriere a conturului intonațional pe baza unor forme elementare de contur F0, la nivelul unităților de accentuare. Această modalitate de descriere a contururilor melodice presupune identificarea unor categorii de forme şi, în mod corespunzător, a unui set de forme prototip. Descrierea intonației pe baza acestora face necesară asocierea lor cu un set de etichete care să permită adnotarea unităților de accentuare din componența conturului melodic. Împărțirea în categorii s-a bazat pe stabilirea unei relații între conturul F0 al unităților de accentuare şi funcția acestora în rostirea textului (în formarea discursului).

Necesitatea codificării patternurilor unităților de accentuare, în vederea folosirii lor în sinteza vocală, apare şi în prezentarea altor implementări a intonației în sistemele text-

voce ((Heggtveit P. O., Natvig J. E. (2001))). În cadrul abordării noastre am crescut numărul de categorii pentru o descriere mai nuanțată a conturului F0.

Conform ierarhiei intonaționale din figura 1, în cazul general, o rostire constă din mai multe fraze intonaționale / intermediare (IP/ip) care la rândul lor sunt formate din secvențe de unități de accentuare (***A**ccentul **U**nit* - AU) şi de grupuri de unități de accentuare (***A**ccentul **U**nit **G**rup* - AUG) aflate pe acelaşi nivel ierarhic. Unitățile de accentuare în cadrul AUG-urilor au funcții la nivelul grupurilor iar grupurile au funcții la nivelul frazelor.

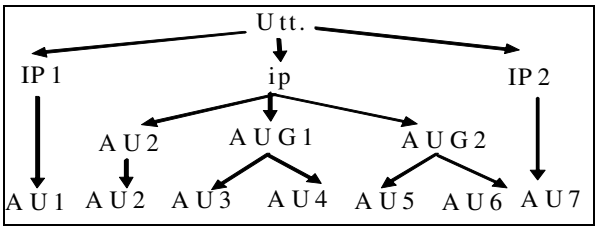


Figura 1: Ierarhia intonațională

La baza definirii categoriilor pentru unitățile de accentuare a stat ideea conform căreia conturul F0 este concretizarea corelației dintre evenimentele rostirii discursului şi evenimentele acustice.

Pornind de la funcțiile unităților de accentuare în cadrul rostirii discursului am definit un set de etichete cu mnemonice sugestive. În cadrul structurii XML aceste etichete devin un atribut funcțional pentru tag-ul <AU> folosit pentru marcarea unităților de accentuare. Variabilitatea pattern-urilor de contur F0 asociate unui anumit tip funcțional de unitate de accentuare poate fi controlată prin folosirea de atribute suplimentare, cum ar fi cel de tip *pitch accent* (în sensul sistemului de notare ToBI), de *nivel tonal* în scara frecvenței tonale, etc., atribute a căror valoare implicită stabilită pentru fiecare categorie poate fi modificată.

Considerăm că perspectiva creată de descrierea intonațională prin acest set de etichete la nivelul unităților de accentuare creează premise de a realiza mai uşor asocierile automate ale structurilor sintactice cu cele intonaționale, cât şi asocierile structurilor intonaționale cu segmentele elementare de contur F0.

2. Prezentarea setului de etichete ale unităților de accentuare

Analiza sistematică a intonației, pe corpusul de voce rezultat din rostirea unui fragment din romanul „1984” a autorului G. Orwel, a condus la identificarea următoarelor categorii de pattern-uri de contur F0 pentru unitățile de accentuare şi la definirea, în mod corespunzător, a unor etichete care pot fi grupate după cum urmează:

- etichete pentru unități de accentuare aflate la începutul sau sfârşitul frazelor intonaționale de tip IP/ip, cu şi fără rol de focalizare;
- etichete pentru unități de accentuare care realizează evidențierea unor paliere tonale implicate în realizarea focalizărilor;

- etichete pentru unități de accentuare aflate în cadrul grupurilor de accentuare (AUG);
- etichete pentru unități de accentuare care se desfăşoară pe linia de interpolare dintre două paliere tonale.

Etichetele din fiecare categorie sunt prezentate în secțiunile următoare. Pe baza lor se poate face o descriere a intonației prin secvențe de etichete AU separate prin "/" şi grupate prin paranteze rotunde în cadrul AUG şi prin paranteze pătrate în unități IP/ip.

2.1 Etichete pentru unitățile de accentuare de la începutul sau sfârşitul frazelor intonaționale

Începutul rostirii unui unități de discurs este efectuat în mod uzual printr-o unitate de accentuare care prezintă o variație semnificativă pe durata silabei accentuate în urma căreia se ating ținte tonale la nivelul cel mai ridicat al unui IP. Am numit aceste unități ca fiind de tip "PUSH" şi am etichetat pattern-ul corespunzător cu eticheta "PH".

În mod asemănător un *IP/ip* trebuie să conțină o unitate de accentuare de tip "POP" care să exprime sfârşitul unității de discurs iar pentru aceasta am folosit etichetele "PO%" şi "PO" pentru cazul *IP*, şi respectiv în cazul *ip* cu accent de frază de tip "low".

Un alt tip de pattern, pentru unitățile de accentuare, care apare la sfârşitul rostirii unei unități de discurs este cel corespunzător marcării atât a sfârşitului unității curente cât şi a începutului celei următoare. Am numit aceste unități de accentuare ca fiind de tip "**P**OP-**U**P" şi le-am etichetat cu mnemonica "PU%" în cazul unui *IP* şi respectiv "PU" în cazul unui *ip* cu accent de frază de tip *High*.

O altă categorie de patternuri de contur care realizează creşteri până la nivelul maxim al frecvenței F0 (Top level) pe silaba accentuată, urmate de coborâri pe silabele ne-accentuate următoare până la nivele tonale scăzute, este cea corespunzătoare evenimentelor **P**USH-**D**OWN pe care l-am etichetat cu "PD".

Pentru patternurile de conturul F0, care se ridică la nivelul tonal maxim pe silaba neaccentuată inițială şi care apoi coboară până înaintea silabei accentuate unde realizează un eveniment de tip focus ("f" sau „F"), am introdus etichetele "PD+f" şi "PD+F".

Există situații când o unitate de accentuare poate avea atât funcție de focalizare cât si una din funcțiile PH, PO%, PU% . În acest caz descrierea lor se face cu etichete derivate de tipul PH+F, PO%+F, PU%+F, PO+F, PU+F.

2.2 Etichete pentru unități de accentuare care realizează focalizările

Într-o frază intonațională IP/ip, între o unitate de tip "PUSH" şi cea de tip "POP", se pot afla unități de accentuare ale căror contururi F0 se desfăşoară în jurul unor paliere tonale. În figura 2, două tipuri de astfel de contururi elementare sunt prezente, corespunzătoare cuvintelor *lucruri* şi *discutat.*
Primul pattern ilustrat de cuvântul *lucruri* este caracterizat de mici variații în jurul unui palier tonal, după atingerea nivelului tonal respectiv. Acest pattern se etichetează cu

eticheta "f", cu precizarea că acest pattern poate apare în cadrul unităților de accentuare care participă la formarea focusului (accentului semantic).

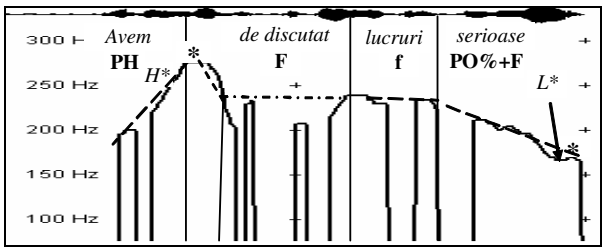


Figura 2: Conturul natural şi cel stilizat al rostirii „Avem de discutat lucruri serioase"

Al doilea pattern, ilustrat de cuvântul *discutat,* caracterizează un focus puternic realizat în cadrul unei singure unități de accentuare generate de un accent de pitch de tip H*. Unitatea de accentuare a ultimului cuvânt *serioase* încheie unitatea de discurs prin realizarea unui accent de pitch L* mai proeminent decât în mod obişnuit, contribuind la crearea focusului semantic pe grupul nominal "*lucruri serioase*". De aceea a fost etichetat cu eticheta derivată "PO%+F".

Cu ajutorul etichetelor definite conturul melodic natural din figura 2 poate fi descris cu următoarea secvență de etichete.

PH / F / f / PO%+F

2.3 Etichete pentru unități de accentuare aflate în cadrul unor grupuri

Într-o frază intonațională IP/ip, între o unitate de tip "PUSH" şi cea de tip "POP", se pot identifica unități de accentuare care realizează accente de pitch şi tonuri țintă semnificative ce se grupează din punct de vedere tonal. Adnotarea grupurilor implică o secvență de etichete corespunzătoare unităților de accentuare componente şi o etichetă care să caracterizeze funcția grupului în cadrul IP/ip.

Pentru etichetarea componentelor AU am introdus un set de etichete echivalente celor folosite la nivelul IP în baza corespondenței dintre funcțiile acestora la cele două nivele. Acestea sunt următoarele: „*ph*" pentru prima unitate din grup, "po/pu" pentru cele ce încheie grupul şi "F"/"f" pentru cele care generează focus-ul, la fel ca şi la nivelul IP/ip. Unitățile de accentuare care au funcție şi la nivelul IP/ip primesc eticheta pentru funcția de la acest nivel. Unitățile "*ph*"şi "*po*" conțin în general accente de pitch de tip H* iar cele de tip "*pu*" accent de pitch de tip L*.

Etichetele de grup AUG sunt aceleaşi cu cele folosite în adnotarea unităților negrupate exprimând faptul că sunt echivalente acestora din punct de vedere funcțional. Spre exemplu, un grup la începutul unui IP cu componența (PH/po) primeşte eticheta *„PH"*, un grup ce conține unitățile (ph/PO%) primeşte eticheta „*PO%"* iar un grup focalizat ce conține două accente de pitch de tip H* (ph/po) primeşte eticheta *„F"*.

2.4 Etichete pentru unități de accentuare care se desfăşoară pe linia de interpolare ce leagă două paliere tonale

Pentru adnotarea unităților de accentuare al căror contur F0 se desfăşoară pe linia de interpolare dintre două paliere tonale am folosit eticheta "L". Aceste patternuri pot avea tendințe de scădere sau creştere a frecvenței F0.

Figura 3 ilustrează conturul F0 al rostirii textului *Sunt destule scaune?*. După focalizarea nivelului *high* de început, conturul F0 are o tendință descrescătoare până la atingerea unui nivel minim înaintea creşterii finale a interogației totale.

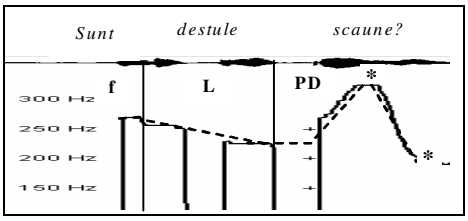


Figura 3: Conturul natural şi stilizat al rostirii textului „Sunt destule scaune"

Conturul unității corespunzătoare cuvântului *destule* urmăreşte această tendință descrescătoare după ce părăseşte nivelul mediu inițial la care a focalizat auxiliarul "*sunt*".

Descrierea conturului melodic din figura 3 este următoarea :

f -l : m/ L / PD

unde atributului de nivel (-l: m), pentru eticheta "f", i s-a asociat valoarea „m" (mediu).

3. Adnotarea XML a intonației pe corpusul voce

Din perspectiva acestui model intonațional, conturul melodic al frazelor intonaționale *IP/ ip* poate fi interpretat printr-o secvență de patternuri care prezintă tonuri țintă, paliere tonale şi patternuri de legătură. Tonurile țintă sunt atinse în cadrul evenimentelor PH/PO/PD/PU/ ph / po iar palierele tonale se formează în cadrul unităților de accentuare de tip "f sau F". Patternurile de legătură, marcate prin eticheta "L", sunt stilizate prin liniile de interpolare între două nivele tonale.

Adnotarea corpus-ului de voce a constat în împărțirea rostirilor în fraze intonaționale (IP) şi intermediare (ip) iar în cadrul acestora textul a fost structurat într-un secvență de unități de accentuare, dintre care unele grupate în AUG-uri. Structurarea textului în unități intonaționale în cadrul fişierului de ieşire XML s-a realizat cu ajutorul tag-urilor prezentate în tabelul 1.

Prin atributul "*PunctSign*" se indică tipul semnului de punctuație prin care se face delimitarea frazei intonaționale.

Atributul "*Break*" indică absența sau prezența pauzei după unitatea de accentuare. Prezența pauzei este marcată în termenii *Short* sau *Large* după cum aceasta este de durată mai scurtă sau mai lungă. Valoarea implicită este *No*, adică absența pauzei.

Atributul "*Function*" asociat unui grup de accentuare indică funcția grupului de accentuare în cadrul unei fraze intonaționale. Atributul "*Function*" asociat unei unități de accentuare indică funcția unității în cadrul unei fraze intonaționale. Valorile acestui parametru sunt chiar etichetele descrise secțiunea 2 (PH, PO, PU, ph, po, pu PD, f, F, PU%, PO%, L, PH+f, PO+f, PU+f, PD+f, L+f, PH+F, PO+F, PU+F, PD+F,L+F).

Cu ajutorul atributului "*Pitch_Accent*" se pot modifica valorile implicite ale accentului de pitch care este asociat unei funcții a unității de accentuare.

Cu ajutorul atributului "*Level* " se indică nivelul mediu al frecvenței F0, relativ la gama de variație a frazei intonaționale (atributul „*Range*"), la care se situează paternul unei unități cu focalizare.

Cu ajutorul atributului "*span*" se indică variația frecvenței F0, pe durata unității de accentuare.

Tabel 1 Tag-urile şi valorile atributelor utilizate în adnotarea conturului intonațional

Tag	**Atribut**	**Valoare**	**Tip unitate intonațională**
<IP/ip>	*Range*	*H, M, L*	Frază intonațională
	Base line	*M, L*	
	PunctSign	*/, / : / ; / . / ! / ? /*	
<AUG>	*Range*	*H, M, L*	Grup de accentuare
	Function	*PH, F, PO, PO%, PU, PU%*	
<AU>	*Function*	*PH, PO, PU, PD, f, F, ph, po, pu, PU%, PO%, L* *PH+f, PO+f, PU+f, PD+f,L+f,* *PH+F, PO+F, PU+F, PD+F,L+F,*	Unitate de accentuare
	Pitch_accent	*H*, L*, L+H*, H+!H* , H+L*,^H**	
	Level	*H,M,L*	
	Break	*No, Short, Large*	
	span	*H, M, L*	

Folosind tag-urile prezentate în această secțiune, se poate marca un text cu informație relativă la intonația unei rostiri a acestuia.

3.1 Exemplu de adnotare a intonației la nivelul unităților de accentuare

Adnotarea la nivelul unității de accentuare creează în conturul F0 nişte repere de formă mai largi decât cele formate de reperele tonale marcate de sistemul ToBI care se desfăşurau numai pe durata silabelor accentuate şi pe tonurile de sfârşit ale frazelor Ip/ip. Precizăm că înțelegerea sistemului ToBI este esențială şi în această perspectivă de modelare a frazelor intonaționale, fără de care nu se pot înțelege realizările particulare ale prototipurilor de contur F0 avute în vedere. Prin atributul de funcție (*Function*) ale AU-urilor se doreşte crearea unor categorii cărora să li se asocieze caracteristici de formă ale conturului F0 şi ca urmare o perspectivă care să permită observarea

asemănărilor dintre realizările acestora indiferent de contextul lexical. În plus paternul pe durata unităților de accentuare reprezintă şi o unitate melodică spre deosebire de cel al evenimentelor tonale marcate ToBI. Astfel conturul unei fraze poate fi privit ca o concatenare a contururilor unităților de accentuare componente.

Din această perspectivă, dacă analizăm contururile frecvenței F0 pentru rostirile corespunzătoare propozițiilor "*Vedeai că lucea de culoarea rubiniului"* şi "*Avem la dispoziție vreo douăzeci de minute"*, prezentate în figura 4 şi respectiv figura 5, se pot observa următoarele:

- ambele rostiri formează o singură frază intonațională care are în componență patru unități de accentuare;
- unitatea de accentuare din poziție inițială (corespunzătoare verbelor *vedeai* şi *avem*) realizează ridicarea tonului de la nivelul de *low* până la nivelul cel mai înalt din fraza intonațională, pe durata silabei accentuate. Ambele corespund astfel prototipului etichetei "PH" de început a unității de discurs
- unitățile din poziția a doua focalizează în grade diferite cuvintele corespunzătoare. În cazul verbului "*lucea*", aceasta este slabă şi se realizează prin mici variații în jurul unui nivel tonal care coboară aproape până la cel de început. În celălalt caz, complementul circumstanțial "*dispoziție*" este focalizat cu accent de pitch proeminent, cu variație între minima de 212 Hz şi maxima de 260 Hz , deci în jurul valorii medii de 225 Hz.
- în ambele cazuri ultimele două substantive formează grupuri nominale, relație de grup ce este exprimată în intonație de contrastul tonal al țintelor din cele două unități componente, accentul H* al primei (cu eticheta "ph+F") față de tonul de *low* din cea de-a doua (cu eticheta "PO%"). În plus primele unități realizează şi focalizarea cuvintelor corespunzătoare prin păstrarea tonului de final egal cu cel de la începutul cuvintelor. Gradul de focalizare este mai mic decât al cuvintelor din unitățile de accentuare anterioare (f/F).
- căderea către tonul final al frazei nu se face cu accent de pitch semnificativ şi ultima unitate din grup fiind şi ultima în fraza de tip terminal are eticheta PO%.

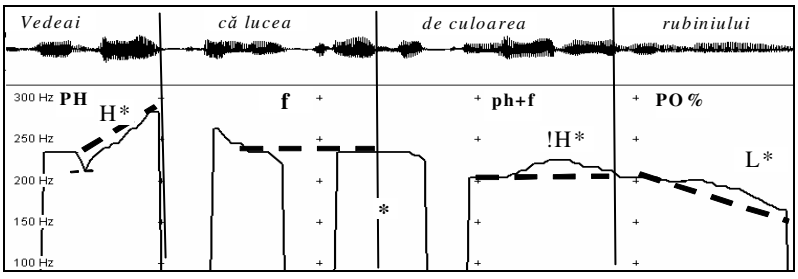


Figura 4: Conturul frecvenței F0 pentru rostirea propoziției „ Vedeai că lucea de culoarea rubiniului"

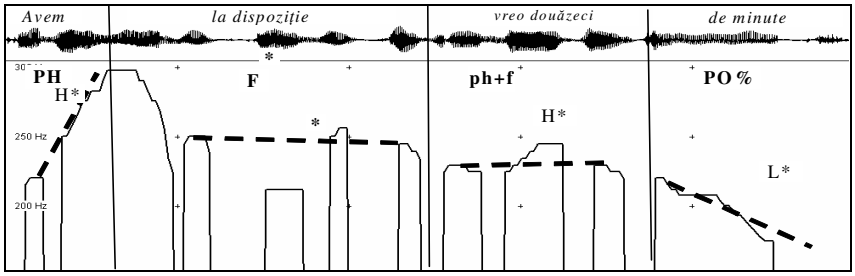


Figura 5: Conturul frecvenței F0 pentru rostirea propoziției
„Avem la disapoziție vreo douăzeci de minute"

În consecință, contururile melodice ale celor două rostiri pot fi descrise cu următoarele secvențe de etichete:

PH / f / (ph+f / PO%) şi respectiv *PH / F / (ph+f / PO%)*

care nu diferă decât prin proeminența focalizării celei de-a doua unități şi nu evidențiază prin intonație diferențele între structurile sintactice ale celor două propoziții.

4. Concluzii

Schimbările propuse în ierarhia intonațională se referă la: introducerea noțiunii de „grup de unități de accentuare – Accentual Unit Grup (AUG)" în loc de unitate ritmică; înlăturarea convenției care impunea ca orice frază intonațională să conțină o unitate ritmică; folosirea etichetelor ToBI cu semnificația de valori ale atributelor asociate etichetelor de pattern de contur F0 la nivelul unităților de accentuare. Noțiunea de „grup de unități de accentuare" exprimă faptul că unitățile de accentuare se grupează nu numai pentru respectarea unor formule ritmice ci şi datorită legăturilor sintactice sau semantice în cadrul grupurilor determinate la nivelul textului. În urma acestor modificări, o frază intonațională poate conține numai unități de accentuare negrupate la nivel AUG sau secvențe de unități negrupate şi grupate la nivel AUG. Patternul de contur F0 al unităților de accentuare negrupate se raportează direct la coordonatele tonale ale frazei intonaționale, pe când patternurile celor grupate se raportează la coordonatele tonale asociate grupului de accentuare.

Această convenție simplifică mult înțelegerea intonației la nivel lingvistic, deoarece gruparea unităților de accentuare la nivel AUG apare doar atunci când acest lucru se observă la nivelul melodiei frazei intonaționale. La nivelul generări conturului frecvenței F0, această descriere va permite o mai bună mapare a patternurilor asociate unităților de accentuare în spațiul (frecvență, timp).

Referințe bibliografice

Heggtveit P. O., Natvig J. E. (2001). Intonation Modelling with a Lexicon Natural F0 Contours", *Proceedings of Eurospeech2001,* p. 1163-1166

Mertens P. (2002). Synthesizing elaborate Intonation Contour in Text-to-speech For French, *Proceedings of the Speech Prosody Conference*, p. 499-502

Curteanu N., Trandabăț D, Moruz A (2007). Syntax-Prosody Interface for Romanian within Information Structure Theories, *Advances in Spoken Language Techno-logy, Romanian Academy, 207-216*.

Sun-Ah Jun (2004). Intonational phonology of Korean Revisted, *Japanese-Korean Linguistics Conference*, Tucson, Arizona, nov.5-7, 2004

Apopei V., Jitcă D. (2007). Module for generating the F0 Contour using as input a Text structured by prosodic information, *Advances in Spoken Language Techno-logy, Romanian Academy, 119-126*.

Apopei V., Jitcă D. (2006). Schema XML de adnotare a intonației în cadrul corpusurilor de text, *Lucrările atelierului „Resurse ligvistice şi instrumente pentru prelucrarea limbii române*, p. 9-14

Ladd D. R. (1996). Intonational Phonology, *Cambridge University Press*

CAPITOLUL 2

DICȚIONARE ŞI CORPUSURI ADNOTATE PENTRU PRELUCRAREA TEXTELOR

ACHIZIȚIE LEXICALĂ NESUPERVIZATĂ PENTRU ADNOTARE MORFO-LEXICALĂ

DAN TUFIŞ, RADU ION, ELENA IRIMIA, ALEXANDRU CEAUŞU

Institutul de Cercetări pentru Inteligență Artificială

Str. 13 Septembrie, nr. 13, Bucureşti 050711, România

{tufis, radu, elena, aceausu}@racai.ro

Rezumat

Articolul prezintă o strategie de achiziție lexicală argumentată de necesitatea unui lexicon de dimensiuni mari, validat, pentru îmbunătățirea rezultatelor procesului de adnotare morfo-lexicală a unui text. Metoda descrisă este complet automată şi, deşi implementată doar pentru limba română, ea beneficiază de o arhitectură generală care poate fi preluată pentru orice altă limbă. În încheierea articolului sunt prezentate rezultatele unui experiment care a relevat faptul că din aproximativ 9.5K de text selectat aleatoriu de pe Internet, 0.85K de cuvinte noi (nu sunt prezente în datele de antrenare ale taggerului), împreună cu lemele şi etichetele POS, pot fi adăugate automat în lexiconul românesc prin această strategie.

1. Introducere

Adnotarea morfo-lexicală (POS tagging) este unul dintre acei paşi de pre-procesare din ingineria limbajului care pot fi efectuați cu rezultate destul de precise. Pentru limba engleză, rezultatele experimentale arată o acuratețe mai mare de 96% folosind diverse seturi de etichete şi corpusuri de antrenare (Brill, 1996; Ratnaparkhi, 1998; Brants, 2000). Câțiva cercetători au observat că sarcina cea mai dificil de rezolvat a tehnologiilor de adnotare actuale rămâne dezambiguizarea lexicală a cuvintelor care nu se află în lexicoanele modelelor de limbă. Această problemă este relevantă în special pentru taggerele bazate pe HMM (Hidden Markov Models – Modele Markov Ascunse), unde poate fi remediată uşor din moment ce probabilitățile de tranziție şi cele de emisie lexicală sunt calculate independent. Unul dintre cei mai bune taggere HMM publice, TnT (Brants, 2000), permite utilizarea, la runtime, a unui lexicon adițional care este consultat ori de câte ori un cuvânt necunoscut este întâlnit. Similar, taggerul TTL (Ion, 2007) permite adăugarea unor dicționare adiționale specifice în momentul în care modelul de limbă este construit. O problemă cu o astfel de abordare este că dacă un cuvânt a fost văzut în datele de antrenare cu o clasă de ambiguitate (mulțimea tuturor etichetelor POS posibile) incompletă, cuvântul va fi cunoscut tagger-ului iar eticheta (tag-ul) pe care cuvântul o va primi într-un context nou va fi una dintre cele din lista de ambiguitate incompletă. Este evident faptul că cea mai simplă rezolvare a acestei probleme este extinderea lexicoanelor suport cu intrări noi pentru completarea claselor de ambiguitate. De altfel, este de preferat ca pentru o anumită lemă împreună cu o parte de vorbire a sa posibilă, toate formele ocurență ale acesteia să fie prezente în lexicon.

Un impediment pentru extinderea lexicoanelor în acest fel în vederea îmbunătățirii adnotării morfo-lexicale HMM, o constituie distribuția uniformă a probabilităților lexicale ale intrărilor noi (cu alte cuvinte, fiecare formă ocurență apare o singură dată cu

fiecare etichetă). O soluție pentru redistribuția acestor probabilități în conformitate cu evidența corpusului, este aceea că probabilitățile lexicale pentru etichetele unor cuvinte noi sunt calculate pe baza distribuției etichetelor în clase de ambiguitate identice sau asemănătoare ale unor cuvinte cunoscute din corpusul de antrenament. Aceste clase de ambiguitate trebuie însă căutate printre cele ale cuvintelor rare din corpus. Rațiunea acestei tehnici este următoarea: dacă un cuvânt este folosit intens, contextele sale de apariție sunt diverse şi, în consecință, cuvântul trebuie să aibă mai mult de o categorie gramaticală pentru a se încadra sintactic în aceste contexte şi viceversa, ne putem aştepta ca, dacă un cuvânt este folosit rar, ambiguitatea sa morfo-lexicală să fie redusă. Astfel, un cuvânt nou poate fi considerat că apare rar şi în consecință, distribuția etichetelor în clasa sa de ambiguitate ar trebui să fie aproximativ aceeaşi cu distribuția etichetelor într-o clasă de ambiguitate identică sau similară a unui cuvânt rar.

Pentru a verifica că clasa de ambiguitate a unui cuvânt se reduce în funcție de frecvența acestuia în corpus, am făcut un experiment pe un corpus paralel englez-român al ICIA, adnotat morfo-lexical şi validat manual (aproximativ un milion de cuvinte pentru fiecare limbă). Astfel, am extras toate cuvintele (atât în engleză cât şi în română) şi le-am sortat în ordine descrescătoare a frecvențelor ocurențelor. Pentru fiecare cuvânt din această listă, s-a calculat ambiguitatea sa morfologică (numărul de etichete POS diferite cu care apare cuvântul).

Figurile 1 şi 2 descriu un grafic al rangurilor de frecvență (axa X) cu medii ale ambiguității POS în ferestre succesive de câte 100 de cuvinte din lista de frecvențe (axa Y) pentru engleză şi română. Se poate observa o descreştere clară a mediei ambiguității POS în timp ce rangul frecvenței creşte (evident, frecvența descreşte).

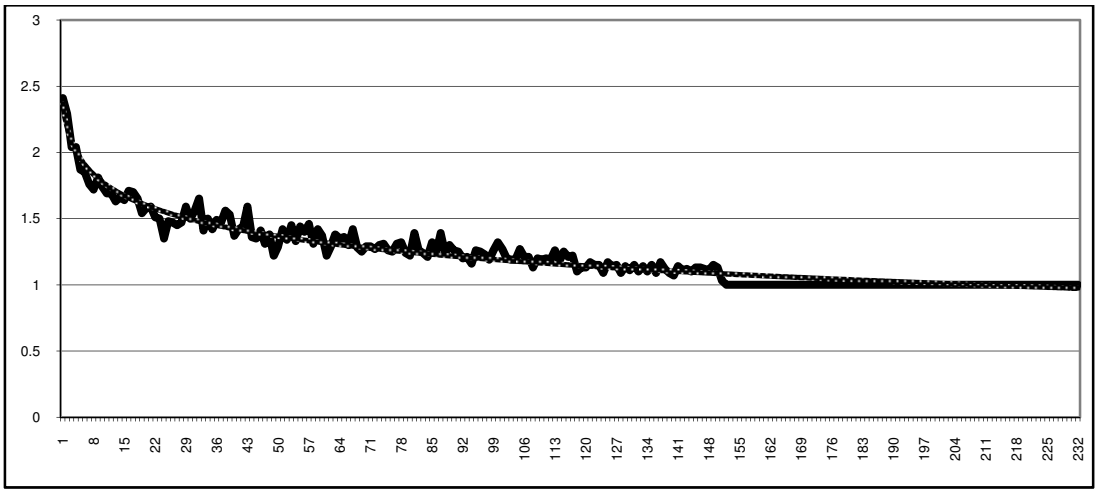


Figura 1: Medii ale ambiguității POS ale ferestrelor conținând 100 de cuvinte succesive din lista de frecvențe a corpusului ICIA în engleză, în raport cu rangurile frecvențelor.

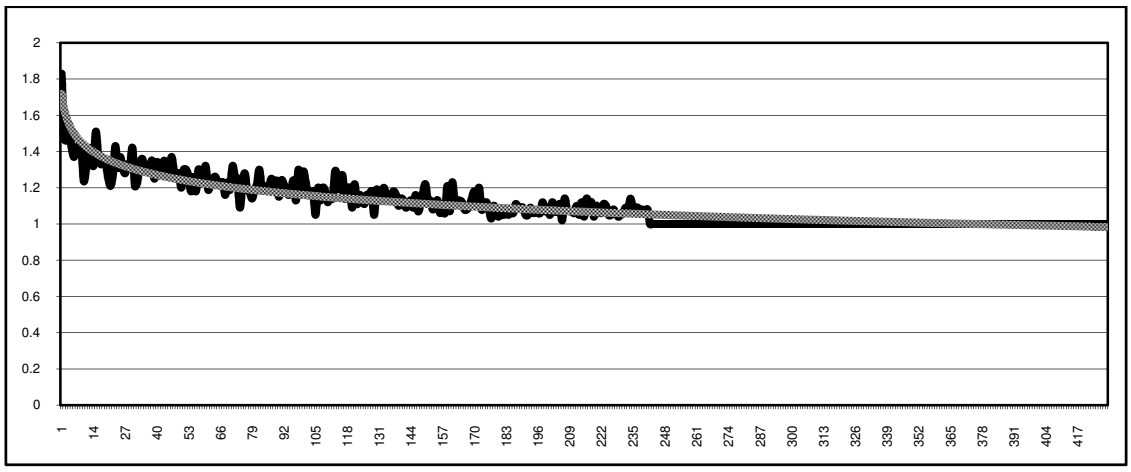


Figura 2: Medii ale ambiguității POS ale ferestrelor conținând 100 de cuvinte succesive din lista de frecvențe a corpusului ICIA în română, în raport cu rangurile frecvențelor.

Analizând toate aceste considerații, am concluzionat că adnotarea morfo-lexicală poate beneficia foarte mult de prezența unui lexicon (cu cât mai mare, cu atât mai bine) de forme flexionare, împreună cu etichetele morfologice posibile corespunzătoare.

2. Achiziția lexicală

2.1 Considerente generale

Vom descrie aici o metodă complet automată pentru îmbogățirea unor astfel de lexicoane cu forme flexionare noi, împreună cu etichetele morfo-lexicale asociate, ce au fost achiziționate de către taggerul POS în adnotarea de texte noi. Metoda noastră este, deocamdată, aplicabilă doar pentru limba română, dar arhitectura este generală şi poate fi implementată pentru orice limbă. Ea se bazează pe teoria morfologiei paradigmatice şi pe implementarea ei pentru limba română (Tufiş, 1989) precum şi pe generatorul morfologic paradigmatic ROG (Irimia, 2007). Conform teoriei morfologiei paradigmatice, un cuvânt este compus dintr-o *rădăcină* şi o *terminație,* care poate conține la rândul ei un sufix derivațional şi o terminație flexionară. De exemplu, substantivele *ceasornicar* şi *cărbunar* flexionează conform aceleiaşi paradigme: nominativ masculin sufix 1 (nomsuf1). Astfel, *ceasornicar* este format din rădăcina morfologică *ceasornic* şi sufixul derivațional *ar*. În mod similar, *cărbunar* are rădăcina *cărbun* şi acelaşi sufix derivațional. La nominative/acuzativ, singular, definit, ambele cuvinte (forme) sunt compuse prin adăugarea sufixului flexionar *ul,* astfel încât obținem *ceasornic+ar+ul* şi *cărbun+ar+ul* şi aşa mai departe, pentru întreaga paradigmă. Odată ce rădăcina pentru întreaga paradigmă este detectată, putem genera toate formele din familia tematică a unei anumite rădăcini, cunoscând şi paradigma sa flexionară.

Componenta de achiziție lexicală nesupervizată este bazată pe următoarea idee: când un text de dimensiuni mari este adnotat, o lemă (forma din dicționar a unui cuvânt) nouă (luând în considerare doar categoriile gramaticale clasă deschisă) are mari şanse să apară cu mai multe forme flexionare ale sale. În cele mai multe cazuri, două sau trei forme flexionare ale aceleiaşi leme pot identifica paradigma acesteia. În caz contrar, prin analiza sufixelor formelor disponibile se poate identifica o mulțime restrânsă de paradigme relevante. Atunci când ROG este invocat cu o lemă şi un identificator de

paradigmă, el generează întreaga familie paradigmatică a lemei date. Plecând de la formele flexionare ale unei leme disponibile într-un text dat şi de la familiile paradigmatice generate de ROG pentru această lemă, se poate detecta paradigma corectă şi, ulterior, formele corecte pot fi adăugate în lexicon.

Următorii paşi sunt parcurşi pentru a introduce automat intrări adnotate morfo-lexical în lexicon:

1. rulăm POS-taggerul pe textul de intrare şi extragem toate cuvintele necunoscute (care nu au fost văzute în datele de antrenare). Folosim TTL, (Ion, 2007) care realizează adnotarea morfo-lexicală şi lematizarea textului. Dacă textului îi lipsesc diacritice, este rulat prin recuperatorul de diacritice pentru limba română DIAC (Tufiş & Ceauşu, articol în Atelierul ConsILR 2007);
2. grupăm cuvintele din lista de cuvinte necunoscute după lemă şi categoria gramaticală; de exemplu, *ceasornicar* (singular, indefinit, forma lemă), *ceasornicarul* (singular, definit, nominativ/acuzativ), *ceasornicarului* (singular, definit, genitiv/dativ) sunt toate forme valide ale substantivului *ceasornicar*;
3. pentru fiecare grup identificat la pasul precedent, generăm formele care lipsesc potrivit paradigmei care se aplică întregului grup. Aici, partea mai greu de realizat este determinarea paradigmei conform căreia flexează o anumită lemă. Dacă grupul de ocurențe ale aceleiaşi leme conține suficiente forme, atunci elementele sale pot prezice paradigma relevantă. Altfel, mai mult de o paradigmă este aplicabilă; în acest caz, testăm fiecare dintre paradigme şi filtrăm rezultatele invalide cu Google™. Această filtrare ne asigură că doar forme corecte sunt generate (sau forme pentru care Google™ întoarce un număr mare de ocurențe). Algoritmul de identificare a paradigmei este descris în secțiunea 2.2. Pentru mai multe detalii, a se vedea (Irimia, 2007). Luând în considerare exemplul de la pasul precedent, putem adăuga la acest pas încă 3 forme, obținând paradigma completă a lemei *ceasornicar*: *ceasornicari* (plural, indefinit), *ceasornicarii* (plural, definit, nominativ/acuzativ) şi *ceasornicarilor* (plural, definit, genitiv/dativ);
4. adăugăm lexiconului toate grupurile complete de forme obținute la pasul precedent.

Procedura de mai sus se aplică formelor flexionare corecte din limba română. Dacă anumite forme flexionare conțin greşeli ortografice, pasul 3 întoarce frecvența zero sau foarte mică pe Google™ a membrilor familiilor paradigmatice generate.

2.2 Descrierea algoritmului de identificare a paradigmei

Pentru a explica algoritmul de funcționare a aplicației, reproducem în continuare un exemplu de intrare în fişierul care descrie morfologia limbii române în conformitate cu teoria morfologiei paradigmatice menționată mai devreme:

```
<PARADIGM PARADIGM="nomneu1" INTENSIFY="none">
    <TYPE TYPE="{proper common}">
      <NUM NUM="singular" GEN="masculine">
        <ENCL ENCL="no">
```

```
            <CASE CASE="{nom, gen, dat, acc, voc}">
              <TERM TERM="" ALT = "1"/>
            </CASE>
          </ENCL>
…
    </NUM>
…
  </TYPE>
…
</PARADIGM>
```

Figura 3: Fragment de intrare în fişierul care descrie morfologia paradigmatică a limbii române

Se poate observa că o astfel de intrare are o structură arborescentă şi specifică toate informaţiile necesare identificării etichetei morfo-lexicale a formei unui cuvânt (vom folosi tagset-ul MSD), dacă vom coborî din rădăcină pe ramura corespunzătoare a arborelui; frunzele conţin informaţii despre terminaţia ce trebuie lipită de rădăcina cuvântului pentru a obţine forma flexionată. Atributul ALT specifică rădăcina pe care o vom folosi în cazul în care avem de a face cu un cuvânt care suferă alternanţe la nivelul rădăcinii (în limba română, multe substantive au două rădăcini, una pentru singular, alta pentru plural – ex.: fereastră/ferestre – în timp ce numărul rădăcinilor verbului poate varia de la unu la şapte). Valoarea atributului ALT este un număr care reprezintă poziţia rădăcinii potrivite din lista rădăcinilor posibile pentru o lemă şi o etichetă morfo-lexicală. Exploatând structura unei astfel de intrări, este uşor de generat familia de forme flexionate a unei leme, dacă aceasta se regăseşte în baza de date, care îi poate asocia atât paradigma căreia îi aparţine, cât şi rădăcina/rădăcinile.

Adevărata problemă de rezolvat o reprezintă însă lemele care nu se regăsesc în baza de date. Pentru îmbogăţirea acesteia, am dezvoltat un modul care să identifice rădăcina şi paradigma asociată pentru cuvinte noi, având ca date de intrare cât mai multe forme flexionate posibile, extrase din tbl.wordform.ro (toate formele conţinute de acesta, adnotate cu etichete MSD şi leme, au fost validate manual) sau din corpusuri. Dacă procesul de identificare se realizează corect, putem îmbogăţi şi tbl.wordform.ro cu forme noi.

În continuare vom face descrierea algoritmului de identificare a rădăcinii (funcţionează deocamdată doar pentru cuvinte fără alternanţă la nivelul rădăcinii, dar, în prezent, se lucrează la varianta care suportă şi alternanţe) şi a paradigmei corecte:

Date de intrare:

$$L_1 \longrightarrow \begin{matrix} w_1 & l & M_1 \\ w_2 & l & M_2 \\ \vdots & \vdots & \\ w_n & l & M_n \end{matrix}$$

- lista formelor (w) disponibile în tbl.wordform.ro pentru o lemă dată l, împreună cu etichetele MSD (M) corespunzătoare;

$$L_2 \longrightarrow \begin{matrix} s_1 & M_1 & p_1 \\ \vdots & \vdots & \vdots \\ s_k & M_k & p_k \end{matrix}$$

- o listă a tuturor sufixelor flexionare (s) posibile în limba romană, împreună cu etichetele MSD (M) şi paradigmele asociate (p), extrasă din fişierul ce conține descrierea morfologică completă a limbii române.

Date de ieşire: rădăcina (R) şi paradigmele (lista PAR) care corespund listei de forme de intrare.

Descrierea procesului de identificare a rădăcinii:

Pentru fiecare w_i, se construieşte mulțimea S_i formată din toate tripletele $(s_j, M_i, p_j) \subset L_2$, unde $j \in \overline{1,k}$ iar s_j este un sufix al lui w_i. Pentru identificarea rădăcinii mulțimii $\{w_1, \cdots, w_n\}$am implementat următoarea procedură:

Pentru fiecare w_i, calculăm mulțimea $R_{w_i} = \{w_i - s_1 = r_1, \cdots, w_i - s_p = r_p\}$

Calculăm $R = \bigcap_{i=1}^{n} R_{w_i}$; dacă $|R| > 1$, alegem R astfel încât lungimea lui R este minimă.

Funcția de identificare a paradigmei:

```
FindMSDandPARADIGM(MPi)
{
Foreach Si{
   Foreach (s, M, p) in Si{
      If (wi − s == R) {
         Pi{p} = M;
         //Pi este o structură hash;
      }
   }
}
```

$$PAR = \bigcap_{i=1}^{n} keys(P_i)$$

Pentru situațiile în care nu se poate identifica în mod exact paradigma (PAR este o listă cu mai mult de o paradigmă), se generează toate formele pentru toate paradigmele din listă şi este aleasă paradigma ale cărei forme sunt validate de Google™.

Pentru cazul în care, după extinderea bazei de date, încă nu putem identifica în ea lema pentru care trebuie să generăm, a fost necesar un modul care să prezică paradigma şi rădăcina cuvântului folosind similarități între terminația lemei noi şi terminațiile lemelor din baza de date (tehnică simplă de pattern matching între cel mai lung subşir-terminație al cuvântului nou şi subşiruri-terminație ale lemelor cunoscute). Pentru neologisme şi cuvinte compuse prin prefixare acest modul dă rezultate foarte bune.

5. Evaluări şi concluzii

Lexiconul de forme flexionare pentru limba română (referit mai sus ca tbl.wordform.ro) conține în prezent peste 800,000 de intrări şi a fost construit pornind de la un lexicon cu 450,000 intrări validate manual. Fiecare intrare conține o formă împreună cu lema şi

eticheta sa morfo-sintactică. Cu un lexicon de asemenea dimensiuni, ale cărui forme sunt extrase din texte jurnalistice şi de ficţiune editate cu atenţie, cuvintele necunoscute nu sunt foarte frecvente. Totuşi, atunci când avem de-a face cu alte registre literare sau cu texte mai puţin bine editate (precum cele de pe web), frecvenţa cuvintelor necunoscute s-a dovedit semnificativă (aproximativ 2%) iar ea nu este doar o sursă de propagare a erorilor de adnotare ci, în acelaşi timp, o sursă importantă pentru extinderea lexiconului de forme flexionare. Am colectat de pe Internet în mod aleator 6 texte în limba română aparţinând unor domenii diferite şi totalizând aproximativ 9.5K de unităţi lexicale şi am calculat statistici ale numărului de unităţi lexicale şi ale numărului de cuvinte necunoscute. Am fost de asemenea interesaţi şi de acurateţea de adnotare morfo-lexicală şi de lematizare a acestora din urmă. Rezultatele experimentelor sunt rezumate în Tabelul 1. În coloana **Unităţi Lexicale**, prima cifră indică numărul de unităţi lexicale, cea de-a doua (separată prin '/') numărul de unităţi lexicale unice. Coloana **Necunoscute** prezintă numărul de cuvinte-formă care nu au fost văzute în datele de antrenare, iar dintre acestea, coloana Er. Ort. numără erorile de ortografie. Ultimele două coloane numără erorile de adnotare morfo-lexicală şi de lematizare ale cuvintelor necunoscute. Astfel, acurateţea POS pe cuvinte necunoscute este 100 – 8.24 = 91.76% iar acurateţea lematizării pe acelaşi set de cuvinte este 92.86%. Evident, aceste date sunt afectate de erorile de ortografie, care produc atât erori de adnotare morfo-lexicală cât şi erori de lematizare.

Tabela 1: Proporţia cuvintelor necunoscute din 6 texte în limba română alese aleator, împreună cu procentele de erori pentru adnotare şi lematizare.

	Unităţi lexicale	**Necunoscute**	**Er. Ort.**	**Er. POS**	**Err. Lema**
Filosofice	1922/880	26/24	0	2	3
Şt. Calc.	1018/488	26/22	12	5	5
Medicale	2601/1002	106/73	1	3	2
Religioase	1312/540	10/10	1	1	1
Jurnalistice	1080/527	2/2	0	0	0
Enciclopedice	1559/737	12/12	4	4	2
TOTAL	9492/4174	182/143	18 (9.89%)	15 (8.24%)	13 (7.14%)

După adnotarea morfo-lexicală, am trimis lista de cuvinte-formă necunoscute (143 de intrări, conţinând leme şi etichete POS pentru fiecare intrare) ca date de intrare pentru generatorul morfologic ROG. Acesta a întors o listă cu 843 intrări corecte (împreună cu lema şi eticheta POS pentru fiecare intrare) corespunzând la 117 forme necunoscute. Cuvintele necunoscute rămase (143 – 117 = 26) fie erau ortografiate greşit (18) fie lema sau eticheta POS greşite erau cauze pentru care paradigma nu a putut fi identificată (toate erorile de ortografie au generat erori de lematizare iar cele mai multe dintre ele erau incorect adnotate morfo-lexical). Acest experiment a arătat că din aproximativ 9.5K de text nou, 0.85K de cuvinte noi, împreună cu lemele şi etichetele POS, pot fi adăugate automat în lexiconul românesc. Plănuim să implementăm această procedură nesupervizată serviciului web de adnotare şi lematizare care a fost deja folosit pentru procesarea unor texte care conţin mai mult de 2G de cuvinte.

Referințe bibliografice

Brants, T. (2000). TnT – A Statistical Part-Of-Speech Tagger. In Proceedings of the 6th Applied NLP Conference, ANLP-2000, pages 224–231, Seattle, WA, April 29 – May 3, 2000.

Brill, E. (1996). A Simple Rule-Based Part-Of-Speech Tagger. In Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing, pages 152–155, Trento, Italy, April 1992.

Ion, R. (2007). Metode de dezambiguizare semantică automată. Aplicații pentru limbile engleză şi română. Teză de doctorat, Academia Română, Bucureşti 2007

Irimia, I. (2007). ROG - A Paradigmatic Morphological Generator for Romanian. In Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, October 5-7, 2007, Poznań, Poland, 2007.

Ratnaparkhi, A. (1998). Maximum Entropy Models for Natural Language Ambiguity Resolution. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998

Tufiş, D. (1989). It Would Be Much Easier If WENT Were GOED. In Harry Somers, Mary McGee Wood (eds.), Proceedings of the 4th European Conference of the Association for Computational Linguistics, Manchester, 1989.

RESURSE LINGVISTICE ÎN FORMAT ELECTRONIC. BIBLIA 1688. REGI I, REGI II - PROBLEME, SOLUȚII

VLAD SEBASTIAN PATRAŞ[1, 2], GABRIELA PAVEL[1, 2], GABRIELA HAJA [2]

[1] *Universitatea "Al.I.Cuza", Facultatea de Informatică, Iaşi – România*
[2]*Institutul de Filologie al Academiei Române, Filiala Iaşi*

{vlad.patras, pavelg}@info.uaic.ro, gabihaja@gmail.com}

Rezumat

Lucrarea prezintă rezultatele cercetării din cadrul grantului CNCSIS 1454 *Resurse lingvistice în format electronic. Monumenta linguae Dacoromanorum. Biblia 1688. Pars VII. Regum I, Regum II - ediție critică și corpus adnotat*, desfăşurat în perioada 2006-2007 în cadrul Institutului de Filologie Română „A. Philippide" al Academiei Române, Filiala Iaşi. Au fost realizate următoarele: program de parsare, adnotare, indexare automată a textului românesc vechi şi interfață de verificare/corectare a adnotării morfologice, la nivel de cuvânt.

1. Introducere

Omenirea parcurge o nouă vârstă, a comunicării necondiționate spațial şi temporal. Faptul este posibil deoarece mijloacele electronice, într-o explozie evolutivă, permit stocarea, conservarea şi transferul imediat al informației. Crearea şi utilizarea acestor mijloace este comparabilă ca importanță, din perspectiva socio-culturală, cu descoperirea scrisului. Trăim, altfel spus, o revoluție. Astfel, cartea tipărită, ca suport pentru transmiterea cunoştințelor, a ideilor, a experiențelor, cunoaşte un proces de transformare. Deocamdată, este tot mai des însoțită de formatul său electronic. Încă înainte de această etapă, au fost lansate pe piață cărțile în format audio, înregistrate pe suport magnetic. Însă, având în vedere, pe de o parte, perisabilitatea cărților şi, pe de altă parte, necesitatea şi posiblitatea căutării inteligente a datelor necesare cercetărilor de orice tip, ne-am propus să realizăm o versiune electronică a uneia dintre cele mai importante realizări ale culturii române, prima versiune integrală în limba română a Bibliei, tipărită în anul 1688 la Bucureşti. Pentru început, ne-am stabilit ca obiectiv editarea în formă electronică şi tradițională (tipărită) a două părți: A împărăției I şi A împărăției II, în trei variante de traducere din secolul al XVII-lea: Biblia 1688, Ms. 45 şi Ms. 4389.

Finalitatea proiectului nostru este nu doar de a scrie aceste texte în documente digitale, ci, mai ales, crearea / adaptarea unor instrumente care să permită şi indexarea automată cu posibilitate de acces la diverse variante, capitole şi versete ale acestei lucrări. Inițierea acestui proiect a fost cu putință datorită, în primul rând, uneltelor de prelucrare ale limbajului natural create pentru limba română.

Folosind serviciile web ale Institutului de Cercetări pentru Inteligență Artificială al Academiei Române, Bucureşti (şi anume parserul morfologic TTL pentru limba română) s-au putut adnota textele propuse. Totuşi, ne-am confruntat cu o problemă:

adnotatorul este antrenat pe limba română actuală şi nu cunoaşte cuvintele şi formele din limba secolului al XVII-lea.

Soluția la care am ajuns a fost corectarea semi-automată a textului adnotat cu TnT (Thorsten, 2000). Am optat pentru această soluție deoarece astfel s-a putut face o altă adnotare (ignorându-se adnotarea inițială), în care să se țină cont de indexul cărții. În acelaşi timp, a fost posibilă şi adnotarea de către un grup de lingvişti a textelor, prin intermediul unei interfețe de verificare / corectare a adnotării.

2. Parsarea şi adnotarea versetelor din Biblie

Mai întâi a fost necesară trecerea textelor în format electronic, pentru a fi uşor procesabile, în vederea creării unei resurse electronice de utilizarea căreia se vor bucura specialişti în diverse domenii – de la filologie clasică până la lingvistică computațională.

Pentru o prelucrare eficientă a fost nevoie ca textele să fie împărțite în unități prestabilite şi stocate pentru facilitarea unor operații cum ar fi sortare, filtrare etc.

2.1 Tehnologii folosite

Pentru a accesa documentele Word în Java, textele au fost salvate folosindu-se formatul deschis RTF, pentru care Java oferă suport prin RTFEditorKit, componentă a sistemului Swing pentru procesare şi vizualizare a documentelor cu formatări.

Problema : Datorită faptului că formatul RTF este în continuă schimbare, suportul în Java este compatibil doar cu un subset al formatului. Astfel unele caractere speciale ale editorului Word nu sunt citite.

Soluție : S-au înlocuit in text caracterele nerecunoscute cu varianta unicode sau cu o serie de caractere ce au aceeaşi semnificație.

2.2 Metoda de parsare

Textele au o anumita organizare, cum ar fi: titlul are font bold şi de mărimea 14, o lemă este cu bold şi are mărimea 10, după care urmează detalii, iar pe următoarele linii inflexunile etc. ce aduc o formalitate asupra limbajului natural. Astfel, pentru această operație, s-au putut adapta idei din teoria limbajelor formale şi a automatelor pentru parsare.

Problema 1: Într-un automat determinist, un caracter indică trecerea într-o altă stare a automatului. În cazul parsării *Indicelui*[1], adeseori trebuie citite înainte mai multe caractere pentru a fixa starea următoare, împreuna cu atributele caracterelor.

Soluție : Nu pot exista mai multe variante de parsare, totuşi s-a considerat automatul ca fiind nederminist si s-a folosit tehnica look-ahead si noi reguli. Unde nici in acest mod nu se poate determina unic tipul următorului token, aplicația va semnala acest lucru.

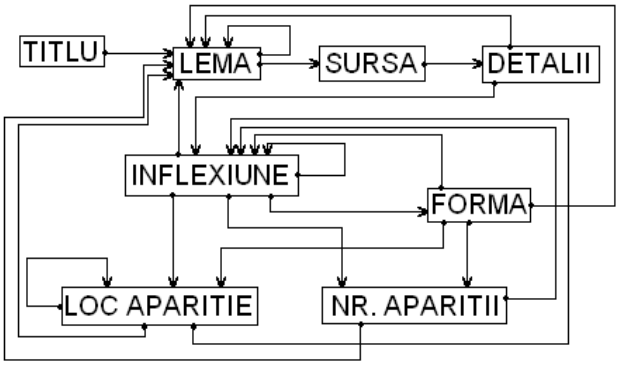
Problema 2: Limbajul uman şi greşelile de redactare. Cu toate că am considerat o anumită organizare a textului, există multe abateri, care pentru specialistul lingvist sunt

[1] S-a folosit un Indice tipărit deja, al volumului anterior din *Monumenta linguae Dacoromanorum. Biblia 1688* şi anume *Pars VI. Isus Navi, Judecătorii, Ruth*, Iaşi, Editura Universității "Alexandru Ioan Cuza", 2004.

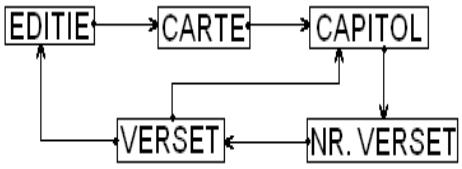
logice sau nu deranjează, cum ar fi omiterea inflexiunii ce se repetă sub diferite forme, adăugarea de precizări şi detalii etc. Pe lângă acestea sunt şi multe inconsecvenţe de redactare, unele aproape invizibile utilizatorului, cum ar fi omiterea stilului italic pentru punct sau pentru o cifră, trecerea pe altă linie aproape de locul unde s-ar fi împărţit implicit etc.

Soluţie : Modificarea programului astfel încât acesta să recunoască cele mai frecvente tipuri de greşeli şi, acolo unde este posibil, să încerce o corectare; de asemenea, în punctul în care nu poate continua, să precizeze motivul şi locaţia.

Rezultatul acestui pas este o listă de token-uri, aranjate în ordinea găsirii lor în text (cu excepţia corectărilor). Aceste unităţi pot fi de diferite feluri, în funcţie de tipul textului (Biblie sau Index) şi respectă o anumită relaţie de precedenţă, spre exemplu: după TITLU urmează LEMA, după LEMA poate urma LEMA (asociere) sau SURSA (sursa detaliilor despre lemă). Relaţia de precedenţă poate fi determinată de schema din figura 1.



precedenţă token-uri in textul cu Indecşi



precedenţă token-uri în textul *Bibliei*

Figura 1: Automatul de la baza parserului

2.3 Ierarhizare şi stocare

Lista de token-uri este dificil de folosit la operaţiile necesare procesării. Spre exemplu, pentru găsirea titlului cărţii din care face parte un VERSET trebuie parcursă calea spre stânga până la găsirea unui token CARTE. Formatul XML este conceput pentru a stoca ierarhic şi, de aceea, rezultatul va putea fi folosit pentru diverse aplicaţii, cu diverse unelte pentru prelucrare. La acest pas se face verificarea listei pentru a se asigura respectarea relaţiei de precedenţă, după care se scrie un fişier XML parcurgând lista şi salvând informaţiile din fiecare token în blocul sau tag-ul destinat lui.

2.4 Adnotarea semi-automată a cuvintelor

Pentru adnotarea corectă a cuvintelor este nevoie de un lingvist atent şi priceput. Însă la volume mari de texte se întâlnesc de multe ori aceleaşi cuvinte în contexte similare. În astfel de cazuri, un program poate ajuta adnotarea, folosind eficient munca lingvistului de până la un moment dat.

Metoda : Se adună rezultatele adnotărilor şi asupra acestora se aplică diverşi algoritmi de căutare, sortare şi comparare, care pot prezice lema şi forma flexionară a unui cuvânt ce a mai fost întâlnit şi adnotat sau a unui cuvânt nou, dar a cărui lemă este cunoscută.

Baza de date : Principala sursă o reprezintă adnotările pe care parserul le-a extras din indecşii existenţi. Pe parcursul adnotării textelor, s-au adăugat rezultatele aflate deja în formatul necesar. Se formează astfel o colecţie de cuvinte vechi pentru care se apreciază lema şi forma. Cuvintele rămase, dacă nu şi-au schimbat forma, pot fi regăsite într-o altă bază de date care provine de la tagger-ul TTL, suplimentată cu numele proprii.

Algoritmii : Pentru căutarea iniţială există câte un algoritm specializat pentru fiecare bază de date (cea a indecşilor şi cea din TTL) care permite căutarea unui cuvânt în funcţie de particularităţile acesteia. În caz de eşec, este folosit un alt algoritm care apreciază lema unui cuvânt nou (negăsit până în acel punct), ţinând seama de terminaţia acestuia. În funcţie de modul de „rezolvare" a cuvântului, aplicaţia precizează dacă adnotarea sa este probabil bună sau probabil greşită.

Folosire : Aplicaţia poate genera un fişier în format text sau XML pentru vizualizarea adnotării. Scopul principal este de a ajuta lingvistul la adnotare. Astfel se poate genera o serie de fişiere XML care respectă structura interfeţei de adnotare (descrisa mai jos).

3. Interfaţa de adnotare

3.1 Necesitatea unei interfeţe

Iniţial ne-am gândit la realizarea unui program în limbajul Java, care să permită utilizatorului să realizeze modificări în text. Pentru o mai bună gestiune a muncii mai multor utilizatori (care nu se pot întâlni simultan), ne-am gândit la posibilitatea unei munci distribuite. De aceea am ales soluţia unei interfeţe Web:

`http://consilr.info.uaic.ro/~biblia/`

3.2 Realizarea aplicaţiei

Interfaţa de corectare a adnotărilor automate este o aplicaţie web, realizată în limbajul PHP şi foloseşte scripturi în limbajul Javascript.

Pentru afişarea capitolelor şi a versetelor conţinute se afişează arborele de directoare din contul utilizatorului (se folosesc funcţiile pentru lucrul cu fişiere folosite de PHP). Pentru un verset selectat, se citeşte conţinutul fişierului XML corespunzător acestuia şi se afişează informaţia într-un tabel. Pentru parsarea versetului s-a folosit un parser DOM.

Pentru alegerea unor caracteristici morfologice, s-a folosit un fişier XML (obţinut din prelucrarea setului de etichete din TnT) în care pentru fiecare parte de vorbire sunt specificate notaţiile ANA posibile (notaţii pentru analiza morfologică).

3.3 Salvarea datelor

Salvarea corecturilor realizate de către un grup de lingvişti presupune suprapunerea fişierului rezultat peste fişierul anterior. În fişierul XML rezultat se adaugă un atribut

pentru a specifica faptul că s-a realizat o modificare. Aceste atribute pot fi folosite pentru o compare a adnotării automate corectate cu o nouă adnotare automată.

3.4 Probleme şi soluții în realizarea interfeței de corectare

Problema 1: Am observat că *diacriticele* din text (codificate corespunzător în fişierul XML; ex.: *&#351* pentru *ş*) sunt afişate pe ecran dar, dacă se doreşte adăugarea altora, fişierul rezultat (cu atributul modificat) nu mai este valid, deoarece în fişier se încearcă scrierea unui caracter « necunoscut ». Încercarea de realiza conversia diacriticelor în notațiile corespunzătoare nu a funcționat (caracterul aşa cum apare pe ecran nu se putea regăsi, chiar printr-o operație simplă de copie-şi-lipeşte (copy-paste), în aceeaşi formă în fişierul PHP în care se face conversia).

Soluție: Pentru a rezolva această problemă s-a introdus în interfață un grup de butoane pentru introducerea de diacritice, corespunzătoare simbolurilor pe care fiecare din acestea le menționează; prin activarea acestor butoane se reține din start simbolul caracterului şi nu caracterul în sine. Această operație este realizată de un fragment de cod Javascript. În plus, pe măsură ce se adaugă diacritice, utilizatorul are şansa previzualizării formei pe care o propune pentru cuvântul de corectat.

Problema 2: Inexistența unor caracteristici morfologice între tagurile care descriu etichetele complexe MULTEXT(de exemplu: *genul neutru*, *verbele copulative*).

Soluție: S-au propus noi etichete, care să respecte întru totul structura etichetelor MULTEXT existente (Erjavec, 2001). Astfel, s-a putut permite alegerea genului neutru pentru substantive şi, respectiv, adjective.

De exemplu, s-a propus notația `afpnsryy` pentru un adjectiv calificativ, gradul pozitiv, singular, cazul direct (nominativ/acuzativ), articulat, clitic (pornind de la structura unei astfel de etichete) (Tufiş, Barbu, 1997):

```
<tag pos="adj." name="afpnsryy">
        <prop name="adj."/><prop name="q."/><prop name="poz."/>
        <prop name="neutru"/><prop name="sg."/><prop name="nom./acc."/>
        <prop name="+def."/><prop name="+clitic"/>
</tag>
```

Problema 3: Din etichetele MULTEXT nu se poate extrage *diateza*.

Soluție: Interfața permite alegerea diatezei, prin prezența unor butoane radio corespunzătoare celor trei diateze din limba română: activă, pasivă şi reflexivă.

Problema 4: *Marcarea relațiilor existente între cuvintele* care apar împreună într-o analiză gramaticală, respectiv a îmbinărilor stabile (*expresii* şi *locuțiuni*).

Soluție: S-a oferit posibilitatea alegerii unui nou atribut pentru cuvântul analizat la un moment dat. Astfel, se poate specifica dacă un cuvânt îl precede sau îl urmează prin selectarea unei opțiuni din interfață (de exemplu: cuvintele *va* şi *veni* sunt marcate pentru a fi analizate împreună).

3.5 Ghidul de utilizare al intefeței de corectare

S-au creat conturi pentru utilizatorii cu drepturi de corectare. În directoarele asociate fiecărui cont s-a introdus un număr de capitole (astfel încât fiecare utilizator să aibă fişiere distincte pentru corectare). Fiecare utilizator înregistrat are posibilitatea de a selecta capitolul şi versetul în care va face modificări.

În momentul selectării unui verset, se va afişa pe ecran textul acelui verset precum şi un tabel cu informații privitoare la cuvinte Pentru un cuvânt se poate modifica lema acestuia şi se pot schimba caracteristicile morfologice ale cuvântului (pentru aceasta se deschide o fereastră din care se poate selecta partea de vorbire şi apoi noua valoarea a atributului ANA – vezi figura 2).

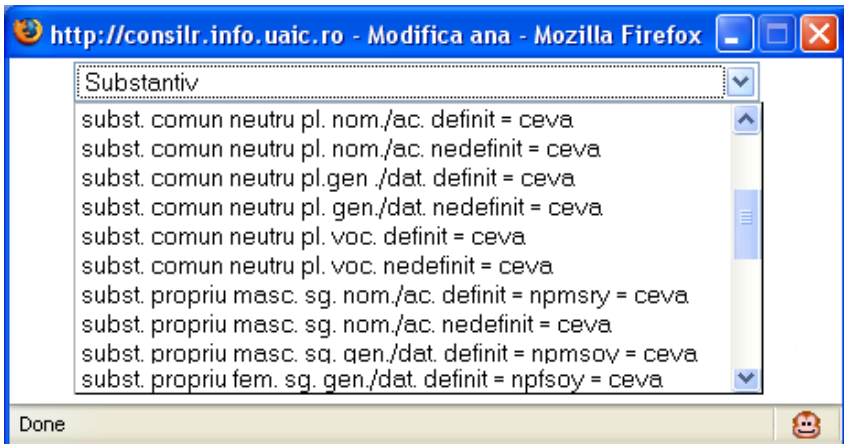


Figura 2: modificarea caracteristicilor morfologice pentru cuvântul selectat

Se pot salva modificările pentru cuvântul analizat sau se poate opta pentru atribuirea aceloraşi modificări tuturor cuvintelor identice cu acesta din verset (modificarea nu se poate propaga în afara fişierului pentru cuvinte identice întrucât nu se doreşte răspândirea unei posibile erori de adnotare).

În mod analog, pentru verbe se poate specifica diateza şi conjugarea, prin selectarea opțiunii corespunzătoare.

De asemenea, lingvistul are posibilitatea de a preciza ordinea cuvintelor care apar împreună într-o analiză gramaticală. De exemplu, pentru formele verbale compuse, se poate spune despre un auxiliar că este într-o relație de „precedență" cu verbul de bază. În cazul în care utilizatorul a specificat din greşeală o astfel de relație, el are posibilitatea ignorării acestui tip de adnotare pentru cuvântul în cauză.

În cazul grupurilor mai lungi, ca de exemplu: `va mai veni`, forma auxiliară `va` îl va precede pe `mai`, iar adverbul precede verbul principal. Ideea este stabilirea ulterioară a unei relații de « precedență » între auxiliar şi verbul de bază prin intermediul tranzitivității şi ulterior eliminarea adverbului din analiză.

Modificările făcute cuvintelor sunt marcate grafic printr-o linie orizontală care taie caracteristicile cuvântului salvat (Figura 3).

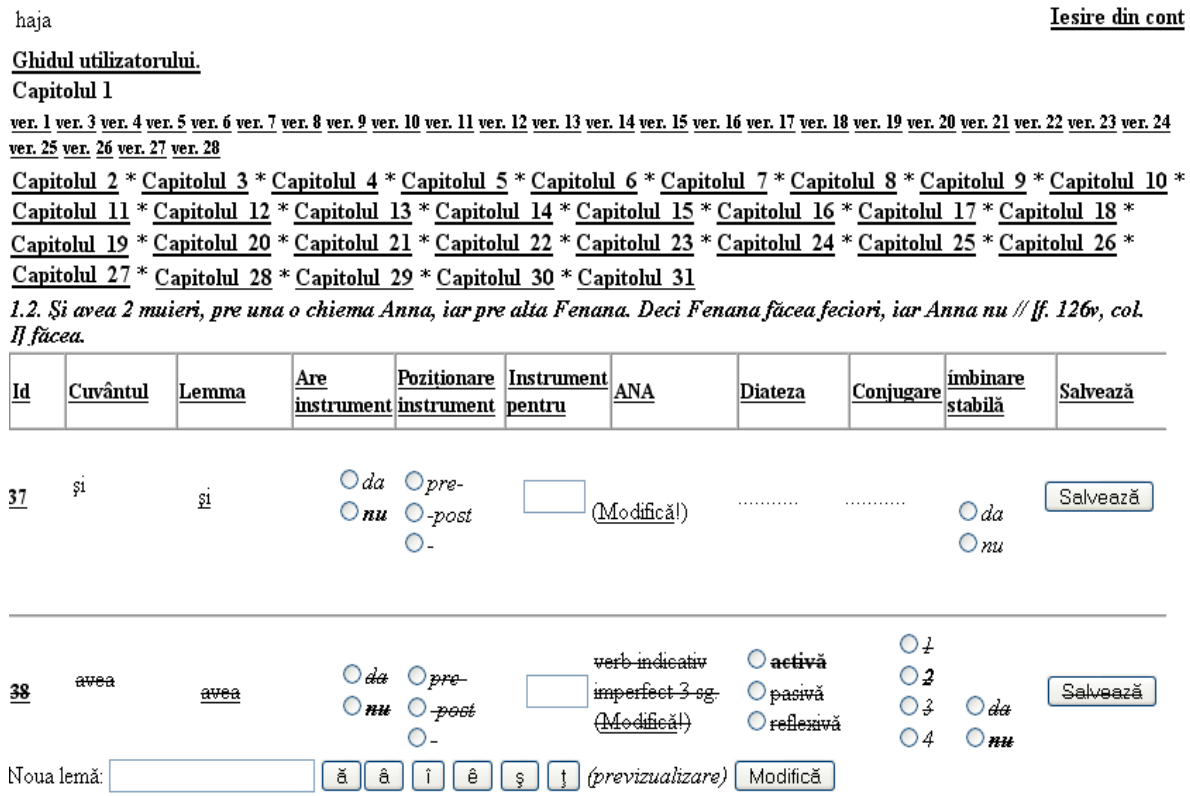


Figura 3: interfaţa de corectare (vizualizarea versetelor, editarea cu diacritice a lemei, specificarea diatezei, relaţionarea cuvintelor)

4. dexul şi interfaţa de vizualizare

Scopul adnotării *Bibliei* este dispunerea acesteia în format electronic după corectare. Rezultatul este într-un format optim pentru interfaţa web însă neprietenos unui utilizator. Interfaţa de vizualizare, aflată în lucru, va permite utilizatorilor să navigheze printre versete în funcţie de cuvântul selectat. În prealabil se face o indexarea a textului biblic pe baza adnotărilor morfologice realizate în etapa anterioară.

Problema: Gruparea după formă. O formă scrisă astfel : „ind.pr.3,sg.” reprezintă acelaşi lucru cu „ind. sg. pr.3” şi nu se pot compara şirurile de caractere ca atare.

Soluţie: Iniţial o mică filtrare a şirului de caractere. Pentru varianta de după corectare această problemă este eliminată din interfaţă, fiind generate formele sub o notaţie standard (etichete MULTEXT).

Detaliile lemelor : În dreptul fiecărei leme vor fi afişate, potrivit tradiţiei impuse de ediţia *Monumenta linguae Dacoromanorum*, *Biblia 1688*, traducerile în germană şi franceză, partea de vorbire, precum şi prima atestare scrisă a cuvântului. Aceste informaţii se caută similar cu adnotarea. Însă varianta corectată va conţine şi leme noi. De aceea, s-a realizat un mic program ce extrage lemele cărora nu li se cunosc detaliile, într-un fişier. Fişierul ce rezultă va fi completat de către lingvişti.

Interfaţa :

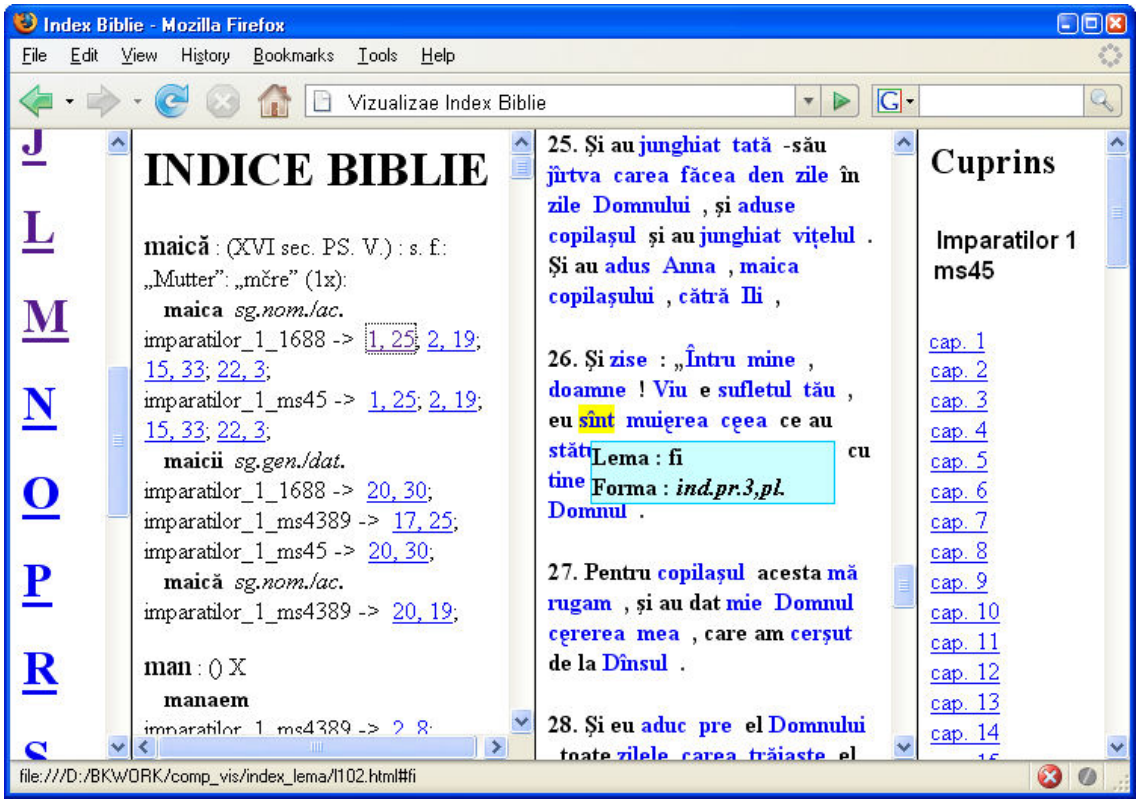


Figura 4: Interfața de vizualizare

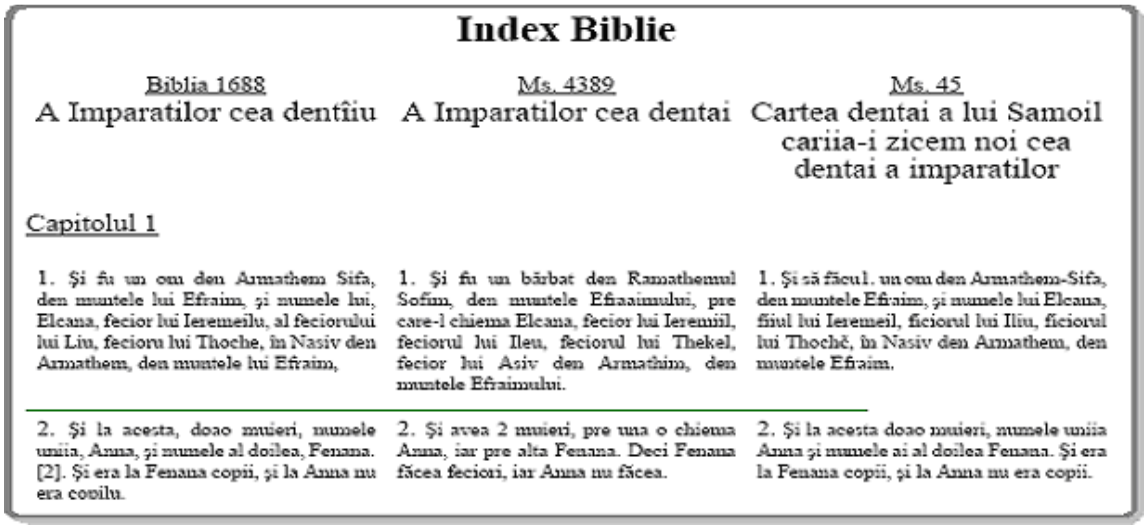
Prima coloană se foloseşte pentru a selecta litera inițială a lemelor. În a doua coloană se găseşte lista lemelor care încep cu litera selectată. Pentru fiecare lemă sunt listate formele flexionare, iar pentru fiecare inflexiune sunt enumerate aparițiile grupate după locul în care se află (variantă de traducere, capitol, verset). Fiecare apariție este un link ce repoziționează textul din coloana a 3-a, ce conține textul propriu-zis al unui capitol. În acest text, cuvintele adnotate sunt şi ele link-uri ce repoziționeaza lista din a doua coloană pe lema cuvântului selectat. Totodată, pentru cuvântul adnotat de sub cursorul mouse-ului sunt afişate lema şi forma. Textul din coloana a 3-a poate fi repoziționat şi folosind coloana a 4-a, selectând un capitol al unei cărți.

Realizarea interfeței : Rezultatul indexorului constă în 2 fişiere, unul cu textul *Bibliei* adnotate şi unul cu *Indexul* în sine. Pentru a transforma aceste fişiere în paginile HTML ce pot fi vizualizate cu un browser, se folosesc transformări XSLT (Extensible Stylesheet Language Transformations) ce reprezintă un limbaj bazat pe XML anume pentru astfel de scopuri. În realizare, s-au întâmpinat diverse probleme legate de limbaj, fiind unul funcțional, iar procesările cu nuanța imperativă a trebuit să fie re-gândite.

Problema: Dimensiunea datelor. Timpul necesar acestor procesări e destul de mare, la fel şi consumul de memorie.

Soluție: Inițial datele au fost preprocesate si erau distribuite paginile HTML. Însă si vizualizarea consuma multe resurse. Soluția finală rezolvă în totalitate problema şi constă în restructurarea transformărilor pentru a împărți vizualizarea în mai multe fişiere, după litera inițială si numărul capitolului la leme respectiv la textul *Bibliei*. În acest mod, vizualizarea poate funcționa şi ca formă publicată on-line, deoarece se încărca în memorie numai porțiunea necesară.

De asemenea, se doreşte oferirea posibilităţii de a vizualiza textul respectiv în forma în care este prezent în cele trei variante de traducere. Acesta va fi şi formatul final împreună cu versiunea în scriere chirilică precum şi cu o versiune actualizată a traducerii. Până atunci, modificând transformările, se pot genera documente FO (Formatting Objects) din care apoi rezultă fişiere RTF sau PDF printabile (figura 5).



Index Biblie

Biblia 1688	Ms. 4389	Ms. 45
A Imparatilor cea dentîiu	A Imparatilor cea dentai	Cartea dentai a lui Samoil cariia-i zicem noi cea dentai a imparatilor
Capitolul 1		
1. Şi fu un om den Armathem Sifa, den muntele lui Efraim, şi numele lui, Elcana, fecior lui Ieremeilu, al feciorului lui Liu, fecioru lui Thoche, în Nasiv den Armathem, den muntele lui Efraim,	1. Şi fu un bărbat den Ramathemul Sofim, den muntele Efraimului, pre care-l chiema Elcana, fecior lui Ieremiil, feciorul lui Ileu, feciorul lui Thekel, fecior lui Asiv den Armathim, den muntele Efraimului.	1. Şi să făcu1. un om den Armathem-Sifa, den muntele Efraim, şi numele lui Elcana, fiiul lui Ieremeil, ficiorul lui Iliu, ficiorul lui Thochĕ, în Nasiv den Armathem, den muntele Efraim.
2. Şi la acesta, doao muieri, numele uniia, Anna, şi numele al doilea, Fenana. [2]. Şi era la Fenana copii, şi la Anna nu era covilu.	2. Şi avea 2 muieri, pre una o chiema Anna, iar pre alta Fenana. Deci Fenana făcea feciori, iar Anna nu făcea.	2. Şi la acesta doao muieri, numele uniia Anna şi numele ai al doilea Fenana. Şi era la Fenana copii, şi la Anna nu era copii.

Figura 5 : Fragment din fisier PDF generat cu XSL-FO

5. Concluzii

Rezultatele obţinute în cadrul grantului *Monumenta linguae Dacoromanorum. Biblia 1688. Pars VII. Regum I, Regum II – ediţie critică şi corpus adnotat* reprezintă un punct de plecare pentru cercetări viitoare în domeniul procesării textelor româneşti vechi. Un prim obiectiv viitor este constituit de realizarea integrală a monumentalei ediţii a *Bibliei* de la 1688 în format electronic, proiectată în douăzeci de volume. Pentru aceasta este necesară achiziţionarea în acest format a volumelor editate deja (*Pentateuh, Iisus Navi, Judecătorii, Ruth* şi *Psaltirea*) şi utilizarea experienţei noastre (B. Aldea, G. Haja, 2006) în editarea volumelor viitoare. Dar, chiar înainte de finalizarea ediţiei, sunt deja posibile o multitudine de aplicaţii computaţionale şi cercetări lingvistice pe baza resursei create deja (cum este de exemplu determinarea paradigmelor morfologice de limba veche).

Prin proiectul nostru s-au creat câteva instrumente necesare prelucrării automate şi semi-automate a textului românesc din secolul al XVII-lea. Aceste instrumente vor putea fi utile în prelucrarea rafinată a resurselor lingvistice în format electronic pentru limba română, resurse create de specialişti lingvişti şi informaticieni, în cadrul altor proiecte, dintre care le amintim aici pe acelea în care sunt implicaţi şi unii dintre autorii acestei lucrări: proiectul PC finanţat prin CNMP, *eDTLR – Dicţionarul Tezaur al Limbii Române în format electronic* şi proiectul CNCSIS *Corpus de referinţă pentru limba română*.

Mulţumiri. Autorii mulţumesc Ministerului Educaţiei, Cercetării şi Tineretului care, prin intermediul Consiliului Naţional al Cercetării Ştiinţifice din Învăţământul Superior (CNCSIS), a susţinut financiar realizarea proiectului *Resurse lingvistice în format electronic. Monumenta linguae Dacoromanorum. Biblia 1688. Pars VII. Regum I,*

Regum II – ediție critică şi corpus adnotat. De asemenea, mulțumesc Institutului de Cercetări pentru Inteligență Artificială (ICIA) din Bucureşti, pentru sprijinul acordat în procesarea textelor.

Referințe bibliografice

Tomaz Erjavec et al. (2001). *Specification and Notation for MULTEXT-East Lexicon Encoding*, Edition Multext-East / Concede. March 21th.

D.Tufis, A.M. Barbu, A. (1997). *Reversible and Reusable Morpho-Lexical Description of Romanian*, în "Recent Advances in Romanian Language Technology", eds. Dan Tufiş, Poul Andersen, Ed. Academiei Române.

Thorsten Brants. (2000). *TnT - A Statistical Part-of-Speech Tagger*, în *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA.

B. Aldea, G. Haja. (2006). Resurse lingvistice româneşti în format electronic. Biblia 1688, în *Lucrările atelierului Resurse Lingvistice şi Intrument pentru prelucrarea limbii române*, Iaşi, noiembrie.

SERVICIII WEB LINGVISTICE ALE ICIA

DAN TUFIŞ, RADU ION, ALEXANDRU CEAUŞU, DAN ŞTEFĂNESCU

Institutul de Cercetări pentru Inteligență Artificială

Str. 13 Septembrie, nr. 13, Bucureşti 050711, România

{tufis, radu, aceausu, danstef}@racai.ro

Rezumat

Lucrarea de față trece în revistă o serie de servicii web destinate Prelucrării Automate a Limbajului Natural (PLN), implementate la Institutul de Cercetări pentru Inteligență Artificială al Academiei Române. Serviciile web sunt destinate incorporării lor directe în aplicații de PLN ca API-uri pentru preprocesarea textelor. Accentul este pus de preprocesarea textelor în limba română fără a elimina totuşi posibilitatea prelucrării textelor în alte limbi (prelucrarea pentru limba engleză este de asemenea asigurată).

1. Introducere

Tehnologiile Web Service, un palier fundamental în filosofia Semantic Web, înlesnesc utilizatorilor dezvoltarea de aplicații ce integrează diverse funcționalități aflate pe alte maşini decât cea locală. Web-ul semantic a apărut dintr-o nevoie imperioasă de a structura şi standardiza colecția eterogenă şi vastă de documente care sunt accesibile pe World Wide Web, în scopul de a facilitata identificarea şi procesarea lor ulterioară din perspectivă semantică. Algoritmii PLN actuali – precum cei de segmentare a textului, de adnotare morfo-lexicală, lematizare, parsing, dezambiguizare semantică, procesare a discursului, ş.a.m.d - capabili să proceseze o cantitate mare de informație pot, dacă sunt dezvoltați ca servicii web, să furnizeze mijloacele necesare pentru adnotarea automată a paginilor HTML cu informație utilă oricărui agent software care doreşte să extragă informație din acestea. Alături de algoritmii de bază pentru procesarea limbajului natural există aplicații de nivel mai înalt, precum sistemele întrebare-răspuns pe domeniu deschis sau sistemele de traducere automată, care pavează drumul către cele mai ambițioase scopuri ale Web-ului Semantic: acces bazat pe cunoştințe la bogăția de informație digitală de pe web printr-o interacțiune naturală fără bariere lingvistice.

În concordanță cu tendința generală promovată de ideologia Semantic Web şi cu cererea în creştere pentru pre-procesarea limbii române, am implementat o platformă de servicii web care permite procesarea multi-lingvală a unor texte arbitrare. În acest moment, în afară de o aplicație care identifică 22 de limbi, cele mai multe servicii oferite de platforma noastră sunt dedicate limbii române (toate serviciile) şi limbii engleze (segmentarea la nivel de propoziție, segmentarea la nivel de cuvânt, adnotarea morfo-lexicală, lematizarea şi analiza legăturilor de dependență). Folosirea SOAP (Simple Object Acces Protocol) pentru comunicare, a WSDL (Web Service Definition Language) pentru descrierea serviciilor şi a UDDI (Universal Description, Discovery, and Integration) pentru înregistrare, asigură paşii de pre-procesare, care sunt esențiali în orice încercare serioasă de dezvoltare a unor aplicații PLN complexe: identificarea

limbajului, segmentarea la nivel de propoziție şi cea lexicală, dezambiguizare morfo-sintactică, analiză sintactică de suprafață (chunking), analiză de legături de dependență şi adnotare automată XML.

Pe măsură ce aplicații noi sunt dezvoltate şi aduse la un stadiu de maturitate, ele sunt codificate ca noi servicii web şi introduse în platforma de servicii Web a ICIA.

2. *Servicii Web lingvistice*

În continuare furnizăm o scurtă descriere a serviciilor pe care platforma noastră de servicii web le oferă în acest moment. Toate aplicațiile descrise în acest capitol, există ca aplicații de sine stătătoare. În prezent, serviciile de identificare a limbii şi adnotare morfo-lexicală sunt accesibile la http://nlp.racai.ro/webservices, dar, în perspectivă, toate serviciile web ale ICIA prezente şi viitoare vor fi accesibile la adresa http://nlp.racai.ro.

2.1 Identificarea limbii

Acest serviciu asigură identificarea automată a limbii unui text scris într-una dintre cele 22 de limbi ale Uniunii Europene. Textul ar trebui să conțină un număr minim de 10-15 cuvinte (în principiu, o propoziție). Am utilizat cu succes această aplicație pentru curățarea corpusului paralel JRC-Acquis (Steinberger et al., 2006) care, în părți specifice unei anumite limbi, include în mod accidental propoziții şi paragrafe din alte limbi. Modulul de Identificare a Limbii este util pentru colectarea textelor de pe Web sau în medii de procesare multi-linguale şi parametrizabile unde instrumentele, modelele şi parametrii potriviți pot fi selectați automat în funcție de limba textului sursă. De exemplu, adnotarea morfo-lexicală are nevoie de informații despre limba pe care o adnotează, deoarece fiecare model de adnotare este construit pentru o anumită limbă. O altă aplicație pe care o avem în vedere este Traducea Automată, în care modelele de traducere din limba sursă în limba țintă depind de perechea de limbi implicate.

Există o multitudine de procedee prin care se poate identifica automat limba unui text. Soluția noastră este una statistică şi are două module: unul de antrenare, iar celălalt de predicție. Modulul de antrenare necesită texte de antrenament în fiecare din limbile pe care dorim ca aplicația să le recunoască. Se realizează câte un model pentru fiecare din aceste limbi pe baza ponderii pe care o au prefixele şi sufixele cuvintelor din textul corespunzător limbii în totalul prefixelor şi sufixelor din acel text. Modulul de predicție realizează un astfel de model pentru un text nevăzut şi-l compară apoi cu cele deja existente. Se calculează un scor de similaritate pentru fiecare din perechile posibile. Identificatorul limbii pentru care modelul are cel mai mare scor de similaritate cu textul analizat este comunicat ca rezultat al predicției.

În experimentele realizate până acum, am utilizat texte de antrenament (cu mărimi ce au variat între 0,5 şi 1,2 Mb) pentru cele 22 limbi oficiale ale Acquis-ului Communautaire. Textele, fiind însă din domeniul juridic şi având o structură mai aparte[2], nu sunt tocmai reprezentative pentru limbile luate în discuție. Cu toate acestea, am obținut rezultate excelente folosind pentru prefixe o lungime de trei caractere iar pentru sufixe de patru.

[2] Textele juridice au adesea o structură formată din multe aliniate în care anumiți termeni se repetă de foarte multe ori afectând acoperirea lingvistică a modelelor de limbă.

Obiectivul pe care îl avem în vedere în continuare în legătură cu această problemă, este utilizarea unor texte mai mari de antrenament pentru ca ponderile calculate în cadrul modelelor obținute să fie cât mai specifice limbilor şi, nu în ultimul rând, rafinarea parametrilor modelului în direcția îmbunătățirii calității clasificării.

2.2 Adnotarea morfo-lexicală şi lematizarea

Adnotarea morfo-lexicală se face cu o mare acuratețe (în jur de 98%) atât pentru limba română cât şi pentru limba engleză. Există două implementări diferite pentru sarcina de adnotare morfo-lexicală: una se bazează pe paradigma HMM (Hidden Markov Models – Modele Markov Ascunse) iar cealaltă foloseşte abordarea Maximum Entropy (Entropie Maximă).

Prima, denumită TTL (Ion, 2007), este un tagger HMM cu 3-grame care implementează TnT-ul lui Brants (2002) şi îl extinde cu câteva trăsături suplimentare:

o euristică de adnotare a substantivelor proprii cu etichetele corespunzătoare dacă acestea nu apar în lexiconul taggerului, încep cu majusculă şi se află la început de propoziție. Dacă textul este din registrul jurnalistic de exemplu (şi conține astfel multe substantive proprii), această euristică salvează multe adnotări greşite datorită faptului că nu se mai calculează clasa de ambiguitate pentru cuvântul respectiv;

expandarea clasei de ambiguitate a unui cuvânt necunoscut numai la etichetele morfo-lexicale ale cuvintelor conținut (eng.: open class). Această măsură asigură reducerea claselor de ambiguitate pentru cuvintele conținut, în baza ipotezei că toate cuvintele funcționale (prepoziții, conjuncții, articole, pronume) se află în lexiconul taggerului *cu clasele lor de ambiguitate complete* şi clasele de ambiguitate ale cuvintelor funcționale nu se suprapun peste cele ale cuvintelor conținut. Implementarea inițială a TnT-ului nu făcea această distincție şi astfel, de exemplu, eticheta morfo-lexicală a unui substantiv care nu era în lexiconul taggerului (deci un cuvânt necunoscut) era aleasă din întreaga mulțime de etichete morfo-lexicale fiind astfel evaluate şi opțiunile de a fi de pildă prepoziție sau articol;

o euristică de adnotare uniformă a unor entități frecvente cum ar fi numerele întregi, reale, procentele, abrevierile etc. pe care TnT-ul le adnota inconsistent greşind astfel în cazuri în care nu există nici o ambiguitate.

TTL lucrează pe limbile română şi engleză folosind abordarea adnotare-stratificată (Tufiş, 1999, 2000) şi diferite seturi de etichete (tagset-uri): tagset-ul lexical compatibil cu specificațiile Multext-East (Erjavec, 2004), cu 614 etichete pentru română şi 133 etichete pentru engleză şi un tagset redus (potrivit modelului adnotării stratificate: 92 etichete pentru română şi 95 etichete pentru engleză). Adnotarea stratificată (eng. tiered-tagging) este o tehnică în doi paşi care adresează problema insuficienței datelor statistice (eng.: data sparseness): (i) adnotare intermediară folosind un tagset redus (CTAG-set), (ii) înlocuirea CTAG-urilor cu etichete MSD adecvate contextului (etapa denumită „recuperarea MSD-urilor" în (Tufiş 1999)). Lexiconul, care se află la baza abordării adnotării stratificate, conține cuvinte adnotate cu etichete MSD iar o intrare în acest lexicon are forma: <cuvânt> <lemă> <msd>. Pentru limba română, acest lexicon conține aproape 800,000 de intrări, în timp ce pentru limba engleză conține în jur de 135,000 de intrări.

Cel de-al doilea adnotator, denumit METT - Maximum Entropy Tiered Tagging (Ceauşu, 2007) este conform modelului ME al lui Ratnaparkhi (1988) şi are modele de limbă de mare acurateţe pentru română şi engleză. Ca şi TTL, tagger-ul METT poate utiliza tagset-ul compatibil cu specificaţiile Multex-East şi tagset-ul redus pentru adnotarea stratificată (Tufiş & Dragomirescu, 2004) însă, spre deosebire de TTL, METT nu utilizează reguli explicite de eliminare a ambiguităţilor de MSD atunci când un CTAG corespunde mai multor MSD-uri posibile ci le învaţă automat din texte adnotate cu MSD-uri folosind algoritmul de maximizare a entropiei (EM). Astfel, METT poate să adnoteze direct cu etichete CTAG sau MSD şi să facă adnotare stratificată cu sau fără un lexicon cu MSD-uri (în tabelul 1 se prezintă atributele contextuale împreună cu valorile lor pentru fiecare stil de adnotare al METT).

De asemenea, în cazul cuvintelor *necunoscute*, s-a înlocuit recuperarea clasică a etichetelor MSD din cadrul adnotării stratificate cu un alt tip de recuperare, care se bazează pe EM. În cadrul acestei abordări, regulile de conversie de la CTAG la MSD sunt învăţate automat din corpus, în aplicarea lor nemaifiind necesară căutarea în lexiconul ce conţine etichete MSD. În acest mod, etichetele CTAG atribuite cuvintelor necunoscute pot fi convertite în etichete MSD. Dacă pentru cuvintele înregistrate în lexiconul modelului statistic (HMM ori EM), recuperarea etichetelor MSD din etichetele CTAG are o acurateţe de aproape 100%, pentru cuvintele necunoscute, estimarea acurateţei recuperării este de 95,2%. Mai mult, modelul EM pentru conversia CTAG-MSD poate ignora etichetarea iniţială CTAG pentru cuvintele necunoscute furnizând direct o etichetă MSD potrivită contextului. În acest fel unele cuvinte, care ar fi fost greşit etichetate cu CTAG, pot fi corect etichetate cu MSD.

Tabelul 1: Predicate contextuale

	Tagger CTAG	**Tagger MSD**	**Convertorul de etichete**
Formă cuvânt	X	X	
lungime (în caractere)	X	X	X
prefix (1 – 2 caractere)	X	X	X
sufix (1 – 4 caractere)	X	X	X
capitalizare (toate caracterele, doar cele iniţiale)	X	X	X
este sau nu abreviere	X	X	X
conţine o linie de subliniere	X	X	X
conţine un număr	X	X	X
poziţia cratimei (iniţială, inclusă, finală)	X	X	X
trăsături MSD anterioare		X	X
etichete CTAG anterioare	X		X
punctuaţia finală a propoziţiei	X	X	X

O observaţie interesantă este că cele două taggere, TTL şi METT, au performanţe similare dar nu fac aceleaşi erori. În consecinţă, o opţiune naturală pentru îmbunătăţirea calităţii serviciului de adnotare morfo-lexicală este combinarea rezultatelor lor aşa cum se arată în (Tufiş, 2000). Există mai multe tehnici de combinare a rezultatelor a două adnotatoare morfo-lexicale complementare, una dintre cele mai performante fiind metoda *credibilităţii* (Tufiş, 2000), bazată pe matricele de confuzie ale modelelor de limbă ale celor două adnotatoare.

Lematizarea (algoritm implementat de TTL) se face ulterior adnotării morfo-lexicale, datorită ambiguității de categorie gramaticală, care poate conduce la ambiguitatea lemei - în funcție de categoria gramaticală, aceleiaşi forme îi pot corespunde leme diferite. Dacă forma unui cuvânt este cunoscută modelului de limbă, lematizarea este un simplu proces de căutare în lexiconul de forme. Atât în engleză cât şi în română, perechea <cuvânt, msd> identifică, aproape întotdeauna, unic lema (lema este cel de-al treilea element al unei intrări în lexicon). În situațiile rare când identificarea unică nu are loc, cea mai frecventă lemă este selectată în mod automat.

Lematizarea cuvintelor necunoscute este un proces statistic, bazat pe reguli induse din lexicoane. Lema pentru un cuvânt necunoscut este aleasă dintr-un set de leme candidat generate cu aceste reguli. Mecanismul de selecție este bazat pe un Model Markov care a fost antrenat pe leme cu aceeaşi etichetă morfo-sintactică. Acuratețea testată a acestei metode de lematizare este de aproximativ 83% pentru cuvinte necunoscute, atât pentru engleză cât şi pentru română. Considerând, în mod acoperitor, un procent de circa 15% cuvinte necunoscute[3] într-un text arbitrar nou în limba română, erorile de lematizare vor fi sub 1,5% (pentru limba engleză, procentul este chiar mai mic).

2.3 LexPar

LexPar (Ion, 2007) este un analizor de legături bazat pe reguli. Este o extensie firească a algoritmului lui Yuret (1998) care constrânge formarea de legături cu reguli sintactice specifice limbii textului procesat. În plus conține şi un mecanism simplu de generalizare a proprietăților unei legături pentru a elimina inabilitatea algoritmului inițial de a trata cuvintele necunoscute. Principalele diferențe între procesorul lui Yuret şi LexPar sunt:

LexPar rulează pe texte adnotate morfo-sintactic şi lematizate. Lematizarea oferă un prim nivel de generalizare pentru forma ocurentă a cuvântului contribuind la estimări mai bune ale parametrilor modelului.

LexPar calculează scorul unei legături considerând simultan lemele cuvintelor legate cât şi etichetele lor morfo-sintactice. În cazul în care una din leme nu a fost întâlnită la antrenare, scorul legăturii este dat de perechea de etichete morfo-sintactice a cărei apariție este mult mai probabilă decât cea a perechii de leme. Împreună cu lematizarea, luarea în calcul a etichetelor morfo-sintactice ale cuvintelor în formarea unei legături reprezintă principalul mecanism de generalizare al lui LexPar în calculul scorurilor legăturilor între cuvintele necunoscute.

Ca şi în algoritmul lui Yuret, LexPar ia în calcul o legătură care nu produce un ciclu şi care nu încalcă proprietatea de planaritate dar în plus, LexPar nu consideră legătura care este respinsă de filtrul său sintactic[4]. Această filtrare are rolul de a grăbi convergența procesului de antrenament către modelul de atracție lexicală care aproximează structura de dependențe a limbii date. În plus, perechile care nu pot fi relaționate sintactic nu încarcă inutil memoria procesorului.

[3] În prelucrările noastre curente ale textelor în limba română, numărul mediu de cuvinte necunoscute este sub 5%, astfel că erorile de lematizare, sub 0,5% sunt neglijabile. Acest procent scade în mod constant prin creşterea continuă a acoperirii lexicale a lexiconului statistic, a se vedea lucrarea (Tufiş et al., 2007).

[4] Prezența filtrului sintactic nu mai garantează o structură de graf conex a analizei de legături.

Algoritmul LexPar consideră o altă ordine de procesare a cuvintelor unei fraze decât scanarea de la stânga la dreapta. Principala presupunere pe care o face este aceea că cele mai multe legături se stabilesc între cuvinte adiacente iar apoi între grupuri adiacente de cuvinte legate. LexPar construieşte progresiv structura de legături a unei fraze, alcătuind grupuri de cuvinte legate de dimensiuni din ce în ce mai mari.

Serviciul web LexPar oferă deci o analiză a legăturilor de dependență pe o propoziție lematizată şi adnotată morfo-sintactic, determinând structura unui graf planar, aciclic şi conex al propoziției. Algoritmul LexPar (Ion, 2007; Ion & Tufiş, 2007) implementează un model CLAM (eng. Constrained Lexical Attraction Model) care este o rafinare a Modelului de Atracție Lexicală a lui Yuret (1988). Foloseşte reguli sintactice specifice limbii procesate pentru a reduce spațiul de căutare şi pentru a elimina legăturile improbabile. Deocamdată lucrează pe limbile română şi engleză dar, fără filtru sintactic (care este dependent de limbă), poate fi aplicat oricărui text lematizat şi adnotat morfo-lexical.

2.4 XCESGen

Acest serviciu garantează codificarea de corpusuri paralele în format XCES pornind de la texte neprelucrate. Foloseşte serviciile menționate mai sus şi produce următoarele marcaje:

- Adnotarea cu legături de dependență;
- Adnotarea de suprafață: grupuri de cuvinte adiacente, dependente sintactic, sunt marcate şi denumite: grupuri nominale, grupuri verbale, grupuri prepoziționale, etc.;
- Adnotarea lemelor;
- Adnotarea morfo-sintactică;
- Segmentarea textului la nivel de frază şi unitate lexicală;
- Recunoaşterea unor entități textuale cum ar fi numerele întregi, reale, abrevierile, unele nume de persoane, cantități, date, sume de bani etc.

XCESGen a fost incorporat în TTL şi pentru fiecare nivel de procesare (segmentarea textului, adnotare morfo-sintactică etc.) există un nivel de codificare XML. LexPar prelucrează de asemenea un text în format XML adnotat la nivel de etichete morfo-sintactice şi leme şi întoarce acelaşi fişier XML cu informație despre perechile de cuvinte ale unei propoziții care se leagă. În Figura 1 se exemplifică un fragment din corpusul paralel SemCor2.0 (Ion, 2007) codificat în format XML.

În această figură fiecare unitate lexicală (codificată cu eticheta `w`) are atributele `lemma`, `ana`, `chunk` şi `head` care desemnează respectiv lema, codificarea analizei morfo-sintactice a unității lexicale, grupul sintactic din care aceasta face parte (absența atributului `chunk` semnifică faptul că unitatea lexicală nu face parte din vreun grup sintactic recunoscut) şi perechea de legătură a unității lexicale (din nou, dacă atributul `head` lipseşte, această unitate lexicală nu a fost inclusă de LexPar în structura de legături a propoziției curente).

```
- <tu id="50">
  - <seg lang="en">
    - <s id="br-a01.44.50.en">
        <w lemma="Caldwell" ana="8+,Np" chunk="Np#1" wns="ili:ENG20-00006026-n">Caldwell</w>
        <w lemma="'s" ana="21+,St" chunk="Np#1" head="0">'s</w>
        <w lemma="resignation" ana="1+,Ncns" chunk="Np#1" wns="ili:ENG20-06109386-n" head="0">resignation</w>
        <w lemma="have" ana="3+,Vais" chunk="Vp#1" head="5">had</w>
        <w lemma="be" ana="3+,Vaps" chunk="Vp#1" head="5">been</w>
        <w lemma="expect" ana="1+,Vmps" chunk="Vp#1,Ap#1" wns="ili:ENG20-00695861-v" head="2">expected</w>
        <w lemma="for" ana="5+,Sp" chunk="Pp#1" head="8">for</w>
        <w lemma="some" ana="22+,Di3" chunk="Pp#1,Np#2" head="8">some</w>
        <w lemma="time" ana="1+,Ncns" chunk="Pp#1,Np#2" wns="ili:ENG20-14265546-n" head="5">time</w>
        <c>.</c>
      </s>
    </seg>
  - <seg lang="ro">
    - <s id="br-a01.44.50.ro">
        <w lemma="demisie" ana="1+,Ncfsry" chunk="Np#1">Demisia</w>
        <w lemma="lui" ana="21+,Tf-so" chunk="Np#1" head="2">lui</w>
        <w lemma="Caldwell" ana="8+,Np" chunk="Np#1" wns="ili:ENG20-00006026-n" head="0">Caldwell</w>
        <w lemma="fi" ana="3+,Vail3s" chunk="Vp#1" head="4">fusese</w>
        <w lemma="aştepta" ana="1+,Vmp--sf" chunk="Vp#1,Ap#1" wns="ili:ENG20-00695861-v" head="0">aşteptată</w>
        <w lemma="de" ana="5+,Spsa" chunk="Pp#1" head="7">de</w>
        <w lemma="ceva" ana="22+,Di3-sr---e" chunk="Pp#1,Np#2" head="7">ceva</w>
        <w lemma="timp" ana="1+,Ncms-n" chunk="Pp#1,Np#2" wns="ili:ENG20-14265546-n" head="4">timp</w>
        <c>.</c>
      </s>
    </seg>
  </tu>
```

Figura 1: Un exemplu de codificare în format XCES a corpusului paralel englez-român SemCor 2.0.

2.5 Alte servicii web

Browser-ul web cu grafuri hiperbolice oferă utilizatorilor acces la conținutul celei mai mari ontologii lexicale pentru limba română: Ro-Wordnet (Tufiş et al., 2008). Acelaşi browser poate fi utilizat pentru wordnet-ul public de referință, Princeton Wordnet 2.0. În acest moment, serviciul permite doar browsing, dar plănuim să adăugăm facilități de dezvoltare, precum: identificarea seriei sau seriilor sinonimice (sinset) din care face parte un cuvânt dat (fie în română, fie în engleză), găsirea unei distanțe semantice între sinseturi arbitrare (atât monolingual cât şi croslingual, via indexul interlingual), identificarea de echivalenți de traducere pentru un sens dat, eticheta SUMO, Domain sau adnotarea subiectivității.

Un alt serviciu web deosebit de util este DIAC^{+}**.** Acesta este un serviciu care permite recuperarea automată a diacriticelor în texte în limba română scrise fără – sau scrise doar parțial - cu caractere diacritice. DIAC^{+} utilizează instrumentele de pre-procesare descrise mai sus şi un lexicon care conține un număr foarte mare de forme. Pentru limba română, recuperarea automată a diacriticelor este o adevărată provocare, atât datorită frecvenței lor (fiecare al treilea cuvânt poate conține cel puțin un caracter diacritic) cât şi datorită contribuției semnificative pe care o au la dezambiguizarea morfo-sintactică şi semantică a cuvintelor. DIAC^{+} este de asemenea disponibil şi ca o aplicație de sine stătătoare, în forma unui DLL pentru MSOffice.

3. Concluzii

Pe lângă cele descrise mai sus, există alte câteva instrumente de procesare a limbajului (un extractor de colocații, un extractor al structurii predicative, un aliniator la nivel de propoziție pentru corpusuri paralele, un motor de căutare avansată şi un sistem întrebare-răspuns pentru limba română) care sunt deja implementate ca aplicații de sine stătătoare şi pe care intenționăm să le includem în platforma de servicii web.

Accesul la serviciile web este pe bază de licenţă şi a fost utilizat deja de diverşi cercetători din Bulgaria, Canada, Danemarca, Franţa, Italia, Olanda, România şi SUA pentru procesarea de texte în limba română totalizând mai mult de 2 milioane de cuvinte.

Referinţe bibliografice

Brants, T. (2000). TnT – A Statistical Part-Of-Speech Tagger. In Proceedings of the 6th Applied NLP Conference, ANLP-2000, pages 224–231, Seattle, WA, April 29 – May 3, 2000

Ceauşu, Al. (2006). Maximum Entropy Tiered Tagging. In Janneke Huitink & Sophia Katrenko (editors), Proceedings of the Eleventh ESSLLI Student Session, ESSLLI 2006, pp. 173-179

Erjavec, Tomasz (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA, Paris

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. (in Romanian). PhD thesis. Romanian Academy, Bucharest, 2007

Ion, R., Tufiş, D. (2007). Meaning Affinity Models. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), Prague, June 2007, Association for Computational Linguistics, pp. 282–287

Ratnaparkhi, A. (1998). Maximum Entropy Models for Natural Language Ambiguity Resolution. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006). Genoa, Italy, 24-26 May 2006.

Tufiş, D., Ion, R., Bozianu, L., Ceauşu, Al., Ştefănescu, D. (2008). Romanian Wordnet: Current State, New Applications and Prospects. În Proceedings of the 4th Global WordNet Conference, Szeged, Hungary, 22-25 January, 2008.

Tufiş, D. (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. International Conference on Language Resources and Evaluation LREC'2000, Athens, pp. 1105-1112

Tufiş, D. (1999). Tiered Tagging and Combined Classifiers. In F. Jelinek, E. Nth (eds) Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence, Springer, pp. 28-33

Tufiş, D., Ion, R., Irimia, E., Ceauşu, Al. (2007). Achiziţie lexicală nesupervizată pentru adnotare morfo-lexicală. În acest volum.

Tufiş, D., Ion, R., Ceauşu, Al., Ştefănescu D. (2006). - Improved Lexical Alignment by Combining Multiple Reified Alignments. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL2006), Trento, Italy, 3-7 April, 2006, pp. 153-160.

Tufiş, D., Dragomirescu L. (2004) - Tiered Tagging Revisited. In Proceedings of the 4th LREC Conference, Lisabona, 2004, pp. 39-42.

Yuret. D. (1998). Discovery of linguistic relations using lexical attraction. Ph.D. thesis, Department of Computer Science and Electrical Engineering, MIT, May, 1988.

DESPRE FORMATUL ELECTRONIC AL DILR

CECILIA CĂPĂȚÎNĂ, ANAMARIA PREDA, VLAD PREDA

Universitatea din Craiova, Facultatea de Litere, România

cecilia_capa@yahoo.com

Rezumat

Prezentăm, în comunicarea noastră, noutatea suportului electronic al Dicționarului invers al limbii române şi, mai ales, avantajele acestui format electronic, un adevărat motor de căutare şi găsire a cuvintelor în funcție de criterii bine determinate. Considerăm că printr-o astfel de grupare a cuvintelor, oferită de e-DILR, se evidențiază sistemul derivativ al limbii române actuale şi se poate stabili dinamica productivității formanților lexicali.

1. Introducere

Ne-am propus ca, prin intermediul acestei comunicări, să semnalăm încercarea autorilor *Dicționarului invers al limbii române*[5] de a consolida un model lexicografic şi de a atrage atenția asupra unui nou format electronic al dicționarelor, asupra utilității programului de căutare şi grupare laolaltă a cuvintelor după anumite criterii.

2. Modelul lexicografic

2.1.

În lexicografia românească a existat, până în 2007, un singur dicționar în care cuvintele erau aşezate în ordine alfabetică inversă, şi anume Dicționar invers, lucrare colectivă, întocmită sub conducerea academicianului Alexandru Graur şi publicată la Editura Academiei în 1957.

Preluând ideea alcătuirii unui asemenea dicționar, autorii Dicționarului invers al limbii române (DILR) încearcă să consolideze acest model lexicografic. Cuvintele sunt ordonate după terminații şi nu după inițiale. Terminația, adică secvența finală alcătuită din una sau mai multe litere, poate fi coincidentă sau nu cu unul sau cu mai multe sufixe lexicale, de pildă – tor este sufix în silitor, *muncitor,* nu şi în abator, unde este doar terminație, - ărească reprezintă întotdeauna două sufixe, ca în: bucătărească, cenuşărească, portărească ş.a. Toate cuvintele care au aceeaşi terminație sunt înşiruite prin ordonarea lor alfabetică inversă în raport cu uzanța lexicografică, adică de la dreapta la stânga. Prin urmare, la literele –a, -ă, -b, -c, -d ş.a.m.d. apar toate cuvintele care se termină, nu cele care încep cu aceste litere. Această ordonare după secvența finală este logică şi eficientă, deoarece evidențiază, printr-o asemenea grupare, cuvintele formate în acelaşi fel. De pildă, în seria cuvintelor terminate în –eşte, vom găsi alături, indiferent de originea bazei, cuvinte ca: româneşte, englezeşte, franțuzeşte, italieneşte, turceşte…bărbăteşte, vitejeşte, prosteşte, frățeşte…, fapt important pentru diverse

[5] Cecilia Căpățînă (coordonator), Claudia Drăghici, Ovidiu Drăghici, Alina Gioroceanu, Daniel Ivănuş, Dumitru Ivănuş, Simona Pisoi, Virgil Pisoi, Anamaria Preda, Vlad Preda, Melitta Szathmary, *Dicționar invers al limbii române,* Bucureşti, Editura Niculescu, 2007

abordări şi de negăsit într-un dicţionar obişnuit. Această ordonare, spre deosebire de cea normală, permite specialiştilor observarea, compararea şi studierea cuvintelor după modul comun de formare, de pildă, care este semnificaţia specifică secvenţei derivative adăugată semnificaţiei bazei, ce reguli combinative pot fi deduse ş.a.

Cuvintele sunt precedate de indicarea statutului morfologic, iar substantivele, şi de indicarea genului. Omonimele cu statut morfologic diferit apar ca intrări separate.

Dicţionarul indică, de asemenea, accentul, cu excepţia neologismelor neadaptate în care vocala proeminentă este redată altfel decât în ortografia românească (de exemplu: groom, pound, weekend, yard ş.a.).

2.2.

O noutate, îndrăzneaţă şi riscantă, poate fi considerată abandonarea unui principiu lexicografic tradiţional, acela al menţinerii intacte a inventarului lexical al dicţionarului sau al dicţionarelor-sursă, prin eliminarea cuvintelor ieşite din circulaţie. Am păstrat din seria cuvintelor învechite pe cele necesare pentru o descriere sumară a aspectelor sociale, culturale, administrative etc. ale unei epoci trecute, care sunt încă prezente în manuale de istorie, literatură şi, de aceea, sunt necesare vorbitorului instruit. Ideea de bază a alcătuirii inventarului lexical al dicţionarului nostru a fost aceea a reprezentativităţii acestuia pentru stadiul actual al limbii române. Ne-am propus ca inventarul DILR să cuprindă cuvintele aflate în uzul vorbitorului instruit de azi. Menţinerea cuvintelor ieşite din uz în inventarul dicţionarului ar fi făcut mai dificilă observarea microstructurilor lexicogramaticale ale românei actuale, ar fi creat o falsă impresie asupra dimensiunii lexicului limbii române. Pe de altă parte, aceste cuvinte, care prezintă mare interes pentru istoria limbii, sunt inventariate fie în dicţionare speciale, fie apar în toate celelalte dicţionare ale limbii române.

DILR cuprinde toate cuvintele din DEX2, DOOM2, NDN, cu excepţia celor ieşite din uz, a expresiilor şi a locuţiunilor, neinteresante pentru un astfel de dicţionar. Menţinerea fondului lexical vechi, de multe ori în detrimentul includerii cuvintelor noi în dicţionarele românei actuale prezintă dezavantaje. Autorii acestor dicţionare n-au renunţat la cuvintele vechi din mai multe considerente. Oricare ar fi acestea, s-a creat un precedent primejdios, şi anume faptul că noile dicţionare, explicative sau normative, ale românei dintr-o anumită perioadă, vor avea inventare din ce în ce mai bogate, în discordanţă evidentă cu realitatea. În fond, astfel de dicţionare trebuie să înregistreze cuvintele aflate în circulaţie, din perioada respectivă.

DILR înregistrează împrumuturi şi creaţii româneşti recente din studii importante asupra lexicului actual, ca cele ale Adrianei Stoichiţoiu-Ichim ş.a. şi din dicţionare de cuvinte recente, ca cel al Floricăi Dimitrescu, Elenei Trifan ş.a.

Preluarea selectivă a inventarului din dicţionarele-sursă amintite şi introducerea de cuvinte noi, care, deşi sunt folosite de vorbitorul instruit, n-au fost încă înregistrate într-un dicţionar explicativ, au contribuit la realizarea unui inventar reprezentativ al românei actuale. Am considerat indispensabilă introducerea în acest dicţionar a majorităţii cuvintelor (derivate sau împrumutate ca atare) cu prefixe ca ne-, re- (neacreditat, neangajare, nearticulare, necomunist, neimputabil, nesponsorizat, neatestare, neautorizare, necalificare, reaccesa, reancheta, reamplasare, reamplasat, înmatriculat, înnorat ş.a.), cu sufixul –re (accesare, autoperfecţionare, autodepăşire, autointitulare,

electrostimulare, macrostabilizare, megapetrecere, ofertare, printare ş.a.) şi a formelor verbale participiale, cuvinte în uzul actual, a căror frecvenţă sporită, atestată şi prin textele puse la îndemână de motorul de căutare Google, ne-a obligat la includerea acestora în inventar. Neintroducerea lor în dicţionare nu se poate justifica în niciun fel, câtă vreme DOOM2 înregistrează cuvinte formate în acelaşi fel, dar care sunt foarte rar folosite: autoînsămânțare, neadormire, neavere, nebunire, nechezare, nedormire, negativizare, osebire, îndrăznire, îndumnezeire, înfățare, îngălare, înnemurire ş.a.

Dicţionarul invers al limbii române are un inventar lexical de circa 100.000 de cuvinte, reprezentativ pentru româna actuală. Ar fi fost necesar ca derivatele incluse de noi în inventar, inexistente în vreun dicţionar românesc, existente în uz însă, să fi fost marcate cu asterisc, lucru care se va realiza în viitoarea ediţie a acestui dicţionar.

3. Formatul electronic al DILR

3.1.

O noutate absolută o constituie formatul electronic al dicţionarului, deoarece niciunul dintre formatele electronice ale dicţionarelor româneşti nu reprezintă altceva decât varianta electronică a inventarului respectiv. În fapt, e-DILR este chiar un motor electronic de căutare şi găsire, care dispune de un program capabil să afişeze, pe baza unor comenzi, liste complete de cuvinte sau de părţi de vorbire indicate care conţin o anumită secvenţă de litere în poziţiile: iniţială, interioară şi/sau finală. Această secvenţă indicată coincide sau nu cu unul sau cu mai multe prefixe/sufixe, chiar cu o temă lexicală, de aceea, în lista afişată pe baza comenzii vor apărea, alături de cuvintele derivate (pe care nu numai specialistul le recunoaşte, ci oricare persoană instruită), şi cuvinte nederivate care conţin secvenţa respectivă.

Cu ajutorul acestui program original, vor fi afişate:

- toate cuvintele terminate într-o secvenţă indicată, de ex. în –tor: abator, actor, creator, silitor, vizitator, numitor etc.

- o anumită parte de vorbire terminată într-o secvenţă indicată, de ex. toate substantivele terminate în –ime: golănime, mulţime, adâncime, înălţime etc.; toate adjectivele terminate în –bil: abordabil, acceptabil, accesibil, stabil, locuibil etc.; toate verbele terminate în –ui: aflui, alcătui, asemui, dărui, locui, vărui etc., toate adverbele terminate în –eşte: vitejeşte, bărbăteşte, frăţeşte, latineşte etc.

- toate cuvintele care încep cu o anumită secvenţă, de ex. cu com- sau cu nepre-: combate, combatant, compătimi; neprevăzător, neprecupeţit, neprevestire ş.a.

- toate cuvintele care încep cu o secvenţă şi se termină cu altă secvenţă, de ex. cuvinte care încep cu secvenţa ne- şi se termină cu secvenţa –bil: nevindecabil, netratabil, neinteligibil etc.

- părţi de vorbire indicate care au o anumită secvenţă iniţială şi altă secvenţă finală, de exemplu toate substantivele care încep cu pre- şi se termină cu –re: prevedere, prevestire, preocupare, prevenire ş.a., toate verbele care încep cu des- şi se termină cu –i: descoperi, descotorosi, deservi, despăgubi, destăinui;

- toate cuvintele sau părţile de vorbire indicate care conţin o anumită succesiune de litere într-o poziţie mediană, de exemplu toate cuvintele care conţin secvenţa –nct-: acupunctură, conjunctivită, conjunctură, punct, punctual, punctualitate, punctiform ş.a. sau toate adjectivele care conţin secvenţa –nct-: conjunctiv, disjunctiv, punctat, punctual, punctiform;

- toate cuvintele terminate într-o secvenţă complexă, de exemplu substantive terminate în - alitate: actualitate, normalitate, bestialitate, brutalitate, punctualitate, oralitate, dualitate etc.

- numărul de cuvinte din lista cerută.

3.2.

Dicţionarul oferă imaginea sistemului morfolexical al românei actuale, deoarece, pe baza lui se pot identifica microsistemele lexico-gramaticale. Considerăm că gruparea cuvintelor în microsisteme este realizabilă cu ajutorul e-DILR şi va putea conduce la stabilizarea normativă de care e atâta nevoie. Se ştie că o comunicare corectă se bazează pe norme ferme şi valabile pe termen lung, prin logica alcătuirii, şi nu se poate baza pe o puzderie de variante admise, tolerate, resuscitate sau inversate din timp în timp.

Informaţiile furnizate de programul nostru sunt indispensabile oricăror cercetări privitoare la dinamica derivativă a românei actuale, la productivitatea unui anumit sufix/prefix/element de compunere, la specificitatea morfologică a unei anumite terminaţii sau a unui anumit sufix, la existenţa, în diferite poziţii din cuvânt, a unor secvenţe fonetice, la constrângerile fonetice impuse vecinătăţii de un sufix sau de o secvenţă finală ş.a.

Cu ajutorul listelor puse la dispoziţie, se pot întocmi diferite statistici asupra frecvenţelor unor sufixe substantivale, verbale, adjectivale, adverbiale, asupra frecvenţelor unor terminaţii formate dintr-o anumită succesiune de litere/foneme şi se pot întreprinde cercetări asupra unor aspecte felurite ale limbii române. Frecvenţele date ierarhizează formanţii lexicali şi evidenţiază gradul de productivitate a acestora. Oricare persoană interesată de dinamica vocabularului actual, în general, şi de dinamica productivităţii actuale a unor elemente de compunere, sufixe şi/sau prefixe va putea, cu ajutorul statisticilor date de suportul electronic, să le constate şi să le analizeze.

4. Concluzii

DILR (Dicţionar invers al limbii române) este, prin urmare, o lucrare nouă prin concepţie şi prin inventarul de cuvinte. Dacă ideea ordonării cuvintelor limbii române după terminaţie nu e nouă, formatul electronic al DILR nu e unul obişnuit, ci este conceput să faciliteze orice fel de cercetare asupra dinamicii lexicului românesc actual. În plus, DILR este al doilea dicţionar bazat pe o asemenea ordonare a cuvintelor care poate fi utilă cercetării şi astfel, poate consolida un model lexicografic. În comparaţie cu alte dicţionare româneşti pe suport electronic, e-DILR nu e doar versiunea electronică a inventarului lexical, ci e un adevărat program, menit studierii limbii române din perspectiva dinamicii sale derivative, în special.

Referințe bibliografice

Academia Republicii Populare Române, Institutul de Lingvistică din Bucureşti. (1957). Dicționar invers, Editura Academiei Republicii Populare Române, Bucureşti, 1957.

Academia Română, Institutul de Lingvistică „Iorgu Iordan" din Bucureşti (1996). Dicționar explicativ al limbii române (DEX), ediția a II-a, Editura Univers Enciclopedic, Bucureşti, 1996.

Academia Română, Institutul de Lingvistică „Iorgu Iordan" din Bucureşti. (2005). Dicționar ortografic, ortoepic şi morfologic al limbii române, ediția a II-a revăzută şi adăugită, Editura Univers Enciclopedic, Bucureşti, 2005.

Căpățînă, Cecilia (coord.), Drăghici, Claudia, Drăghici, Ovidiu, Gioroceanu, Alina, Ivănuş, Daniel, Ivănuş, Dumitru, Pisoi, Simona, Pisoi, Virgil, Preda, Anamaria, Preda, Vlad, Szathmary, Melitta. (2007). Dicționar invers al limbii române, Bucureşti, Editura Niculescu, 2007.

Dimitrescu, Florica. (1997). Dicționar de cuvinte recente (DCR), ediția a II-a, Editura Logos, Bucureşti, 1997.

Marcu, Florian. (1997). Noul dicționar de neologisme, Bucureşti, Editura Academiei Române, 1997.

Stoichițoiu Ichim, Adriana. (2001). Vocabularul limbii române actuale. Dinamică, influențe, creativitate, Editura All, Bucureşti, 2001.

Stoichițoiu Ichim, Adriana. (2006). Creativitatea lexicală în româna actuală, Editura Universității din Bucureşti, Bucureşti, 2006.

Stoichițoiu Ichim, Adriana. (2006). Aspecte ale influenței engleze în româna actuală, Bucureşti, Editura Universității, 2006.

Trifan, Elena, Adrian, Trifan. (2003). Dicționarul de neologisme şi abrevieri recente (DNAR), Ceraşu, Editura Scrisul Prahovean, 2003.

DLRI. BAZĂ LEXICALĂ INFORMATIZATĂ. DERIVATE

BOGDAN ALDEA[2], MARIUS CLIM[1], ELENA DĂNILĂ[1],

CRISTINA FLORESCU[1], LAURA MANEA[1]

[1] *Academia Română, Institutul de Filologie Română „A. Philippide", Iaşi – România*

[2] *Universitatea „Alexandru Ioan Cuza", Facultatea de Informatică, Iaşi – România*

bogdan.aldea@gmail.com, mariusradu_ro@yahoo.com, isabelle.danila@gmail.com, cristinafl24@yahoo.fr, l_manea2002@yahoo.com

Rezumat

Lucrarea se referă la prima serie de rezultate ale proiectului DLRI. BAZĂ LEXICALĂ INFORMATIZATĂ. DERIVATE (cod CNCSIS 1609), desfăşurat în perioada 2007-2008.Subiectul analizei este cuvântul lexicografic – articol de dicţionar – achiziţionat electronic, bază a analizei lingvistice şi informatice a fenomenului derivării cu sufixele *-ime* şi *-işte* pe teren românesc; se realizează astfel primul eşantion semnificativ, în lexicografia românească informatizată, cuprinzând un corpus de articole DLRI (DA[6]+DLR[7] informatizat, unificat şi actualizat).

1. Introducere

În filologia românească s-au iniţiat în ultima vreme proiecte menite să ducă la informatizarea, unificarea şi actualizarea lucrării fundamentale a lexicografiei româneşti, *Dicţionarul (Tezaur) al Limbii Române* (DA + DLR).

Demersul de faţă prezintă rezultatele primei etape a proiectului CNCSIS nr. 1609, *DLRI. Bază lexicală informatizată. Derivate*, proiect finanţat de Ministerul Educaţiei şi Cercetării (MEC), desfăşurat în perioada 2007–2008 în Institutul de Filologie Română „A. Philippide" al Academiei Române – Filiala Iaşi şi condus de CS I dr. Cristina Florescu. Echipa de cercetare este formată din autorii articolului de faţă şi din: acad. Dan Tufiş (RACAI), prof. univ. Dan Cristea (UAIC–FII), lector drd. Corina Forăscu (UAIC–FII).

2. Scopul cercetării

Proiectul menţionat îşi propune: a) realizarea unui eşantion lexicografic tip DLR, format din derivatele pe terenul limbii române cu sufixul *-ime* (de origine latină) şi cele cu *-işte* (de origine veche slavă), reprezentând cca 550 de articole lexicografice (din seria veche

[6] DA = *Dicţionarul limbii române* (DA), tom I-II, Tipografia ziarului „Universul", Imprimeria Naţională, Bucureşti, 1913-1949.

[7] *Dicţionarul limbii române* (DLR), Serie nouă, tom VI-XIV, Editura Academiei, Bucureşti, 1965-2007.

DA şi din seria nouă DLR), prelucrate în format XML cu ajutorul DLRex[8], concretizat într-o bază de date lexicale informatizată (cuprinzând aceste derivate) şi rafinarea, în funcţie de materialul achiziţionat în format electronic, a instrumentului de lucru DLRex; b) unificarea tehnico-lexicografică a articolelor DA – DLR conform normelor DLR; c) redactarea unui volum cuprinzând studii de specialitate privind unele aspecte informatice şi lingvistice relevate de materialul lexicografic implicat analizei.

3. Elemente inedite

În cadrul cercetării prezentate se întreprinde pentru prima dată achiziţionarea electronică a unor texte lexicografice atât din seria veche DA a *Dicţionarului limbii române*, cât şi din seria nouă DLR; în egală măsură, analiza vizează actualizarea şi unificarea, cu ajutorul instrumentelor şi tehnologiei create, a unui grup lexical semnificativ pentru limba română care înglobează actualmente articole inegale în ceea ce priveşte tehnica lexicografică şi informaţia lingvistică din DA şi DLR.

4. Faza actuală de lucru

Până în prezent, în cadrul grantului s-au realizat etapele programate pentru acest prim an de lucru:

1) stabilirea listei de intrări (s-a plecat de la un număr de cca 300 lexeme, stabilite în funcţie de lucrările de specialitate anterioare şi s-a ajuns la o cifră de peste 550 de lexeme excerptate manual din volumele DA şi DLR) – operaţiune făcută de specialiştii lingvişti;

2) scanarea articolelor care cuprind derivate în *-ime* şi *-işte* din DA şi din DLR (o parte din materialul scanat a fost selectat din corpusul de date lexicografice DA+DLR din proiectul complex eDTLR[9]),

3) OCR-izarea[10] materialului rezultat (ultimele două operaţiuni au fost făcute de specialiştii informaticieni);

4) corectarea materialului OCR-izat; precizăm că, dacă pentru DLR – în cadrul grantului *Dicţionarul limbii romane în format electronic. Studii privind achiziţionarea* – operaţia de OCR-izare mai fusese întreprinsă, OCR-izare unor articole compacte, grupate lexicologic, din DA este făcută pentru prima dată;

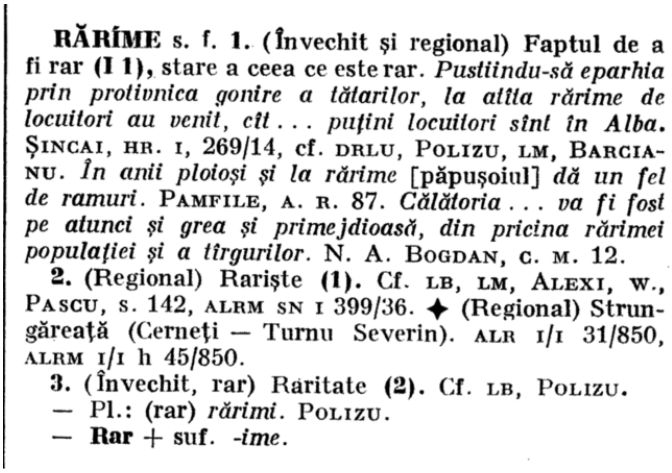
5) redactarea, în funcţie de normele DLR, a unui număr de peste 130 articole cu derivate în *-ime* şi *-işte* din DA;

6) rafinarea, în funcţie de materialul lexicografic, a DLRex-ului.

[8] DLRex este un instrument de achiziţionare, prelucrare şi consultare a DLR, creat în cadrul grantului *Dicţionarul limbii române în format electronic. Studii privind achiziţionarea* (cod CNCSIS 1815), proiect desfăşurat în perioada 2003 – 2005.
[9] Proiect complex *eDTLR – Dicţionarul Tezaur al Limbii Române în format electronic* (2007 – 2010).
[10] Transpunerea din format imagine (.tif) in format text (.rtf).

Ilustrăm succint etapele de achiziționare a materialului din DA-DLR în cadrul etapelor 2), 3) şi 4). Pentru derivatele din DLR procesul de prelucrare nu a necesitat un timp de lucru prea mare datorită calității bune a hârtiei şi a cernelii utilizate pentru tipărire.

Cuvânt scanat din DLR:



RĂRÍME s. f. **1.** (Învechit şi regional) Faptul de a fi rar **(I 1),** stare a ceea ce este rar. *Pustiindu-să eparhia prin protivnica gonire a tătarilor, la atîta rărime de locuitori au venit, cît . . . puțini locuitori sînt în Alba.* ŞINCAI, HR. I, 269/14, cf. DRLU, POLIZU, LM, BARCIANU. *În anii ploioşi şi la rărime* [păpuşoiul] *dă un fel de ramuri.* PAMFILE, A. R. 87. *Călătoria . . . va fi fost pe atunci şi grea şi primejdioasă, din pricina rărimei populației şi a tîrgurilor.* N. A. BOGDAN, C. M. 12.

2. (Regional) Rariște **(1).** Cf. LB, LM, ALEXI, W., PASCU, S. 142, ALRM SN I 399/36. ✦ (Regional) Strungăreață (Cerneți — Turnu Severin). ALR I/I 31/850, ALRM I/I h 45/850.

3. (Învechit, rar) Raritate **(2).** Cf. LB, POLIZU.

— Pl.: (rar) *rărimi.* POLIZU.

— **Rar** + suf. *-ime.*

Forma corectată în urma OCR-izării:

RĂRÍME s. f. **1.** (Învechit şi regional) Faptul de a fi rar **(I 1),** stare a ceea ce este rar. *Pustiindu-să eparhia prin protivnica gonire a tătarilor, la atîta rărime de locuitori au venit, cît... puțini locuitori sînt în Alba.* ŞINCAI, HR. I, 269/14, cf. DRLU, POLIZU, LM, BARCIANU. *În anii ploioşi şi la rărime* [păpuşoiul] *dă un fel de ramuri.* PAMFILE, A. R. 87. *Călătoria... va fi fost pe atunci şi grea şi primejdioasă, din pricina rărimei populației şi a tîrgurilor.* N. A. BOGDAN, C. M. 12.

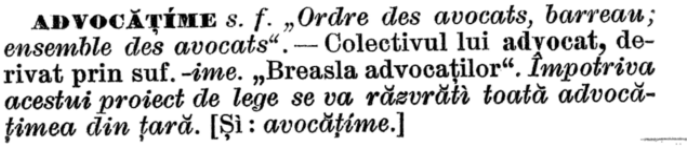
2. (Regional) Rariște **(1).** Cf. LB, LM, ALEXI, W., PASCU, S. 142, ALRM SN I 399/36. ◊ (Regional) Strungăreață (Cerneți – Turnu Severin). ALR I/I 31/850, ALRM I/I h 45/850.

3. (Învechit, rar) Raritate **(2)**. Cf. LB, POLIZU.

– Pl: (rar) *rărimi.* POLIZU.

Achiziționarea articolelor DA a necesitat un timp de lucru mai mare datorită problemelor întâmpinate în procesul OCR-izării mai ales din cauza calității hârtiei şi a cernelii (volumele din DA au fost tipărite între anii 1913-1949). Ilustrăm etapele parcurse în cadrul unui articol DA, inclusiv etapa de redactare şi refacere a acestuia după normele lexicografice din DLR:

Cuvânt scanat din DA:



ADVOCĂȚÍME *s. f.* „*Ordre des avocats, barreau; ensemble des avocats*“. — Colectivul lui **advocat,** derivat prin suf. *-ime.* „**Breasla advocaților**“. *Împotriva acestui proiect de lege se va răzvrăti toată advocățimea din țară.* [Şi : *avocățíme.*]

Forma corectată în urma OCR-izării:

ADVOCĂȚÍME *s. f. „Ordre des avocats, barreau; ensemble des avocats”.* — Colectivul lui **advocat**, derivat prin suf. *-ime.* „Breasla advocaților”. *Împotriva acestui proiect de lege se va răzvrăti toată advocățimea din țară.* [Şi: *avocățime.*]

Forma actualizată după normele de redactare DLR:

AVOCĂȚÍME s. f. Mulțime de avocați; (p. ext.) totalitatea avocaților (dintr-o unitate administrativă); breaslă a avocaților. *Împotriva acestui proiect de lege se va răzvrăti toată advocățimea din țară.* DA, cf. CADE, SCRIBAN, D., CIORĂNESCU, D. ET. 100, MDA. _ *Avocățimea baroului ieşean s-a întrunit ieri.*

– Şi: (învechit) **advocățíme**.
– **Avocat** + suf. *-ime.*

5. Repere analitice lingvistice

Din punct de vedere lingvistic, în proiectul de față se vizează analiza contrastivă a două grupuri lexicale (cel al derivatelor cu sufixul *-ime* şi cel al derivatelor cu *-işte*), pe baza tratării lor lexicografice în *Dicționarul limbii române* (DA + DLR).

Prin actualizarea lexicologică şi lexicografică a listei de cuvinte (a intrărilor) din DA şi DLR, prin informarea şi completarea bibliografică cu privire la grupul lexical în studiu şi prin analiza semantico-lingvistică întreprinsă, cele două grupuri ale derivatelor au început să fie taxonomizate şi în funcție de achiziționarea electronică a faptelor. Studiul lingvistic al lexemelor în discuție este întreprins din punctul de vedere al etimologiei, al structurii semantice şi noționale, al categoriei gramaticale, al repartiției dialectale, stilistice etc.

Aşadar, avem în vedere situația din limba română a derivatelor cu sufixul *-ime* de tip *românime* „poporul român; număr mare de români; teritoriu locuit de români” < *român* + *-ime* etc. şi a derivatelor cu sufixul *-işte* – de exemplu, *alunişte* < *alun* + *-işte*.

Prin acest proiect se valorifică cercetările lingvistice anterioare care analizează aspecte semnificative privind formarea şi structura derivatelor în limba română (Pascu, 1916, Paşca, 1948, Carabulea, 1959, Sădeanu, 1962).

În continuare, prezentăm câteva dintre observațiile de natură lingvistică rezultate din cercetările efectuate până în prezent asupra eşantionului vizat.

a) Derivatele cu -ime

În română, sufixul *-ime* se ataşează unor cuvinte de origine diversă (latină, veche slavă, turcă, maghiară etc.)[11]. O analiză statistică a elementelor derivate relevă o situație specială pentru limba română veche (unde sufixul *-ime* formează derivate substantive abstracte – de tipul *cruzime* < *crud*, *înălțime* < *înalt* etc.), față de limba română

[11] Vezi şi Carabulea, 1959:67

contemporană (unde sufixul *-ime* are mai ales o valoare colectivă – *românime* < *român*)[12].

Din punct de vedere semantic, sufixul *-ime* poate forma substantive abstracte feminine care exprimă calitatea, ataşându-se unor adjective (*acrime* < *acru*, *cruzime* < *crud* etc.) sau poate forma numerale fracţionare, ataşându-se unor numerale cardinale (*doime* < *doi* + *-ime*; *treime* < *trei* + *-ime* etc.); acest sufix poate avea şi o valoare colectivă pe care o conferă unor derivate pe care le formează.

În contextul limbilor romanice, în literatura de specialitate se consideră că mai ales limba română a păstrat sufixul de origine latină *-ime* în derivate colective; în afara ariei romanice, acest sufix era identificat de cunoscutul romanist Meyer Lübke şi în albaneză[13]. Din această cauză, dată fiind condiţia complexă a derivatelor pe teren românesc în *-ime*, vom lua în discuţie în continuare mai ales valoarea colectivă a acestui sufix (care formează aproximativ 236 de substantive colective feminine[14] din totalul de cca 414 derivate cu acest sufix).

Atunci când derivatul substantiv colectiv feminin format desemnează o mulţime sau o colectivitate de persoane ori o stare, din punct de vedere gramatical sufixul colectiv *-ime* se poate ataşa:

– unor substantive, nume de persoane (*arăbime* „mulţime de arabi”[15]*, băieţime* „număr mare de băieţi”*, ciobănime , ţărănime* etc.);

– unor adjective (*albăstrime* „oameni de la oraş îmbrăcaţi în albastru”*, greime* „mulţime, grosul (oştii)”*, vechime* „oameni din trecut (vechi)”);

– unor adverbe (*josime* „oameni de jos”*, călărime* „oameni călări”);

– unui verb (*însoţime* „grup de oameni, ceată”).

Atunci cînd derivatul substantiv colectiv feminin format desemnează o mulţime sau o colectivitate de lucruri, ori o stare, din punct de vedere gramatical sufixul colectiv *-ime* se poate ataşa:

– unor substantive, nume de lucruri (*păime* „paie de nutreţ”*, pietrime* „mulţime, grămadă de pietriş” etc.);

– unor adjective (*desime* „desiş”*, gălbenime* „cantitate sau mulţime de lucruri de culoare galbenă”*, acrime* „aguridă, fructe verzi);

– unor adverbe (*împrejurime* „locul sau ţinutul dimprejur, din apropiere”);

– unor verbe (*arzime* „febră”).

Atunci când derivatul substantiv colectiv feminin format desemnează o mulţime sau o colectivitate de animale, din punct de vedere gramatical sufixul colectiv *-ime* se poate ataşa:

[12] Vezi Iordan, 1956:311; Ivănescu, 2000 :701.
[13] Vezi Meyer-Lübke, 1895:531.
[14] În cursul cercetărilor întreprinse în cadrul grantului s-a îmbogăţit lista de derivate cu sufixul colectiv *-ime* de la 182 (menţionate în studiul Florenţei Sădeanu şi în cel al Elenei Carabulea) la cca 236.
[15] Precizăm faptul că definiţiile au fost simplificate şi că spaţiul nu ne permite să citǎm sursele.

– unor substantive, nume de animale (*bondărime, broştime, păsărime* etc.);

– unor adjective (*sălbăticime* „mulțime de fiare sălbatice”).

Sufixul colectiv *-ime* se poate ataşa şi unor substantive nume de plante, derivatul rezultat desemnând o mulțime de plante (*nucime, rugime, stejărime*).

Cea mai mare parte a acestor derivate cu sufixul colectiv *-ime* au un singur sens cu valoare colectivă, dar există şi derivate care au mai multe sensuri dintre care doar unul are un semantism colectiv [de exemplu, *întunecime* 1. „întuneric (adânc), obscuritate; 2. (despre lună sau soare) eclipsă”, 3. (fig.) „lipsă de cultură, barbarie”; *4. (rar) „mulțime nenumărată”*; *prostime* 1. „simplitate, modestie; sărăcie”, 2. (învechit) „neştiință, ignoranță, nepricepere” 3. „prostie; ceea ce denotă prostie”, *4. (cu sens colectiv) „oameni de rând, marea masă a populației; s p e c . țărănime; norod, gloată, mulțime”* etc.]. Anumite derivate au un sens colectiv dat chiar de cuvântul de bază însuşi (vezi *mulțime* < *mult*, *desime* deja menționat etc.), deci sufixul colectiv nu face decât să întărească semantismul colectiv al derivatului. În ceea ce priveşte valoarea peiorativă a unor lexeme, aceasta poate fi considerată ca inclusă în nucleul lor lexical central [(peior.) *burtăverzime* «burghezie» < *burtăverde* «burghez», (peior.) *calicime* < *calic*, (peior.) *golănime* < *golan* etc.] sau poate exista la nivel semantic secundar, accentuată în funcție de context [(peior.) *popime* < *popă*].

b) Derivatele cu -işte

Spre deosebire de situația derivatelor cu sufixul *-ime*, cazul multor derivate cu sufixul *-işte* a fost studiat punctual şi la diverse nivele de generalitate exegetică. Taxonomizarea bazată pe un număr extins de cazuri se întâlneşte la Pascu (1916), Paşca (1948), Sădeanu (1962). În Florescu (2007) se stabilesc, în funcție de ultimele analize (cuprinse lingvistic şi în materialul lexical al proiectului), repere taxonomice amănunțite ale grupului derivatelor cu sufixul *-işte*.

Pentru a sublinia complexitatea derivatelor în limba română (complexitate care va fi cunoscută prin cercetările lingvistice şi informatice ce formează scopul proiectului de față), prezentăm actuala taxonomie a categoriilor de derivate în *-işte* (cele mai semnificative statistic) din punct de vedere al dominantei semantice.

În funcție de realitatea desemnată, în Florescu (2007: 140-142), aceste lexeme sunt grupate în nouăsprezece categorii semantice. Menționăm numai unsprezece dintre aceste categorii, cele mai semnificative statistic:

– locul pe care cresc (fiind cultivate) sau au crescut plante, arbori etc.: *barabulişte, cânepişte*;

– loc (amenajat) pentru vite, loc unde stau sau îşi au sălaşul animale sălbatice: *bourişte*, *lupişte* „loc unde stau lupii”;

– loc, teren cu anume caracteristici, calități, trăsături (geomorfologice): *bătelişte* „loc bătătorit”, *golişte* „loc lipsit de vegetație”;

– loc pe care se fac sau pe care s-au făcut anumite construcții, amenajări: *cotişte* „cătun”, *cuptorişte* „loc pentru cuptor”;

– locul pe care se desfăşoară ori s-a desfăşurat o acțiune, o activitate etc.: *alergărişte* „hipodrom”, *mulgărişte* „loc în care se mulg oile”;

– plantele care cresc (fiind cultivate) sau se depozitează pe un anume teren: *arinişte*, *curpenişte*, *făgişte*;

– grup de ființe: *roielişte* „mulțime de pui de albină”, *porumbărişte*;

– construcții, amenajări etc. şi părți ale acestora (cu anumite trăsături caracteristice): *măierişte* „construcție unde se păstrează recolta, uneltele”, *molişte* „porțiune din fagure în care s-au instalat moliile”;

– o acțiune: *măsorişte, oprelişte*, *pieişte*;

– stare fizică, sufletească, trăsătură caracteristică etc.: *firişte* „soi, viță”, *linişte*;

– caracteristică a naturii, fenomen (sau stare) atmosferic(ă) etc.: *norişte* „cer înnorat”, *prigorişte* „caniculă”, *sorişte*.

Din punct de vedere gramatical, sufixul *-işte* se poate ataşa:

– unor substantive (cea mai frecventă situație): *alergărişte* „alergare; hipodrom”, *barabulişte* „ogor pe care se cultivă cartofi” < *barabulă* „cartof”; *făgişte* „pădure de fag”, *vraişte* < *vrah* (varianta lui *vraf*) (cu sensul învechit şi popular: „snopi de cereale desfăcuți şi împrăştiați pe arie pentru a fi treierați cu ajutorul vitelor)”;

– unor adjective: *linişte*, *desişte* „teren acoperit cu mulțime deasă de arbori”, *golişte* „loc neacoperit (de vegetație)”

– unor verbe: *împărțişte* „împărțire”, *pribegişte*, *zăcărişte* < *a zăcări* „a zăcea”.

6. Cadru informatic

La baza parsării DA şi DLR, cu ajutorul instrumentul special creat DLRex, stă formatarea textului:

CUVÂNT parte de vorbire, sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
A sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
I sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
1 sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
a) sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
b) sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
2) sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
II sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
III sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
B sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
C sensul cuvântului [*exemplu care ilustrează sensul menționat* SIGLA]
- informații ortoepice, gramaticale, de circulație …
- informații etimologice

Figura 1: Schema generală a unei intrări în DLR

Figura 1: O intrare din dicționar păstrează, în linii mari, acest format

După citirea unei intrări din dicționar s-a construit un vector în care este pus (în ordinea citirii din fişier) fiecare fragment care are o formatare diferită față de fragmentul ce îl precedă şi respectiv cel de după el, astfel că o parsare a vectorului, ținând seama de

modul în care este scrisă o intrare din dicționar, ar putea duce la formarea fişierului XML dorit.

Chiar dacă multe erori de formatare pot fi evitate în etapa de prelucrare automată, vor exista întotdeauna factori ce nu pot fi schimbați. Printre aceştia se numără şi calitatea hârtiei şi a cernelii folosite la tipărirea dicționarului, dar şi modul în care acesta a fost editat, mod ce diferă în unele cazuri de la un volum la altul.

Pentru a evita toate aceste probleme care pot genera erori la parsare, s-a impus o prelucrare a vectorului înainte ca acesta să fie parsat şi aducerea lui într-o formă mai restrânsă (contopirea într-unul singur a elementelor din vector, determinarea formatării corecte a caracterelor ce nu-şi păstrau formatarea inițială şi aducerea la aceeaşi fomatare a informațiilor din listele ce încheie o intrare).

Parsarea vectorului are la bază succesiunea stilurilor fragmentelor, prezentată în schema de mai sus, la care se adaugă şi tratarea cazurilor particulare ce au fost constatate pe parcursul testării aplicației pe un eşantion cât mai larg de pagini din DA şi DLR. Acest vector s-a împărțit în mai multe elemente astfel încât, fiecare element să conțină câte o intrare din fişierul prelucrat. Astfel, în cazul aparițiilor unor erori de parsare sau formatare a unei intrări să nu se pericliteze întreaga parsare.

Precizăm funcționalitatea acestui instrument, pentru a dezvolta ulterior descrierea caracteristicilor principale:

- permite trecerea textului DA-DLR din format RTF (Word) în format XML;
- permite vizualizarea şi corectarea fişierelor XML;
- funcționează ca interfață de consultare şi realizează interogarea DA-DLR în format electronic;
- permite actualizarea şi unificarea DA-DLR.

Funcționalitatea de bază a aplicației este aceea de transpunere a DA-DLR în format electronic (XML). În prima fază, aplicația are un fişier XML gol. Pentru crearea DA-DLR electronic sau pentru adăugarea de noi pagini la cele existente deja, se încarcă în program fişierele RTF, după cum este ilustrat în Figura 2.

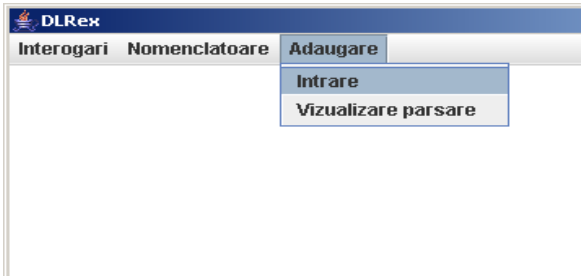


Figura 2: Captură de ecran: încărcarea în DLRex a fişierelor

Pentru a se verifica dacă parsarea s-a realizat cu succes se deschide fişierul XML rezultat în urma parsării ca în Fig 4 şi 5.

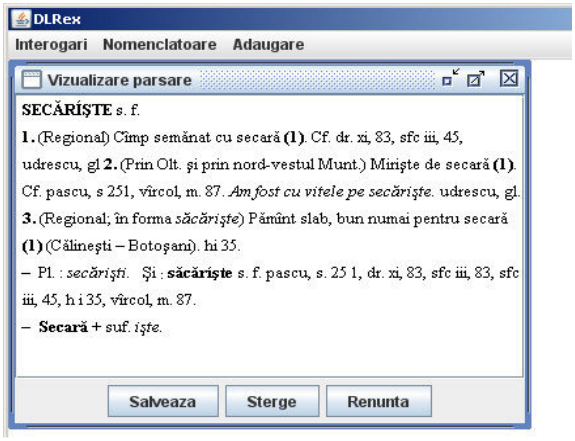


Figura 3: Rezultat în urma parsării unui derivat cu suf. *-işte*.

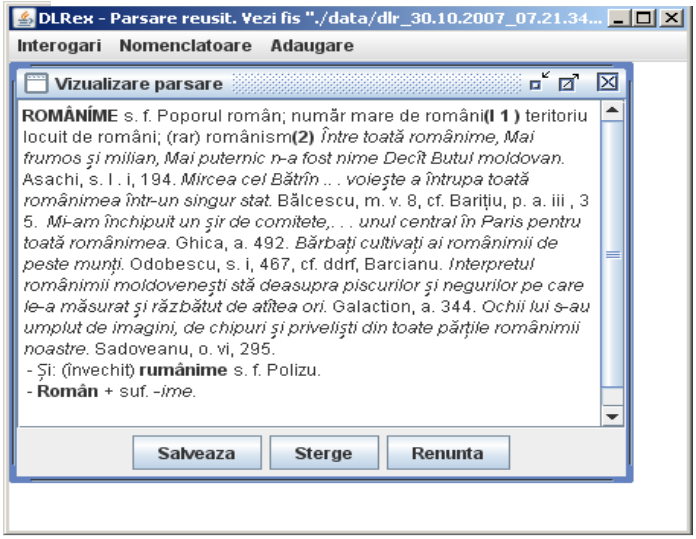


Figura 4: Rezultat în urma parsării unui derivat cu sufixul *-ime*.

O problemă apărută în cazul derivatelor este faptul că unele spații, dar şi „new line” (simbolul de sfârşit de linie), au caracteristici diferite de text.

Aceasta problemă apare frecvent la începutul şi sfârşitul unei intrări. Pe lângă erorile de scanare şi OCR-izare întâlnite, o posibilă cauză a acestui fenomen ar putea fi faptul că fişiere parsate conţin mai multe intrări din DA şi DLR care au fost copiate din paginile originale ale dicţionarului şi apoi lipite la colecţia de derivate într-un singur fişier.

Tot acestui fapt se datorează, în unele cazuri, şi diferenţele de formatare ce nu pot fi sesizate cu ochiul liber şi anume acele caractere/simboluri care nu îşi păstrează formatarea şi pe care le-am amintit anterior. Datorită acestui lucru, împărţirea unui document în intrări a trebuit, de asemenea, finisată. Copierea şi alipirea pe rând a câte unei intrări pot avea efecte şi asupra codificării începutului/sfârşitului unei intrări. Drept urmare s-a pus accent pe a defini cât mai general trecerea de la o intrare la alta, astfel încât să nu se modifice nici rezultatele unor parsări precedente din DA sau DLR.

Una din problemele cu care DLRex se confrunta era legată de spațiere, anume alipirea de cuvinte care apăreau consecutiv sau împărțirea unora prin inserarea de spații. Astfel, în unele cazuri, o seamă de cuvinte puteau fi interpretate ca doi termeni diferiți. Deoarece unele cuvinte puteau fi interpretate în urma OCR-rizării că având mai mult de un singur format pe cuvânt, chiar dacă tipul formatării era identic, în unele cazuri erau interpretate ca fiind cuvinte diferite.

Prelucrarea derivatelor cu DLRex a fost totodată şi o testare a acestuia. Dacă acesta a fost antrenat pentru anumite pagini din DLR, iată că, odată cu parsarea derivatelor, DLRex-ul a trebuit să facă față unei plaje mai largi de volume din dicționar.

Pentru că volumele dicționarului tezaur al limbii române au fost redactate de autori diferiți, precum şi faptul că unele volume sunt mai vechi iar altele mai noi, aplicația ar putea fi mereu îmbunătățită pentru ca parsarea să aibă un procentaj cât mai mare de reuşită, fără a fi nevoie de prea multe intervenții din partea factorului uman care să trateze manual eventualele cazuri particulare.

7. Concluzii

Articolul prezintă o serie de rezultate intermediare ale cercetării lingvistice şi informatice privind *DLRI. Bază lexicală informatizată. Derivate*. Articolele lexicografice analizate sunt derivatele pe terenul limbii române cu sufixele *-ime* şi *-işte* pe baza materialului din DA + DLR, material achiziționat electronic şi prelucrat lingvistic şi informatic.

Pentru întâia oară a fost stabilit, în lingvistica românească, un inventar lexicografic cât mai complet al cuvintelor derivate cu sufixele *-ime* şi *-işte*.

Cercetarea prezentată valorifică: 1) studii lingvistice anterioare consacrate fenomenului derivării în limba română; se conturează puncte de vedere lexicologice noi, puncte de vedere care, şi prin lexicometrie (statistică), îşi modifică substanțial perspectiva diacronică (etimologică) şi semantică (semasiologică) asupra obiectului de studiu; 2) performanțele instrumentului de lucru DLRex creat în cadrul unui grant anterior (2003-2005) *Dicționarul limbii române. Studii privind achiziționarea*, grant care a deschis seria proiectelor de cercetare informatică şi lingvistică privind achiziționarea şi prelucrarea în format electronic a Dicționarului (Tezaur) al Limbii Române; rafinarea acestui instrument de lucru, pe baza eşantionului lexicografic decelat în proiectul de față, va putea crea premizele unor rafinări ulterioare cu un înalt grad de aplicabilitate.

Se preconizează că rezultatele finale ale proiectului vor completa, unifica şi rafina din punct de vedere lingvistico-lexicografic şi informatic faptele de limbă studiate.

Referințe bibliografice

Carabulea, Elena (1959). -AME şi -IME în limba română, în Studii şi materiale privitoare la formarea cuvintelor în limba română, vol. I, Bucureşti, Editura Academiei, p. 65-75.

Cristea, Dan, Răschip, Marius, Forăscu, Corina, Haja, Gabriela, Florescu, Cristina, Aldea, Bogdan, Dănilă, Elena (2007). The Digital Form of the Thesaurus Dictionary of the Romanian Language, în vol. Advances in Spoken Language Technology (editors Corneliu Burileanu, Horia-Nicolai Teodorescu), Bucureşti, Editura Academiei Române, p. 195-206.

Dănilă, Elena (2007). Le traitement lexicographique des dérivés aux suffixes collectifs en roumain et en français, en DLR et en TLFi (roum. -ime, -iste et fr. -aille, -erie), în XXV CILPR 2007 Congres International de Linguistique et de Philologie Roumaine. Communications: Résumés, 3-8 septembre 2007, Innsbruck, Innsbruck University press, p. 572-573.

Florescu, Cristina (2006). Linişte şi derivatele pe teren românesc în -işte, în Volum omagial „Mioara Avram", Bucureşti, Editura Academiei, p. 151-160.

Florescu, Cristina (2007), Probleme de semantică a limbii române (capitolul I, §8, p. 128-150), Editura Universității „Al. I. Cuza" – Iaşi, 395 p.

Haja, Gabriela, Dănilă, Elena, Forăscu, Corina, Aldea, Bogdan-Mihai (2005). Dicționarul limbii române (DLR) în format electronic. Studii privind achiziționarea, Iaşi, Editura Alfa, 76 p., publicat şi electronic pe www.consilr.info.uaic.ro.

Iordan, Iorgu (1956). Limba română contemporană, Bucureşti, Editura Ministerului Învățământului.

Ivănescu, G. (2000). Istoria limbii române, Iaşi, Editura Junimea.

Meyer-Lübke, W. (1895). Grammaire des langues romanes, II, Paris.

Pascu, Gheorghe (1916). Sufixele româneşti, Bucureşti, Ediționea Academiei Române.

Sădeanu, Florența (1962). Sufixele colective din limba română cu specială privire asupra repartiției lor, în Studii şi materiale privitoare la formarea cuvintelor în limba română, vol. III, Bucureşti, Editura Academiei, p. 41-88.

Tufiş, D., Diaconu, L., Barbu, A.M., Diaconu, C. (1995). The Mac-ELU implementation of derivative morphology for Romanian, Research Report, I.C.I, iunie 1995.

*** Dicționarul limbii române (DA), tom I-II, Tipografia ziarului „Universul", Imprimeria Națională, Bucureşti, 1913-1937; Dicționarul limbii române (DLR), Serie nouă, tom VI-XIV, Editura Academiei, Bucureşti, 1965-2007.

PARSAREA EDTLR CU GRAMATICI ÎN MEDIUL JAVACC. STADIUL ACTUAL, PROBLEME ŞI SOLUȚII DE DEZVOLTARE

NECULAI CURTEANU[1], GABRIELA PAVEL[1,2],
CRISTINA VEREŞTIUC[2], DIANA TRANDABĂȚ[1,2]

[1]*Institutul de Informatică Teoretică Iaşi, Academia Română*
[2]*Facultatea de Informatică, Universitatea "Al. I. Cuza", Iaşi*

curteanu@iit.tuiasi.ro, pavelg@info.uaic.ro,
cciocarlau@info.uaic.ro, dtrandabat@info.uaic.ro

Rezumat

Lucrarea prezintă experienţe de parsare a dicţionarelor DEX şi DTLR cu trei versiuni de gramatici implementate în mediul JavaCC. O abordare complementară acestor experimente este parsarea de suprafaţă prin Segmentare-Dependenţă la marcherii de sensuri dintr-o intrare de dicţionar şi construirea arborelui de sensuri în mod independent de parsarea individuală a definiţiilor de sensuri. Arborii de sensuri obţinuţi (prin fiecare dintre metode sau prin combinarea lor) sunt cumulaţi într-o bază de cunoştinţe cu multiple aplicaţii de natură (computaţional) lingvistică.

1. Introducere

Scopul parsării articolelor de dicţionar este obţinerea arborelui lexico-semantic al unei intrări. Această lucrare descrie principalele probleme întâlnite la transformarea unei intrări de dicţionar, mai întâi DEX (Dicţionarul EXplicativ) şi apoi DTLR (Dicţionarul Tezaur al Limbii Române), într-un fişier XML care reprezintă codificarea definiţiilor din intrarea respectivă.

Intrările lexicale din dicţionar sunt transformate prin scanare şi validare manuală în format electronic, respectând toate convenţiile de notare şi abreviere folosite în formatul tipărit. Principalele etape care trebuie efectuate pentru transformarea unei intrări din DEX / DTLR într-un fişier XML conform specificaţiilor CONCEDE-TEI (Erjavec *et al.*; 2000) sunt: **(1)** *Transformarea intrărilor* din dicţionar în *format* HTML: Această etapă este necesară deoarece gramatica Java folosită iniţial (Tufiş; 2001) pentru parsarea de dicţionar (notată în continuare JavaINI) lucrează pe etichetele HTML rezultate în urma exportului din mediul MS-Word 97. **(2)** *Curăţarea fişierului* HTML: Parserul pentru dicţionar foloseşte în varianta existentă numai anumite etichete HTML. **(3)** *Parsarea* bazată pe seturi de reguli din gramatici implementate în mediul JavaCC. Aceste reguli de bază (aprox. 400), cărora li se adaugă un număr de reguli subsidiare, particulare, vor defini arborele de sensuri pentru intrarea lexicală. Abordarea tradiţională a parsării unei intrări de dicţionar este una de tip *Depth-First*, deoarece sensurile şi definiţiile lor sunt parsate secvenţial, de la un capăt la altul al articolului, construcţia arborelui de sensuri având loc în mod dinamic, odată cu înaintarea parserului în corpul articolului. Dezavantajul major al acestui tip de abordare este acela că arborele de sensuri ajunge să fie construit în final doar dacă parserul ajunge să accepte efectiv *toate* definiţiile sensurilor descrise în intrarea de dicţionar. Neparsarea

unei singure definiţii de sens (fie ea şi ultima) duce la respingerea articolului şi la abandonarea construcţiei arborelui de sensuri. **(4)** Crearea fişierului XML conţinând *arborele lexico-semantic* ce codifică principalele definiţii ale sensurilor pentru un înţeles al cuvântului de intrare respectând standardul CONCEDE-TEI. Implementarea parserului eDTLR pe care o avem în vedere va trebui să respecte noile specificaţii de codificare actuale, standardul de etichetare XCES-TEI, versiunea P5 (2007).

Structura lucrării este următoarea: după o trecere în revistă a formatului DTLR şi a diferenţelor faţă de formatul DEX, se vor prezenta tipuri de erori ale parserului generat de gramatici de tip JavaCC şi soluţii propuse, în cazul dicţionarelor DEX şi DTRL. O abordare *complementară* acestor experimente este o parsare de suprafaţă a marcherilor de sensuri dintr-o intrare de dicţionar şi construirea arborelui de sensuri pentru această intrare în mod independent de parsarea individuală a definiţiilor de sensuri, prin recunoaşterea marcherilor la sensuri, a secvenţelor de marcheri şi a dependenţelor dintre aceşti marcheri (algoritmul DSSD – Dictionary Sense Segmentation & Dependency, schiţat în secţiunile 3 şi 4).

2. Parsarea DEX şi DTLR cu gramatici în mediul JavaCC

2.1 Structura articolelor DTLR

Structura articolelor DTLR este următoarea: • Cuvântul-titlu: scris îngroşat, cu litere mari, cu precizarea accentului; • Formele sale flexionare (dacă există), despărţite între ele prin ”, – ”; • Un indice superior pentru specificarea omonimiei; • Partea de vorbire (sau părţile de vorbire) a(le) acelui cuvânt, scrisă(e) cu caractere normale; • Definiţia sensului (sensurilor) din cuvântul-titlu, primul sens fiind scris în continuarea cuvântului-titlu, fără alineat nou. Celelalte sensuri se termină cu punct. Explicaţiile corespunzătoare unui sens sunt scrise într-un paragraf nou. Sensurile se numerotează *ierarhic* astfel: sensurile principale cu litere mari de tipar (A., B., C., ...); cifre romane (I., II., III., ...); cifre arabe (1., 2., 3., ...); sensurile principale pot fi divizate în sensuri secundare, denotate cu enumerarea de litere mici a), b), c), ... şi romb-plin sau romb-gol. Există uneori şi forme de reprezentare a (sub)sensurilor printr-o expresie sau mai multe, fără o subordonare explicită la un sens principal, denotate de obicei cu romb-gol (E. Dănilă; 2007). • Un sens are următoarea structură: definiţia propriu-zisă; explicaţii suplimentare sau de utilizare, incluse sau nu în paranteză; citate, urmate de siglă (*i. e.* autorul, opera din care a fost preluat citatul, pagina, etc.); expresii urmate de explicaţii (separate prin ”=”); referinţe la alte cuvinte. • În cadrul unui sens, după numerotarea respectivă, se pot da informaţii de natură sintactico-semantică asupra cuvântului (partea de vorbire, dacă este verb tranzitiv sau nu, dacă are formă de plural), informaţii referitoare la modul sau aria de utilizare, etc. Un exemplu de articol DTLR este prezentat mai jos:

> **VENIÁL, -Ă** adj. (Livresc; despre păcate[2], greşeli etc.) Care poate fi iertat (de Biserică); uşor, fără importanţă deosebită. Cf. PONTBRIANT, D., LM. Flămînzilă, Setilă sînt păcate veniale ale omului, pe care le-a personificat amabil Rabelais în Grandgousier, Gargantua şi Pantagruel. CĂLINESCU, B. 59. Ierarhia sufletelor în eternitate, în acord cu doctrina virtuţilor teologice şi a păcatelor mortale şi veniale, nu lasă... nici o îndoială asupra caracterului eticii danteşti. VIANU, L.U. 15. Uriaşii [sînt]... simbolizări ale forţelor, anomaliilor şi ale unor păcate veniale ale omului. IST. LIT. ROM. I, 223, cf. DN[3].

– Pronunțat: -ni-al. – Pl.: veniali, -e.
– Din lat. **venialis, -e,** fr. **véniel.**

2.2 Diferențe între structura articolelor din DEX şi DTLR

Diferențele cele mai importante: **1.** Forma de plural ce urmează imediat după cuvântul-titlu în cadrul unei intrări din DEX nu se regăseşte şi în DTLR. **2.** Fiecare nou sens al unei intrări DTLR este tratat într-un *paragraf nou*, în timp ce în DEX toate sensurile sunt tratate în cadrul aceluiaşi paragraf. **3.** În structura definiției unui sens, în cadrul unei intrări DTLR, citatele dintr-o operă au precizată sigla – care urmează după citat, elementul siglă nefiind prezent în DEX. **4.** Penultimul paragraf dintr-o intrare DTLR are o structură specială. **5.** Etimologia cuvântului-titlu este dată în ultimul paragraf ce tratează o intrare DTLR.

2.3 Preprocesarea

Pentru a putea fi procesat, fişierul de intrare pentru parser (articolul de dicționar) trebuie să fie în format HTML. Pentru parsare s-au folosit câteva fişiere de intrare în format document. Acestea au fost convertite în format HTML folosind Microsoft Word 2003, obținându-se fişiere cu marcaje HTML complexe. Deoarece parserul nu acceptă diacritice sau alte caractere speciale, acestea trebuie convertite în entități HTML. S-a dezvoltat un program de preprocesare, în limbajul PHP, care elimină toate marcajele inutile parserului şi efectuează conversiile necesare, astfel încât să se obțină un fişier de intrare corect construit pentru parsare.

O situație specială o constituie caracterele romb-plin (♦) şi romb-gol (◊). În conversia realizată de Microsoft Word 2003, codificarea pentru romb-gol este „♧”. Pentru a fi corect interpretat de parser, programul de preprocesare converteşte acest caracter în următoarea codificare: <FONT FACE="Monotype Sorts">G</FONT>.

Un exemplu de rulare a *programului de preprocesare* este prezentat mai jos, pentru exemplul din secțiunea 2.1, precedând parsarea la arborele de sensuri.

<B><P>VENIAL, -Ă</B>adj. (Livresc; despre păcate2, greşeli etc.) Care poate fi iertat (de Biserică); uşor, fără importanță deosebită. Cf. pontbriant, d., lm. <I>Flămînzilă, Setilă sînt păcate veniale ale omului, pe care le-a personificat amabil Rabelais în Grandgousier, Gargantua şi Pantagruel.</I>călinescu, b. 59. <I>Ierarhia sufletelor în eternitate, în acord cu doctrina virtuților teologice şi a păcatelor mortale şi veniale, nu lasă... nici o îndoială asupra caracterului eticii danteşti.</I>vianu, l.u. 15. <I>Uriaşii... simbolizări ale forțelor, anomaliilor şi ale unor păcate veniale ale omului.</I>ist. lit. rom. i, 223, cf. dn3. -- Din lat. <B>venialis, -e</B>, fr.<B> veniel.</B></P>

Rularea parserului (arborele de sensuri) pentru exemplul din secțiunea 2.1 este:

```
<?xml version='1.0' encoding='utf-8'?><document>
 <entry>
    <hw>VENIAL</hw>
    <gram> nominativ_masculin_singular_indefinit </gram>
    <orth>VENIAL</orth>
    <gram> nominativ_feminin_singular_indefinit </gram>
    <orth>-Ă</orth>
    <pos>adjectiv</pos>
    <struc>
        <usg>Livresc; despre păcate2, greşeli etc.</usg>
        <alt>
```

```
            <def> Care poate fi iertat ( de Biserică) </def>
            <def> uşor, fără importanţă deosebită. Cf. pontbriant, d., lm. </def>
          </at>
    <struc type="Phrase">
            <orth>Flămînzilă, Setilă sînt păcate veniale ale omului, pe care le-a personificat amabil
Rabelais în Grandgousier, Gargantua şi Pantagruel. </orth>
            <def> călinescu, b. </def>
    </struc>
    <struc type="Phrase"> <def> 59. </def></struc>
    <struc type="Phrase">
            <orth>Ierarhia sufletelor în eternitate, în acord cu doctrina virtuţilor teologice şi a
păcatelor mortale şi veniale, nu lasă ... nici o îndoială asupra caracterului eticii danteşti. </orth>
            <def> vianu, l. </def>
    </struc>
    <struc type="Phrase"><def> u. 15. </def> </struc>
    <struc type="Phrase">
            <orth>Uriaşii ... simbolizări ale forţelor, anomaliilor şi ale unor păcate veniale ale
omului. </orth><def>ist. </def>
 </struc>
    <struc type="Phrase"><def> lit. rom. i, 223, cf. dn 3.- </def></struc>
  </struc>
        <etym> Din limba<lang>lat.</lang> venialis, - e </etym>
        <etym> Din limba <lang>fr.</lang> veniel.</etym>
 </entry>
```

2.4 Probleme de parsare a DEX şi DTLR cu gramatici JavaCC

Probleme identificate la parsarea DTLR cu gramatica JavaDTLR:

1. *Erori* care ţin de structura fişierelor HTML (intrările DTLR): nerecunoaşterea elementelor care pot codifica sigle (<SPAN>) şi a atributelor HTML introduse de Word 2003 (*rezolvare*: eliminarea tagurilor/atributelor inutile); problemele de acest tip au fost soluţionate în etapa de preprocesare. **2.** *Erori* care ţin de structura articolelor DTLR: **(a)** nerecunoaşterea siglelor (considerate fie definiţii, fie o structură complexă de tip frază), pentru exemplele oferite de editor (codificate prin “ ”); **(b)** nerecunoaşterea subsensurilor / sensurilor secundare (marcate prin romb-gol sau romb-plin; problema a fost rezolvată prin identificarea codificării aferente acestor două notaţii); **(c)** nerecunoaşterea referinţelor către sensuri ale aceluiaşi cuvânt. **3.** Erori care ţin de structura regulilor din gramatică: nerecunoaşterea sensurilor unor expresii / locuţiuni (apar confuzii dacă sensul subliniat sau scris aldin, este un început de definiţie sau o expresie urmată de o definiţie; soluţia propusă este modificarea regulii prin care sunt identificate sensurile). De exemplu, putem întâlni câmpuri de forma:

◊ E x p r. **A turna venin în cineva** (sau **în sîngele cuiva**) = a produce cuiva un rău, o suferinţă. *Afacerea bazilicii turnă... venin... în sîngele lui Pomponescu.* Călinescu, b. i. 476. respectiv, *N-are nimic !* = **a)** nu i s-a întâmplat nici un rău; *folosire in context,* CARAGIALE, o. iii, 86.

unde expresia poate fi scrisă fie aldin, fie italic.

Articolele DTLR se disting de cele din DEX prin prezenţa citatelor. Dacă în DEX aveam doar exemple (ale editorului sau din alţi autori), în DTLR avem două tipuri de exemple: • exemple ale editorului (mai rare, codificate în mod grafic prin “ ”); • exemple formate din citate şi sursele acestora.

Gramatica JavaDTLR pentru parsarea DTLR este rezultată din modificarea gramaticii JavaDEX, folosită pentru a genera parserul pentru dicționarul DEX (Curteanu, Amihăesei; 2004). Gramatica JavaDTLR este modificată astfel încât regulile ei să urmeze structura unei intrări DTLR, în cadrul căreia se disting multiple diferențe față de o intrare DEX. În continuare prezentăm câteva probleme şi soluții de parsare cu gramatica JavaDTLR; pentru mai multe detalii a se vedea (Curteanu *et al.*; 2007).

1) Problemă: Salvarea intrărilor în format HTML. Setul de etichete folosite pentru codificarea HTML este mult mai mare şi mai variat decât etichetele parsate de gramatica JavaDTLR. Totuşi, adnotarea mai rafinată permite identificarea parțială a unor probleme importante în codificarea unui articol, de exemplu, începutul de siglă este marcat prin . Dar *această codificare* nu este recunoscută de parser. **Soluție:** Formatul HTML trebuie curățat printr-un program de eliminare a etichetelor nefolosite. Etapa de rescriere este menționată mai jos printr-o listă de probleme apărute la curățare.

a. Subproblemă: Parserul acceptă doar *etichete scrise cu majuscule*. De asemenea, atributele acestora se scriu tot cu majuscule. Word 2003 furnizează doar etichete scrise cu litere mici, conform standardului XML. **Soluție:** Se modifică în textul articolului (prin intermediul programului de curățare) caracterele utilizate. Astfel <p …> devine <P> etc. De asemenea se scot toate atributele corespunzătoare elementelor întâlnite în codificarea HTML.

b. Subproblemă: Nu sunt recunoscute anumite etichete (<SPAN>) şi atribute care țin de formatarea documentului: mărime, font etc. **Soluție:** Se înlătură din fişierul HTML etichetele nerecunoscute de parser prin intermediul programului de preprocesare. Se scoate un fişier curățat de elemente de formatare legate de stiluri, fonturi, porțiuni de text (elemente de tip sau). Se curăță informația legată de descrierea documentului şi de stiluri (totul până la primul paragraf: conținutul elementului <head> şi totul până la primul <p>, style=’...’; de asemenea se şterg elementele şi .

2) Problemă. Nu sunt recunoscute anumite caractere, de exemplu cratima ”-”, care specifică un interval: în cadrul unei sigle (scrisă necompactat, fără un marcaj specific), cratima blochează parserul (simbolul „-” este acceptat doar în paragrafele ultim şi penultim, unde se dau indicații cu privire la formele de plural etc.). **Soluție**: Pentru parsarea textelor, cratima poate fi considerată ca făcând parte din structura unui token.

3) Problemă: Confuzii între intrepretările care trebuie atribuite diverselor şiruri de caractere. **Exemplu:** Ambiguitate în interpretarea cifrelor romane (care identifică un sens principal), ce pot fi considerate majuscule (prezente în sigle la numele autorului). **Soluție:** Acest exemplu este un caz particular al problemei mai generale de parsare a siglelor, ce se rezolvă prin aplicarea unui procedeu eficient de pattern-matching cu o listă generală de sigle siglelor (vezi problema 6).

4) Problema *referințelor* la alte sensuri / subsensuri descrise in DTLR, fie că este vorba de aceeaşi intrare sau de intrări diferite, fie că este vorba de sensurile rezultate prin aplicarea unor ”funcții” unui anumit sens, cum ar fi sinonimia, antonimia, paronimia etc. (mai multe exemplificări sunt date în (Curteanu *et al.*; 2007)).

5) Problema *citatelor.* Nu există în gramatică elemente precise pentru codificarea citatelor. Există trei tipuri de *citate*: **a.** Exemple ale editorului (precedate de simbolul

„□”); simbolul □ nu este recunoscut de către parser deoarece nu a fost introdus în gramatica JavaDLR un simbol care să reprezinte codificarea aferentă acestuia; **b.** Exemple conform altei surse (un citat urmat de o referință de tipul „Cf. sursă”); **c.** Exemple din *siglă*, structură constituită dintr-o listă de triplete (autor, carte, pagină/şir-de-pagini/volum). **Soluție**: Este necesară introducerea în gramatică a unui nou simbol, similar cu cel corespuzător romburilor, şi crearea unei reguli care să marcheze începutul unui astfel de citat.

6) Problemă: *Recunoaşterea automată a siglelor*. În ieşirea HTML din Word, elementele prin atributul *text-uppercase* conțin informații foarte importante – încadrează siglele (etichetele din Word 2003 sunt mai bogate şi mai rafinate). **Soluție:** Aplicarea unui algoritm de pattern-matching pentru recunoaşterea elementelor de tip siglă folosindu-se un fişier cu toate siglele din DTLR.

7) Problema *parsării arborelui de sensuri* *de mai multe niveluri*. **Soluție:** Experimentele au arătat că gramatica permite parsarea articolelor DTLR şi crearea unui arbore de sensuri de orice adâncime. Faptul că gramatica JavaDTLR poate construi recursiv şi incremental acest arbore dovedeşte că aceste gramatici reprezentă o platformă de plecare corectă pentru parserul arborelui de sensuri. Pentru a testa parsarea recursivă la arborele de sensuri pe adâncimi mai mari decât trei (3) s-a construit o intrare artificială care să respecte structura articolelor DEX şi care a fost trecută prin parser. S-au obținut arbori pe 2-3-4 niveluri (etape succesive de lucru pe fişiere cu structura sensurilor cât mai complicată).

În parsarea arborelui de sensuri folosind gramatica JavaDTLR rămân încă multe probleme nerezolvate la nivelul subcâmpurilor din definițiile unor sensuri: de exemplu, sunt parsate doar fragmente din citate (până la ”;”). Acestea sunt recunoscute doar dacă sunt precedate de „v.” şi „cf.”, deci pentru cazuri particulare.

Prin utilizarea gramaticii JavaDEX, parsarea corectă a intrărilor de dicționar DEX s-a putut realiza într-un procentaj de 90-93%. Acest procentaj a fost obținut prin parsarea câtorva mii de intrări DEX (Curteanu, Amihăesei; 2004). Plecând apoi de la gramatica JavaDEX şi utilizând aceeaşi tehnologie s-a putut trece la dezvoltarea unei gramatici JavaDTLR, specializată pe parsarea dicționarului DTLR. Dat fiind că structura şi diversitatea informațiilor codificate în DTLR sunt substanțial mai complexe, adaptarea noii gramatici JavaDTLR la parsarea DTLR s-a putut face până în acest moment pentru aproximativ 10% din intrări DTLR, procent estimat pe un număr mic de articole DTLR, având însă structuri diverse. O problemă importantă şi dificil de rezolvat este şi transformarea actualei ieşiri a parserului cu gramatici în mediul JavaCC din standardul CONCEDE-TEI (Erjavec *et al.*; 2000) în standardul XCES-TEI, versiunea P5 (2007).

3. Două abordări ale parsării arborelui de sensuri în DTLR

Din experimentele şi problemele ridicate de analiza automată a unei intrări de dicționar rezultă limpede că *parsarea arborelui de sensuri* pentru articolele DTLR rămâne problema „strategică” a construcției unui parser performant pentru DTLR (şi, în general, pentru marile dicționare). *Soluția* pe care o considerăm viabilă pentru rezolvarea acestei probleme este, în condițiile date, o analiză aprofundată a gramaticilor JavaDEX şi JavaDTLR, şi identificarea principalelor module (pachete) de reguli de producție astfel încât să devină transparente *etapele mari* ale parsării unui articol.

Ar fi important să putem separa încă de la început construcţia arborelui de sensuri de parsarea subcâmpurilor din definiţiile sensurilor şi a câmpurilor asociate întregului articol; altfel spus, să putem extrage *mai întâi* arborele de sensuri şi *apoi* să se realizeze recunoaşterea (parsarea) câmpurilor din definiţii.

Construcţia unui astfel de parser are un caracter complementar actualelor gramatici în mediul JavaCC, integrând elemente din actuala gramatică JavaDTLR. Un astfel de parser trebuie să realizeze mai întâi *segmentarea* la elementele care introduc definiţiile de (sub)sensuri, să recunoască *secvenţele de marcheri* care introduc aceste sensuri, să facă *ierarhia* acestor secvenţe (ordinea lor parţială reprezentând chiar nodurile arborelui de sensuri ale intrării de dicţionar respective), după care să se revină la *parsarea definiţiilor* cuprinse în sensuri, una câte una. Această separare a obţinerii arborelui de sensuri chiar înainte de parsarea definiţiilor poate creşte substanţial procentajul de parsare deoarece poate accepta articole DTLR ale căror definiţii pe (sub)sensuri, considerate individual, să nu fie complet parsabile.

Problema parsării unui articol de dicţionar presupune o bună delimitare a sensurilor în cadrul intrării respective. De aceea considerăm că sunt posibile (cel puţin) două abordări: **(1)** O parcurgere în adâncime (un algoritm de tip *Depth-First*) în care identificarea şi parsarea sensurilor se realizează în mod dinamic, în aceeaşi secvenţă: pentru o intrare se caută un început de sens şi, dacă se găseşte, se încearcă parsarea definiţiei şi exemplelor asociate acelui sens; această manieră de lucru a fost descrisă în secţiunea 2.4, prin explicarea modului de lucru al parserului JavaDTLR; **(2)** O analiză de suprafaţă a intrării, în care accentul să cadă pe identificarea sensurilor, fără o parsare a definiţiilor aferente sensurilor identificate. Aceste două abordări ale problemei parsării DTLR sunt complementare.

4. Parsarea prin Segmentare-Dependenţă la marcheri de sensuri

Pentru *parsarea arborelui de sensuri* luăm în considerare implementarea unui algoritm de segmentare a intrării la secvenţele de marcheri pentru codificarea sensurilor. Marcherii de sensuri pot fi secvenţe de tipul: • o etichetă <B> urmată de un număr scris cu cifre romane şi de punct, pentru codificarea sensurilor principale; • etichetele <P> şi <B> în această ordine urmate de un număr (scris cu cifre romane sau arabe) şi de punct, pentru codificarea unui sens principal sau a unui subsens. • etichetele specifice codificării romburilor : <FONT FACE='MONOTYPE SORT' >&#71 ;</FONT> ; romburile constituie de obicei sensuri secundare, alături de sensurile marcate prin litere mici scrise boltit şi urmate de paranteza închisă, de exemplu **a)**. În plus, romburile pot fi singurele sensuri în cazul unei intrări de dicţionar (vezi exemplul 3). • referinţe (definiţii prin trimiteri la alte sensuri ale intrării sau la sensuri ale altor cuvinte din dicţionar). • expresiile şi locuţiunile, în cazul în care acestea nu sunt precedate de un romb, dar sunt scrise boltit şi sunt urmate de "=" sau de ", ".

Stabilirea dependenţelor dintre sensuri se face în conformitate cu ierarhia cunoscută a delimitatorilor de sensuri (majuscule, cifre romane, cifre arabe, litere mici, romburi etc.), redată în Fig. 1. Menţionăm că în faza aceasta nu se face încă parsarea *propriu-zisă* a câmpurilor din definiţiile de sensuri. Pentru verificarea corectitudinii arborelui de sensuri obţinut în acest fel se va realiza recuperarea automată a structurii liniare din

structura arborele de sensuri parsat (de exemplu, cu un script VBA inclus în editorul Word), realizând o comparație cu textul inițial al intrării.

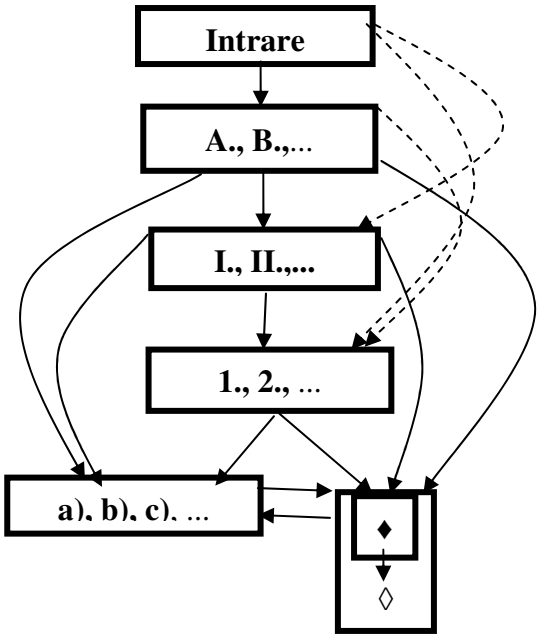


Figura 1: Hiper-graful de dependență între clasele de marcheri la sensuri DTLR

Marcarea sensurilor în intrarea de dicționar nu ține cont de nivelul acestora în ierarhia sensurilor. Pentru a putea obține arborele de sensuri corespunzător unei intrări DTLR, în parsarea de suprafață se stabilesc depedențele dintre (sub)sensuri cu un algoritm relativ simplu, denumit *Dictionary Sense Segmentation & Dependency* (DSSD), de examinare *Breadth-First* a macherilor la sensuri. DSSD analizează secvențele de marcheri la sensuri pentru o intrare DTLR şi stabileşte arborele de sensuri pe baza ierarhiei claselor de marcheri din Fig. 1. Hipergraful dependențelor nu prezintă o structură complet aciclică, iar rezolvarea ambiguităților de dependență la nivelul hipernodurilor finale (sensurile secundare) se poate rezolva printr-o căutare *lookahead* cu mai multe poziții în secvența de marcheri, dublată de o analiză contextuală a câmpurilor din vecinătatea marcherilor de sensuri (Dănilă; 2007). Iată două exemple de articole DTLR concrete ce demonstrează ciclicitatea claselor de marcheri la sensuri secundare din Fig. 1.

Aceste tipuri de sensuri sunt greu de ierarhizat întrucât pot produce situații ambigue, datorită flexibilității sensurilor în codificare. De exemplu, un romb poate conține subsensuri de tipul **a)** (a se vedea exemplul 1)**,** aşa cum şi un marcher de tipul **a)** poate precede un romb (pe care îl conține într-o ierarhie a sensurilor) (vezi exemplul 2) :

Exemple de codificare a sensurilor (conform grafului de dependențe din Fig. 1):

(1) **ÚNU**… **B. …I…. 2.** (La f., cu valoare neutră…)

a) (în legătură cu verbe ca "a da" ...)...

b) (în legătură cu verbe ca "a spune"...)...◊E x p r . **Ştii (**sau **ştiți) una ?,** se spune despre**…♦** (în legătură cu verbele "a cânta", ...) …

c) (în e x p r.= **A)+i= face** (cuiva) **una (şi bună)** sau ...**3.**...

(2) **ÚNU**… **B. …I…. 3.** ... ◊…◊…◊…◊ L o c . a d v. **Unul peste altul = a)** în dezordine, de-a valma. *Feciorii au sărit...*BARAC, T. 18/15....; **b)** (regional) în total....

(3) **VÎLCELÚŞĂ** s. f. (Atestat prima dată în 1519, cf. MIHĂILĂ, D.) Diminutiv al lui v î l c e a (1). ...*şi-l întrebă dacă-i mai lasă cele cinci pogoane de fînețe ce se aflau într-o vîlceluşă.* SĂM. VI, 884, cf. CADE. ... ◊ (F i g .) *O vîlceluşe de carne, albă şi fragedă.* CAMIL PETRESCU, P. 65.

Avantajul parsării de suprafață a arborelui de sensuri cu algoritmul DSSD față de parsarea cu gramatici JavaCC este acela că în parsarea de suprafață prin Segmentare-Dependență nu mai întâlnim problema ieşirilor din analizor prin neparsarea corespunzătoare a definițiilor. Desigur, problema parsării unei intrări nu se rezumă doar la identificarea arborelui, necesitând o nouă parsare de "adâncime", pentru identificarea corectă a elementelor unei definiții. Parsarea DSSD de suprafață poate fi continuată cu pachetul de reguli din gramatica JavaDTLR care realizează parsarea câmpurilor fiecărei definiții (sens) din arborele de sensuri stabilit prin DSSD.

5. Concluzii

Prin utilizarea gramaticii JavaDEX în mediul JavaCC, parsarea corectă a intrărilor de dicționar DEX s-a putut realiza într-un procentaj de 90-93% din totalul articolelor DEX (Curteanu, Amihăesei; 2004). Plecând apoi de la gramatica JavaDEX şi folosind aceeaşi tehnologie s-a putut trece la dezvoltarea unei gramatici JavaDTLR, specializată pe parsarea dicționarului DTLR. Dat fiind că structura şi diversitatea informațiilor codificate în DTLR sunt substanțial mai complexe, adaptarea noii gramatici JavaDTLR la parsarea DTLR s-a putut face până în acest moment pentru aproximativ 10% din intrări DTLR, procent estimat pe un număr mic de articole DTLR, având însă structuri diverse. Procesul de transformare şi adaptare a gramaticii Java ar trebui continuat astfel încât gramatica JavaDTLR să parseze corect aproximativ toate intrările DTLR. Rămâne, de asemenea, transformarea actualei ieşiri a parserului cu gramatici în mediul JavaCC din standardul CONCEDE-TEI în standardul de adnotare XCES-TEI, versiunea P5.

Abordarea complementară la parsarea cu gramatici în mediul JavaCC este propunerea de parsare de suprafață a arborelui de sensuri cu algoritmul DSSD. Această parsare se bazează pe ierarhia marcherilor la sensuri din Fig. 1 şi realizează mai întâi *segmentarea* la elementele care delimitează definițiile între ele (marcherii de sensuri), recunoaşte *secvențele de marcheri* care introduc aceste sensuri, stabileşte *ierarhia* acestor secvențe, ordinarea lor parțială reprezentând chiar nodurile arborelui de sensuri ale intrării de dicționar respective, deci arborele de sensuri. Ulterior se poate reveni la parsarea definițiilor cuprinse în sensuri, una câte una. Această separare a obținerii arborelui de sensuri chiar înainte de parsarea definițiilor ce reprezintă conținutul acestor sensuri poate creşte substanțial procentajul articolelor parsate deoarece poate accepta articole DTLR ale căror definiții pe (sub)sensuri, considerate individual, să nu fie complet parsabile.

Mulțumiri. Cercetarea prezentată în acest articol a fost finanțată prin grantul PNCDI 2 eDTLR.

Referințe bibliografice

Cristea, D., Răschip, M., Forăscu, C., Haja, G., Florescu, C., Aldea, B., Dănilă, E. (2007). The Digital Form of the Thesaurus Dictionary of the Romanian Language. In Proceedings of the 4th International IEEE Conference SpeD 2007

N. Curteanu, E. Amihăesei. (2004). Grammar-based Java Parsers for DEX and DTLR Romanian Dictionaries. *ECIT-2004 Conference*, Iasi, Romania.

N. Curteanu, D. Trandabăț, G. Pavel, C. Vereştiuc, C. Bolea (2007). Raport ştiinţific şi tehnic la proiectul PNCDI II eDTLR- Dicționarul Tezaur al Limbii Române în format electronic, faza 2007

Dănilă, Elena. (dec. 2007). Comunicare personală.

Kilgarriff, Adam. (1999). Generic encoding principles. *CONCEDE Project Deliverable 2.1*, University of Brighton, UK

Normele de redactare a DLR (1952). Institutul de Filologie al Academiei Române, Colectivul de revizie a DLR.

Tomaž Erjavec, Roger Evans, Nancy Ide and Adam Kilgariff. (2000). The CONCEDE Model for Lexical Databases. *Research Report on TEI-CONCEDE LDB Project*, Univ. of Ljubljana, Slovenia.

Dan Tufiş (2001). *From Machine Readable Dictionaries to Lexical Databases*, RACAI, Romanian Academy, Bucharest, Romania.

Dan Tufiş, Ana-Maria Barbu. (2001). Computational bilingual lexicography: automatic extraction of translation dictionaries, In Romanian Journal on Information Science and Technology, vol. 4, no. 3

CAPITOLUL 3

APLICAȚII ALE TEHNOLOGIILOR LINGVISTICE

DESCOPERIREA RELAȚIILOR ÎNTRE ENTITĂȚI DE TIP NUME FOLOSIND WIKIPEDIA ÎN LIMBA ROMÂNĂ

ADRIAN IFTENE[1], ALEXANDRA BALAHUR-DOBRESCU[1,2]

[1]*Universitatea "Al.I.Cuza", Facultatea de Informatică, Iaşi – România;*

[2]*Universitatea Alicante, Departamentul de Limbaje şi Sisteme Informatice, Alicante-Spania;*

{adiftene, abalahur}@info.uaic.ro

Rezumat

Descoperirea relațiilor dintre entitățile de tip nume din resurse mari de informație este atât o provocare, cât şi o activitate utilă în sfera procesării limbajului natural, în cadrul unor aplicații cum sunt căutarea documentară, sumarizarea, găsirea răspunsurilor la întrebări puse în limbaj natural şi în realizarea inferențelor textuale.

Ceea ce o să prezentăm în continuare, a rezultat din încercarea de a găsi soluții viabile la problemele care au intervenit în timpul construirii unor sisteme înscrise în competiții precum recunoaşterea inferențelor textuale, respectiv găsirea răspunsului la întrebări puse în limbaj natural.

1. Introducere

În cadrul competiției de recunoaştere a inferențelor textuale (Dagan et al., 2006), provocarea constă în stabilirea faptului dacă un text (denumit ipoteză - *I*) poate fi dedus sau nu dintr-un text mai lung (denumit text - *T*). Performanțele sistemului construit de noi la competiția de anul acesta au depins foarte mult de regula privind prezența entităților de tip nume din ipoteză sau a unor forme echivalente ale acestora printre entitățile de tip nume din text. Din acest motiv, după marcarea entităților de tip nume în cele două texte folosind Lingpipe[16], sistemul verifică dacă toate entitățile de tip nume din *I* se găsesc de asemenea în *T*, şi dacă acest lucru nu se întâmplă, se foloseşte un modul care încearcă să găsească o legătură între entități. Modulul achiziționează semi-automat o colecție de legături între entitățile de tip nume sub forma unei cunoaşteri suplimentare. Această cunoaştere nu exista disponibilă sub forma unei resurse, aşa că obținerea ei a devenit o cerință practică a cărei rezolvare a dus la o creştere semnificativă a performanțelor sistemului.

În cadrul competiției QA@CLEF de găsire a răspunsurilor la întrebări formulate în limbaj natural, am folosit atât o bază de date de acronime cât şi o resursă de cunoaştere suplimentară cu relații între entitățile de tip nume. Aceasta din urmă ne-a permis expandarea entităților de tip nume din întrebare, mărind în acest fel probabilitatea sistemului nostru de a găsi răspunsul corect. Această expansiune pare a fi utilă, nu doar în cadrul acestei competiții, în care întrebările au fost formulate folosind Wikipedia ca resursă, ci şi în sistemele de întrebare-răspuns din lumea reală. Acest fapt vine din faptul că entitățile de tip nume nu sunt prea des înlocuite cu omonimele lor (ca India cu Asia etc.), şi din faptul că întrebările utilizatorului nu folosesc tot timpul exact aceleaşi

[16] Lingpipe – http://www.alias-i.com/lingpipe/

entități de tip nume ca acelea care există în documentele din care poate fi extras răspunsul.

Abordarea noastră este similară celei descrise în (Hasegawa et al., 2004), care caută relații între entități de tip nume din resurse mari de informație precum Wikipedia sau Web-ul fără a avea o adnotare în prealabil a acestora. De asemenea, am folosit un identificator de entități de tip nume care împreună cu o taxonomie ne-a ajutat în clasificarea entităților, spre deosebire de (Weaver et al., 2006) care au făcut clasificarea pe baza unor clase predefinite. Programul nostru are două opțiuni: fie extrage din Wikipedia fragmente de text care conțin o entitate de tip nume specificată, fie extrage o listă de entități de tip nume care au legătură cu entitatea specificată. Pentru ambele cazuri am construit relații între entitățile de tip nume identificate şi am încercat să evaluăm rezultatele. În plus, pentru a descoperi relațiile dintre entitățile din fragmentele de text extrase, am folosit o gramatică care identifică contextele de definiții.

În continuare, în capitolul 2 vom prezenta modalitatea în care am extras entitățile de tip nume relativ la o entitate de start. Capitolele 3 şi 4 vin cu două moduri de evaluare a muncii noastre: calitativ (verificând câte din rezultatele extrase sunt corecte) şi cantitativ (prin comparație cu WordNet-ul).

2. Extragerea entităților de tip nume din Wikipedia relativ la o entitate de tip nume

Pentru o entitate dată, utilizăm un modul special construit de noi pentru a extrage din Wikipedia[17] fragmente de text cu informații care au legătură cu ea. În fragmentele de text extrase din Wikipedia identificăm contextele de definiții, urmând aceeaşi idee ca în (Iftene et al., 2007). Pentru fiecare astfel de context de definiție:

a) Identificăm "nucleul" definiției, care este fie verbul "a fi" sau alt verb care introduce o definiție sau un semn de punctuație care introduce o definiție;

b) Extragem din partea stângă a "nucleului": toate entitățile de tip nume (entități stânga);

c) Extragem din partea dreaptă a "nucleului": toate entitățile de tip nume (entități dreapta);

d) Calculăm produsul cartezian dintre entitățile stânga şi entitățile dreapta şi adăugăm perechile rezultate la baza de date cu rezultate existentă.

De exemplu pentru entitatea de tip ORAŞ Oradea fişierul cu fragmente de text arată ca în tabela următoare:

Tabela 1: Fragmente de text extrase pentru Oradea

'Oradea', mai demult 'Oradea Mare', este un municipiu situat în vestul României, pe râul Crişul Repede, ... În imediata apropiere a graniței cu Ungaria, Oradea, reşedință de județ a

[17] http://en.wikipedia.org/wiki/Main_Page

Bihorului, ... acesta fiind cel mai important oraş din regiune istorică Crişana ...

Pentru primul fragment identificăm următoarele elemente: nucleul este verbul "a fi" care apare la prezent, entitățile din partea dreaptă sunt "Oradea" şi "Oradea Mare", iar în partea dreaptă entitățile "România" şi "Crişul Repede" (vezi figura de mai jos):

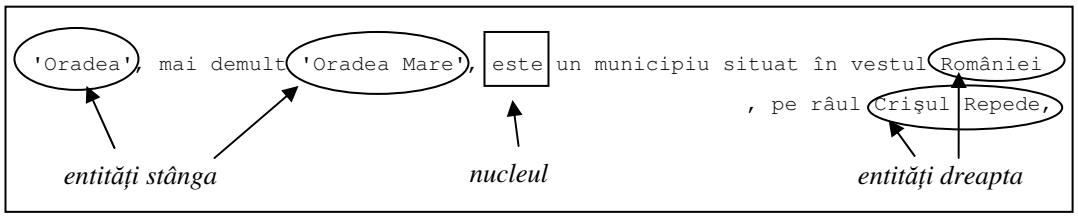


Figura 1: Identificarea nucleului şi a entităților pentru localitatea Oradea

În urma produsului cartezian se generează următoarele relații:

Tabela 2: Relațiile identificate pentru localitatea Oradea

Oradea [in] România
Oradea Mare [in] România
Oradea [in] Crişul Repede
Oradea Mare [in] Crişul Repede

Alte exemple obținute folosind această metodă sunt prezentate în tabelul următor:

Tabela 3: Alte exemple de relații identificate în Wikipedia românească

Iaşi [in] România
Eminescu [in] Iaşi
Moldova [is] Republica Moldova
August [is] Gustar

După cum s-a observat, şabloanele create identifică două tipuri de relații între entitățile de tip nume:

- "is", atunci când modulul extrage informații folosind identificarea contextelor de definiții;
- "in", atunci când în contextele de definiții avem în plus cuvinte de forma ***în, localizat în, de la, din, regiune, etc***.

La finalizarea extragerii tuturor entităților de tip nume corelate cu o entitate inițială, vom spune că am obținut primul nivel de entități de tip nume pentru entitatea inițială. Mai apoi, extragem pentru fiecare dintre entitățile de tip nume de pe primul nivel entitățile de tip nume corelate cu acestea şi obținem al doilea nivel de entități de tip nume. Acest proces de extragere continuă până în momentul în care nu mai obținem noi entități pentru o entitate inițială.

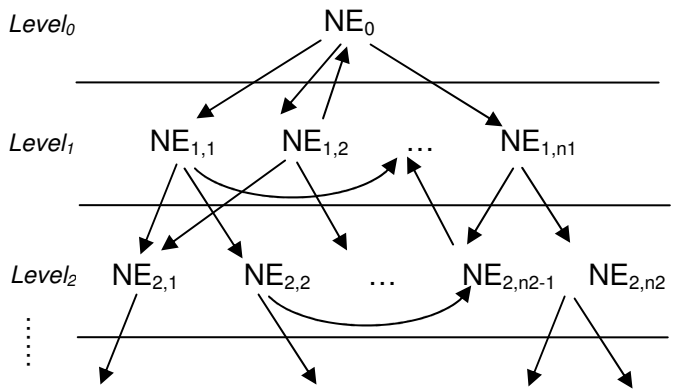


Figura 2: Nivelurile obținute pornind de la o entitate de start

Situațiile care pot apare:

1. pentru o entitate de tip nume avem entități corelate pe nivelul următor;
2. pentru o entitate de tip nume este posibil să avem entități corelate pe acelaşi nivel;
3. pentru o entitate de tip nume este posibil să avem entități corelate pe nivelul precedent.

Procesul recursiv progresează doar pentru cazul 1.

Un rezultat sugestiv a fost obținut atunci când am folosit ca entitate de pornire entitatea de tip localitate "*Iaşi*". Rezultatul parțial este arătat în figura 3 de mai jos.

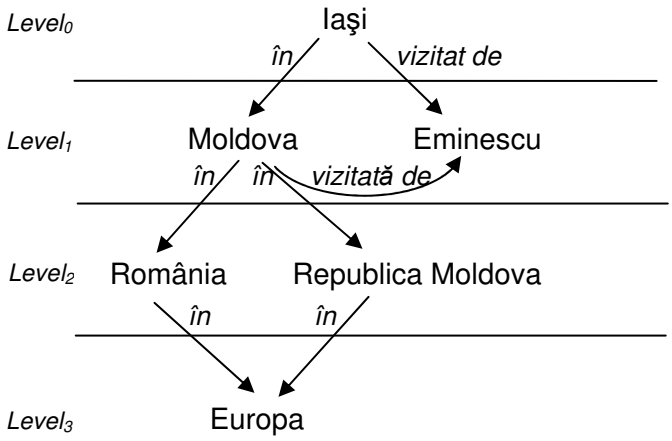


Figura 3: Nivelurile obținute pornind de la localitatea Iaşi

3. *Tipuri de relații între entitățile extrase şi entitatea inițială – evaluare calitativă*

Pentru entitățile extrase din Wikipedia care sunt corelate cu o entitate dată, vom realiza o clasificare şi grupare. Pentru a realiza acest proces, utilizăm GATE (General Architecture for Text Engineering) setat pe limba română (Hamza et al., 2002) şi identificăm următoarele tipuri de entități de tip nume: țară, oraş, ocean, mare, râu, munte, regiune (normală şi extinsă), limbă, monedă, nume de persoană, organizație, şi slujbă.

Clasificarea depinde de tipul entității de tip nume inițială şi tipul entităților corelate cu aceasta. De exemplu, pornind de la entitatea de start "*România*", care este *Țară* am extras 324 entități din care 213 entități diferite (am specificat frecvența de apariție a entităților prin notația la putere, atunci când aceasta este mai mare de o apariție):

- *Monezi*: *Leu* şi *Ban*, care sunt monezile care sunt folosite în prezent în *România*.
- *Persoane*: preşedinți ai României (*Emil Constantinescu, Gheorghe Gheorghiu-Dej, Nicolae Ceauşescu*[2]*, Ion Iliescu*[4]*, Traian Băsescu*[5]*, Constantin Parhon*), regi ai României (*Burebista*[2]*, Carol I*[2], *Carol al II-lea al României, Decebal, Ferdinand I al României, Mihai I*[2]), prim-miniştri ai României (*Călin Popescu-Tăriceanu, Nicolae Văcăroiu, Nicolae Iorga, Petru Groza, Ion Gheorghe Maurer, Chivu Stoica*[2]), sportivi români de renume (*Dorinel Munteanu, Gheorghe Hagi, Ilie Năstase, Ion Țiriac, Nadia Comăneci*), scriitori români de renume (*Andrei Mureşanu*[2], *Mihai Eminescu, Mihail Sadoveanu, Mircea Eliade, Emil Cioran, Eugen Ionesco, Panait Istrati, Anton Pann*), compozitor român (*Ciprian Porumbescu*[2]), general român (*Ioan Sion*). În două cazuri entitățile nu au fost Persoane care au trăit în România: *Constanța*[3] (care este de fapt un oraş din România), iar *Woodrow Wilson* (care este cel de-al 28-lea preşedinte al Statelor Unite ale Americii).
- *Regiuni extinse*: corecte (*Europa de Est, Peninsula Balcanică, Europa*[4]), incorecte (*Europa Occidentală, Europa Centrală*).
- *Oraşe*: corecte (*Aiud, Alba Iulia, Arad, Băile Herculane, Baziaş, Borzeşti, Braşov*[6]*, Brăila*[2]*, Bucureşti*[16]*, Cluj-Napoca*[5]*, Craiova*[2]*, Făgăraş*[2]*, Focşani, Galați*[3]*, Hunedoara, Iaşi*[5]*, Iernut, Mangalia, Miercurea Ciuc, Nădlac*[2]*, Oradea, Piatra Neamț, Ploieşti*[2]*, Râmnicu Vâlcea, Reghin, Sarmizegetusa, Sfântu-Gheorghe, Sibiu, Sighişoara*[2]*, Slatina, Târgu-Jiu, Târgu-Mureş, Timişoara*[4]), incorecte (*Bratislava, Budapesta, Belgrad, Viena*). Interesant de observat este faptul că frecvența cea mai mare o are *Bucureştiul* care este capitala țării, iar următoarele oraşe ca frecvență sunt oraşe mari din România. De asemenea, se observă ca toate oraşele incorecte au frecvența de 1.
- *Munți*: valori corecte (*Carpați, Carpații de Curbură, Carpații Meridionali*[2]*, Carpații Occidentali, Carpații Orientali*[2]*, Munții Apuseni, Munții Buzăului, Munții Făgăraş, Munții Hăşmaş, Munții Harghita, Munții Vrancei, Vârful Moldoveanu*), valori incorecte (*Munții Pădurea Neagră, Munții Ural*).

- *Organizații*: partide politice din România (A*lianța D.A., Alianța dreptate şi Adevăr, Consiliul Frontului Salvării Naționale*), organizații din care România face parte (*NATO*[3]*, ONU, Organizația Națiunilor Unite, OSCE, UE*[2]*, Uniunea Europeană*[4]), alte organizații din România (*Palatul Parlamentului, Universitatea din Bucureşti*).
- *Regiuni*: corecte (*Țara Almăjului, Țara Moților, Țara Românească, Bărăgan, Banat*[2]*, Biserica Neagră, Bucovina*[2]*, Câmpia de Vest, Câmpia Română, Crişana*[2]*, Dobrogea*[2]*, Harghita*, *Maramureş, Mehedinți, Moldova*[4]*, Muntenia, Oltenia, Transilvania*[7]*, Valahia*[2]*, Valea Prahovei,* 41 de județe) , parțial corectă (*Basarabia* care a fost inclusă la un moment dat în România).
- *Râuri*: corecte (*Argeş*[2]*, Delta Dunării*[4]*, Dunăre*[2]*, Dunărea*[2]*, Prut, Râul Argeş, Râul Ialomița, Râul Jiu*[2]*, Râul Mureş, Râul Olt*[2]*, Râul Prut, Râul Siret, Râul Someş, Râul Timiş, Tisa*[2]), incorecte (*Nistru, Volga*).
- *Țări*: denumiri mai vechi ale României (*Republica Populară Română, Republica Socialistă România,*), țări vecine României (*URSS, Uniunea Sovietică*[2]*, Ucraina*[3]*, Serbia*[4]*, Bulgaria*[4]*, Ungaria*[5]*, Republica Moldova*[5]), altele (*Austria, Slovacia, Franța, Luxemburg, Croația, Germania*). Se observă că țările vecine României sunt cele care au frecvența de apariție cea mai mare.
- *Mări*: mare vecină României (*Marea Neagră*[2]), altele (*Marea Marmara, Marea Adriatică, Marea Egee*).
- *Limbi*: vorbite pe teritoriul României (*limbile indo-europene, latina vulgară, limba germană, limba română*[2]*, limba sârbă, limbi romanice*), altele (*limba engleză, limba franceză*).

Pentru obținerea rezultatelor din tabelul de mai jos am folosit peste 1000 de perechi de forma (*entitate de start, entitate extrasă*), pe care le-am evaluat manual. În această primă fază ne-am concentrat mai mult pe identificarea tipurilor de relații care pot fi extrase, iar apoi ne-am concentrat doar pe relațiile pentru care precizia era cât mai mare. Evident în anumite cazuri se pot deduce informații suplimentare (cum am observat în exemplul de mai sus că la oraşe, oraşul cu frecvența cea mai mare este chiar capitala României), dar pentru a putea generaliza astfel de relații trebuie să testăm pe mult mai multe entități de acelaşi tip.

Tabela 4: Tipuri de relații identificate între entități

Tipul entității inițiale	**Tipul entităților corelate**	**Relație**	**Precizie**
Țară	Persoană	Persoană <a fost în> Țară	98 %
	Regiune	Regiune <inclusă în> Țară	100 %
	Regiune Extinsă	Țară <inclusă în> Regiune extinsă	75 %
	Țară	Țară <vecină cu> Țară	82 %
	Limbă	Limbă <vorbită în> Țară	56 %
	Moneda	Moneda <este moneda din> Țară	100 %
	Mare	Țară <vecină cu> Mare	40 %
	Râu	Râu <inclus în> Țară	92 %

Tipul entității inițiale	**Tipul entităților corelate**	**Relație**	**Precizie**
	Oraş Organizație	Oraş <inclus în> Țară Organizație <in> Țară	95 % 88 %
Organizație	Țară Limbă	Țară <component al> Organizație Limbă <vorbită în> Organizație	85 % 90 %
Persoană	Persoană Oraş Slujbă Limbă	Persoană <a auzit de> Persoană Persoană <a fost în> Oraş Persoană <a lucrat> Slujbă Persoană <vorbeşte în> Limbă	100 % 94 % 70 % 100 %
Oraş	Regiune Râu Limbă Persoană	Regiune <inclusă în> Oraş Râu <traversează> Oraş Limbă <vorbită în> Oraş Persoană <a trecut prin> Oraş	100 % 25 % 50 % 100 %

Unde precizia a fost calculată în felul următor:

$$precizia = \frac{\sum numar\_aparitii\_entitate}{\sum_{toate\_entitatile} numar\_aparitii\_entitate}$$

De exemplu, pentru țara România cele 4 regiuni extinse extrase au următoarele numere de apariții:

- Europa Occidentală – 1 apariție
- Europa de Est – 1 apariție
- Peninsula Balcanică – 1 apariție
- Europa Centrală – 1 apariție
- Europa – 4 apariții

Deoarece corecte sunt doar Europa de Est, Peninsula Balcanică şi Europa, precizia în acest caz este:

$$precizia = \frac{1+1+4}{1+1+1+1+4} = \frac{6}{8} = 0.75$$

Relații specifice

În testările pe care le-am făcut pentru entitățile de tip *Țară* am observat că pentru entitățile extrase avem următoarele tipuri de relații specifice: orașul cu cea mai mare frecvență este *capitala țării*, iar următoarele orașe cu frecvențe mari sunt *orașele mari*, țările cu cea mai mare frecvență sunt *țările vecine*, iar persoana cu cea mai mare frecvență este *președintele țării*.

Pentru entitățile de tip *Persoană*, putem deduce de asemenea informații adiționale: orașul și țara cu cea mai mare frecvență sunt *orașul natal* și respectiv *țara natală* ale acesteia.

Pentru unele cazuri este dificil să identificăm relația corectă dintre entități de tip nume. De exemplu, pentru două entități de tip oraş, nu este posibil să precizăm corect ce relație există între ele.

4. Comparație cu WordNet-ul românesc – evaluare cantitativă

După cum se observă în capitolul precedent rezultatele extrase prezintă destul de bine relațiile dintre entitățile de tip nume. Întrebările pe care ni le punem în continuare sunt: *Entitățile de tip nume care au legătură cu o altă entitate de tip nume au fost extrase destul de bine, relațiile dintre entități sunt identificate cu o precizie suficient de mare, dar sunt ele suficiente*? *Câte entități am pierdut din vedere folosind Wikipedia*? Pentru a răspunde la aceste întrebări am luat câteva exemple, şi pentru rezultatele obținute din Wikipedia am încercat să obținem aceleaşi tip de informații folosind WordNet-ul românesc, iar în final am comparat rezultatele. Pentru următoarele teste am considerat entitatea de tip Organizație "Uniunea Europeană".

Tabela 5: Diferențele dintre Wikipedia şi WordNet pentru entitatea Uniunea Europeană

Entitate de tip Nume	**Lipsă în ambele**	**Corecte în WordNet şi în Wikipedia**	**În plus**	
			Corecte	**Greşite**
Uniunea Europeană	Albania, Bosnia şi Herțegovina, Muntenegru, Serbia	Anglia, Austria, Danemarca, Franța, Germania, Portugalia, Spania, Suedia	Belgia, Bulgaria, *Croația,* Cipru, Cehia, Finlanda, Estonia, Grecia, Ungaria, Irlanda, Italia, Letonia, Lituania, Luxemburg, Malta, *Macedonia,* Olanda, România, Slovacia, Slovenia, *Turcia*	Elveția, Moldova, Islanda, Liechtenstein, Norvegia, Statele Unite

Ce avem în plus în Wikipedia? Țările membre din Uniunea Europeană care au aderat în anul 2004 ca *Cipru, Estonia, Ungaria*, apoi țările care au aderat în 2007 precum *România, Bulgaria*, şi cele trei noi candidate recunoscute: *Croația, Macedonia* şi *Turcia*. Ce lipsesc din ambele resurse sunt candidatele potențialele recunoscute oficial: *Albania, Bosnia şi Herțegovina, Muntenegru* şi *Serbia*, care în schimb se găsesc în Wikipedia englezească.

Legat de frecvența apariției țărilor în rezultatele noastre am observat că valorile cele mai mari sunt obținute pentru țările care apar atât in Wikipedia cât şi-n WordNet (a căror frecvență este mult mai mare în colecția folosită din Wikipedia) şi valorile cele mai mici corespund valorile în plus greşite (care apar accidental în colecție). De asemenea, se mai poate observa că am extras de patru ori mai multă informație din Wikipedia decât din WordNet, şi doar 15 % din această informație este greşită. O altă problemă de care am depins a fost calitatea resursei GATE care a fost folosită de programul nostru pentru a extrage entitățile de tip nume.

5. Concluzii

Acest articol prezintă metoda pe care am utilizat-o pentru determinarea de relații dintre entitățile de tip nume, utilizând corpusul Wikipedia. Rezultatele preliminare demonstrează o calitate şi cantitate bună a informațiilor extrase şi, de asemenea, arată modul în care o resursă precum WordNet nu poate acoperi în timp real întreaga suită de schimbări ce au loc în plan mondial. Ideea utilizării enciclopediei Wikipedia s-a născut din necesitatea de a construi astfel de resurse uşor adaptabile la schimbare pentru o gamă largă de limbi şi motivată de faptul că Wikipedia este accesibilă gratuit în mai mult de 253 de limbi, având peste 10 milioane de utilizatori. WordNet este o resursă importantă şi extrem de utilă, dar există doar pentru 15 limbi, iar numărul de sinseturi este scăzut pentru majoritatea limbilor exceptând engleza. Metoda prezentată este independentă de limbă şi poate fi aplicată pentru articole din Wikipedia în orice limbă. Problema cea mai mare rămâne însă calitatea informațiilor extrase şi de aceea, pentru a îmbunătăți calitatea se poate folosi suplimentar şi WordNet-ul.

În viitor, dorim să testăm calitatea informațiilor extrase pentru mai multe tipuri de entități de tip nume şi un volum mai mare de date. Mai apoi, vom construi o resursă generală ce va fi utilizată în sistemele de tip Întrebare-Răspuns, pentru găsirea formelor echivalente a entităților de tip nume din întrebare sau atunci când dorim să extragem din fişiere entități de tip nume corelate cu cele din întrebare. Această abordare va ameliora calitatea sistemelor ÎR, aşa cum demonstrează testele preliminare pe care le-am realizat pentru limba română.

Mulțumiri. Autorii mulțumesc membrilor grupului de lingvistică computațională din Iaşi pentru ajutorul şi ideile oferite de-a lungul lucrului la acest proiect.
Lucrul din cadrul acestui proiect este parțial finanțat de Siemens VDO Iaşi, de proiectul CEEX Rotel numărul 29 şi de proiectul FP6 LT4eL (Learning Technologies for e-Learning).

Referințe bibliografice

Dagan I., Glickman O., Magnini B. (2006). The PASCAL Recognising Textual Entailment Challenge. *In Quiñonero-Candela et al., editors, MLCW 2005, LNAI Volume 3944*. Springer-Verlag 177-190.

Hamza, O., Tablan, V., Maynard, D., Ursu, C., Cunningham, H. and Wilks, Y. (2002). Name entity recognition in Romanian. *Technical report, Department of Computer Science, University of Sheffield*. Forthcoming

Hasegawa T., Sekine S., Grisham R. (2004). Discovering Relations among NEs from Large Corpora. *Proceedings of ACL 2004 Conference*.

Iftene, A., Balahur-Dobrescu, A. (2007). Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment. *In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Pp.125-130. 28-29 June, Prague, Czech Republic.

Iftene, A., Trandabăț, D. and Pistol, I. (2007). Grammar-based Automatic Extraction of Definitions and Applications for Romanian. *In Proceedings of RANLP workshop*

"Natural Language Processing and Knowledge Representation for eLearning environments". September 26, Borovets, Bulgaria.

Liu, B., Chin C. W., and Ng H. T. (2003). Mining Topic-Specific Concepts and Definitions on the Web. *Proceedings of the Twelfth International World Wide Web Conference* (WWW'03).

Weaver G., Strickland B., Crane G. (2006). Quantifying the Accuracy of Relational Statements in Wikipedia: A Methodology. *In JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*.

REALIZAREA INFERENȚELOR TEXTUALE PE LIMBA ROMÂNĂ

ADRIAN IFTENE[1], ALEXANDRA BALAHUR-DOBRESCU[1,2]

[1] Universitatea "Al.I.Cuza", Facultatea de Informatică, Iaşi – România;

[2]Universitatea Alicante, Departamentul de Limbaje şi Sisteme Informatice, Alicante-Spania;

{adiftene, abalahur}@info.uaic.ro

Rezumat

Informațiile dintr-un corpus pot fi reprezentate într-o varietate de forme. Sistemele de tip întrebare-răspuns (ÎR) trebuie să rezolve variabilitatea semantică, şi să identifice într-o anumită colecție de date răspunsul la o anumită întrebare. O soluție potrivită la această problemă constă în folosirea unui sistem de inferențe textuale (SIT) care să implementeze paşii descrişi în (Bar-Haim et al., 2006). Anul acesta în cadrul competiției QA@CLEF[18], ne-am confruntat cu problema variabilității semantice şi ne-am decis să includem în sistemul nostru de ÎR un modul care să se ocupe de rezolvarea inferențelor textuale. Rezultatele au fost încurajatoare ducând la o creştere semnificativă a preciziei. Prin urmare, am decis să construim un SIT pentru limba română care să fie la rândul lui parte componentă a unui sistem de ÎR pe limba română.

1. Introducere

Recunoaşterea inferențelor textuale (Textual entailment recognition) RTE[19] (Dagan et al, 2005) este o competiție, în care, fiind date două fragmente de text se cere precizarea dacă înțelesul unuia din texte poate fi dedus din celălalt text. Scopul acestei competiții este de a crea o platformă independentă de aplicație, care să fie capabilă să identifice inferențele semantice atât de folositoare în aplicațiile din lingvistica computațională. Exemple de astfel de aplicații sunt: căutarea documentară (Information Retrieval - IR), sistemele de tip întrebare răspuns (Question Answering - QA), extragerea informației (Information Extraction - IE), şi sumarizarea textuală (Text Summarization - SUM).

Formal, **inferența textuală** – IT - (Textual Entailment) definită în (Dagan et al, 2005) este o relație unidirecțională între două fragmente de text, denumite *T* – textul, şi *H* - ipoteza. Se spune că din *T* se poate infera *H* dacă, un om care citeşte *T* poate infera faptul că *H* este de regulă adevărată. Această definiție se bazează pe faptul (şi presupune) cunoaşterea umană a limbajului şi cunoaşterea suplimentară despre lume.

Sistemele de IT participă în fiecare an în competiția RTE, organizată de PASCAL[20] (Pattern Analysis, Statistical Modelling and Computational Learning), o comisie Europeană de excelență. Acest an, în cadrul competiției RTE3, am participat pentru prima dată în această competiție cu un sistem realizat pe limba engleză.

[18] CLEF: http://clef-qa.itc.it/

[19] RTE: http://www.pascal-network.org/Challenges/RTE/

[20] PASCAL: http://www.pascal-network.org/

Construirea **sistemelor de ÎR** este una din direcțiile cele mai importante în procesarea limbajului natural. Aceste sisteme presupun atât analiza discursului cât şi unelte de procesare avansate, cât şi studii teoretice şi formalizări ale problematicilor limbajului, precum structura întrebărilor şi folosirea cunoaşterii. Sistemele de ÎR primesc întrebările în limbaj natural şi trebuie să găsească pentru fiecare din ele răspunsul exact, nepermițând întoarcerea unui întreg document ca răspuns. Găsirea răspunsului la o întrebare implică prin urmare două lucruri: identificarea informației necesare şi cantitatea, plus calitatea acestei informații.

Informațiile dintr-un corpus pot fi reprezentate într-o varietate de moduri. Sistemele de tip întrebare-răspuns trebuie să rezolve problema variabilității semantice, şi trebuie să identifice textele din care pot fi deduse răspunsurile aşteptate. O soluție acceptabilă la această problemă poate fi folosirea unui SIT şi realizarea paşilor descrişi în continuare:

Pornind de la întrebarea descrisă în (Bar-Haim et al., 2006) se realizează următorii paşi:

Întrebare: "*Cine este văduva lui John Lennon?*"

Putem obține o expresie cu variabila *PERSOANĂ*:

Expresie: *PERSOANĂ este văduva lui John Lennon*.

Printre fragmentele de text ce conțin cuvintele cheie *John Lennon*, putem găsi:

Fragment de Text: "*Yoko Ono a dezvelit o statuie de bronz a fostului ei soț, John Lennon, şi pentru a încheia redenumirea oficială a aeroportului englez Liverpool în aeroportul John Lennon din Liverpool.*" Din acest text, o valoare posibilă pentru variabila *PERSOANĂ* poate fi – *Yoko Ono*. Ipoteza poate fi construită prin înlocuirea variabilei *PERSOANĂ* cu valoarea posibilă.

Ipoteza: "*Yoko Ono este văduva lui John Lennon*".

Obținerea faptului că am făcut alegerea potrivită, şi a faptului că această variabilă poate fi răspunsul corect al întrebării, se face prin evaluarea relației de inferență dintre Text şi Ipoteză.

Una din competițiile în care sunt implicate sistemele de tip Întrebare-răspuns se organizează în cadrul CLEF (Cross-Language Evaluation Forum). CLEF are ca principală preocupare dezvoltarea librăriilor digitale prin crearea unei infrastructuri pentru testare, îmbunătățire şi evaluare a sistemelor de căutare documentară în limbile din Europa, atât în forma mono-lingvă cât şi în forma multi-lingvă. În cadrul exercițiului de evaluare QA@CLEF, am participat încă din anul 2006 cu un sistem multilingv român-englez. Deoarece ne-am confruntat cu problema variabilității semantice, am decis să introducem începând cu acest an sistemul de inferențe textuale de pe limba engleză în sistemul de ÎR.

Rezultatele obținute folosind acest modul de inferențe textuale în cadrul sistemului de ÎR au fost încurajatoare, ducând la o creştere semnificativă a preciziei. Prin urmare, am decis să construim un SIT pe limba română care să poată fi folosit în cadrul sistemului ÎR românesc. În cele ce urmează vom descrie componentele sistemului funcționând pe limba engleză şi paşii pe care i-am parcurs pentru a adapta aceste componente pe limba română.

2. *Sistemul de inferenţe textuale Românesc*

În **SIT-ul englezesc** construit de noi anul acesta pentru competiţia RTE3[21], ideea principală este de a transforma ipoteza folosind cunoaşterea semantică din resurse precum DIRT (Lin and Pantel, 2001), WordNet, Wikipedia, o baza de date de acronime. În plus, am construit un sistem capabil să achiziţioneze cunoaştere suplimentară din Wikipedia englezească. De asemenea, rularea sistemului necesită o parte de pre-procesare realizată cu MINIPAR (care construieşte arborii de dependenţă asociaţi textului şi ipotezei) (Lin, 1998) şi cu LingPipe[22] (care identifică entităţile de tip nume din text şi ipoteză), urmată de încercarea găsirii distanţei minime dintre arborii asociaţi (Kouylekov, Magnini, 2005).

După terminarea competiţiei am construit un **SIT românesc** care să poată fi inclus într-un sistem de ÎR. Pentru a putea face acest lucru am înlocuit majoritatea componentelor din sistemul englezesc cu variante ale acestora care funcţionează pe limba română.

2.1 GATE

Am pornit cu identificarea entităţilor de tip nume, unde am folosit **GATE[23] setat pe limba română**, obţinând o lista cu entităţi de tip nume specifice limbii române. Deoarece, în sistemul englezesc am avut o regulă care se ocupa cu identificarea numelor de entităţi, numere şi date, şi a cărei folosire a dus la o îmbunătăţire a preciziei sistemului cu 16 %, am acordat o atenţie deosebită acestui modul.

În (Hamza et al., 2002), un sistem de identificare a entităţilor de tip nume a fost dezvoltat pe limba română folosind ANNIE, componenta centrală a sistemului de entităţi de tip nume pe limba engleză construit în cadrul arhitecturii GATE, şi prezentată în (Maynard et al., 2001). Sistemul de identificare a entităţilor de tip nume pe limba română foloseşte împărţirea în cuvinte, un dicţionar geografic şi un modul cu reguli gramaticale din ANNIE.

Însă, deoarece multe din întrebările din cadrul competiţiei QA@CLEF au inclus nume de scriitori ai literaturii universale sau personalităţi universale, am rulat şi cu GATE setat pe limba engleză, iar în final am considerat ambele mulţimi de entităţi.

2.2 Acronime

Baza de date cu acronime ne ajută să găsim relaţii între acronim şi semnificaţia lui: "*UE – Uniunea Europeană*". Pentru a găsi **acronimele pentru limba română** am extras automat o listă de acronime dintr-o colecţie de ziare româneşti cu articole din economie şi politică folosind un algoritm asemănător celui prezentat în (Shinyama et al., 2002). De asemenea, am folosit o listă de acronime româneşti de pe Internet[24].

[21] Competiţia RTE3: http://www.pascal-network.org/Challenges/RTE3

[22] Lingpipe: http://www.alias-i.com/lingpipe/

[23] GATE: http://gate.ac.uk/

[24] Acronime: http://www.abbreviations.com/acronyms/ROMANIAN

2.3 Cunoaşterea suplimentară

Cunoaşterea suplimentară pentru entitățile de tip nume şi pentru numere a fost construită semi-automat pentru entitățile din ipoteză fără corespondent în text. Pentru acestea, am folosit un modul asemănător celui din (Iftene, Balahur, 2007) care extrage într-un fişier fragmente de text din Wikipedia[25], dar care are setată limba pe română şi care foloseşte Wikipedia[26] românească.

În continuare am folosit fişierul cu fragmentele extras mai sus şi şabloane cu relații între entitățile de tip nume, cu scopul de a identifica relații cunoscute între entitatea cu probleme şi altă entitate. Aceste şabloane construite pentru limba română sunt asemănătoare şabloanelor construite pentru limba engleză, dar am adăugat în plus şabloane specifice limbii române. Aceste reguli vin în mare parte din regulile de extragere a contextelor de definiții româneşti descrise în (Iftene et al., 2007b).

După cum se poate observa în tabela 1, şabloanele noastre identifică două tipuri de relații între cuvinte:

- "is", când modulul extrage informații folosind exact regulile de identificare a contextelor de definiții româneşti;
- "in", când în plus informațiile extrase conțin cuvinte specifice precum: ***în, din, inclus, regiune*** etc.

Tabela 1: Cunoaşterea suplimentară

Bucureşti [in] România
American [in] America
America [is] Statele Unite ale Americii
II [is] Februarie
Chinez [in] China

Rezultatele pe limba română sunt incomplete, întrucât numărul articolelor Wikipedia în limba română este încă redus. De aceea, atunci când nu avem rezultate satisfăcătoare pe limba română folosim cunoaşterea suplimentară obținută pe limba engleză.

2.4 WordNet

WordNet-ul românesc (Tufiş et al., 2002) a fost folosit pentru a găsi sinseturile cuvintelor din ipoteză fără corespondent în text, urmând ca mai apoi să încercăm să găsim corespondent pentru acestea în text.

Datorită temerilor generale ale unor anumiți lexicografi, conform cărora simpla traducere a sinseturilor din WordNet-ul Princeton (Fellbaum, 1999) nu va avea ca rezultat un dicționar reprezentativ pentru limba vizată, în (Tufiş, 1999) a fost adoptată o metodă centrată pe limbă (în contrast cu o metodă mai simplă bazată pe traducerea cuvintelor din Princeton WordNet), bazată pe resurse lexicografice de referință: Dicționarul Explicativ al Limbii Române, Dicționarul de Sinonime, şi de asemenea un dicționar propriu român-englez.(Tufiş et al., 1999)

[25] Wikipedia englezească: http://en.wikipedia.org

[26] Wikipedia românească: http://ro.wikipedia.org

2.5 Regulile de variabilitate semantică: negații şi termeni contextuali

Regulile de variabilitate semantică pentru sistemul IT pentru limba engleză au inclus reguli de negație pentru termeni specifici ca "*no*", "*never*", "*don't*" etc., utilizarea verbelor modale în formă condițională sau folosirea verbelor la forma infinitivă. Alte reguli au ca scop surprinderea influenței pe care o au cuvintele ***pozitive*** asupra contextului – accentuarea înțelesului unui verb şi a influenței cuvintelor ***negative*** micşorarea gradului de probabilitate a acțiunii reprezentate de verb şi introducerea incertitudinii. În varianta sistemului pentru limba română, am identificat reguli de negație şi cuvinte care influențează contextul şi am introdus reguli similare.

Pentru **regulile de variabilitate semantică** am considerat negația cu următoarele cuvinte "*nu*", "*poate*" (care reprezintă forma pură de negație). De asemenea, subjunctivele au fost identificate prin faptul că sunt precedate de particula "*să*". În acest caz, dacă subjunctivul este precedat de un cuvânt precum "*permite, impune, indica, propune*" sau sinonimele lor, de adjective ca "*necesar*", "*obligatoriu*", "*liber*" sau sinonimele lor, sau substantive precum "*încercare*", "*posibilitate*", "*opțiune*" şi sinonimele lor, înțelesul devine pozitiv. Pentru cazul cuvintelor care influențează contextul, am construit, ca şi în cazul limbii engleze, două liste, una conținând cuvinte precum "*sigur*", "*absolut*", "*categoric*", "*cert*", "*precis*", "*inevitabil*", "*infailibil*" care accentuează certitudinea contextului şi "*probabil*", "*posibil*", "*fezabil*", "*realizabil*", "*practicabil*" – care micşorează certitudinea contextului.

2.6 Calcularea potrivirii globale

Ideea pentru calculul acestei valori constă în verificarea potrivirilor dintre cuvintele din ipoteză cu toate cuvinte din text, urmată de calcularea unei valori globale care reprezintă valoarea normalizată a sumei tuturor valorilor cuvintelor. Toate aceste calcule se fac după folosirea tuturor resurselor prezentate anterior: WordNet, baza de date de acronime, şi cunoaşterea suplimentară.

Ideea principală din cadrul abordării constă în determinarea cuvintelor cheie din ipoteză care se găsesc de asemenea în text şi marcarea pozițiilor în care se află. Cuvintele cheie reprezintă termenii din propoziție în afara stop word-urilor. Inițial asupra ipotezei se execută operațiile de tokenizare, POS-tagging, lematizare şi eliminarea stop word-urilor. Ceea ce rămâne reprezintă cuvintele cheie care se expandează, utilizând următoarele resurse: WordNet, baza de date de acronime şi cunoaşterea suplimentară.

De exemplu, pentru ipoteza:

H: *Ernest Hemingway, faimos romancier, nuvelist, realizator de povestiri American, a trăit între anii 1899 şi 1961.*

După eliminarea stop word-urilor, obținem următoarea listă de termini cheie, ce conține lemele cuvintelor din ipoteză:

{*Ernest Hemingway, faimos, romancier, nuvelist, realizator, povestire, American, trăi, an, 1899, 1961*}

Această listă este mai apoi expandată folosind WordNet-ul Românesc, iar rezultatul este următorul:

{*Ernest Hemingway, {faimos, celebru, excelent}, {romancier, scriitor}, nuvelist, {realizator, producător, creator, participant}, {povestire, mit, parabolă, narațiune}, American, {trăi, exista, viețui}, an, 1899, 1961*}

În faza următoare, lista expandată este completată utilizând cunoaşterea suplimentară. În această colecție, găsim *American [in] America* şi înlocuim *American* cu lista {*American, America*}.

În final, folosind colecția de acronime, expandăm încă o dată lista de termini pentru America cu *US* şi *USA*.

Lista completă rezultată este: {*Ernest Hemingway, {faimos, celebru, excelent}, {romancier, scriitor}, nuvelist, {realizator, producător, creator, participant, autor}, {povestire, mit, parabolă, narațiune}, {American, America, US, USA}, {trăi, exista, viețui}, an, 1899, 1961*}.

Rolul "textului" pentru sistemul de inferențe textuale este jucat de fragmentul de text următor rezultat în urma interogării cu Lucene[27]: "*Ernest Hemingway (n.21 iulie 1899 - d.2 iulie 1961), faimos romancier, nuvelist, realizator de povestiri (short stories în limba engleză), reporter de război, laureat al Premiului Pulitzer în 1953, laureat al Premiului Nobel pentru Literatură în 1954, probabil cel mai cunoscut autor american în întreaga lume.*"

Folosind această listă, construim o matrice care conține aparițiile cuvintelor din ipoteză în textul fără stop word-uri:

Tabela 2: Maparea ipotezei pe text

Nr.	Cuvânt	Poziții în Text
1	*Ernest Hemingway*	**1**
2	*faimos, celebru, excelent*	**8, 30**
3	*romancier, scriitor*	**9**
4	*nuvelist*	**10**
5	*realizator, producător, creator, participant, autor*	**11, 31**
6	*povestire, mit, parabolă, narațiune*	**12**
7	*American, America, US, USA*	**32**
8	*trăi, exista, viețui*	**-**
9	*an*	**-**
10	*1899*	**4**
11	*1961*	**7**

Formula pentru calcularea **potrivirii globale** este următoarea:

$$GF = \frac{\sum_i \max \frac{1}{abs(PositionInText_i - PositionInText_{i-1})}}{NumberOfWords}$$

Pentru cazul considerat, rezultatul aplicării formulei este:

[27] http://lucene.apache.org/

$$GF = \frac{1+\frac{1}{7}+1+1+1+1+\frac{1}{20}+0+0+\frac{1}{28}+\frac{1}{3}}{11} = \frac{5.56}{11} = 0.51$$

Pragul dintre perechile pentru care relaţia de inferenţă textuală este adevărată şi cele pentru care este falsă a fost stabilită utilizând perechile de antrenament de la competiţia RTE3. Valoarea sa a fost stabilită la 0.42. În cazul de faţă, deoarece 0.51 este mai mare decât 0.42, decidem că relaţia de inferenţă textuală este adevărată pentru această pereche.

3. Folosirea sistemului de inferenţe textuale în cadrul competiţiei pentru sisteme de tip ÎR

Scopul utilizării sistemului de inferenţe textuale ca modul în arhitectura generală a unui sistem de tip ÎR este acela de a îmbunătăţi clasificarea dintre răspunsurile posibile pentru întrebări de tip PERSOANĂ, LOCALITATE, DATĂ şi ORGANIZAŢIE.

Ideea este aceea de a selecta toate entităţile nominale din fragmentele de text extrase pentru o întrebare şi de a le înlocui cu variabilele din şabloanele asociate întrebării (Iftene et al., 2007a). În acest mod, vom obţine mai multe ipoteze pentru un singur text (reprezentat de fragmentul de text). Pentru fiecare ipoteză, calculăm scorul de potrivire global şi în final selectăm entitatea nominală pentru care obţinem cea mai mare valoare. În continuare, comparăm cea mai mare valoare din fiecare fragment de text şi în final selectăm cea mai mare valoare globală.

Pentru exemplul dat, scorul de potrivire globală pentru fragmentul de text indicat este 0.51. În acest caz, mai există alte două fragmente de text:

S1: "*Petru Popescu este un romancier, scenarist şi realizator de filme american de origine română. A emigrat în Statele Unite ale Americii în anii 1980, unde s-a impus drept romancier şi autor de scenarii ale unor filme de la Hollywood."*

S2: "*Americanul Ernest Hemingway (1899-1961), autor de povestiri, nuvelist şi romancier, şi romancierul rus Yuri Olesha (1899-1976) s-au născut la aceeaşi dată.*"

Pentru primul fragment de text, S1, avem un singur răspuns posibil, care este *Petru Popescu.* Ipoteza va fi: *Petru Popescu, faimos romancier, nuvelist, realizator de povestiri American, a trăit între anii 1899 şi 1961.* Deoarece în ipoteză avem numerele 1899 şi 1961 care nu apar în fragmentul de text S1, vom utiliza regula referitoare la entităţi nominale şi astfel vom obţine scorul de potrivire global 0.

Al doilea fragment de text conţine două entităţi nominale de tip PERSOANĂ: *Ernest Hemingway* şi *Yuri Olesha*. Urmând paşii din (Iftene et al., 2007a) obţinem două ipoteze:

H2_1: *Ernest Hemingway, faimos romancier, nuvelist, realizator de povestiri American, a trăit între anii 1899 şi 1961.*

H2_2: *YuriOlesha, faimos romancier, nuvelist, realizator de povestiri American, a trăit între anii 1899 şi 1961.*

Scorurile de potrivire globală pentru perechea (*H2_1, S2*) este 0.47, iar pentru perechea (*H2_2, S2*) este 0.44. Ambele sunt peste 0.42, dar din acest fragment de text vom selecta cea mai mare valoare, care este obținută pentru *Ernest Hemingway*.

În final, răspunsurile posibile sunt *Ernest Hemingway* cu scorurile 0.51 şi 0.47, *Yuri Olesha* cu scorul de potrivire global 0.42 şi *Petru Popescu* cu scorul 0. Aceste este clasamentul final obținut de sistem.

Pentru tipurile specificate, construim şabloane specifice, în funcție de tipul aşteptat al răspunsului:

Tabela 3: Transformarea întrebării în şablon

LOCALITATE	Unde s-a născut?	S-a născut **în** LOCALITATE.
DATĂ	Când a fost republicată ediția reorganizată a poemului?	Ediția reorganizată a poemului a fost publicată **pe** DATĂ.
ORGANIZAȚIE	Ce companie de software cu sediul central în San Jose a fost fondată în 1982?	ORGANIZAȚIE, **o** companie de software cu sediul central în San Jose, a fost fondată în 1982.

4. Rezultate

Sistemul de inferențe textuale englezesc are un nivel de acuratețe de 69.13% pe cele 800 de perechi text-ipoteză care reprezintă datele de test din cadrul competiției RTE3[28] şi s-a clasat pe locul 3 în cadrul competiției de anul acesta. O primă evaluare pe limba română s-a făcut traducând cei 1600 de arbori obținuți cu Minipar din engleză în română şi folosind sistemul de inferențe textuale englezesc. În acest prim caz am obținut o precizie de aproximativ 67 % datorită diferențelor existente între resursele englezeşti şi cele româneşti. A doua evaluare am realizat-o folosind sistemul prezentat în această lucrare, iar de această dată am tradus în română perechile text-ipoteză de test din competiția RTE3. Rezultatele în acest caz nu au depăşit 57 %. Prin urmare se observă cum cea mai importantă problemă cu care suntem confruntați în construirea sistemului de inferențe textuale pentru limba română este reprezentată de lipsa de resurse, acesta fiind şi motivul principal pentru diferența dintre rezultatele celor două evaluări.

Prin adăugarea modulului de inferențe textuale la sistemul de tip întrebare răspuns, pentru fragmentele de text ce nu reprezintă texte coerente, sistemul de inferențe textuale este inutil; însă, pentru fragmente de text complexe, care exprimă aceeaşi idee, dar cu actori şi contexte diferite, diferența pentru alegerea răspunsului corect este obținută clar, cu un grad mai mare de certitudine, utilizând modulul de inferențe textuale. În prezent, utilizăm sistemul de inferențe textuale pentru a clasifica mai bine răspunsurile posibile pentru întrebările de tip PERSOANĂ şi LOCALITATE. În cazul acestora, rezultatele demonstrează o creştere a acurateții de până la 5 %.

Pentru viitor, dorim să continuăm dezvoltarea sistemului pentru a putea fi capabil de a procesa întrebări cu răspunsuri de tip DATĂ şi ORGANIZAȚIE. De asemenea, vom utiliza un modul de traducere român-englez, pentru a putea utiliza resursele consistente ce există pentru limba engleză.

[28] http://www.pascal-network.org/Challenges/RTE3/Datasets/

Mulţumiri. Autorii mulţumesc membrilor grupului de lingvistică computaţională din Iaşi pentru ajutorul şi sprijinul acordat la diferite stagii ale dezvoltării sistemului.
Lucrul din cadrul acestui proiect este parţial finanţat de Siemens VDO Iaşi şi de proiectul CEEX Rotel numărul 29.

Referinţe bibliografice

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. (2006) The Second PASCAL Recognising Textual Entailment Challenge. *In Proc. of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*. Venice. Italy.

Dagan, I., Glickman, O. and Magninim, B. (2006). The PASCAL Recognising Textual Entailment Challenge. *In Quiñonero-Candela et al., editors, MLCW 2005*, LNAI Volume 3944, pages 177-190. Springer-Verlag.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Mass.

Kouylekov, M. and Magnini, B. (2005) Recognizing Textual Entailment with Tree Edit Distance Algorithms. *In Proceedings of the First Challenge Workshop Recognising Textual Entailment*, Pages 17-20, 25–28 April, 2005, Southampton, U.K.

Hamza, O., Tablan, V., Maynard, D., Ursu, C., Cunningham, H. and Wilks, Y. (2002). Name entity recognition in Romanian. *Technical report, Department of Computer Science, University of Sheffield*. Forthcoming.

Iftene, A., Balahur-Dobrescu, A. (2007). Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment. *In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Pp.125-130. 28-29 June, Prague, Czech Republic.

Iftene, A., Pistol, I., Forăscu, C., Trandabăţ, D., Balahur-Dobrescu, A., Cotelea, D., Drăghici, I. (2007a). Construirea unui sistem de tip întrebare-răspuns pentru limba română. *The third Workshop on Romanian Linguistic Resources and Tools for Romanian Language Processing*. 14-15 December. Iaşi, România.

Iftene, A., Trandabăţ, D. and Pistol, I. (2007b). Grammar-based Automatic Extraction of Definitions and Applications for Romanian. *In Proceedings of RANLP workshop "Natural Language Processing and Knowledge Representation for eLearning environments"*. September 26, Borovets, Bulgaria.

Lin, D. (1998). Dependency-based Evaluation of MINIPAR. *In Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May.

Lin, D. and Pantel, P. (2001). DIRT - Discovery of Inference Rules from Text. *In Proceedings of ACM Conference on Knowledge Discovery and Data Mining* (KDD-01). pp. 323-328. San Francisco, CA.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H. and Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. *In Recent Advances in Natural Language Processing 2001 Conference*. Pages 257–274, Tzigov Chark, Bulgaria.

Shinyama, Y., Sekine, S., Sudo, K. and Grishman, R. (2002). Automatic Paraphrase Acquisition from News Articles. *Proceedings of Human Language Technology Conference*, San Diego, USA.

Tufiş, D., Barbu, E., Barbu Mititelu, V., Ion, R., Bozianu, L. (2004) The Romanian Wordnet. *Romanian Journal of Information Science and Technology*, Volume 7, Numbers 1-2, pp. 107-124.

Tufiş, D. (1999). Blurring the distinction between machine readable dictionaries and lexical databases, *Research Report*, RACAI-RR56.

Tufiş, D., Rotariu, G., Barbu, A.M. (1999). TEI-Encoding of a Core Explanatory Dictionary of Romanian, *In Papers in Computational Lexicography*, 219-228, Kiefer, F., Pajzs J. (Eds.), Hungarian Academy of Sciences.

UN SISTEM DE EXTRAGERE A COLOCAȚIILOR

AMALIA TODIRAŞCU[1], DAN STEFĂNESCU[2], CHRISTOPHER GLEDHILL[1]

[1]*LILPA, Université Marc Bloch Strasbourg;*

[2]*Institutul de Cercetări pentru Inteligență Artificială, Academia Româna, Bucureşti, România*

todiras@umb.u-strasbg.fr, danstef@racai.ro, gledhill@umb.u-strasbg.fr

Rezumat

Articolul prezintă un proiect de cercetare al cărui obiectiv este de a dezvolta un sistem de extragere semi-automată a colocațiilor, parametrabil pentru mai multe limbi: franceză, română, germană. Vom prezenta proprietățile morfologice şi sintactice contextuale ale unei clase specifice de colocații, construcțiile de tip verb-substantiv (Gledhill, 2007). Aceste proprietăți sunt folosite în cadrul metodei de extragere a colocațiilor care aplică mai întîi metode statistice, iar apoi o etapă de filtrare lingvistică. În acest articol, prezentăm datele extrase pentru limba română.

1. Context

Articolul de față prezintă o parte a cercetărilor realizate în cadrul unui proiect (finanțat parțial de Agenția Universitară pentru francofonie), al cărui obiectiv este de a dezvolta un sistem de extragere semi-automată a colocațiilor. Sistemul va fi folosit pentru a crea un dicționar multilingv francez-român-german.

Colocațiile sunt expresii idiomatice sau combinații libere, care au un sens şi o serie de proprietăți morpho-sintactice proprii, diferite de cele ale elementelor componente. Colocațiile pun probleme deosebite unor sisteme de prelucrare automată a limbajului natural, precum şi persoanelor care traduc sau învață o limbă străină. Astfel, este importantă folosirea colocației într-un context adecvat, expresiile din această categorie jucînd un rol pragmatic bine definit şi avînd un sens de sine stătător. De asemenea se impune utilizarea corectă a unor proprietăți morfologice şi sintactice ale acestora (număr, gen, caz etc.) sau a combinațiilor lexicale corecte (astfel putem spune *a ține o conferință* dar nu **a face o conferință*).

Din acest motiv, în ultimii ani, colocațiile au fost studiate din perspectiva extragerii automate pe baza corpusurilor sau a realizării unor dicționare electronice. Deşi există unele dicționare de colocații monolingve, ele se limitează fie la regruparea expresiilor idiomatice şi explicarea sensului (CoBuild), fie la o caracterizare a proprietăților morfologice, sintactice (BLF, Selva et al., 2003) sau semantice (Dico, Polguère şi Mel'čuk, 2006). De asemenea, există proiecte care vizează realizarea de dicționare de colocații bilingve, dar care studiază o categorie precisă de colocații (francez-german) (Blumenthal, 2007). Unele din aceste dicționare sunt incomplete sau propun doar unele aspecte legate de proprietățile colocațiilor. Pentru limba română, deşi există studii ale colocațiilor şi inițiativa de a crea un dicționar (Căpățînă et al., 2005), totuşi, la ora

actuală nu există resurse electronice complete din această categorie. De aceea, am ales să dezvoltăm un sistem hibrid de extragere semiautomată a colocațiilor.

Pentru constituirea unor dicționare de colocații, multe proiecte de cercetare s-au orientat spre dezvoltarea unor sisteme de extragere a colocațiilor bazate pe exploatarea corpusurilor. Unele dintre acestea aplică o interpretare statistică a colocațiilor considerîndu-le fie simple co-ocurențe (Sinclair, 1996), fie considerînd că există o relație sintactică care se stabileşte între bază şi colocativ, şi deci aplică tehnici de extragere bazate pe folosirea unui analizor sintactic (Seretan et al., 2004). Pe de o parte, metodele statistice au dezavantajul de a produce multe rezultate eronate, metodele lingvistice necesită cunoştințe lingvistice complexe. Nici una din aceste metode nu este suficient de exactă, din această cauză, multe metode de extragere combină metode statistice cu aplicarea unor filtre morfo-sintactice. Astfel, dacă sistemele de extragere a unor termeni dintr-un domeniu tehnic aplică mai întîi o metodă de extragere care foloseşte cunoştințe lingvistice (Daille, 1996), alte metode aplică mai întîi o metodă de extragere statistică, urmată de o etapă de filtrare (Evert, 2005) (Heid, 1998). Metoda noastră de extragere este de asemenea o metodă hibridă, care aplică mai întâi o metodă de extragere statistică, urmată de o selecție bazată pe criterii lingvistice.

Pentru proiectul nostru, am considerat colocațiile ca fiind elemente lexicalizate (Hausmann, 2004), caracterizte prin informațiile contextuale, morfologice şi sintactice (contextul fiind atît elementele colocației cît şi constituienții sintactici cu care se combină colocația) (Ritz şi Heid, 2006). Am studiat comportamentul sintactic al colocațiilor de tip verb-substantiv (VS) în mai multe limbi: franceză, germană, română, dintr-o perspectivă a gramaticii sistemice funcționale (Halliday, 1985). Studiul a fost realizat pe un corpus paralel (AcquisCommunautaire, ACC (Steinberger et al., 2006)), disponibil în cele 3 limbi. Am identificat mai multe clase de construcții VS interesante (predicate complexe şi combinații predicat+complement (A.Todirascu, C.Gledhill, 2007)), caracterizate de o serie de proprietăți morfologice şi sintactice stabile. Pe baza acestor elemente, am creat un sistem de extragere a colocațiilor care aplică mai întîi un modul statistic (Ştefănescu et al., 2006), care calculează perechile VS cele mai frecvente, iar apoi aplică o etapă de filtrare. Articolul prezintă o primă evaluare a rezultatelor sistemului, privind numai extragerea colocațiilor pentru limba română. Pentru aceasta, vom prezenta mai întîi metodologia adoptată în cadrul proiectului, apoi vom argumenta cadrul adoptat pentru analiza lingvistică, iar apoi vom prezenta sistemul de extragere, precum şi primele date extrase automat şi analizate manual din corpusurilor româneşti.

2. Metodologia adoptată

În cadrul proiectului, am adoptat o metodologie bazată pe folosirea corpusurilor multilingve paralele şi a corpusurilor monolingve pentru extragerea unei clase specifice de colocații. Astfel, clasa de colocații studiată este formată dintr-un verb şi un substantiv, care joacă rol de complement (se exclud situațiile în care substantivul joacă rolul de subiect). Metoda de extracție aplicată nu necesită resurse lingvistice complexe. Pentru constituirea acestor resurse, avem nevoie doar de un corpus adnotat cu categoria

lexicală (partea de vorbire) şi eventual o serie de proprietăţi morfologice. Pentru extragerea colocaţiilor am procedat după cum urmează:

1) am aplicat o metodă de extragere statistică a perechilor verb-substantiv, pentru cele trei limbi studiate (română, franceză, germană), folosind un corpus paralel multilingv şi am comparat listele perechilor extrase, identificînd cazurile cînd nu există expresii echivalente în celelalte limbi.

2) am analizat contextele perechilor verb-substantiv pentru cele trei limbi, insistînd asupra unor proprietăţi morfologice şi sintactice specifice. Astfel, substantivul poate fi folosit numai la singular (*a face <u>obiectul</u>*) sau numai la plural, articolul poate lipsi (*pune în <u>aplicare</u>*) sau este folosit numai cu articol definit. Verbul poate fi folosit numai la un anumit timp sau diateză (această observaţie nu se aplică însă pentru limba română). De asemenea, am analizat contextele candidaţilor, remarcînd o serie de proprietăţi morfologice specifice: complementul indirect este folosit numai în cazul dativ, preferinţa pentru anumite prepoziţii care urmează după verb etc. Un studiu similar este realizat folosind de data aceasta corpusuri monolingve, pentru detectarea unor eventuale probleme legate de stilul folosit în corpusul paralel. Anumite proprietăţi pot fi prezente doar în corpusul ACC, influenţînd rezultatele.

3) pe baza rezultatelor obţinute la punctele 1) şi 2), am identificat mai multe clase de construcţii Verb-Substantiv, care sunt interesante pentru aplicaţia noastră:

4) Predicate complexe, în care perechea Verb-Substantiv formează un bloc avînd rolul de predicat. În această categorie intră locuţiunile verbale dar şi combinaţii libere, care au o preferinţă marcată pentru anumite proprietăţi morfologice şi sintactice;

5) Construcţii de tip predicat+complement. Această clasă de construcţii acceptă variaţii în formele substantivului sau al verbului (diateza activă sau pasivă).

6) definirea filtrelor morfosintactice care permit selecţionarea unor perechi verb-substantiv care sunt colocaţii/combinaţii libere sau expresii idiomatice (locuţiuni verbale), definite pentru toate limbile studiate.

7) validarea manuală a candidaţilor, pe baza analizei lingvistice propuse în secţiunea 3.

8) alimentarea dicţionarului. Candidaţii pentru care există echivalenţi între cele trei limbi sunt propuşi pentru a face din dicţionar, dar şi cei pentru care nu există o colocaţie echivalentă.

Pentru o analiză lingvistică aprofundată, am ales corpusul AcquisCommunautaire (Steinberger et al., 2006) disponibil în 21 de limbi diferite. Astfel, am selecţionat aceleaşi documente, disponibile în cele trei limbi pentru a avea conţinut similar. Corpusul are aproximativ 15 milioane de cuvinte pentru fiecare limbă şi a fost folosit pentru a extrage o listă de candidaţi pentru fiecare din cele trei limbi studiate. Cum acest corpus conţine normele europene publicate din 1950 şi pînă astăzi, stilul folosit este unul specific juridic administrativ, impersonal, conţinînd multe formule specifice textelor juridice, expresii predefinite. Pentru a evita ca rezultatele studiului nostru să fie prea dependente de conţinutul corpusului studiat, am folosit de asemenea şi corpusuri monolingve, conţinînd mai ales ziare, texte literare, manuale tehnice. Pentru limba

română, avem la dispoziție un corpus constituit din ziare, un roman, Constituția României care însumează 7 milioane de cuvinte, validate parțial manual (pe care îl numim RoGen). Am folosit atît corpusul neetichetat, precum şi corpusul etichetat cu ajutorul TTL (Ion, 2006). Corpusurile pentru franceză şi germană au fost etichetate cu ajutorul TreeTagger (Schmid, 1994). Deoarece corpusul AcquisCommunautaire este un corpus specializat, a fost necesară corectarea manuală a lemelor sau a etichetelor.

Metodologia aleasă poate fi aplicată şi altor clase de colocații (Substantiv-Adjectiv, Substantiv-Substantiv etc.), dar în cadrul acestui proiect ne-am limitat la studiul unei clase restrînse de colocații (Verbe-Substantiv).

3. Colocații Verb-Substantiv

Definiția noțiunii de colocație pe care am adoptat-o pentru acest proiect este aceea propusă de (Haussman, 2004), care consideră colocațiile ca fiind constituite dintr-un element de bază şi un element asociat (colocativ), elemente care pot fi discontinue. Între aceste elemente se stabilesc relații sintactice de dependență (substantivul este modificat de un adjectiv, verbul se combină cu substantivul care joacă rol de complement, etc.). Pentru a realiza o analiză lingvistică a combinațiilor Verb Substantiv, am adoptat punctul de vedere al gramaticii sistemice funcționale (Halliday, 1985) care propune o analiză completă a combinațiilor Verb-substantiv, atît a locuțiunilor verbale, cît şi a combinațiilor libere. Din perspectiva acestei teorii, propunem o analiză care ține cont de trei aspecte (Gledhill, 2007): structura lexicală a predicatului (astfel, locuțiunile verbale formează un bloc unitar, care joacă rol de predicat, pe cînd celelalte construcții sunt de tip predicat+complementul, interschimbabile), de rolul funcțional jucat de fiecare element al predicatului (subiect, predicat sau obiect) precum şi de procesul exprimat de către predicat (astfel, complementul completează procesul exprimat de către verb).

Am studiat proprietățile specifice construcțiilor de tip Verb Substantiv, care sunt împrumutate atît de la verbe cît şi de la substantive (Gledhill 2007). Proprietățile specifice substantivului sunt prezența/absența determinantului, nominalizarea posibilă doar pentru combinații libere (a lua o decizie, şi luarea deciziei), modificarea substantivului cu ajutorul unei clauze relative este posibilă doar pentru combinațile libere (a lua decizia care e necesară dar nu *a făcut obiectul care era cerut). Dintre proprietățile specifice verbului, putem aminti că predicatele complexe pot fi uneori înlocuite printr-un singur verb (a se face noapte = a înnopta), pot avea argumente ca orice predicat verbal simplu (subiect, complement), iar diateza pasivă nu este întotdeauna acceptată (pentru expresii sintagmatice fixe, acest lucru nu este posibil).

Putem constata că preferința pentru anumite proprietăți morfologice şi sintactice indică mai degrabă o combinație restricționată sau o locuțiune. Dacă substantivul este folosit în mod sistematic fără articol (a ține seamă de), sau numai la numărul singular (a face obiectul), atunci aceste elemente permit identificarea perechii VS ca fiind un predicat complex (joacă în bloc rolul de predicat). De asemenea, imposibilitatea de a folosi expresia la diateza pasivă (a făcut obiectul unui contract… dar nu obiectul a fost făcut….) indică de asemenea un grad înalt de rigiditate a expresiei. Nici una din proprietățile amintite nu permite identificarea sistematică a tuturor locuțiunilor şi a

construcţiilor VS, dar putem constata că anumite proprietăţi sunt folosite în mod sistematic şi pot reprezenta un indiciu interesant asupra expresiei. În tabelul I prezentăm o serie de proprietăţi identificate în contextele perechilor candidat, în corpsul ACC, dar şi în corpusul monolingv RoGen.

De asemenea, în cazul expresiilor fixe, chiar şi contextele arată o serie de proprietăţi identice: complementul direct sau indirect este folosit mereu în cazul dativ sau în mod sistematic cu o prepoziţie predefinită (la, din).

Tabel 1: Pentru verbul '*a face*', elementele colocative cele mai frecvente în corpusul ACC şi RoGen, precum şi proprietăţile morfologice identificate în context: fără articol sau numai cu articol definit, numărul singular, caz dativ pentru complementul indirect.

Colocativ	**Frecv ACC**	**Art**	Nr	**Caz**	**Pred. comp.**	**Colocativ**	**Frecv RoGen**	**Art**	**Nr**	**Caz**	**Pred. comp.**
Obiectul	3092	def	Sg	Dativ	+	Parte	1571	-	Sg	Acc (Din)	+
Referire	1416	-	sg, pl	A (la)	+	înscrieri	422	-, def	Pl	A(La)	-
Parte	1268	-	Sg	A (din)	+	Baza	362	-, def	Sg	Dativ	-
Trimitere	691	-	Sg,pl	A(la)	+	Loc	160	-, def	Sg, pl	Dativ	-
Dovada	178	def	Sg	Dativ	-	Cursuri	142	-, def	pl	-	-
Posibilă	170	-	Sg	A	+	Faţă	137	-	sg	Dativ	+
Necesară	155	-	Sg, pl	A/nom	+	Obiectul	127	-, déf, indéf	Sg, pl	Dativ	+
Faţă	150	-	Sg	Dativ, A(la)	+	Precizări	124	-, déf, indéf	Sg, pl	Dativ	-

Nu numai proprietăţile morfologice sunt importante în analiza datelor lingvistice. Astfel, rolul pe care îl joacă substantivul (subiect sau complement) permite identificarea unor clase de combinaţii interesante pe care le vom identifica automat:

predicate complexe (dacă verbul şi substantivul formează predicatul împreună);

construcţii predicat+complement (în care substantivul joacă rol de complement al verbului).

Deşi corpusurile pe care le avem la dispoziţie nu conţin adnotări sintactice pentru a putea folosi aceste informaţii pentru identificarea cazurilor predicat+complement, putem totuşi defini o serie de filtre, bazate pe identificarea proprietăţilor morfologice de mai sus. Analiza manuală, bazată pe identificarea procesului exprimat de verb şi de complemente, permite selectarea candidaţilor din cele două clase.

4. Extragerea colocațiilor

Pentru acest proiect, am adoptat o metodă hibridă care constă în extragerea statistică a perechilor verb-substantiv, iar apoi aplicăm o metodă de filtrare a candidaților, folosind observațiile realizate pe datele extrase din corpusurile disponibile.

4.1 Modulul de extragere statistică

În modelarea noastră (Stefanescu et al., 2006), considerăm colocațiile ca fiind succesiuni de cuvinte (nu neapărat adiacente) care respectă două criterii statistice:

distanța dintre cuvinte este relativ constantă;

apar în aceleaşi contexte de un număr de ori semnificativ din punct de vedere statistic.

Primul criteriu este evaluat folosind abordarea lui Smadja (1990) iar cel de-al doilea se bazează pe calculul raportului Log-Likelihood (LL). Rezultatele obținute folosind o combinație a celor două metode indică un lucru interesant: utilizarea scorului LL calculat pentru perechi de cuvinte care îndeplinesc anumite criterii ce țin de partea de vorbire, cât şi de media distanței dintre cuvinte, constituie o abordare eficientă. Inițial, textul este lematizat şi adnotat la părți de vorbire. Apoi, o fereastră de 11 cuvinte (acesta este contextul în care se consideră co-ocurențele) parcurge fiecare propoziție din text în aşa fel încât fiecare cuvânt devine la un moment dat centrul ferestrei[29]. Cuvintele ce se introduc în fereastră sunt substantive sau verbe; celelalte părți de vorbire sunt ignorate. Lungimea a fost aleasă astfel încât fereastra să poată cuprinde orice pereche de cuvinte interesantă care ar exista. Am considerat că o distanță de 5 (stânga/dreapta) pentru o astfel de fereastră, în care se găsesc doar cuvinte ce au ataşate doar anumite etichete morfo-sintactice este suficientă pentru a găsi perechile interesante. Deşi ar putea exista exemple în care distanța dintre cuvinte este mai mare de 5 (numărând doar cuvintele din categoriile gramaticale care ne interesează pe noi), aceste cazuri sunt rare şi se datorează probabil intercalării unor expresii lungi între cuvintele ce formează perechea interesantă. Considerând apoi toate perechile de cuvinte de tip SV formate de cuvântul din centrul ferestrei cu celelalte cuvinte de interes din fereastră, parcurgem tot corpusul numărând aceste perechi la diferitele distanțe la care cuvintele ce formează perechile apar în text. Putem calcula apoi pentru fiecare pereche de cuvinte media şi deviația standard a distanțelor dintre ele. O deviație standard mică indică faptul că cele două cuvinte (din pereche) se găsesc într-o poziție aproximativ fixă în text, la o distanță indicată de medie. Perechile ale căror deviație standard este mai mică de un anumit prag (Stefănescu et al., 2006) sunt păstrate într-o listă de perechi candidat pentru care apoi calculăm scorul LL. Dorim să vedem care dintre cuvinte apar împreună în corpus mai des decât ne-am aştepta să apară întâmplător. Considerând un prag minim (Ştefănescu et al., 2006) obținem o listă finală de perechi candidate pe care o ordonăm în funcție de scorul LL. Această listă va fi analizată în secțiunea următoare. Trebuie să amintim că alegerea celor două praguri influențează precizia şi completitudinea sistemului. Pentru

[29] Folosim metoda lui Smadja. Aceasta ne permite să identificăm perechi interesante de cuvinte ce nu sunt neapărat adiacente

problema de faţă ele au fost setate la valorile de 1,5 pentru deviaţia standard şi de 9.0 pentru scorul LL (Ştefănescu et al., 2006).

Tabel 2: Candidaţii cei mai frecvenţi şi proprietăţile asociate: articol, număr, cazul complementului şi statutul (predicat complex, predicat+complement)

Baza	**Colocativ**	**LL**	**Art**	**Nr**	**Comp.**	**Categorie**
Aduce	Atingere	51567.34864	-	Sg	Dativ	Predicat complex
Înlocui	Text	43992.3067	Def	Sg, pl	Acuzativ	predicat+complement
Intra	Vigoare	42527.03736	-	Sg	Acuzativ (în)	predicat complex
Avea	Tratat	32050.11219	Def	Sg, pl	Acuzativ	structură invalidă (Predicat+Adjunct)
Face	Obiectul	30729.47663	Def	Sg	Dativ	predicat complex
Modifica	Regulamentul	29141.39454	def, -	Sg, pl	Acuzativ (la, din)	structură invalidă (Predicat+Comp. Indirect)
Lua	Considerare	27062.0349	-	Sg	Nom	Predicat complex
Ţine	Cont	26635.12649	-	Sg	Acuzativ (de)	Predicat complex

O parte din acestea, cum ar fi *a intra în vigoare*, *a lua în considerare, a fi adoptat la bruxelles* pot fi consideraţi termeni specifici limbajului juridic reflectat în corpusul ACC. Cele mai multe din perechile prezentate în tabelul 2 sunt predicate complexe sau construcţii VS valide.

4.2 Filtrarea candidaţilor

Pentru a putea selecta o mulţime de perechi VS interesante, după ordonarea candidaţilor în funcţie de scorul LL, este necesar să definim o serie de filtre morfo-sintactice. Astfel, am analizat primii 1000 de candidaţi, precum şi contextele acestora. Am identificat apoi manual construcţiile valide (predicate complexe, structuri de tipul predicat+complement). Din analiza perechilor VS, am constatat că mai multe tipuri de construcţii invalide care pot fi eliminate prin definirea unor filtre simple. Astfel, construcţiile invalide pot fi identificate prin următoarele clase:

1) Predicat+subiect: substantivul joacă rolul de subiect, iar în acest caz, putem elimina o parte din aceşti candidaţi numai pe baza corpusului etichetat şi pe baza aplicării unor reguli euristice: subiectul fiind în general înaintea predicatului (*Comisia decide suspendarea articolului 4, deoarece…*). Nu vom putea însă elimina cazurile în care subiectul este precedat de către verb, decît cu ajutorul unui corpus adnotat cu relaţii de tip dependenţă.
2) Predicat+adjunct: în acest caz, este posibil să eliminăm candidaţii, dacă distanţa dintre verb şi substantiv este prea mare (ar trebui să limităm căutările numai la perechi care se află la distanţa de 1 sau 2 cuvinte) *(…modificat ultima dată de regulamentul…*).
3) Predicat+complement indirect: deşi este posibil uneori să identificăm cazul dativ sau o prepoziţie care marchează complementul indirect, această informaţie poate

elimina şi candidaţi care sunt valabili *(…modificat ultima dată de regulamentul…)*.

4) Reziduuri: în acest caz, putem clasa mai multe situaţii: verbul şi substantivul sunt separate de un separator de frază (*Articolul 4 Comisia întrunită a hotărît….*) sau de o conjuncție (*Articolul modifică şi textele…*), sau de mai multe prepoziții (*Comisia a răspuns la cererea tribunalului, cu argumente în favoarea deciziei…*). În aceste cazuri, candidaţii pot fi eliminaţi din lista finală prin identificarea separatorilor sau a criteriilor de distanță.

5) Grup nominal, compus din substantiv şi verb la participiu (*Un raport realizat de către Comisie …)*. Pentru limba română, putem aplica un filtru care elimină această categorie de construcții.

Pentru identificarea filtrelor, ne bazăm pe studiul contextelor extrase pentru fiecare din perechile candidat:

Exemplu.

A) Pentru perechea ***intra-vigoare*** (este predicat complex), observăm că substantivul este mereu în aceeaşi formă – singular, nearticulat (etichetat *nsrn* de către TTL), iar prepoziția 'în' este folosită în mod sistematic:

intră/v3 în/s vigoare/nsrn

intrat/vp în/s vigoare/nsrn

intre/v3 în/s vigoare/nsrn

intra/vn în/s vigoare/nsrn

intrând/vg în/s vigoare/nsrn

B) pentru perechea ***adopta-regulament*** (este de data aceasta o construcție predicat+complement), substantivul poate fi folosit în orice forma – singular, plural, articulat sau modificat de un adjectiv:

adoptă/v3 prezentul/asry regulament/nsn

adoptă/v3 propriul/asry regulament/nsn

adoptă/v3 următorul/asry regulament/nsn

adoptă/v3 un/tsr regulament/nsn

adoptat/vp aceste/dmsr regulamente/npn

Cu cît avem mai puține contexte posibile, putem constata că proprietățile morfologice ale substantivului, ale complementelor se repetă în marea majoritate a contextelor asociată fiecărei perechi. Pe baza acestor contexte, putem identifica filtrele care permit selectarea claselor de colocații care ne interesează (cu ajutorul metodelor statistice de învațare automată (Claveau et Sébillot, 2004)).

Cîteva exemple de filtre aplicate pentru identificarea predicatelor complexe:

1) `face *{0,1} NSRY *{1,5} NxOY`

NSRY - substantiv, articol definit, singular, acuzativ;

NxOY – substantiv, articol definit, d; *{x,y} – pot apare minimum x şi maximum y cuvinte între elementele filtrului.

Cîteva exemple de candidaţi selecţionaţi de acest filtru: *face obiectul unui contract, face dovada unui curaj*

2) `V în NSRN *{1,3} NxRY`

NSRN - substantiv, nearticulat, singular, acuzativ;

NxRY – substantiv, articol definit, acuzativ; *{x,y} – pot apare minimum x şi maximum y cuvinte între elementele filtrului (prepoziţii, adverbe).

Printre candidaţii selecţionaţi de acest filtru: *intra în vigoare la data, ia în considerare o situaţie*..

3) `not (NxRY AxRN{0,1} VP)`

NxRY – substantiv, articulat, acuzativ

ASRN – adjectiv, nearticulat

{0,1} – poate apare cel mult un singur adjectiv între substantiv şi verb

Printre candidaţii selecţionaţi de acest filtru (aplicat numai pentru candidaţii care nu acceptă şi contexte cu verbul la o formă finită): *regulamentul adoptat, textul modificat*

O evaluare manuală a sistemului a fost realizată, pentru 1000 de candidaţi examinaţi. Deşi proprietăţile morfologice şi sinactice au fost folosite pentru a identifica perechile interesante, aceste proprietăţi nu sunt suficiente pentru a putea decide clasa din care face parte construcţia Verb-Substantiv. Astfel, în cazul unui predicat complex, complementul este integrat (nu putem aplica pasivul) în cadrul grupului verbal şi joacă un rol important în precizarea procesului exprimat de predicat (proces de tip relaţie care se stabileşte între un agent şi beneficiar: *a face faţă, a face legătura* etc.). În cazul construcţiilor predicat+complement, complementul este independent (forma pasivă este acceptată), iar rolul complementului este acela de beneficiar sau de obiect. Deocamdată, în lipsa resurselor semantice complete, nu putem identifica clasele în mod automat. Am obţinut o precizie de 36,7%, care reprezintă proporţia de perechi Verb-Substantiv interesante (predicate complexe + construcţii predicat+complement). Acest scor poate fi îmbunătăţit dacă un corpus adnotat sintactic ar fi disponibil pentru limba română, deoarece nu am eliminat decît parţial cazurile cu probleme: cazurile în care avem predicat+complement indirect sau predicat+adjunct nu sunt deocamdată filtrate. Acest lucru este studiat în următoarea etapă a proiectului. Pentru celelalte limbi dispunem de corpusuri adnotate la nivel sintactic, şi putem folosi corpusurile aliniate la nivel de cuvînt pentru a recupera transfera informaţiile dintr-o limbă în cealaltă.

5. Concluzii şi perspective

În acest articol, am prezentat o metodă de extragere a colocaţiilor care se aplică pentru franceză, germană şi română insistînd asupra rezultatelor obţinute pentru limba română. Metoda adoptată combină metode statistice şi o etapă de filtrare bazată pe identificarea

automată a unor proprietăți morfologice simple, în contextele acestora. În viitor, metodologia va fi aplicată şi altor clase de colocații.

Mulțumiri. Autorii sunt recunoscători organizației AUF (Agence Universitaire pour la Francophonie), care finanțează acest proiect în cadrul rețelei «Lexicologie, Terminologie, Traduction» pe durata iunie 2007-martie 2008. Autorii mulțumesc doamnei Rada Mihalcea pentru corpusul românesc pus la dispoziție de către aceasta, d-lor Dan Tufiş şi colegilor de la Institutul de Cercetări în Inteligență Artificială (Academia Română, Bucureşti) pentru corpusurile AcquisCommunautaire şi corpusul general etichetate şi lematizate pentru română. Corpusul general românesc a fost completat cu o parte din corpusul creat în cadrul proiectului L2TE, pentru care îi mulțumim domnului Dan Cristea.

Referințe bibliografice

Blumenthal, P., (2007). A Usage-based French Dictionary of Collocations, in: Y. Kawaguchi/T. Takagaki/N. Tomimori/Y. Tsuruga (éds.): Corpus-Based Perspectives in Linguistics, Amsterdam u.a.: Benjamins (Usage-Based Linguistic Informatics 6), 67-83.

Căpățînă, C., (2005). Despre colocații, în Analele Ştiințifice ale Universității Al.I.Cuza din Iaşi (serie nouă), Secțiunea a III-a Lingvistică „Studia linguistica et philologica in honorem Constantin Frâncu", tomul LI.

Claveau, V., Sébillot, P., (2004). Apprentissage semi-supervisé de patrons d'extraction de couples nom-verbe, in TAL (traitement automatique des langues), Hermès, Vol. 45, No. 1.

Daille, B., (1996). Study and Implementation of Combined Techniques for Automatic Extraction of Terminology, in Resnik, P. (ed.) The Balancing Act: Combining Symbolic and Statistical Approaches to Language, MIT Press, page 49—66.

Evert, S., (2005). The Statistics of Word Cooccurrences: Word Pairs and Collocations, Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Gledhill, C., (2007). La portée : seul dénominateur commun dans les constructions verbo-nominales, in Frath, P., Pauchard, J. & Gledhill, C. (éds) Actes du 1er colloque Res per nomen, Reims 24-36 mai 2007, Université de Reims, Champagne, 113-124.

Gledhill C., Heid U., Mihăilă C., Rousselot F., Ştefănescu D., Todiraşcu A., Tufiş D. & Weller M. 2007. Collocations en contexte: extraction et analyse contrastive, Project Report for the Agence Universitaire pour la Francophonie 'Réseau Lexicologie, Terminologie, Traduction', Paris :1-38.

Halliday, M.A.K., (1985). An Introduction to Functional Grammar, London, Arnold.

Hausmann, F.J., (2004). Was sind eigentlich Kollokationnen?, en K.Steyer (eds.) Wortverbindungen – mehr oder weniger fest, 309-334.

Heid, U., (1998). Towards a corpus-based dictionary of German noun-verb collocations, in: Proceedings of the Euralex International Congress 1998, (Liège), 1998, SS. 301 -- 312.

Ion, R., (2007). TTL: A portable framework for to-kenization, tagging and lemmatization of large corpora, Research Institute for Artificial Intelligence, Romanian Academy, Bucharest (in Romanian), 22p.

Manning, C. D., Schütze, H., (1999). Foundations of statistical natural language processing, MIT Press.

Mel'čuk, I., Polguère, A., (2006). Dérivations sémantiques et collocations dans le DiCo/LAF, Langue française, special issue on collocations « Collocations, corpus, .dictionnaires », edited by P. Blumenthal and F. J. Hausmann, 150, June 2006, 66-83

Ritz, J., Heid, U., (2006). Extraction tools for collocations and their morphosyntactic specificities, in: Proceedings of LREC'2006, Genova, Italia.

Schmid, D., (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees, in: Proceedings of International Conference on New Methods in Language Processing.

Seretan, V., Nerima, L., Wehrli, E., (2004). A tool for multi-word collocation extraction and visualization in multilingual corpora, in: Proceedings of EURALEX'2004, Lorient, France, Vol2, pp.755-766.

Sinclair, J., (1991). Corpus, Concordance, Collocation, Oxford, Oxford University Press.

Smadja, F. A., McKeown, K. R., (1990). Automatically extracting and representing collocations for language generation, in: Proceedings of the 28th annual meeting on Association for Computational Linguistics, 252-259, Pittsburgh, Pennsylvania.

Ştefănescu, D., Tufiş, D., Irimia, E., (2006). Extragerea colocaţiilor dintr-un text, în Resurse lingvistice si instrumente pentru prelucrarea limbii române, Universitatea Al.I.Cuza Iasi, 89-95.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga, D., (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages, in: Proceedings of LREC'2006, 2142-2147.

Todiraşcu, A., Gledhill, C., (2007). Extracting Collocations in Context: The case of Verb-Noun Constructions in English and Romanian, RANAM, submitted.

Tutin, A., (2004). Pour une modélisation dynamique des collocations dans les textes, Actes du congrès EURALEX'2004, Lorient, France, 2004, Vol. 1, 207-221.

Verlinde, S., Selva, T., Binon, J., (2003). Les collocations dans les dictionnaires d'apprentissage: repérage, présentation et accès, en Grosman F., Tutin, A. (eds.). Les collocations: analyse et traitement / - Amsterdam: De Werelt, 2003. - p. 105-115.

EXTRAGEREA AUTOMATĂ A DEFINIȚIILOR DIN TEXTE ÎN LIMBA ROMÂNĂ

ADRIAN IFTENE[1], DIANA TRANDABĂȚ[1,2], IONUȚ PISTOL[1]

[1] *Facultatea de Informatică, Universitatea "Al.I.Cuza", Iaşi*
[2] *Institutul de Informatică Teoretică, Academia Română*

{adiftene, dtrandabat, ipistol}@info.uaic.ro

Rezumat

Acest articol prezintă dezvoltarea unei gramatici pentru extragerea automată a definițiilor. Pentru a putea evalua regulile gramaticii noastre vom prezenta câteva rezultate calitative şi aplicații posibile. Printre aplicațiile unei astfel de gramatici se numără cele din cadrul sistemelor de tip întrebare-răspuns, pentru regăsirea răspunsurilor la întrebări de tip definiție, şi aplicațiile în care este necesară extragerea de cunoaştere suplimentară din resurse precum Wikipedia. Avantajele unei resurse ce oferă cunoaştere suplimentară sunt evidente în sistemele de inferențe textuale, unde resurse precum WordNet sau baze de date de acronime nu pot acoperi toate cerințele sistemului.

1. Introducere

În contextul proiectului european FP6 LT4eL[30] (Language Technology for e-Learning), a fost creat un mediu pentru colectarea şi exploatarea (semi)automată a resurselor lingvistice. Scopul principal al proiectului este de a oferi funcționalități bazate pe tehnologiile limbajului şi de a integra cunoaşterea semantică în sisteme de coordonare a învățării. Primul pas a constat în crearea, pentru cele 9 limbi implicate (bulgară, cehă, olandeză, engleză, germană, malteză, poloneză, portugheză şi română), a un corpus multilingv, parțial paralel, de aproape 5.5 milioane de cuvinte, adnotate şi încărcate pe portalul proiectului[31] (Monachesi et al., 2006).

Pentru a îmbunătăți gestionarea, distribuirea şi căutarea materialului pentru învățare şi pentru a permite adăugarea automată de meta-informații (precum cuvinte cheie şi definiții) oricărui text, a fost necesară o cunoaştere atentă a acestor meta-informații din materialele adnotate. Prin urmare, corpusul a fost adnotat manual la cuvinte cheie (cuvinte sau expresii pe care utilizatorul unui sistem de coordonare a învățării le foloseşte pentru a căuta documente ce fac referire la acea noțiune), definiții sau diverşi termeni sau concepte semantice. Folosind documentele adnotate manual, a fost creată o gramatică care identifică automat definițiile din text. Pe lângă folosirea în acest proiect, vom mai prezenta încă două aplicații ale acestei gramatici.

După o descriere sumară a tipurilor materialelor de învățare folosite în proiectul LT4eL şi descrierea adnotării definițiilor, secțiunea 3 va descrie gramatica românească. Secțiunea 4 prezintă câteva posibile aplicații ale gramaticii care au scopul de a îmbunătăți calitatea unor sisteme complexe precum cel de tip întrebare-răspuns şi cel de

[30] LT4eL: http://www.lt4el.eu/

[31] Consilr: http://consilr.info.uaic.ro/uploads_lt4el/

inferențe textuale, iar în final sunt prezentate câteva concluzii şi direcții de lucru viitoare.

2. Materialul de învățare

Resursele lingvistice (*learning objects* – LO – obiecte de învățare) au fost selectate în funcție de domeniu (în general folosirea calculatorului în educație), format sau drepturi de autor. După conversia automată într-un format XML comun tuturor limbilor implicate în proiect care păstra numai informațiile minimale de formatare formatul XML (Pistol et al., 2006), obiectele au fost adnotate lingvistic (parte de vorbire, lemă, grupuri sintactice). Corpusul colectat pentru limba română conține 56 de documente însumând aproximativ 700.000 cuvinte.

Pentru adnotarea manuală, s-a înțeles prin definiție o explicație concisă, o descriere a înțelesului sau tipului unui concept. O definiție are două părți: un *termen definit* şi un *context de definire*. Un exemplu de definiție extrasă din corpusul românesc este:

[{Cetățenia Uniunii Europene}$_{Def\_term}$*]*$_{DEF\_PART1}$*, prevăzută în tratatul de la Roma şi mai apoi în cel de la Maastricht [este caracterizată de drepturi, de obligații şi de implicarea în viața politică]* $_{DEF\_PART2}$.

unde termenul definit este *Cetățenia Uniunii Europene* iar definiția este marcată între paranteze pătrate [ ]. Se observă că nu toată fraza este considerată a face parte din definiție, clauza atributiva fiind lăsată la o parte. Pentru a marca acest aspect s-a folosit împărțirea definiției în părți marcate succesiv. Astfel, adnotarea definiției de mai sus în format XML (Tufis, 2004) este prezentată în figura 1:

```
<definingText comment="" id="def37" status="" continue="y" def="dt35"
part="1">
<markedTerm id="dt35" comment="" dt="y" kw="n" status="">
<tok rend=" /b, /p, p" base="cetățenie" ctag="Ncfsry" id="t960">
Cetățenia </tok>
<markedTerm id="k36" comment="" dt="n" kw="y" status="">
<tok rend="" base="Uniunii_Europene" ctag="Ed"
id="t961">Uniunii_Europene</tok>
</markedTerm>
</markedTerm>
</definingText>
<tok rend="" base="," ctag="COMMA" id="t962">, </tok>
<tok rend="" base="prevedea" ctag="Vmp--sf" id="t963">prevăzută</tok>
<tok rend="" base="în" ctag="Spsa" id="t964">în</tok>
<tok rend="" base="tratat" ctag="Ncmsry" id="t965">Tratatul</tok>
<tok rend="" base="de_la" ctag="Spca" id="t966">de_la</tok>
<tok rend="" base="Roma" ctag="Np" id="t967">Roma</tok>
<tok rend="" base="(0.67)ş" ctag="Vmis1s" id="t968">şi</tok>
<tok rend="" base="mai" ctag="Rp" id="t969">mai</tok>
<tok rend="" base="apoi" ctag="Rgp" id="t970">apoi</tok>
<tok rend="" base="în" ctag="Spsa" id="t971">în</tok>
<tok rend="" base="acela" ctag="Pd3msr" id="t972">cel</tok>
<tok rend="" base="de_la" ctag="Spca" id="t973">de_la</tok>
<tok rend="" base="Maastricht" ctag="Np" id="t974">Maastricht</tok>
<definingText comment="" id="def38" status="" continue="y" def="dt35"
part="2">
<tok rend="" base="fi" ctag="Vaip3s" id="t975">este </tok>
<tok rend="" base="caracteriza" ctag="Vmp--sf"
```

```
id="t976">caracterizată</tok>
<tok rend="" base="de" ctag="Spsa" id="t977">de</tok>
<tok rend="" base="drept" ctag="Ncfp-n" id="t978">drepturi</tok>
<tok rend="" base="," ctag="COMMA" id="t979">,</tok>
<tok rend="" base="de" ctag="Spsa" id="t980">de</tok>
<tok rend="" base="obligaţie" ctag="Ncfp-n" id="t981">obligaţii</tok>
<tok rend="" base="(0.62)ş" ctag="Ncmpry" id="t982">şi</tok>
<tok rend="" base="de" ctag="Spsa" id="t983">de</tok>
<tok rend="" base="implicare" ctag="Ncfsrn" id="t984">implicare</tok>
<tok rend="" base="în" ctag="Spsa" id="t985">în</tok>
<tok rend="" base="viaţă" ctag="Ncfsry" id="t986">viaţa</tok>
<tok rend="" base="politic" ctag="Afpfsrn" id="t987">politică</tok>
</definingText>
```

Figura 1. Exemplu de definiție adnotată manual

3. Gramatica Românească

Pentru adnotarea automată a definițiilor din obiectele de învățare, soluția abordată în cadrul proiectului LT4eL a fost dezvoltarea de gramatici locale pentru cele 9 limbi ale proiectului care să surprindă şabloane de definiții. Dificultățile majore au fost evidențierea diferitelor metode de a exprima definițiile, păstrând o lexicalizare minimă a cuvintelor care introduc definițiile (precum verbele „a fi", „a reprezenta" etc.). Alte probleme au fost definițiile întrerupte şi marcherul de terminare a unei definiții, în special în cazul în care acesta nu coincide cu semnele de punctuație.

Informația lingvistică din definițiile marcate automat este folosită ca punct de plecare în identificarea posibilelor şabloane. Cercetările anterioare în acest domeniu au arătat că folosirea gramaticilor locale bazate pe şabloane sintactice sunt foarte utile atunci când analiza semantică lipseşte (Mureşan şi Klavans, 2002), (Liu et al., 2003).

Crearea gramaticii pentru limba română a început cu descrierea unor reguli simple şi aplicarea acestora pentru definițiile extrase manual. Observând în mod repetat erorile s-a îmbunătățit gramatica pentru a trata toate cazurile. Dezavantajul acestei metode este că a devenit dependentă de corpus.

3.1 Clasificarea definițiilor

Definițiile au fost clasificate în şase categorii cu scopul de a reduce spațiul de căutare şi complexitatea regulilor. Tipurile de definiții identificate în textele româneşti au fost clasificate după cum urmează:

5. **"is_def"** – Definiții conținând verbul "a fi":

Exemplu: *"Prescurtare pentru Hyper Text Mark Up Language, HTML* ***este*** *tot un protocol folosit de World Wide Web."*

6. "**verb_def**" – Definiții introduse de verbe specifice, diferite de "a fi". Verbele considerate pentru română sunt "a indica", "a arăta", "a preciza", "a reprezenta", "a defini", "a specifica", "a consta", "a fixa", "a permite".

Exemplu: "*Poşta electronică* ***reprezintă*** *transmisia mesajelor prin intermediul unor rețele electronice*.".

7. "**punct_def**" – Definiții introduse de semne de punctuație precum cratima "-", paranteze rotunde "()", virgula "," etc.

Exemplu: "*Bit* **–** *prescurtarea pentru binary digit...*"

8. "**layout_def**" – Definiții care pot fi deduse din aranjarea în pagină: aici pot fi incluse tabelele în care termenul definit şi definiția sunt în celule separate sau termenul definit este cuvânt titlu şi definiția este pe alt rând.

Organizarea secvențială	*Cel mai simplu mod de organizare a datelor.*

9. "**pron_def**" – Definiții anaforice, când termenul definit este prezent într-o propoziție anterioară şi el este doar referit în definiție, de obicei prin pronume.

Exemplu: *"…definirii conceptului de baze de date.* ***Acesta*** *descrie metode de modelare ale problemelor reale în scopul definirii unor structuri care să elimine redundanțele în stocarea datelor.*"

10. "**other_def**" – Alte tipuri de definiții, care nu pot fi incluse în nici una din categoriile anterioare. În această categorie sunt construcții care nu folosesc verbe pentru introducerea termenului, ci construcții specifice precum "adică".

Exemplu: "*triunghi echilateral,* ***adică*** *cu toate laturile egale*".

Distribuția tipurilor de definiții în corpusul românesc este prezentată în tabela 1:

Tabel 1: Distribuția definițiilor pe categorii

Tip	**Manual**	**%**	**Automatic**	**%**
is_def	70	33.8	204	32.8
verb_def	116	56.0	272	43.8
punct_def	15	7.2	124	20.0
layout_def	2	1.0	21	3.4
pron_def	4	2.0	0	0.0
Total	**207**		**621**	

Tabelul de mai sus arată că 33% din numărul total de definiții sunt introduse de verbul „*a fi*". Definițiile introduse de altceva decât un verb sunt aproximativ 10% din definițiile manuale şi în jur de 23% din definițiile automate. Diferența mare sugerează faptul că multe definiții au fost scăpate de adnotatori.

3.2 Gramatica

Aplicația *lxtransduce* prezentată în (Tobin, 2005) este folosită pentru a identifica în fişiere XML definițiile descrise în gramatica românească. În gramatica pentru limba română am creat reguli pentru fiecare tip de definiție din cele prezentate anterior şi o regulă principală folosită pentru a apela regulile individuale. Toate aceste reguli au fost construite pe baza observațiilor făcute asupra definițiilor adnotate manual.

3. 3 Regulile gramaticii

Construirea gramaticii folosite pentru extragerea definițiilor româneşti a început cu construirea unor reguli simple care identifică părțile de vorbire. De exemplu, regula prezentată în Figura 2 identifică adverbele cerând ca atributul "ctag" să aibă prima literă "r" (eticheta morfologică pentru adverbe):

```
<rule name="Adv">
 <query match="tok[@ctag[starts-with(.,'r')]"/>
</rule>
```

Figura 2: Exemplu de regulă pentru identificarea adverbelor.

Aceste reguli pot fi combinate pentru a se obține reguli mai complexe. Figura 3 prezintă o regulă obținută din combinația unor reguli ce identifică entități simple:

```
<rule name="Nominal">
 <seq>
  <ref name="undef" mult="?" />
  <ref name="AdjP" mult="?" />
  <ref name="Noun" />
  <ref name="AdjP" mult="?" />
 </seq>
</rule>
```

Figura 3: Regulă compusă pentru identificarea unui grup nominal.

După crearea regulilor care identifică diverse structuri se apelează regulile care identifică definiții. În Figura 4 sunt prezentate regulile necesare identificării definițiilor de tip "is_def". Lema pentru verb trebuie să fie "fi" iar eticheta părții de vorbire (tag-ul *ctag*) trebuie să fie "vmip3" (verb la timpul indicativ prezent, persoana a treia). O altă condiție este aceea ca înainte de verb să existe o entitate de tip "DefNominal" sau o entitate de tip "UndefNominal" (grup nominal articulat definit sau nedefinit), entități identificate prin reguli complexe precum cea din Fig. 3.

```
<rule name="may_be_term">
 <seq>
   <query match="tok[@base='fi' and
        substring(@ctag,1,5)='vmip3']"/>
   <first>
    <ref name="UndefNominal" />
    <ref name="DefNominal" />
   </first>
</seq>
```

Figura 4: Regula pentru identificarea definițiilor de tip "is_def"

Un alt tip de regulă este cea care identifică sfârşitul definiției. Deocamdată am identificat sfârşitul definiției cu sfârşitul propoziției (Figura 5):

```
<rule name="main" wrap="definingText" attrs="def_type1='punct_def'">
 <seq>
  <ref name="NP" wrap="markedTerm" attrs="dt='y'"/>
  <ref name="may_be_term" />
  <repeat-until name="anything">
      <query match="tok[(@base='.' and @ctag='period') or (@base=';' and @ctag='scolon')]"
/>
  </repeat-until>
  <query match="tok[(@base='.' and @ctag='period') or (@base=';' and @ctag='scolon')]" />
 </seq>
</rule>
```

Figura 5: Identificarea limitelor propoziției

3.4 Evaluarea gramaticii

Folosind *lxtransduce* identificăm porțiunile din fişier care corespund unei reguli şi marcăm corespunzător acele zone ca fiind definiții. Aplicația a fost rulată pentru fiecare tip de definiție şi rezultatele sunt prezentate în tabelul 2:

Tabel 2: Evaluarea gramaticii româneşti (P = precizie, R= recall şi F2 = F-measure)

Tip de Definiție	Rezultat
is_def	Potrivire la nivel de propoziție: P: 0.5366, R: 1.0, F2: 0.7765 Potrivire la nivel de cuvânt: P: 0.0648, R: 0.3328, F2: 0.14
verb_def	Potrivire la nivel de propoziție: P: 0.7561, R: 1.0, F2: 0.9029 Potrivire la nivel de cuvânt: P: 0.0471, R: 0.1422, F2: 0.085
punct_def	Potrivire la nivel de propoziție: P: 0.1463, R: 1.0, F2: 0.3396 Potrivire la nivel de cuvânt: P: 0.0025, R: 0.1163, F2: 0.0072
layout_def	Potrivire la nivel de propoziție: P: 0.0488, R: 1.0, F2: 0.1333 Potrivire la nivel de cuvânt: P: 0.0007, R: 0.1020, F2: 0.0022

Pentru fiecare tip de definiție, precizia şi recall au fost calculate în două moduri: la nivel de cuvânt şi la nivel de propoziție (Carletta, 1996). La ***nivel de cuvânt,*** precizia este înțeleasă ca fiind numărul de cuvinte care se găsesc în acelaşi timp în definițiile adnotate manual şi în cele identificate automat, împărțit la numărul de cuvinte din definițiile identificate automat. Corespunzător acestei formule, recall este raportul dintre numărul de cuvinte găsite în cele două tipuri de definiții, şi numărul total de cuvinte din definițiile adnotate manual. La ***nivel de propoziție***, considerăm că o propoziție face parte dintr-o definiție manuală sau automată dacă şi numai dacă ea conține o parte dintr-o definiție manuală sau automată. În continuare precizia şi recall sunt calculate asemănător valorilor calculate la nivel de cuvânt.

Rezultatele cele mai bune sunt obținute pentru definițiile care sunt identificate folosind verbe (majoritatea cazurilor). Dintre acestea, definițiile introduse de verbul *„a fi”* sunt cel mai greu de identificat, deoarece acest verb apare foarte frecvent în limba română şi astfel sunt luate în considerare foarte multe cazuri care nu reprezintă definiții. Un astfel de exemplu este:

<definiție>o asemenea practica este recomandată în cadrul documentelor complexe . </definiție>

Pentru *pron_def* şi *other_def* este necesară îmbunătățirea modului de extragere deoarece exemplele prea puține din corpusul de antrenament nu permit extragerea unor şabloane corecte.

4. Aplicații ale extragerii definițiilor

4.1 Extragerea definițiilor în sistemele de tip Întrebare-Răspuns

Sistemele de tip Întrebare-Răspuns (ÎR) sunt sisteme care primesc o întrebare în limbaj natural şi oferă unul sau mai multe răspunsuri ordonate, folosind o colecție de documente din care se extrage răspunsul. Sistemele ÎR au o arhitectură liniară, fiind

compuse din trei module principale: analiza întrebării, căutarea documentară şi extragerea răspunsului (Harabagiu şi Moldovan, 2003).

Primul modul se ocupă de analiza întrebării. Intrarea acestui modul este o întrebare în limbaj natural şi ieşirea una sau mai multe reprezentări ale întrebării care vor fi utilizate în etapele următoare. În această fază majoritatea sistemelor identifică tipurile semantice ale entităţilor din întrebare, constrângeri suplimentare legate de tipul întrebării şi al răspunsului, şi cuvintele cheie ce vor fi folosite de modulul de căutare.

Modulul de căutarea documentară este de regulă bazat pe un motor clasic de căutare şi are scopul de a identifica şi extrage o colecţie de paragrafe sau propoziţii relevante din colecţia de documente.

Ultima fază constă în extragerea şi ordonarea răspunsurilor. Din documentele obţinute în faza anterioară se extrag entităţile care au acelaşi tip ca tipul răspunsului căutat. În final, în funcţie de distanţa dintre entităţile extrase şi cuvintele cheie folosite de motorul de căutare se obţine o listă ordonată a răspunsurilor posibile.

Conform tipului răspunsului, avem următoarele tipuri de întrebări:

- ***Factoid*** – La întrebarea respectivă se aşteaptă un singur răspuns, ca în exemplele: "*Cine a descoperit oxigenul?*" sau "*Când s-a născut Eminescu?*" sau "*Care este căpitanul echipei de fotbal a României?*".
- ***Listă*** – Răspunsul la o astfel de întrebare este o enumerare: "*Ce judeţe au fost devastate de inundaţii?*" sau "*Care sunt cei mai bogaţi oameni din lume?*". Dificultatea în acest caz constă în faptul că de cele mai multe ori răspunsul nu se află într-o singură propoziţie şi este necesar să-l extragem din mai multe propoziţii, fraze sau documente.
- ***Definiţie*** – Acest tip de întrebare necesită o procesare mai complexă a textelor şi răspunsul final constă fie dintr-un fragment de text, fie este o combinaţie de mai multe documente: "*Ce este indigestia?*" sau "*Cine a fost Brâncuşi?*"

Vom prezenta în continuare câteva din caracteristicile sistemului dezvoltat de echipa noastră în cadrul competiţiei QA@CLEF2007[32] şi modalităţile în care am abordat întrebările de tip definiţie. Sistemul se bazează pe sistemul inter-lingual construit de noi anul trecut pentru engleză-română (Iftene şi Balahur-Dobrescu, 2007a).

În cazul întrebărilor de tip DEFINIŢIE, paragrafelor candidate extrase în faza de căutare documentară li se aplică un set de reguli din gramatica românească. Regulile din gramatica românească sunt transformate din formatul lxtransduce în şabloane Perl. Motivul acestei transformări a venit din faptul că sistemul de ÎR foloseşte un tip de adnotare a corpusului (format din lemă, parte de vorbire, entităţi de tip nume, etc.) care este diferit faţă de formatul fişierului XML folosit de lxtransduce, iar mărimea foarte mare a corpusului nu ne-a permis o transformarea dintr-un format într-altul.

Astfel, fiecare definiţie posibilă, având ca noţiune definită focusul întrebării, a fost extrasă şi adăugată la o mulţime de răspunsuri posibile, împreună cu un scor care reprezenta încrederea că aceasta reprezintă răspunsul final.

[32] Cross Language Evaluation Forum - http://www.clef-campaign.org/2007.html

Mulțimea grupurilor substantivale din fragmentul de text a fost examinată de asemenea cu atenție, în scopul identificării acelor grupuri substantivale care conțin noțiunea definită în jurul altor cuvinte funcționale (motivația acestei operații vine din situațiilor precum *racheta spațială Atlantis*, în care definiția corectă este *Atlantis este o rachetă spațială*). Mulțimea grupurilor substantivale este adăugată la o mulțime de răspunsuri posibile, dar cu un scor mai mic.

Mulțimea răspunsurilor posibile este ulterior ordonată folosind scorul asociat fiecărui răspuns. Răspunsurile care au cel mai mare scor sunt oferite în final ca răspunsuri posibile.

4.2 Construirea unei baze de date de cunoştinţe pentru realizarea inferențelor textuale

În cadrul competiției de inferențe textuale[33] (Dagan et al., 2006), participanților din exercițiul de evaluare li se pun la dispoziție perechi de fragmente de text (una sau mai multe propoziții în limba engleză), denumite perechi *text-ipoteză* (*T-H*). Participanții trebuie să construiască un sistem care, pentru fiecare pereche, trebuie să precizeze dacă avem inferență textuală sau nu (adică dacă ipoteza poate fi dedusă din text).

Ideea principală a sistemului construit de noi constă în transformarea ipotezei folosind cunoaştere semantică suplimentară din resurse precum WordNet, DIRT (Lin şi Pantel, 2001), baze de date de acronime, etc. În plus, am construit un sistem care achiziționează cunoaştere suplimentară din Wikipedia[34] (Iftene şi Balahur-Dobrescu, 2007b). Apoi calculăm distanța dintre arborii de dependență asociați textului inițial şi ipotezei obținute în urma transformărilor. În final, în funcție de această distanță decidem dacă avem inferență sau nu.

Deoarece exista informație care nu putea fi dedusă din resursele existente am cules informație suplimentară sub forma celei prezentate în tabela 3.

Tabela 3: Cunoaştere suplimentară

Argentine [is] Argentina
Netherlands [is] Holland
2 [is] two
Los Angeles [in] California
Chinese [in] China

Aceste informație suplimentară a fost extrasă din Wikipedia românească folosind gramatica prezentată în secțiunea 3 (vezi lucrarea Iftene şi Balahur-Dobrescu, 2007b pentru detalii suplimentare).

5. Concluzii

Lucrarea de față a prezentat gramatica dezvoltată în cadrul proiectului LT4eL pentru extragerea automată a definițiilor. Definițiile au fost împărțite în şase categorii şi am prezentat rezultatele sistemului pentru fiecare categorie. Extragere automată a

[33] Competiția RTE: http://www.pascal-network.org/Challenges/RTE/

[34] http://ro.wikipedia.org

definiţiilor din text poate îmbunătăţi semnificativ performanţele unui sistem de Întrebare-Răspuns sau baza de cunoştinţe ataşată unui sistem de Inferenţe textuale.

Pentru îmbunătăţirea gramaticii pentru limba română, o etapă necesară este validarea acesteia pe un corpus nou, pentru a verifica corectitudinea sistemului.

Pe viitor, dorim să completăm mulţimea de reguli identificate manual din secţiunea 3 automat prin tehnici de bootstrapping asemănătoare celor prezentate în lucrarea (Riloff şi Jones, 1999).

Mulţumiri. Lucrarea prezintă rezultatele obţinute de echipa românească în cadrul proiectul european LT4eL (Language Technologies for e-Learning), STREP FP6-2004-IST-4, şi a proiectului naţional CEEX Rotel 29/2006. Mulţumiri speciale sunt adresate celorlalţi membri ai echipei româneşti din cadrul proiectului: Dan Cristea şi Corina Forăscu.

Referinţe bibliografice

Carletta, J. (1996) Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22: 249–254.

Dagan, I., Glickman, O. and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. *In Quiñonero-Candela et al., editors, MLCW 2005*, LNAI Volume 3944, pages 177-190. Springer-Verlag.

Harabagiu, S. and Moldovan, D. (2003) Question answering. *In Ruslan Mitkov, editor, Oxford Handbook of Computational Linguistics*, chapter 31, pages 560 – 582. Oxford University Press.

Iftene, A. and Balahur-Dobrescu, A. (2007a) Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment. *In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Pp. 125-130. 28-29 June, Prague, Czech Republic.

Iftene, A. and Balahur-Dobrescu, A. (2007b) Name entity relations discovery using Wikipedia for Romanian. *The third Workshop on Romanian Linguistic Resources and Tools for Romanian Language Processing*. 14-15 Decembrie. Iaşi, România.

Lin, J. (2005) Evaluation of Resources for Question Answering Evaluation. *In Proceedings of the 28th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2005), Salvador, Brazil.

Lin, D., and Pantel, P. (2001) *DIRT - Discovery of Inference Rules from Text*. In Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01). pp. 323-328. San Francisco, CA.

Liu, B., Chin, C. W., Ng, H. T. (2003) Mining Topic-Specific Concepts and Definitions on the Web. *In Proc. of the Twelfth Int. WWW Conference 2003*.

Monachesi, P., Lemnitzer, L. and Simov, K. (2006) Language Technology for eLearning *Poster presentation at First European Conference on Technology Enhanced Learning*, 1-4 October, Crete, Greece. http://www.ectel06.org/.

Mureşan, S. and Klavans, J. (2002) A Method for Automatically Building and Evaluating Dictionary Resources. *Proceedings of LREC 2002*.

Pistol, I., Trandabăț, D., Iftene, A., Cristea, D., Forăscu, C. (2006) Processing Romanian linguistic Resources in the LT4eL project (in Romanian). *In Proc. of the Wokshop Linguistic Resources and Tools for Processing Romanian Language*, C. Forăscu, D. Tufiş, D. Cristea (eds.). Iasi, Romania, November 2006. University "Al.I. Cuza" Publishing House.

Riloff, E. and Jones, R. (1999) Learning Dictionaries for Information Extraction Using Multi-level Boot-strapping. *In Proceedings of AAAI-99 Workshop on Machine Learning for Information Extraction.*

Tobin. R. (2005) Lxtransduce A replacement for fsgmatch. http://www.ltg.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html.

Tufiş, D., Dragomirescu L. (2004) - Tiered Tagging Revisited. In Proceedings of the 4th LREC Conference, Lisabona, 2004, pp. 39-42.

CONSTRUIREA UNUI SISTEM DE ÎNTREBARE RĂSPUNS PENTRU LIMBA ROMÂNĂ

ADRIAN IFTENE[1], IONUȚ PISTOL[1], CORINA FORĂSCU[1,2], DIANA TRANDABĂȚ[1,3], ALEXANDRA BALAHUR-DOBRESCU[1,4], DIANA COTELEA[1], IULIANA DRĂGHICI[1]

[1]Facultatea de Informatică, Universitatea "Al. I. Cuza" Iaşi, [2]Institutul de Cercetare în Inteligență Artificială Bucureşti, [3]Institul de Informatică Teoretică Iaşi, [4]Universitatea Alicante, Departamentul de Limbaje şi Sisteme Informatice, Spania

{adiftene, ipistol, corinfor, dtrandabat, abalahur, dcotelea, idraghici }@info.uaic.ro

Rezumat

Începând cu anul 2007, în cadrul competiției QA@CLEF a fost introdusă o colecție de documente în română pentru regăsirea răspunsului, o variantă înghețată a Wikipediei româneşti din luna decembrie anul 2006. Articolul de față prezintă etapele construirii sistemului de tip întrebare-răspuns care funcționează pentru limba română. Spre deosebire de competiția similară din 2006, accentul a căzut pe o nouă caracteristică introdusă anul acesta în competiție: gruparea întrebărilor pe focus, ceea ce a presupus rezoluția anaforei la nivel de întrebare. De asemenea un modul important al sistemului îl constituie sistemul de inferențe textuale care poate duce la o ordonare mai bună a răspunsurilor.

1. Introducere

Primul sistem de Întrebare-Răspuns[35] (ÎR) românesc a fost dezvoltat în anii '80 (Tufiş şi Cristea 1985) şi era reprezentat de o interfață ce facilita comunicarea cu o rețea semantică (care codifica cunoaşterea). Astăzi sistemele de ÎR folosesc documente text ca bază de cunoaştere şi integrează tehnici de prelucrare a limbajului natural (PLN) pentru a găsi (într-o colecție dată de documente sau prin căutare pe web) răspunsul la o întrebare pusă în limbaj natural.

România a participat pentru prima dată la o competiție CLEF în 2006, în cadrul secțiunii QA@CLEF[36] cu un sistem dezvoltat de UAIC[37] şi RACAI[38] (Puşcaşu et al., 2006) pentru perechea de limbi ROmână-ENgleză. În anul 2007 organizatorii români din QA@CLEF au dat posibilitatea participanților să aleagă între exercițiile RO-RO, EN-RO, RO-EN (prima reprezentând limba sursă, a întrebărilor şi cea de a doua – limba țintă, cea a documentelor în care se caută răspunsurile). Anul acesta pentru limba română colecția de documente în care s-a căutat răspunsul a fost formată dintr-o variantă înghețată a Wikipediei româneşti din luna decembrie 2006.

Sistemele de ÎR folosesc o arhitectură generală de tip *pipe-line*, în care prelucrarea parcurge trei etape principale: analiza întrebării, căutarea documentară şi extragerea răspunsului (Harabagiu, Moldovan, 2003). Sistemul prezentat este o variantă a

35 Question Answering (QA) – rom.: Întrebare-Răspuns (ÎR)
36 Multilingual Question Answering at CLEF: http://clef-qa.itc.it/
37 Universitatea "Al.I.Cuza" Iaşi: http://www.uaic.ro/
38 Romanian Academy Center for Artificial Intelligence: http://www.racai.ro/

arhitecturii generale, cu particularizări specifice legate de reprezentare şi procesare pentru fiecare din componentele amintite mai sus.

De la an la an în competiția QA@CLEF se adaugă noi caracteristici, cu scopul de simula cât mai fidel situații concrete. Anul acesta provocarea a fost gruparea întrebărilor pe domenii. În cadrul unui domeniu toate întrebările se referă la o anumită temă, prezentă fie în prima întrebare, fie în răspunsul acesteia; există posibilitatea folosirii legăturilor anaforice între întrebările din aceeaşi topică.

Arhitectura generală a sistemului este ilustrată în Figura 1, iar modulele mai importante vor fi prezentate în capitolele următoare, insistând pe componentele nou introduse. În ultima parte vom prezenta rezultatele obținute la competiția de anul acesta, concluziile, precum şi direcțiile de lucru viitoare.

2. Etapele parcurse de Sistemul ÎR

2.1 Pre - procesări asupra corpusului

Filtrarea documentelor

Corpusul utilizat pentru extragerea răspunsului a fost reprezentat de o colecție completă a documentelor în limba română disponibile pe Wikipedia[39] în octombrie 2006. Acest set de documente a fost pus la dispoziție de organizatorii competiției şi însumează 180.500 fişiere cu o dimensiune totală de 1.9 GB. Documentele includ discuții de pe forum, imagini şi profiluri de utilizatori.

Documentele au fost disponibile în format *XML* si *html*. Datorită specificului instrumentelor de pre-procesare, colecția de documente wiki a fost preluată în format *html*. O primă etapă de procesare a constat în filtrarea fişierelor nerelevante pentru extragerea răspunsului, păstrând doar articolele propriu-zise. O a doua etapă de filtrare a constat în transformarea fişierelor din format *html* în format *txt*, păstrând totuşi unele informații cum ar fi titlul articolului şi marcajele de paragraf.

În scopul reducerii duratei de procesare am rulat lanțul de pre-procesare pe un sistem capabil de a rula 10 procese în paralel cu o viteză considerabil superioară unui sistem desktop mediu. Acest lucru a redus timpul de procesare pentru întreg corpusul românesc de la aproximativ 6 ore la 30 de minute.

După aceste etape de pre-procesare, dimensiunea corpusului s-a redus semnificativ până la 175 MB. Fără această etapă, procesarea lingvistică ar fi fost mult mai costisitoare, atât ca timp cât şi ca spațiu.

[39] http://ro.wikipedia.org/

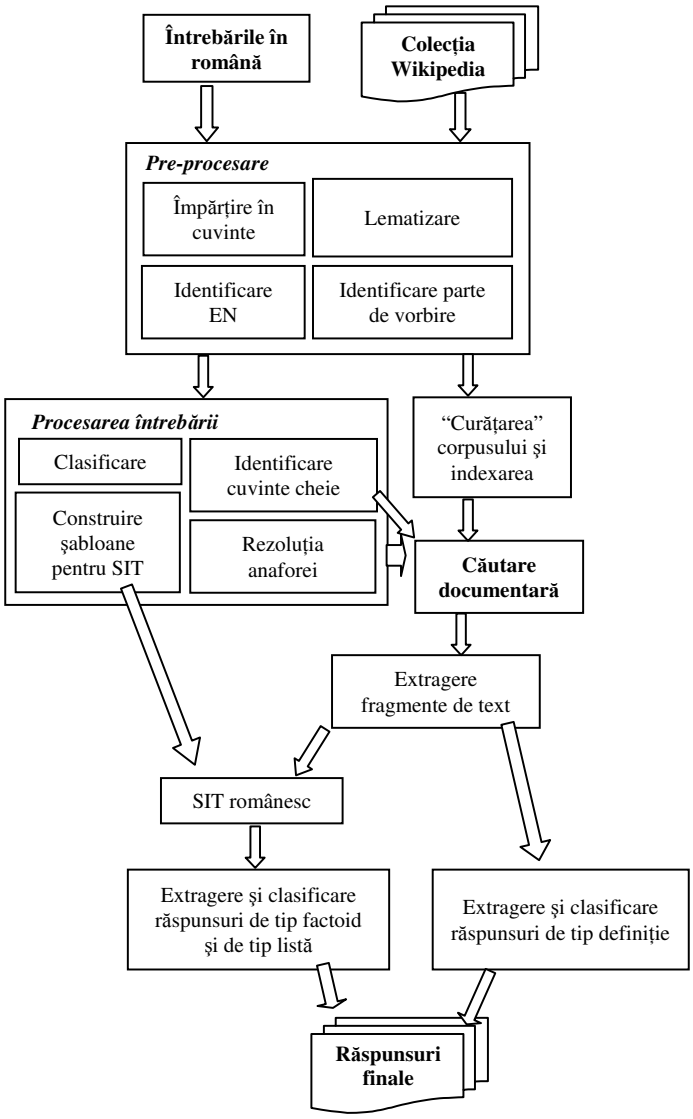


Figura 1: Arhitectura generală a sistemului de întrebare-răspuns românesc

Pre-procesarea lingvistică

Corpusul Wikipedia şi setul de întrebări au trecut prin aceleaşi etape de procesare lingvistică:

6. tokenizare: s-a folosit un tokenizator implementat în Java;

7. POS-tagging: am utilizat un model de limbă dezvoltat la RACAI de colectivul prof. Dr. Dan Tufiş;

8. lematizare: am utilizat un lematizator implementat în Perl ce utilizează un dicționar de forme flexionate pentru limba română realizat de grupul de la Chişinău;

9. marcarea EN (Entități de tip nume) de tip Persoană, Locație, Măsură şi Dată a fost realizată folosind aplicația ANNIE inclusă în GATE (Cunningham et. al. 2002).

Această etapă a produs versiunea adnotată a corpusului ce a fost indexată ulterior (vezi secțiunea 2.3). Întrebările astfel procesate au fost utilizate în etapa de analiză.

2.2 Analiza întrebării

Această etapă are ca scop identificarea tipului răspunsului aşteptat. În plus, se identifică tipul întrebării, focusul întrebării, şi mulţimea cuvintelor cheie relevante pentru întrebare. Pentru atingerea acestor obiective, s-au parcurs următorii paşi:

- **Identificarea grupurilor nominale. Extragerea entităților de tip nume**

Folosind mulțimea de întrebări ca intrare, pe baza adnotărilor morfo-sintactice, s-au implementat reguli de identificare a grupurilor nominale. Acelaşi instrument folosit în faza de pre-procesare a permis identificarea entităților de tip nume din întrebare.

- **Focusul întrebării**

Focusul întrebării este cuvântul sau secvența de cuvinte care precizează ce anume se caută ca răspuns sau despre ce anume este vorba. Pentru aceasta am considerat fie primul substantiv din întrebare (ca în *Ce țară*) sau prima componentă a primului grup substantival atunci când acesta apare după verbul principal al întrebării sau dacă apare după verbul "*a fi*".

- **Tipul răspunsului**

Sistemul dezvoltat este capabil să identifice următoarele tipuri de răspuns: PERSOANĂ, LOCAȚIE, ORGANIZAȚIE, TEMPORALĂ, NUMERICĂ, DEFINIȚIE şi ALTELE. Ataşarea unuia dintre aceste tipuri unei întrebări analizate s-a realizat folosind şabloane specifice pentru fiecare tip în parte. În cazul întrebărilor ambigue (de exemplu cele care încep cu *Ce*), am folosit focusul întrebării (de exemplu, în cazul întrebării *Ce oraş este identificat cu Troia Homerică?*, care are focusul *oraş*, folosim tipul asociat focusului care este LOCAȚIE).

- **Tipul întrebării**

Tipul întrebării poate fi unul din următoarele FACTOID, DEFINIȚIE sau LISTĂ. Pentru identificarea acestuia am folosit două reguli simple: dacă tipul răspunsului este DEFINIȚIE, atunci evident tipul întrebării este DEFINIȚIE; dacă focusul întrebării este un substantiv la plural, atunci tipul întrebării este LISTĂ, în celelalte cazuri fiind FACTOID.

- **Rezoluția anaforei**

Noua caracteristică introdusă anul acesta în competiția QA@CLEF, gruparea întrebărilor pe domenii, a dus la mărirea gradului de dificultate, prin faptul că a fost necesară introducerea unui modul special în arhitectura generală a sistemului, care să se ocupe de rezoluția anaforei. De exemplu, pentru primul grup din setul de întrebări avem:

Tabel 1: Primul grup din setul de întrebări

<Group id=1>
<Question id=1> Ce faimos romancier, nuvelist şi realizator american de povestiri a trăit între anii 1899 şi 1961?
</Question>
<Question id=2> Pentru ce premiu a fost laureat în anul 1954?
</Question>
<Question id=3> În ce an a fost laureat al Premiului Pulitzer?
</Question>
</Group>

Se observă că în întrebările 2 şi 3 trebuie înlocuit pronumele "*el*" cu răspunsul de la prima întrebare. Pentru a rezolva această problemă am adoptat două metode de rezoluție a anaforei prezentate în cele ce urmează:

11. Forța Brută

În această situație toate cuvintele cheie de la prima întrebare din grup sunt adăugate la cuvintele cheie ale următoarelor întrebări din grup. Iată ce ar însemna pentru exemplul prezentat mai sus acest lucru:

Cuvintele cheie pentru prima întrebare din grup sunt {*faimos, romancier, nuvelist, realizator, american, a trăi, 1899, 1961*}, şi doar acestea vor fi folosite în procesul de căutare a răspunsului.

Cuvintele cheie pentru a doua întrebare sunt {*premiu, laureat, 1954*}, dar în procesul de căutare a răspunsului pentru această întrebare, vom folosi în virtutea regulii de mai sus şi cuvintele cheie de la prima întrebare: {*faimos, romancier, nuvelist, realizator, american, a trăi, 1899, 1961, premiu, laureat, 1954*}.

12. Folosirea răspunsurilor

În acest caz folosim răspunsul de la prima întrebare din grup care se adaugă la lista curentă de cuvinte cheie. Pentru exemplul nostru ar însemna că la cuvintele cheie de la întrebarea a doua din grup s-ar adăuga răspunsul de la prima, care este "Ernest Hemingway" şi s-ar obține mulțimea: {*premiu, laureat, 1954, Ernest Hemingway*}.

Desigur a doua metodă este mai bună, dar ea depinde de capacitatea sistemului nostru de a identifica răspunsul corect. Deoarece, în multe cazuri nu am reuşit să extragem răspunsul corect pentru prima întrebare din grup, am preferat să folosim o combinație a celor două metode.

- **Generarea cuvintelor cheie**

Inițial se porneşte cu o mulțime formată din: focus, entitățile de tip nume, celelalte substantive din întrebare, şi toate verbele care nu sunt auxiliare în întrebare. Ulterior se completează lista cu sinonimele fiecărui cuvânt.

2.3 Crearea indexului şi căutarea documentară

În aceasta etapă se urmăreşte extragerea paragrafelor relevante ataşate fiecărei întrebări. Pentru obținerea acestora s-au parcurs următorii paşi:

a) Formarea interogărilor

Interogările sunt formate dintr-o succesiune de cuvinte cheie, fiecare precedat de un operator Lucene[40] opțional, obținând în acest fel o expresie regulată pe care am folosit-o în căutare. Atât focusul cât şi sinonimele cuvintelor cheie au fost incluse în interogări. De exemplu, pentru prima întrebare din setul de intrare interogarea obținută este:

+romancier (faimos renumit vestit) (nuvelist nuvelistic) (realizator înfăptuitor) american (trăi viețuit) 1899 1961

unde operatorul "+" înseamnă obligativitate, iar absența lui indică caracterul opțional, grupurile de cuvinte dintre parantezele rotunde semnifică faptul că e posibil ca în timpul căutării să obținem doar unul din cuvintele.

b) Indexarea colecției de documente

Indexarea colecției de documente s-a făcut folosind lemele cuvintelor, stabilite în etapa de pre-procesare. Indexarea s-a făcut atât la nivel de paragraf cât şi la nivel de articol, după cum urmează:

1. *Indexare la nivel de paragraf*

   Scopul acestui tip de indexare este de a putea identifica şi extrage o cantitate cât mai redusă de informație relativ la o anumită întrebare. Această metodă s-a dovedit în multe cazuri ineficientă, deoarece în documentele html pe care le-am indexat, numărul paragrafelor a fost foarte mare (în jur de 200.000) şi deoarece un paragraf era de multe ori doar o propoziție, fiind foarte puține cazurile în care găseam toate cuvintele cheie căutate într-un paragraf. Avantajul acestei metode a fost evident în cazurile de succes, deoarece paragraful fiind de dimensiune redusă am putut identifica relativ uşor răspunsul căutat.

2. *Indexare la nivel de articol Wikipedia*

   Pentru a primi totuşi un răspuns în cazurile de insucces de mai sus, am realizat şi acest tip de indexare. Dezavantajul acestei metode a fost evident în momentul în care a trebuit să extragem răspunsul la întrebarea pusă, cantitatea mare de informație făcând necesară crearea unor algoritmi mai complecşi pentru identificarea şi extragerea răspunsurilor finale.

c) Extragerea paragrafelor relevante

Folosind interogările create la punctul a) şi indexul creat la punctul b), am folosit utilitarul Lucene care, în funcție de interogările primite, a extras din documentele indexate părțile de text relevante pentru fiecare întrebare.

2.4 Extragerea Răspunsului

Operația se bazează pe tipul răspunsului aşteptat, focusul întrebării, mulțimea de cuvinte cheie, părțile de text obținute în urma căutării pe partea de vorbire, lemă şi informații de tip entități de tip nume şi indicatorul de relevanță al paragrafelor determinat de Lucene. Procesul de extragere depinde de tipul aşteptat al răspunsului: când răspunsul are un anumit tip de entitate de tip nume, modulul de extragere a răspunsului identifică aceste entități în fiecare propoziție întoarsă de Lucene; când tipul răspunsului nu este un nume

[40] http://lucene.apache.org/

de entitate, procesul de extragere se bazează în principal pe recunoaşterea focusului, în acest caz şabloanele sintactice de găsire a răspunsului bazate pe focus fiind foarte importante.

2.5 Extragerea răspunsurilor de tip MĂSURĂ

Deoarece această categorie de răspunsuri poate fi divizată in funcție de elementul măsurabil, au fost identificate 5 categorii pentru care s-au construit şabloane de căutare. Categoriile găsite au fost: suprafață; lungime, lățime, înălțime; procentaj; viteză; altele. Răspunsul la o întrebare face parte din una dintre primele 4 categorii dacă focusul întrebării este unul dintre cuvintele care definesc respectiva categorie. În funcție de categorie este căutat în paragrafele relevante un număr, întreg sau zecimal, urmat de un anumit caracter sau grup de caractere.

De exemplu, pentru cazul lungime, lățime, înălțime, este căutat un număr urmat de *m, km, metri, metru, kilometri, kilometru, ar sau ari.*

2.6 Extragerea răspunsurilor utilizând modulul de inferențe textuale

Motivație

Recunoaşterea inferențelor textuale (*Recognizing Textual Entailment – RTE*) reprezintă o sarcină generică propusă recent (Dagan et al., 2006) cu scopul de a crea un cadru de lucru independent de aplicație ce surprinde mijloacele de capturare a inferențelor semantice majore necesare în multe din aplicațiile pentru procesarea limbajului natural. Mai concret, noțiunea aplicată de recunoaştere a inferențelor textuale este definită (Dagan et al., 2006) ca o relație direcțională între perechi de texte, notate prin *T* – textul din care se va face inferența şi *H* – ipoteza posibil inferată. Spunem că *H* este dedusă-inferată din *T* dacă, citind *T* putem deduce că *H* este cel mai probabil adevărată.

Un sistem ÎR trebuie să identifice texte din care răspunsul aşteptat poate fi inferat. Fiind dată întrebarea :

Întrebare: „Cine a scris ´Oda pentru Joy´?"

se poate proceda la transformarea ei într-un enunț cu o necunoscută de tip PERSOANĂ astfel:

Enunț: PERSOANĂ a scris ´Odă pentru Joy´.

Printre fragmentele de text care conțin sintagma cheie *´Odă pentru Joy´* se găseşte de exemplu textul:

Text: "*Ode pentru Joy este o odă scrisă în 1785 de poetul, dramaturgul şi istoricul german, Friedrich Schiller*." De unde poate fi determinat un candidat pentru necunoscuta *PERSOANĂ* – Friedrich Schiller. Ipoteza este construită prin înlocuirea necunoscutei cu candidatul găsit.

Ipoteza: "Friedrich Schiller a scris ´Ode pentru Joy´."

Pentru a verifica dacă termenul candidat găsit este corect, şi prin urmare dacă răspunsul dat la întrebare este corect, folosim sistemul de inferenţe textuale cu *Textul* şi *Ipoteza* identificate mai sus.

Din testele pe care le-a făcut pe limba engleză cu sistemul construit pentru competiţia RTE3 (Iftene şi Balahur-Dobrescu, 2007a), am remarcat că utilizarea unul modul de inferenţe textuale are ca rezultat o îmbunătăţire a clasificării răspunsurilor posibile. Acest lucru este posibil deoarece sistemul de inferenţe textuale face o analiză semantică a contextului întrebării şi nu doar prin calculează distanţe lexicale între cuvinte. Datorită acestui lucru am creat un sistem de inferenţe textuale pentru limba română (Iftene şi Balahur-Dobrescu, 2007b), pe care l-am folosit ca modul în cadrul sistemului de întrebare-răspuns românesc. Testele, care au fost efectuate din păcate după terminarea competiţiei, au relevat o creştere a preciziei cu aproximativ 5 %.

Construirea şabloanelor

Pentru a putea folosi sistemul de inferenţe textuale românesc, am construit şabloane de transformare a întrebărilor cu răspunsuri de tip PERSOANĂ, LOCALITATE, DATĂ şi ORGANIZAŢIE în enunţuri cu necunoscute, astfel:

1. Întrebări cu răspuns de tip PERSOANĂ. Fie exemplul:

Întrebare: Cine a creat serialul Twin Peaks?

prin eliminarea expresiei de interogare şi adăugarea necunoscutei *PERSOANĂ*, ce va fi înlocuită pentru crearea ipotezelor cu entităţile nominale de tip *PERSOANĂ* găsite în răspunsurile posibile, întrebarea se transformă în:

Şablon: PERSOANĂ a creat serialul Twin Peaks.

2. Întrebări cu răspuns de tip LOCALITATE. Fie exemplul:

Î: Unde s-au desfăşurat Jocurilor Olimpice în anul 1976?

întrebarea se transformă pe modelul de mai sus, folosind în plus cuvântul "***în***" în:

Ş: În LOCALITATE s-au desfăşurat Jocurilor Olimpice în anul 1976.

3. Întrebări cu răspuns de tip DATĂ. Fie exemplul:

Î: Când a domnit Alexandru Ioan Cuza?

întrebarea se transformă în:

Ş: În DATĂ a domnit Alexandru Ioan Cuza.

4. Întrebări cu răspuns de tip ORGANIZAŢIE. Fie exemplul:

Î: Ce organizaţie mondială a fost construită în anul 1945 cu scopul promovării unei economii globale sănătoase?

pe acelaşi model ca mai sus avem:

Ş: ORGANIZAŢIE organizaţie mondială a fost construită în anul 1945 cu scopul promovării unei economii globale sănătoase.

3. Analiza Rezultatelor

Rezultatele evaluării pentru sistemul nostru sunt prezentate în tabelul de mai jos.

Tabel 2: Rezultatele oficiale

Evaluarea Rezultatelor		
Z	NECUNOSCUT	0
R	CORECT	24
U	NEJUSTIFICAT	1
W	INCORECT	171
X	INEXACT	4
	TOTAL	**200**

Fiecare răspuns a fost evaluat ca fiind NECUNOSCUT (răspuns neevaluat), CORECT (răspuns corect), NEJUSTIFICAT (răspuns care nu putea fi găsit în secțiunile de text justificatoare), INCORECT (răspuns greşit) sau INEXACT (răspuns incomplet). Precizia sistemului a fost de 12%, valoare care este foarte apropiată de valoarea obținută anul trecut. Deşi sistemul a fost mult îmbunătățit, menținerea preciziei trebuie pusă în seama dificultății crescute a sarcinilor, mai ales datorită grupării întrebărilor pe domenii.

4. Concluzii

În configurația actuală, realizarea implementează cele trei niveluri esențiale ale unui astfel de sistem. Evaluarea arată o precizie de aproximativ 12%, care, deşi suficient de scăzută comparativ cu celelalte sisteme RO-RO, de 30% (Tufiş et al., 2007), indică o plasare în rând cu alte sisteme participante la competiția QA@CLEF (Giampiccolo et al., 2007).

Analizând rezultatele obținute se pot observa două probleme majore: prima este la întrebările de tip listă (unde grupurile implicate nu au reuşit să răspundă corect), iar a doua este la gruparea întrebărilor pe domenii (unde sistemele implicate în competiție au reuşit de regulă să răspundă corect doar la prima întrebare dintr-un grup de întrebări grupate sub aceeaşi topică). De asemenea, faptul că găsirea răspunsurilor la anumite întrebări depindea de succesul găsirii răspunsului corect la întrebările anterioare din grup, le-a scăzut considerabil probabilitatea găsirii răspunsului.

Un aspect pozitiv a fost folosirea experimentală a noii componente care realizează inferențele textuale, a cărei utilitate sperăm să se dovedească benefică pe viitor.

Experiența câştigată va fi folosită la îmbunătățirea sistemului pentru participări la ediții viitoare ale QA@CLEF cât şi în cadrul proiectului SIR-RESDEC, aprobat recent spre finanțare de CMNP.

Mulțumiri. Lucrarea prezintă rezultatele obținute de echipa românească în cadrul proiectul european LT4eL (Language Technologies for e-Learning), STREP FP6-2004-IST-4, şi a proiectului național CEEX Rotel 29/2006. Autorii mulțumesc celorlalți membri ai echipei de la UAIC: Dan Cristea, Iustin Dornescu, Alexandru Moruz, Marius Răschip.

Referințe bibliografice

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002) GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July

Dagan, I., Glickman, O. and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. *In Quiñonero-Candela et al., editors, MLCW 2005*, LNAI Volume 3944, pages 177-190. Springer-Verlag.

Giampiccolo, D., Forner, P., Peñas, A., Ayache, C., Cristea, D., Jijkoun, V., Osenova, P., Rocha, P., Săcăleanu, B., Sutcliffe, R. (2007) Overview of the CLEF 2007 Multilingual Question Answering Track. *In Alessandro Nardi and Carol Peters (eds.) Working Notes for the CLEF 2007 Workshop*, 19-21 September, Budapest, Hungary.

Harabagiu, S., Moldovan, D. (2003) *Question Answering*. In: The Oxford Handbook of Computational Linguistics. Oxford; New York: Oxford University Press.

Iftene, A., Balahur-Dobrescu, A. (2007a) Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment. *In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Pp.125-130. 28-29 June, Prague, Czech Republic.

Iftene, A., Balahur-Dobrescu, A. (2007b) Improving a QA System for Romanian Using Textual Entailment. *In Proceedings of RANLP workshop "A Common Natural Language Processing Paradigm For Balkan Languages"*. ISBN 978-954-91743-8-0, Pp. 7-14, September 26, 2007, Borovets, Bulgaria.

Puşcaşu, G., Iftene, A., Pistol, I., Trandabăț, D., Tufiş, D., Ceauşu, A., Stefănescu, D., Ion, R., Dornescu, I., Moruz, A., Cristea, D. (2007): Cross-Lingual Romanian to English Question Answering at CLEF 2006. CLEF 2006, Revised Selected Papers, *Lecture Notes in Computer Science* vol. 4730/2007, pp. 385-394.

Tufiş, D., Cristea, D. (1985). *IURES: A Human Engineering Approach to Natural Language Question Answering*, in W. Bibel, B.Petkoff (eds), Artificial Intelligence: Systems, Applications, Methodology, North Holland.

Tufiş, Ștefănescu, D., Ion, R., Ceauşu, A. (2007). *RACAI's Question Answering System at QA@CLEF 2007*. In Alessandro Nardi and Carol Peters (eds.) Working Notes for the CLEF 2007 Workshop, 19-21 September, Budapest, Hungary.

DIAC+: UN SISTEM PROFESIONAL DE RECUPERARE A DIACRITICELOR

DAN TUFIŞ, ALEXANDRU CEAUŞU

Institutul de Cercetări pentru Inteligență Artificială
Str. 13 Septembrie, nr. 13, Bucureşti 050711, România

{tufis, aceausu}@racai.ro

Rezumat

Această lucrare prezintă pe scurt o variantă îmbunătățită a soluției de inserție de diacritice – DIAC – introdus în (Tufiş & Chițu, 1999) care beneficiază de noile metode de adnotare morfo-sintactică şi de recuperare a MSD-urilor (etichete morfo-lexicale) bazate pe folosirea principiului maximizării entropiei. Se evidențiază de asemenea oportunitatea folosirii setului extins de etichete (MSD) pentru operația de recuperare în comparație cu cel redus (Ctag).

1. Introducere

Există mai multe limbi care folosesc caractere diacritice care nu se află în setul de caractere ASCII. Pentru unele dintre aceste limbi, cele mai multe diacritice pot fi recuperate în mod determinist, dar aceste cazuri nu reprezintă norma. Mai mult, dificultatea acestei sarcini diferă de la o limbă la alta în funcție de rolul funcțional al acestor caractere. Pentru limba română, restaurarea automată a diacriticelor este o adevărată provocare, atât datorită frecvenței lor, cât şi contribuției semnificative pe care acestea o au la dezambiguizarea morfo-lexicală şi semantică a cuvintelor. Găsirea unei metode de inserție automată a diacriticelor este importantă nu doar pentru textele vechi valoroase stocate în format electronic, dar şi pentru cele contemporane, care continuă să fie produse într-o formă fără diacritice. Vasta majoritate a textelor româneşti publicate pe web sunt scrise parțial cu diacritice sau complet fără caractere diacritice. De aceea, colectarea de texte de pe web, pentru realizarea de corpusuri electronice relevante ale limbii române scrise, este serios îngreunată.

Verificarea automată şi corectarea greşelilor de ortografie este una dintre cele mai vechi aplicații ale procesării limbajului natural. În cazul celor mai multe aplicații de corectare a ortografiei, corectarea se face independent de contextul cuvântului corectat. Cele mai multe erori tipografice sunt cele în care caracterele lipsesc, apar inversate, sau sunt introduse din greşeală caractere în plus. Sunt însă şi erori de ortografie care nu pot fi rezolvate independent de contextul cuvântului analizat, printre acestea se numără şi inserarea automată a diacriticelor.

Vom descrie un sistem de recuperare a diacriticelor pentru limba română, bazat pe progrese recente din tehnologia adnotării morfo-lexicale probabilistice. Abordări similare au fost propuse în (Tufiş & Chițu, 1999) pentru română, (Simard, 1998) pentru franceză, (El-Bèze et al., 1994) (cf. Simard, 1998) de asemenea pentru franceză. Mihalcea (2001) prezintă o metodă de recuperare a caracterelor diacritice în română folosind un model cu n-grame. Yarowsky (1994) rezolvă aceeaşi problemă pentru spaniolă (în special) şi franceză dar, în locul adnotării morfo-lexicale, foloseşte o teorie

a listelor de decizie care oferă performanţe satisfăcătoare (viteză şi acurateţe) cu preţul unui model de limbă care este "destul de slab: în absenţa unui corpus de antrenare adnotat manual, el şi-a construit modelul pe seturi *ad hoc* de etichete" (Simard, 1998). În comparaţie cu franceza, româna foloseşte mult mai des caractere diacritice, iar absenţa lor creează şi mai multe dificultăţi.

2. Caracterele diacritice în limba română

Limba română are 5 caractere diacritice: *ă,â,î,ş* şi *ţ* (plus variantele lor majuscule). Un text fără diacritice va avea aceste caractere substituite prin caractere ASCII: a (pentru *ă* şi *â*), respectiv *i*, *s* şi *t*. Acest lucru se întâmplă, de exemplu, când exportăm dintr-un editor de texte care recunoaşte diacritice, într-un format text. Pentru un număr semnificativ dintre cuvintele care ar trebui, dar nu conţin diacritice, recuperarea este deterministă deoarece variantele fără diacritice ale acestor cuvinte nu sunt lexeme în limba română. Dar în cele mai multe dintre cazuri, absenţa diacriticelor creează ambiguitate autentică, greu de rezolvat chiar şi de om dacă i se prezintă doar un context limitat al ocurenţei cuvântului.

Cuvintele limbii române pot fi împărţite în două mari clase: cuvinte ce nu conţin diacritice în nici una din formele lor omografe (carte, autor, paragraf etc.) şi cuvinte în care prezenţa sau absenţa unuia sau a mai multor caractere diacritice fie exclud cuvântul respectiv din lexicul limbii române fie îi schimbă categoria gramaticală, atributele lexicale sau chiar sensul. Prima categorie de cuvinte este cea mai numeroasă şi o vom numi clasa cuvintelor N (Non-diacritice). Cea de a doua categorie de cuvinte o vom numi clasa cuvintelor D (Diacritice). Facem observaţia că un cuvânt legal al limbii române chiar dacă nu conţine diacritice nu este neapărat un cuvânt de tip N ori altfel spus mulţimea cuvintelor fără diacritice dintru text este un superset al cuvintelor de tip N din acel text. De pildă în cuvântul *lat* (adjectiv sau substantiv referitor la lăţimea unui obiect), substituţia caracterului *t* cu diacriticul *ţ* generează un cuvânt legal al limbii române: *laţ* (substantiv: Nod larg la capătul unei sfori, întocmit în aşa fel încât să se poată strânge în jurul unui punct fix; instrument pentru prins păsări sau animale, constând dintr-un ochi de sfoară, de sârmă etc.). Pe de altă parte, un cuvânt de tip N nu este neapărat neambiguu (de pildă cuvântul *mare,* deşi de tip N, în diferite contexte este fie substantiv fie adjectiv). Prin urmare, distincţia între cuvintele de tip N şi cele de tip D se poate face doar în raport cu un lexicon de referinţă cu o cât mai largă acoperire lexicală. Procedura este relativ simplă:

a) fie un lexicon de conţinând intrări unice de tipul:

<formă-ocurenţă><formă-lemă : descriere morfo-sintactică>$^{+}$

în care:

forma-ocurenţă şi *formă-lemă* sunt scrise în conformitate cu normele tipografice ale limbii (adică, atunci când formele respective trebuie să conţină unul sau mai multe caractere diacritice, ele le şi conţin);

descriere morfo-sintactică (una sau mai multe) reprezintă o codificare neambiguă a proprietăţilor gramaticale a formei ocurenţă a unei anumite leme (de pildă, pentru forma ocurenţă "manifestaţiilor" perechea *manifestaţie : Ncfpry* indică faptul că acest cuvânt este un substantiv

feminin, cazul genitiv/dativ, numărul plural, cu articulat, a cărui lemă este *manifestație*); în cazul mai multor descrieri *morfo-sintactice*, forma ocurență este ambiguă (de pildă pentru cuvântul *vin* vor exista trei interpretări distincte; una pentru substantiv cu lema *vin* si două pentru verb cu lema *veni*, prezent, persoana 1 singular respectiv persoana a 3-a plural)

b) prin eliminarea caracterelor diacritice din toate câmpurile <formă-ocurență> ale lexiconului intrările lexicale se pot modifica după cum urmează:

b1) anumite intrări rămân neschimbate: situația corespunde intrărilor pentru cuvintele de tip N.

b2) anumite intrări îşi schimbă câmpul <formă-ocurență> dar nu şi câmpul <formă-lemă : descriere morfo-sintactică>+: situația corespunde cuvintelor de tip D, pe care în (Tufiş & Chițu, 1999) le-am numit cuvinte cu diacritice neambigue (U-words). Aceste cuvinte conțin unul dau mai multe diacritice, dar prin eliminarea acestora nu se obține un cuvânt legal al limbii. Recuperarea acestora este independentă de context şi se poate face prin simpla inspecție a lexiconului. Exemple tipice sunt padure (*pădure*), tufis (*tufiş*), autorizatie (*autorizație*), cantar (*cântar*), carare (*cărare*), macar (*măcar*), fara (*fără*), cati (câți) etc.

b3) anumite intrări îşi schimbă atât câmpul <formă-ocurență> cât şi câmpul <formă-lemă : descriere morfo-sintactică>+. Aceste cuvinte, tot de tip D, pe care în (Tufiş & Chițu, 1999) le-am numit cuvinte cu diacritice ambigue (A-words) sunt cele mai problematice pentru că prezența sau absența diacriticelor afectează fie categoria gramaticală sau atributele morfologice, fie chiar semantica cuvântului. Exemple tipice sunt formele lemelor *fată, a făta, fâță* pentru care şirul ortografic *fata* este susceptibil a corespunde, în absența diacriticelor, la nu mai puțin de 11 interpretări diferite (substantivele fata/fată, față/față, fâța/fâță, verbele făta (infinitiv)/făta (imperfect)/fată (prezent pers.3 singular)/ /fată (prezent pers.3 plural)/fătă (perfect simplu).

În cazul majorității cuvintelor cu diacritice ambigue (A-words) informația morfo-lexicală dezambiguizează forma corectă a cuvântului. Totuşi, există o submulțime a acestora, pentru care descrierile morfo-sintactică sunt identice, diferența fiind dată doar de leme, iar dezambiguizarea poate fi făcută doar la nivelul de semnificație a cuvântului. In exemplul de mai sus, cele 11 interpretări posibile ale aceleaşi forme ocurență (fără diacritice) sunt descrise de 7 coduri morfo-sintactice distincte:

fata⇒ *fata, fâța, fața* (Ncfsry – substantiv comun, feminin, singular, caz direct, articol hotărât)

fata⇒ *fată, fâță, față* (Ncfsrn – substantiv comun, feminin, singular, caz direct, nearticulat)

fata⇒ *făta* (Vmn - verb principal, infinitiv)

fata ⇒ *fată* (Vmip3s - verb principal, indicativ, prezent, persoana 3 singular)

fata ⇒ *fată* (Vmip3s - verb principal, indicativ, prezent, persoana 3 plural)

fata ⇒ *făta* (Vmii3s - verb principal, indicativ, imperfect, persoana 3 singular)

fata ⇒ *fătă* (Vmis3s - verb principal, indicativ, perfect simplu, persoana 3 singular)

Se observă că dacă pentru cele 5 ocurențe verbale codificarea morfo-sintactică face distincție atât asupra interpretarii gramaticale şi semantice, în cazul celor 6 interpretări nominale, codificarea morfo-sintactică poate decide numai asupra literei a finale a cuvintelor, care poate fi *ă* (pentru codul Ncfsrn) sau *a* (pentru codul Ncfsry). Pentru grupul imbricat de litere *at,* recuperarea sa sub forma *âț*, *at* sau *aț* necesită identificarea sensului cuvântului țintă. Aceste cuvinte formează o subclasă a cuvintelor de tip A, şi în continuare le vom numi cuvinte de tip S (S-words).

O analiză a cuvintelor de tipul celei de mai sus necesită, aşa cum s-a arătat, un dicționar cu mare acoperire lexicală. În absența unei astfel de resurse, prin analiza unui corpus insuficient de mare, este foarte dificil a face distincția între cuvintele de tip N (pe care în (Tufiş & Chițu, 1999) le-am numit *diacritics-free words*, cuvinte libere de diacritice) şi cuvintele de tip D.

Din acest motiv, în lucrarea de față am optat pentru o clasificare a cuvintelor mult mai facilă, bazată pe corpus conținând texte în grafia corectă, respectiv cuvinte cu diacritice şi cuvinte fără diacritice[41]. În continuare vom folosi terminologia U-cuvinte, A-cuvinte şi S-cuvinte cu următoarele amendamente:

a) un U-cuvânt este un cuvânt legal al corpusului pentru care atunci când îi sunt eliminate unul sau mai multe semne diacritice se obțin "cuvinte" care nu există în corpus.

b) un A-cuvânt este un cuvânt legal al corpusului pentru care atunci când îi sunt eliminate unul sau mai multe semne diacritice se obțin "cuvinte" care există în corpus.

c) un S-cuvânt este un cuvânt legal al corpusului pentru care atunci când îi sunt eliminate unul sau mai multe semne diacritice se obțin "cuvinte" care există în corpus cu aceeaşi etichetă morfo-sintactică ca şi cuvântul original.

Tabelul 1 prezintă datele extrase din corpusuri constituite din texte din domenii diferite. Corpusul jurnalistic se compune din articolele revistei săptămânale „Agenda" din Timişoara (anii 2003-2006) iar corpusul juridic este format din colecția de documente româneşti (aproape 6000) a JRC-Acquis (Steinberger et al., 2006). Adnotarea etichetelor morfo-lexicale a fost făcută automat folosind o adnotare stratificată. Numărul total de cuvinte din tabelul 1 (linia 1) nu include punctuația, numele proprii, cuvintele care nu aparțin limbii române, abrevierile şi secvențele de caractere conținând numere. Din numărul total de atomi lexicali au fost înlăturați 36% şi, respectiv, 26%. Aceste categorii de atomi lexicali nu sunt semnificative pentru recuperarea diacriticelor deoarece, în marea majoritate a cazurilor, acestea nu conțin semne diacritice.

În Tabelul 1 sunt prezentate două numere diferite pentru S-cuvinte, depinzând de setul de etichete folosit pentru adnotarea morfo-lexicală: un set redus de etichete (Ctag-set în

[41] Cuvintele de tip N constituie marea majoritate (circa 75-80%) a cuvintelor fără diacritice.

linia 5) şi setul maximal de etichete morfo-lexicale (MSDtag-set în linia 6). Diferența dintre cele două numere demonstrează faptul că recuperarea diacriticelor are o acuratețe mai mare atunci când sistemul are acces la mai multe informații care dezambiguizează contextul morfo-sintactic.

Tabelul 1. Distribuția cuvintelor cu diacritice în texte din domenii diferite

Corpus	**Jurnalism**	**Juridic**
1. Cuvinte	6 680 448	3 511 093
1* Caractere	37 008 236	21 404 666
2. Cuvinte cu diacritice (din 1.)	2 004 763 (30,01%)	1 026 385 (29,23%)
2*. Caractere diacritice	2 351 220	1 192 875
3. U-cuvinte (din 1)	927 013 (13.88%)	430 862 (12.27%)
4. A-cuvinte (din 1)	1 766 631 (26,44%)	850 563 (24,22%)
5. S-cuvinte (Ctag-set, din 4)	58 420 (3,31%)	38 323 (4,51%)
6. S-cuvinte (MSDtag-set, din 4)	24 916 (1,41%)	16 463 (1,94%)

Aşa cum se poate observa din tabelul de mai sus, în limba română, cel puțin o treime din cuvinte (fără a lua în considerare atomii lexicali enumerați anterior) conțin semne diacritice (30.01% din cuvintele din corpusul jurnalistic cu o medie de 1.17 semne diacritice per cuvânt cu diacritice, iar în corpusul juridic 29.23% cu o medie de 1.16 caractere diacritice per cuvânt cu diacritice). Doar o mică parte din numărul total de cuvinte sunt U-cuvinte (13.88% în corpusul jurnalistic şi 12.27% în corpusul juridic).

3. DIAC+

Pentru a ameliora efectele negative date de existența insuficienței datelor indusă de folosirea unui tagset mare şi cea a insuficienței informației într-un tagset redus, am folosit metodologia adnotării stratificate (Tufiş, 1999, 2000). Aceasta este o tehnică în doi paşi care se referă la problema insuficienței datelor: (i) adnotare intermediară folosind un tagset redus (Ctag-set), (ii) înlocuirea Ctag-urilor cu descriptorii morfo-lexicali contextuali potriviți (MSD-uri, tagset-ul extins). Cea de-a doua etapă poate întâmpina anumite ambiguități care sunt rezolvate folosind resurse adiționale de cunoştințe. În (Tufiş, 1999), această nouă resursă este un set de reguli de dezambiguizare contextuală scris de mână. Recuperarea etichetelor MSD, atât cea deterministă cât şi cea bazată pe reguli, sunt aplicabile doar cuvintelor înregistrate în lexicon. Am înlocuit cea de-a doua etapă a procesului de adnotare stratificată printr-o recuperare a MSD-urilor bazată pe entropie (Ceauşu, 2006). În această abordare, regulile pentru conversia din Ctag-uri în MSD-uri sunt învățate în mod automat din corpus iar aplicarea lor nu cere căutarea în lexicon. Astfel, chiar şi Ctag-urile atribuite unor cuvinte necunoscute pot fi convertite în tag-uri MSD. Dacă un lexicon cu adnotări MSD este disponibil, înlocuirea Ctag-urilor pentru cuvinte cunoscute se face cu acuratețe de aproximativ 100%. Pentru etichetarea morfo-sintactică a textelor fără diacritice, din modelul de limbă HMM standard pentru limba română este păstrată matricea de tranziție, iar lexiconul probabilist (probabilitățile de emisie) sunt recalculate

pe corpusul de antrenare standard, din care sunt eliminate diacriticele. În acest fel, pentru cuvintele din lexiconul modelului de limbă clasele de ambiguitate se pot modifica (de pildă cuvântul "peşte" care este numai substantiv, va avea în noul dicţionar probabilistic şi interpretarea corespunzătoare prepoziţiei "peste"). Restricţiile gramaticale surprinse de matricea originală de tranziţie vor permite în marea majoritate a cazurilor rezolvarea acestor ambiguităţi lexicale artificial introduse prin eliminarea diacriticelor. Desigur, acurateţea dezambiguizării morfo-sintactice a textelor fără diacritice scade în comparaţie cu cea a textelor normalizate, dar diferenţa nu este foarte mare (a se vedea mai jos)

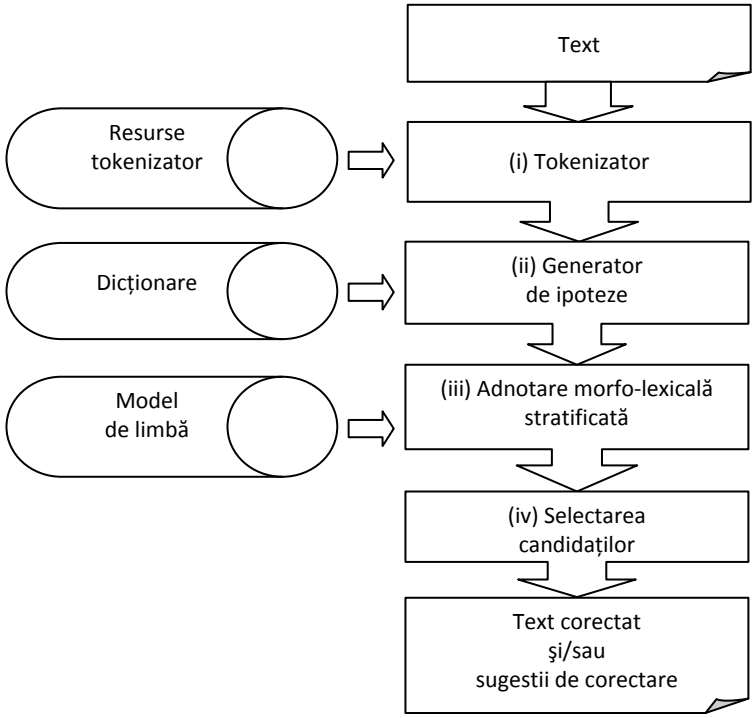


Figura 2. Arhitectura generală a sistemului DIAC+

În sistemul conceput de noi, procesul de inserare a diacriticelor are patru paşi: (i) textul de intrare este segmentat în unităţi lexicale potrivit regulilor specificate ca resurse externe; (ii) fiecare cuvânt este căutat în dicţionar după forma sa de suprafaţă fără diacritice, iar rezultatul acestei căutări este o listă a formelor corecte şi a posibilelor etichete morfo-lexicale pe care le poate avea cuvântul; (ii) textul segmentat este supus procesului de adnotate stratificată; (iv) formele neambigue sunt înlocuite cu formele cu diacritice corespunzătoare, în timp ce pentru cazurile de ambiguitate sunt preferate formele cele mai frecvente; în acest caz, se generează în mod automat un jurnal al procesului, iar utilizatorul poate reface o înlocuire şi selecta o altă variantă. Cuvintele care nu se găsesc în lexicon rămân neprelucrate, dar sunt listate în jurnal pentru a fi inspectate ulterior de către utilizator. Cuvintele necunoscute sunt de asemenea colectate pentru validare şi, dacă sunt corect lematizate şi adnotate, sunt introduse în lexiconul de forme flexionare al aplicaţiei.

Erorile introduse de procesul de adnotare morfo-lexicală (aproximativ 3%) au scăzut acurateţea în comparaţie cu un scenariu ideal (în care adnotarea este perfectă) cu o medie de 1.3%. Merită să menţionăm că rata erorii în recuperarea diacriticelor a fost cu

mult mai mică decât cea din faza intermediară de adnotare, iar explicația acestui fapt este că anumite erori ale tagger-ului nu sunt relevante pentru recuperarea diacriticelor. De exemplu, eroarea de adnotare destul de frecventă care apare datorită confuziei dintre participii, adjective şi uneori substantive, nu afectează restaurarea diacriticelor dacă genul şi articolul hotărât sunt corect identificate de tagger.

O aplicație tradițională de corectare a ortografiei evidențiază cuvintele care nu se regăsesc în dicționar, utilizatorul trebuind să opteze pentru una din formele corecte sugerate de aplicație. În cazul DIAC+ corectarea formelor greşite se face în majoritatea cazurilor automat. În cazul particular al C-cuvintelor, DIAC+ se comportă ca un corector ortografic tradițional sugerând utilizatorului variante de corectare. Aceste variante sunt conforme restricțiilor lingvistice impuse de contextul morfo-sintactic de ocurență a cuvântului. De exemplu, daca C-cuvântul „fata" a fost adnotat ca substantiv feminin articulat cu articolul hotărât, într-unul din cazurile directe, soluțiile de corectare sunt „fata", „fața", „fâța" toate caracterizate de aceleaşi atribute morfo-lexicale. Celelalte variante („fată", „față", „fâță", „făta", „fată") sunt ignorate datorită diferențelor de atribute morfo-lexicale (aceste variante fiind fie substantive în formă nearticulată, fie verbe).

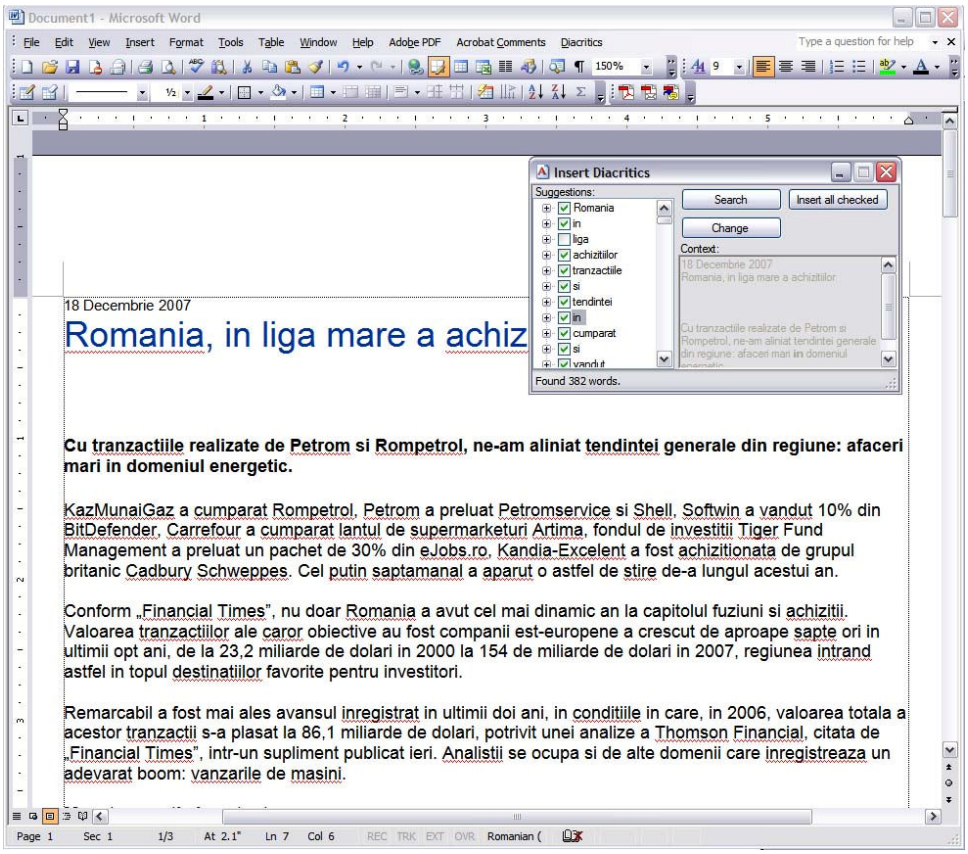


Figura 3. Recuperarea diacriticelor în Microsoft Word

Versiunea curentă a acestui sistem este complet operațională şi s-a dovedit extrem de utilă dar poate fi îmbunătățită în diverse moduri. Cum rata de succes depinde foarte mult de acoperirea lexiconului, cea mai semnificativă îmbunătățire este extinderea automatică a acestuia. Un astfel de instrument este în dezvoltare folosind generatorul morfologic paradigmatic pentru limba română, ROG, şi rezultatele adnotării şi lematizării a numeroase texte în limba română, conținând mai mult de 2 G cuvinte. Toate cuvintele necunoscute sunt clasificate în funcție de lema lor. Atunci când două sau mai multe cuvinte sunt clasificate sub aceeaşi lemă, prin analiza sufixelor acestora se poate identifica, în cele mai multe dintre cazuri în mod unic, paradigma tuturor cuvintelor din acea clasă. În celelalte cazuri, doar câteva clase de paradigme sunt supuse analizei. Generatorul morfologic este capabil să genereze toate variantele flexionare ale unei leme de intrare, dacă se cunoaşte paradigma căreia aceasta îi aparține. Formele generate sunt căutate pe web (în site-urile româneşti care folosesc diacritice) şi dacă toți membrii unei presupuse familii paradigmatice sunt identificați, întreaga familie poate fi stocată automat în lexiconul de forme. Acest proces depinde de acuratețea cu care sunt adnotate cuvintele necunoscute. Nu toate trăsăturile atribut-valoare din structura unei etichete sunt relevante în acest proces şi de aceea nu toate erorile de adnotare generează actualizări automate ale lexiconului.

Când creşterea lexiconului depăşeşte un anumit nivel (acum de 2%, ceea ce reprezintă aproximativ 25,000 forme noi) modelul de limbă al motorului de adnotare şi lematizare, care stă la baza procesului de recuperare a diacriticelor, este reconstruit incluzând noile cuvinte. Versiunea actuală a programului DIAC^{+} verifică platforma de servicii web a ICIA iar dacă există vreo versiune actualizată, utilizatorul este informat şi o poate descărca.

4. Evaluare

Pentru evaluare am folosit un corpus de referință – R – conținând aproximativ 100'000 de atomi lexicali. Corpusul de referință a fost adnotat şi lematizat manual. Au fost înlăturate toate caracterele diacritice dar a fost păstrată adnotarea originală. Această versiune a corpusului de referință o vom numi corpus de referință ideal (RA): nu are erori de adnotare morfo-lexicală. Procesarea cu DIAC^{+} a acestui text ne va da limita superioară a performanței sistemului, performanță neafectată de erorile de adnotare automată. Pentru o evaluare a performanțelor reale am înlăturat din RA şi descrierile morfo-lexicale. Acest text fără diacritice (RN) a fost prelucrat folosind toate etapele de procesare ale DIAC^{+}: adnotare cu set restrâns de etichete, conversia setului restrâns de etichete la setul maximal de etichete morfo-lexicale, recuperarea diacriticelor chiar şi în cazul C-cuvintelor.

Rezultatele acestor experimente sunt sintetizate în Tabelul 2. Față de statistica din Tabelul 1, din statistica din Tabelul 2 nu lipsesc atomi lexicali ca punctuația, numele proprii, abrevierile etc. Diferența de 1.28% dintre formele corecte dintre cele două experimente se datorează în întregime erorilor adnotării automate, deşi doar o mică parte dintre erorile de adnotare se regăsesc şi în erorile de recuperare a diacriticelor. Sistemul de referință (baseline) din ultima coloană este bazat pe un dicționar de cuvinte construit din corpusul Agenda (de aproximativ 10 000 000 de atomi lexicali) prezentat în statistica din secțiunea 2. Dicționarul este indexat după forma fără diacritice a

cuvintelor. Sistemul de referinţă înlocuieşte în corpusul de evaluare forma fără diacritice a cuvântului cu cea mai frecventă formă de suprafaţă găsită în acest dicţionar.

S-cuvintele (361) se constituie într-o parte semnificativă a erorilor de recuperare a diacriticelor. Restul erorilor de recuperare de diacritice se datorează erorilor de adnotare.

Tabelul 2. Evaluarea pe atomi lexicali a DIAC+

	Text adnotat (RA)	**Text neadnotat (RN)**	**Sistem de referinţă**
Atomi lexicali	117 909	117 909	117 909
Cuvinte cu diacritice	34 745 (29,47%)	34 745 (29,47%)	34 745 (29,47%)
Forme corecte	116 810 (99,06%)	115 262 (97,75%)	113 491 (96,25%)
Forme incorecte	1 092 (0,94%)	2 609 (2,21%)	4 418 (3,75%)
S-cuvinte	361	361	361

În cazul S-cuvintelor, în modul interactiv de funcţionare, aplicaţia de recuperare de diacritice oferă utilizatorului sugestii de corectare, alegerea formei corecte fiind făcută manual. În modul "batch", foarte util prelucrării volumelor mari de texte, S-cuvintele sunt corectate în baza frecvenţei statistice, dar toate modificările de acest tip sunt înregistrate într-un jurnal, dând posibilitatea verificării individuale, la un moment ulterior.

Tabelul 3. Evaluarea pe caractere a DIAC+

	Text adnotat (RA)	**Text neadnotat (RN)**	**Sistem de referinţă**
Caractere (fără spaţii)	501 735	501 735	501 735
Caractere diacritice	41 144	41 144	41 144
Caractere corecte (fără spaţii)	500 400 (99.73%)	498 764 (99.40%)	497 096 (99,07%)
Caractere incorecte (fără spaţii)	1 335 (0,27%)	2 971 (0,6%)	4 639 (0,93%)

Se observă că evaluarea la nivelul caracterelor indică performanţe de peste 99% chiar la nivelul sistemului de referinţă (baseline). Ceea ce se remarcă este faptul că o diferenţă de acurateţe de 0,33% în evaluarea pe caractere se reflectă într-o diferenţă de acurateţe de 1,5% atunci când evaluarea este efectuată la nivelul cuvântului.

5. Concluzii

În comparaţie cu versiunile precedente, (Tufiş & Chiţu, 1999), implementarea DIAC+ actuală include un corector ortografic şi are o acurateţe mult mai bună datorită îmbunătăţirilor aduse modelului de limbă precum şi datorită performanţelor crescute ale adnotării stratificate.

Sistemul DIAC^{+} este disponibil în două implementări: un serviciu web bazat pe licenţă şi o variantă off-line pentru recuperarea locală a diacriticelor în cazul documentelor secrete, pe care autorul nu doreşte să le transmită prin internet. Versiunea off-line este implementată ca un DLL pentru suita Microsoft Office şi este un bun complement pentru corectorul ortografic Microsoft, dar nu este la fel de rapid şi lucrează cu documente de dimensiuni mai mici decât versiunea web.

Referinţe bibliografice

Bèze, M., Mérialdo, B., Rozeron, B., Serouault, A., M. (1994): Accentuation automatique de texte par des méthodes probabilistes. *Technique et sciances informatiques*, 13(6):797-815

Ceauşu, Al. (2006): Maximum Entropy Tiered Tagging, Janneke Huitink & Sophia Katrenko (editors), *Proceedings of the Eleventh ESSLLI Student Session*, ESSLLI 2006, pp. 173-179

Mihalcea, R., Năstase, V. A. (2001): An automatic method for diacritics insertion into Romanian texts (O metodă automată pentru inserarea diacriticelor în texte în limba română), Tufiş, D., Filip, Gh. (coord.) Limba română în Societatea Informaţională, Expert, Bucharest, pp. 191-205

Simard, M. (1998): Automatic Insertion of Accents in French Texts. In *Ide & Vuotilainen (eds) Proceedings of the Third Conference on Empirical Methods in Natural Language Processing ,* Granada, Spain, 27-35

Tufiş, D., Chiţu, A. (1999): Automatic Insertion of Diacritics in Romanian Texts. In *Proceedings of the 5th International Workshop on Computational Lexicography COMPLEX*, Pecs, Ungaria, 1999, pp. 185-194

Tufiş, Dan (1999). Tiered Tagging and Combined Classifiers. In F. Jelinek, E. Nth (eds) *Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence*, Springer, pp. 28-33

Tufiş, D. (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging, *International Conference on Language Resources and Evaluation LREC'2000*, Athens, 2000, pp. 1105-1112

Yarowsky, D. (1994): A Comparison of Corpus-based for Restoring Accents in Spanish and French Texts. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, Kyoto, Japan

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC'2006). Genoa, Italy, 24-26 May 2006

CĂUTAREA INFORMAȚIEI PE RESURSE LINGVISTICE TEXTUALE CU FILTRU DE RELEVANȚĂ FUZZY

SILVIU IONIȚĂ

Universitatea din Piteşti, Facultatea de Electronică, Comunicaţii şi Calculatoare, Piteşti - România

ionis@upit.ro

Rezumat

Lucrarea tratează problema clasificării informației accesibilă pe arhive electronice de documente, inclusiv pe Internet, utilizând drept funcție de filtrare, relevanța ca măsură a informației construită după un model fuzzy.

1. Introducere

Accesul în timp util la resurse informaționale relevante reprezintă un aspect crucial într-o lume în curs de globalizare, cu o economie tot mai dependentă de servicii electronice. Cea mai mare cantitate de informație se regăseşte încă sub formă textuală în arhive distribuite pe milioane de servere din întreaga lume (Ulieru & Ionita, 2001). Problematica accesării, regăsirii şi extragerii informației din arhive multilingve de documente stocate în format electronic, depinde în mare măsură de factorul uman. Acesta intervine atât în faza preliminară, de preprocesare a fondului documentar, realizând clasificarea documentelor într-un sistem standardizat, cât şi ulterior în procesul de căutare a informației, în ipostaza de beneficiar al resurselor informaționale. În ambele sensuri menționate, omul efectuează un proces de clasificare în raport cu anumite criterii convenționale, mai mult sau mai puțin subiective. În procesul de clasificare a informației se efectuează un ansamblu de operații consacrate sub denumirea de *filtrare*. Problematica clasificării este complexă sub mai multe aspecte. În primul rând, clasificarea unor obiecte se face într-un spațiu de *descriptori* definit printr-un proces de indexare preliminară, convențională a obiectelor. Acest lucru conduce inevitabil la o insuficiență de modelare, deci la incertitudine. În al doilea rând, pentru teoriile de clasificare un rol esențial îl are *informația subiectivă*, care este legată de ceea ce numim *relevanță*. Dincolo de criteriile cantitative furnizate de modelul probabilist, bazate pe măsuri energetice (Onicescu, Ştefănescu, 1979), (Nicolescu, Stoka, 1971), relevanța constituie în opinia noastră cea mai obiectivă măsură pentru conceptul de informație. Astfel, problematica căutării şi clasificării arhivelor lingvistice textuale în funcție de interesul utilizatorului constă în construirea unui model plauzibil pentru relevanța informației.

Lucrarea de față îşi propune să prezinte unele rezultate obținute pe baza unui model de relevanță bazat pe concepte fuzzy.

1. Modelul relevanței

1.1. O interpretare conceptuală intuitivă

În accepțiunea noastră (Ioniță, 2004), interpretarea grafică a relevanței ca măsură a utilității informației aplicată în procesul de căutare pe resurse documentare se prezintă ca în Figura 1. Modelul prezentat arată comparativ cazul ideal, în care funcția de decizie pentru clasificare prezintă un punct clar de selecție şi cazul real, în care decizia se ia gradual pe nivele de relevanță. Justificarea modelului propus se bazează intuitiv pe distribuția statistică normală, potrivit căreia într-o arhivă de resurse documentare disponibile pe o anumită tematică, utilizatorul (subiectul uman) va clasifica relevanța informațională a acestora după o curbă aproximativ gaussiană. Cu alte cuvinte, doar un număr redus de resurse vor satisface criteriul de relevanță al unui utilizator inevitabil subiectiv. Forma curbei ce descrie clasificarea documentelor în raport cu relevanța lor poate fi discutată, astfel că apare justificată abordarea propusă de aplicare a conceptelor fuzzy pentru elaborarea unui filtru de relevanță destinat clasificării resurselor documentare.

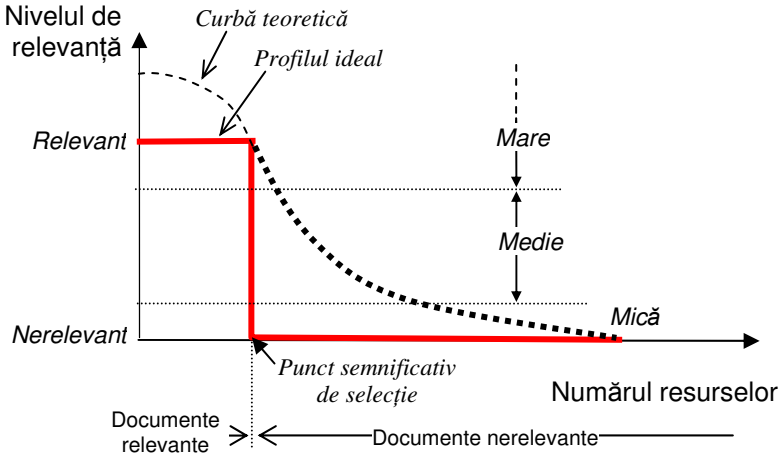


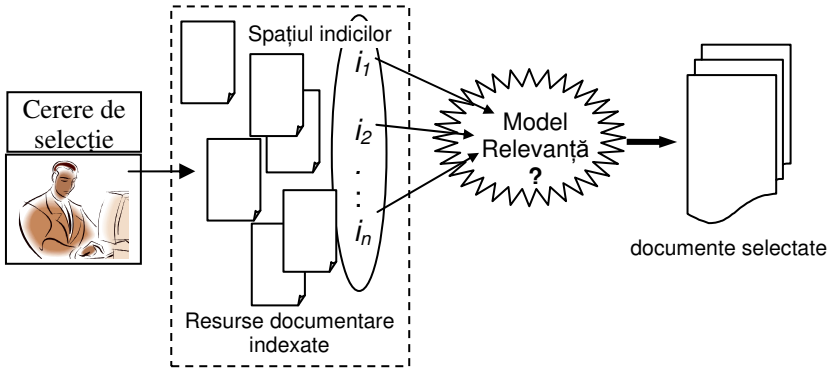
Figura 1: Modelul relevanței pe un set de resurse date.

1.2. Exprimări analitice

În principiu, problema căutării automate a informației în resurse lingvistice multilingve arhivate electronic este sarcina motoarelor de căutare. Acestea funcționează pe principiul indexării documentelor electronice distribuite în rețeaua globală. Selectarea şi ierarhizarea documentelor se face pe baza unui filtru de căutare în concordanță cu un sistem de indici descriptori, aşa cum este ilustrat sugestiv în Figura 2. Clasificarea resurselor documentare depinde de modelul filtrului de selecție, adică de o funcție de relevanță, în raport cu care se face ierarhizarea documentelor (ranking), potrivit cererii de căutare a utilizatorului. O tratare sistematică a problematicii motoarelor de căutare este accesibilă la resursa web (www.searchtools.com), unde se afirmă că relevanța este o măsură pentru cât de bine răspunde un document indexat la cererea utilizatorului, şi numai acesta poate să o definească. Decizia cu privire la relevanța informațională a unui document pe baza unui singur criteriu (indice), de exemplu cuantificarea numărului de

potriviri stricte a cuvântului căutat, poate fi considerată de încredere cu un anumit grad, destul de redus. Scopul nostru este de a regăsi şi extrage resursele informaționale cu un grad de încredere cât mai mare, ceea ce implică luarea în considerare a unui număr mai mare de criterii pe bază de indici suplimentari. Problema modelării relevanței rezidă în găsirea unei dependențe funcționale pertinente privind sistemul de indici aleşi ca descriptori.

Figura 2: Selecția pe bază de indici de descriere



Vom defini relevanța ca pe o funcție în spațiul descriptorilor exprimând formal o dependență de un sistem convențional de indici $i_1, i_2, ... i_n$, prin relația următoare:

$$relevanta \quad = r(i_1, i_2, ..., i_n) \tag{1}$$

În general, spațiul descriptorilor pentru resurse text cuprinde un sistem de indici folosiți pentru cuantificarea relevanței documentelor în procesul de căutare aşa cum se prezintă în Tabelul 1. Fiecare indice constituie practic un criteriu de apreciere a relevanței, pe baza căruia se elaborează o funcție de căutare. Astfel de indici sunt specifici motoarelor actuale de regăsire a informației pe Internet, valoarea acestora constituind măsuri ale relevanței parțiale pe baza cărora se realizează ordonarea documentelor electronice în urma căutării pe web.

Există relații analitice pentru evaluarea statistică a documentelor în raport cu criteriile menționate mai sus, care se referă la frecvența de apariție, astfel :

$$i_k = \left(\frac{n_k}{n_{total}} \cdot 100 \right), [\%] \tag{2}$$

unde n_k , $k = 1,3,5,6$ este numărul de potriviri exacte ale termenului în textul analizat şi n_{total} este numărul unităților/entităților lingvistice în document ce pot fi propoziții, linii (sau rânduri de text), paragrafe, cuvinte. În mod uzual, când se efectuează statistica unui text entitatea lingvistică luată în considerare este cuvântul. Pentru căutarea în texte cu dimensiuni mari, ori în cazul identificării colocațiilor (Ştefănescu et al, 2006) pot fi folosite entități lingvistice ce includ mai multe cuvinte.

Tabel 1 : Indici pentru clasificarea documentelor

k	Index i_k	Semnificația
1	Frecvența de apariție a termenului	Dacă termenul apare de mai multe ori se presupune că acesta conține informația utilă
2	Poziția termenului în raport cu începutul documentului	Dacă cuvântul apare mai devreme în text (în antet, de exemplu) atunci probabilitatea ca documentul să fie cel căutat este mai mare.
3	Frecvența apariției legăturilor (links)	Dacă şi alți (potențiali) utilizatori consideră documentul important, atunci este mult mai probabil ca el să fie relevant.
4	Distribuția termenilor în conținutul documentului	Dacă distribuția cuvântului cheie este uniformă atunci relevanța ar putea fi mai mare.
5	Frecvența diferitelor forme lexicale ale termenului	Dacă termenul apare în forme gramaticale variate atunci documentul este într-o măsură mai mare focalizat pe subiect.
6	Frecvența sinonimelor	Dacă sinonimele termenului cheie apar şi ele de mai multe ori, se presupune că documentul conține cu atât mai mult informație utilă.
7	Distanța (în cuvinte) dintre aparițiile consecutive ale termenului	Dacă distanțele sunt scurte, atunci documentul este mai relevant din punctul de vedere al subiectului căutat.

Pentru criteriile ce nu se referă la frecvența de apariție se folosesc relații consacrate ale indicatorilor statistici de bază. Astfel, poziția termenului căutat în text se determină pe baza distanței între aparițiile succesive ale termenului respectiv d_j , cu ajutorul cărora se estimează distanța medie la nivelul întregului document:

$$d_{mean} = \frac{\sum_{j=1}^{m} d_j}{m-1} \quad [\hat{i}n\ cuvinte] \tag{3}$$

Numărul de apariții ale termenului (m) poate fi considerat ca fiind suma dintre numărul de apariții ale termenului exact (n_{term}) şi numărul termenilor în forme diferite (n_{dtf}). În acest caz indicele $i_7 = d_{mean}$. Indicele nr. 4 indică împrăştierea termenilor potriviți. Ca măsură statistică pentru distribuția cuvintelor se utilizează deviația standard - σ, deci indicele $i_4 = \sigma$. Aceasta poate fi calculată relativ la distanțele dintre termenii potriviți, ca variabile aleatoare pe întregul document:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m} (d_j - d_{mean})^2}{m-1}} \tag{4}$$

Deviația standard a distanțelor dintre potrivirile de termen reflectă gradul de uniformitate al prezenței termenului pe întregul document analizat: dacă distanțele d_j sunt apropiate de valoarea lor medie d_{mean} atunci termenul este distribuit uniform, iar aceasta indică o relevanță mai bună a textului cu privire la subiectul sugerat prin termenul respectiv.

2. *Filtrul de relevanță fuzzy*

Motivația introducerii unui filtru fuzzy pentru căutarea informației este dată de următoarele aserțiuni: (a) informația are diferite grade de ambiguitate, (b) criteriile de căutare sunt relative, (c) modelul relevanței este incert. Modelul fuzzy propus pentru filtrarea documentelor în funcție de relevanță foloseşte ponderarea sistemului de indecşi asociați fiecărui document ca variabile fuzzy de intrare, cărora li se atribuie mulțimi fuzzy cu anumite etichete lingvistice asociate. De exemplu, în Figura 3a am definit trei mulțimi fuzzy, descrise cu funcții de apartenență liniare şi adnotate cu termenii ligvistici: *slab*, *mediu*, *bun*. Ca ieşire, modelul are drept variabilă fuzzy chiar relevanța prin credibilitatea (gradul de încredere al) acesteia, definită pe un univers de discurs notat de exemplu convențional de la 1 la 10, aşa cum se arată în Figura 3b. Pe universul

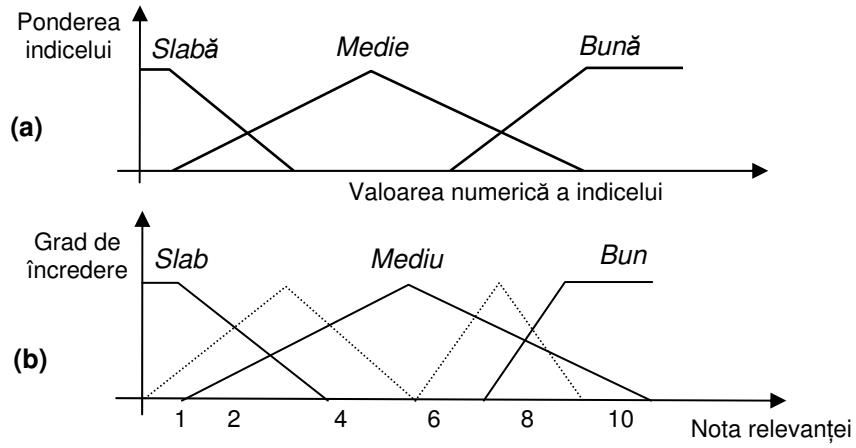


Figura 3: Fuzificarea variabilelor, a-indici, b-relevanță

de discurs convențional se aleg intuitiv un număr de mulțimi fuzzy etichetate cu termeni lingvistici: *slab*, *mediu*, *bun,* etc.. În general, funcția de relevanță în sine este o combinație neliniară a indicilor ce intervin cu ponderi diferite, astfel că modelul fuzzy va juca rolul de aproximator universal pentru o dependență greu de exprimat analitic. Fuzzyficarea valorilor variabile ale indicilor se face în concordanță cu funcțiile de apartenență din Figura 3a şi relațiile (5), unde N', N'', N''' sunt submulțimi pe **N**. În acelaşi mod se tratează şi variabilele de tip relevanță.

$$\begin{aligned} &m_{slab}(i_k) : N' \to [0,1], \\ &m_{mediu}(i_k) : N'' \to [0,1], \\ &m_{bun}(i_k) : N''' \to [0,1]. \end{aligned} \tag{5}$$

Un rol esențial în construirea funcției de relevanță pe baza logicii fuzzy îl are constituirea regulilor de inferență pentru combinarea indicilor. Regulile fuzzy reprezintă cunoştințe intuitive, în care premisele capătă valori fuzzy potrivit termenilor lingvistici stabiliți pentru fiecare indice, iar consecința conține o singură concluzie, ce furnizează valoarea încrederii relevanței pe domeniul de discurs convențional ales. Stabilirea regulilor fuzzy comportă unele discuții referitoare la principiul obținerii regulilor pe de o parte, şi la implementarea efectivă a acestora, pe de altă parte. În aplicația de față obținerea regulilor s-a făcut în mod manual pe baza de cunoştințe apriorice.

Implementarea regulilor, depinde de alegerea unor strategii de asociere a indicilor de relevanță cu configurarea unor structuri de reguli înlănțuite pe mai multe nivele. Abordarea euristică predomină atât în stabilirea structurii regulilor în sine, cât şi a structurii sistemului de reguli. Structura generală a unei reguli ce agregă toți indicii pe acelaşi nivel este de forma:

DACĂ (i_1 =<*termen lingvistic*> ŞI i_2 =<*termen lingvistic*> ŞI . . .
. . . ŞI i_7 =<*termen lingvistic*>) ATUNCI (r =<*termen lingvistic*>)

Aceasta presupune însă o structură liniară, pe un singur nivel ierarhic de combinare a criteriilor de relevanță, care pentru toți cei şapte indici de relevanță menționați aici are aspectul din Figura 4. Procedeul este însă ineficient, deoarece conduce la creşterea rapidă a numărului de reguli odată cu numărul de termeni lingvistici atribuiți variabilelor fuzzy. De exemplu, considerând doar câte trei termeni lingvistici pentru fiecare variabilă, din cele şapte de intrare se obțin $N_R=3^7=2187$ reguli, deci o bază de reguli fuzzy (BRF) inacceptabil de mare din punct de vedere practic. O astfel de bază de reguli multidimensională este greu sau aproape imposibil de implementat şi la fel de greu se lucrează cu ea. Pentru a evita problemele legate de dimensiunea BRF se utilizează noțiunea de *relevanță parțială* şi se adoptă unele scheme arborescente de combinare a indicilor de relevanță aşa cum se arată în Figura 5. Intuitiv, se pot combina câte două sau trei variabile pe nivel, rezultând diferite structuri de agregare ierarhizate, care se pot descrie cu ajutorul unui număr rezonabil de reguli. În situații particulare schemele de compunere se stabilesc fără a mai lua în considerare toți indicii de relevanță. Pentru aplicația de față s-a decis renunțarea la indicele i_3 (privitor la frecvența legăturilor/referințelor în document), adoptându-se de pildă, schema de agregare din Figura 6. Schema propusă se bazează pe patru nivele de relevanță parțiale descrise de următoarele relații :

$$\begin{aligned} r_1 &= r_1(i_1, i_4) \\ r_2 &= r_2(i_2, i_7) \\ r_3 &= r_3(i_5, i_6) \\ r_4 &= r_4(r_2, r_3) \end{aligned} \tag{6}$$

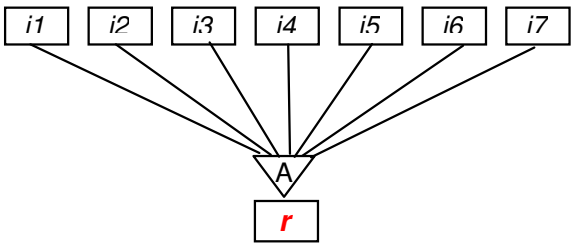


Figura 4: Combinarea (agregarea) liniară a indicilor de relevanță

În final, relevanța este dată de relația funcțională următoare:

$$r = r(r_1, r_4) \tag{7}$$

Pe baza acestor relații, descrierea prin reguli se reduce la cinci baze de reguli fuzzy parțiale, bidimensionale (vezi Tabelele 2 - 6), ceea ce reduce considerabil numărul de reguli necesare, în cazul de față la numai 45. Etapele de implementare a sistemului de inferență fuzzy respectă metodologia tipică pentru sistemele fuzzy de tip Mamdani.

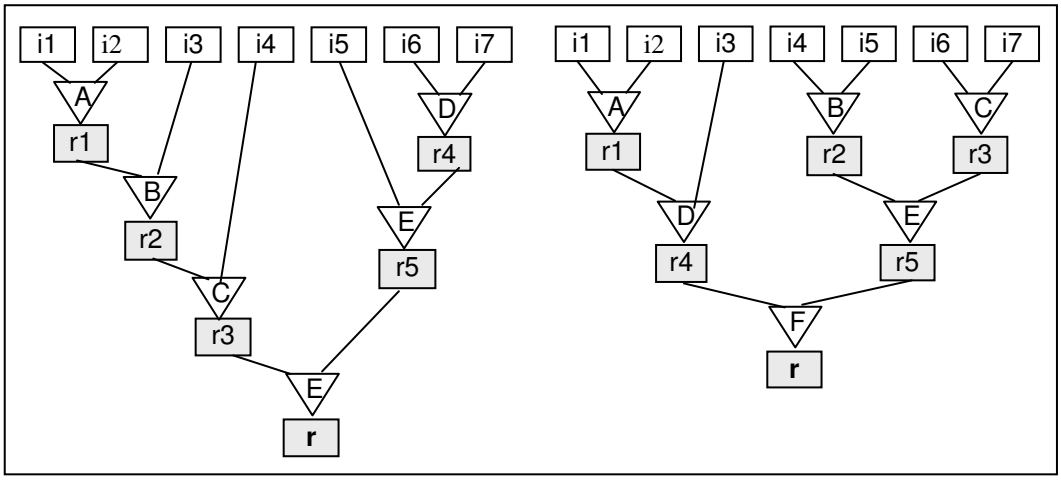


Figura 5 : Exemple de scheme arborescente pentru compunerea regulilor

3. Rezultate experimentale

Modelul propus a fost testat pentru un set de zece documente sub forma unor resurse lingvistice cu conținut preponderent text. Acestea au fost preselectate din primele două pagini returnate de motorul de căutare Google [www.google.com] la o cerere de căutare pentru cuvântul cheie compus *"data mining"*. Ca posibile sinonime s-au luat în considerare *"data discovery"* şi *"data retrieve"*, iar termenii cu diferite forme flexionare au fost detectați ca orice şir de caractere ce conține *"data mining"*.

Tabel 2: reguli fuzzy pentru relevanța parțială *r1(i1,i4)*

		Indice i_1		
		Slabă	Medie	Bună
Indice i_4	Slabă	*Slabă*	*Medie*	*Bună*
	Medie	*Slabă*	*Bună*	*Bună*
	Bună	*Medie*	*Bună*	*Bună*

Tabel 3: reguli fuzzy pentru relevanța parțială *r2(i2,i7)*

		Indice i_2		
		Slabă	Medie	Bună
Indice i_7	Slabă	*Bună*	*Medie*	*Slabă*
	Medie	*Medie*	*Slabă*	*Slabă*
	Bună	*Slabă*	*Slabă*	*Slabă*

Tabel 4: reguli fuzzy pentru relevanța parțială *r3(i5,i6)*

		Indice i_5		
		Slabă	Medie	Bună
Indice i_6	Slabă	*Slabă*	*Medie*	*Medie*
	Medie	*Slabă*	*Medie*	*Bună*
	Bună	*Medie*	*Medie*	*Bună*

Tabel 5: reguli fuzzy pentru relevanța parțială *r4(r2,r3)*

		Relevanță parțială *r2*		
		Slabă	Medie	Bună
Relevanță parțială *r3*	Slabă	*Slabă*	*Medie*	*Bună*
	Medie	*Slabă*	*Medie*	*Bună*
	Bună	*Medie*	*Bună*	*Bună*

Tabel 6: reguli fuzzy pentru relevanța finală *r(r1,r4)*

		Relevanța parțială *r1*		
		Slabă	Medie	Bună
Relevanță parțială *r4*	Slabă	*Slabă*	*Medie*	*Bună*
	Medie	*Medie*	*Bună*	*Bună*
	Bună	*Medie*	*Bună*	*Bună*

Documentele au fost prelucrate manual cu ajutorul instrumentelor pentru statistica textului în MSWord, rezultatele fiind date în Tabelul 7. Pe baza relațiilor (2) - (4), în care numărul total al entităților lingvistice n_{total} s-a considerat a fi numărul de propoziții din fiecare document în parte s-au calculat valorile indicilor din Tabelul 8. În urma aplicării filtrului de clasificare bazat pe modelului fuzzy descris mai sus s-a obținut o clasificare a documentelor pe baza relevanței în raport cu termenul cheie impus (vezi Tabelul 9). Teoretic, pentru cazul considerat există maxim 10 clase de relevanță posibile, dar se observă că sistemul propus a găsit 8 clase, notate de la C1 la C8, ca urmare a situațiilor de egalitate a valorii criteriului de clasificare (adică a relevanței). Practic, utilizatorul poate defini numărul dorit de clase de relevanță, prin acceptarea unor intervale valorice convenabile ale acesteia. Experimentul efectuat a avut rolul de a evalua funcționarea unui filtru de relevanță pe baza unui model fuzzy. Credibilitatea modelului propus este verificată în prima fază prin raportare la rezultatul obținut cu motorul de căutare al companiei Google.

Tabel 7: Statistica documentelor analizate

Nr. document	n_{total}	n_{term}	n_{syn}	n_{dtf}	d_{med}
1	*34*	*4*	*0*	*4*	*35.8*
2	*494*	*64*	*0*	*5*	*73.1*
3	*44*	*3*	*0*	*0*	*73.5*
4	*122*	*18*	*0*	*0*	*88.4*
5	*45*	*11*	*0*	*0*	*21.2*
6	*260*	*76*	*0*	*1*	*32.2*
7	*20*	*5*	*0*	*5*	*26.5*
8	*240*	*27*	*0*	*3*	*75.2*
9	*237*	*48*	*0*	*9*	*59.7*
10	*433*	*41*	*0*	*1*	*110.4*

Tabel 8: Valorile calculate ale indicilor de relevanță

Nr. document	i_1	i_2	i_3	i_4	i_5	i_6	i_7
1	*11.7*	*1*	-	*31.4*	*11.7*	*0*	*35.8*
2	*12.9*	*4*	-	*100.8*	*1.0*	*0*	*73.1*
3	*6.8*	*3*	-	*74.4*	*0.0*	*0*	*73.5*
4	*14.7*	*4*	-	*84.1*	*0.0*	*0*	*88.4*
5	*24.4*	*2*	-	*15.5*	*0.0*	*0*	*21.2*
6	*29.2*	*14*	-	*26.2*	*0.4*	*0*	*32.2*
7	*25.0*	*48*	-	*15.4*	*25.0*	*0*	*26.5*
8	*11.2*	*5*	-	*83.9*	*1.3*	*0*	*75.2*
9	*20.2*	*0*	-	*57.4*	*3.8*	*0*	*59.7*
10	*9.4*	*8*	-	*162.4*	*0.2*	*0*	*110.4*

Ordonarea celor zece documente în Tab.7-9 este potrivit ierarhizării returnată de Google, în sensul descreşterii relevanței. În schimb, ierarhizarea documentelor cu filtrul fuzzy propus se face după nota de relevanță calculată (Tab. 9) în ordine descrescătoare a valorilor, aşa cum se arată în Tabelul 10.

Tabel 9: Note/grade grade de relevanță calculate şi clase distincte

Nr. document	Nota relevanței	Clasa
1	5.490	C1
2	5.490	
3	5.294	C2
4	5.569	C3
5	5.922	C4
6	5.922	
7	5.608	C5
8	5.451	C6
9	5.804	C7
10	6.039	C8

Tabel 10: Rezultatul clasificării după relevanță

Ordinea de clasificare		Nota relevanței calculată cu modelul fuzzy propus
Google	fuzzy	
10	1	6.039
5	2	5.922
6	3	5.922
8	4	5.804
7	5	5.608
4	6	5.569
1	7	5.490
2	8	5.490
8	9	5.451
3	10	5.294

4. Concluzii

Relevanța, ca trăsătură a informației semnificative este o măsură subiectivă elaborată de intelectul uman, imposibil de modelat cu instrumente exclusiv analitice. Abordarea raționamentului aproximativ din perspectiva sistemelor fuzzy implementează consistența umană în procesul de extragere a informației. Modelul fuzzy propus pentru calculul relevanței oferă o soluție neanalitică de agregare ponderată a indicilor ce descriu relevanța unui document.

Referitor la rezultatele experimentale, deşi în lucrare se prezintă un singur test, se observă că ierarhizarea obținută cu soluția propusă este semnificativ diferită de cea oferită de motorul Google, ceea ce denotă clar principiul diferit de evaluare a relevanței documentelor. Aceast fapt este explicabil, deoarece Google aplică principii comerciale, importanța documentului fiind măsurată în principal pe baza numărului de accesări de

către diferiți utilizatori, în timp ce sistemul bazat pe indici ține seama mai mult de conținutul documentului. În ultimă instanță, decizia corectă asupra relevanței documentului într-un domeniu o are utilizatorul (beneficiarul) avizat. Astfel, cele mai populare (căutate) documente nu sunt în general şi cele mai relevante.

Filtrul de relevanță propus poate fi aplicat la motoare de căutare inteligente, cu scopul de a răspunde mai exact la cererile concrete de informare ale utilizatorilor şi de a reduce timpul de căutare printr-o clasificare mai bună. Concepția şi rezultatele experimentale prezentate în această lucrare pot fi utile în dezvoltarea actualelor motoare de căutare bazate pe potrivirea de cuvinte cheie, mergând către căutarea după concept şi web-ul semantic.

Referințe bibliografice

Ioniță, S. (2004). Sisteme fuzzy, Editura Universității din Piteşti.

Nicolescu, J., Stoka, M. (1971). Matematici pentru ingineri, vol II. Editura Tehnică, Bucureşti.

Onicescu, O., Ştefănescu, V. (1979). Elemente de statistică informațională. Editura Tehnică, Bucureşti.

Ştefănescu, D. Tufiş, D., Irimia, E. (2006). *Resurse lingvistice şi instrumente pentru prelucrarea limbii române*. Corina Forăscu, Dan Tufiş, Dan Cristea (eds.), Editura Universității "Al. I. Cuza" Iaşi, 186 p., ISBN 978-973-703-208-9.

Ulieru, M., and Ionita S. (2001). Soft Computing Techniques for the Holonic Enterprise, FLINT 2001, M. Nikravesh and B. Azvine (Eds.), *New Directions in Enhancing the Power of the Internet*, UC Berkeley Electronics Research Laboratory, Memorandum No. UCB/ERL M01/28, August 2001. pp. 182-187.

* * * Search Tools for Web Sites and Intranets, accesibil la http://www.searchtools.com

CONTRIBUŢII LA PROIECTUL "ROLTECH. PLATFORMĂ PENTRU TEHNOLOGIA LIMBII ROMÂNE: RESURSE, INSTRUMENTE, INTERFEŢE"

C.CIUBOTARU, S.COJOCARU, E.BOIAN, A.COLESNICOV, L.MALAHOVA,

G. MAGARIU, M. PETIC, T. VERLAN, O. BURLACA

Institutul de Matematică şi Informatică, Academia de Ştiinţe a Republicii Moldova

{chebotar, sveta, lena, kae, mal, gmagariu, mirsha, tverlan, oburlaca}@math.md

Rezumat

Lucrarea propune contribuţii la proiectul RoLTech ce ţin de crearea aplicaţiilor bazate pe resurse reutilizabile pentru tehnologia limbii române (RRTLR). Se propun metode pentru automatizarea achiziţionării RRTLR, pentru verificarea formală (inclusiv cu caracter statistic) a RRTLR, precum şi metode bazate pe corespondenţa caracteristicilor morfologice în diferite limbi. Se propune un sistem adaptabil de instruire asistată de calculator pentru morfologia limbii române, un generator de dicţionare şi un motor de căutare Web adaptat la particularităţile morfologice ale limbii române.

1. *Introducere*

Printre obiectivele principale ale proiectului RoLTech se evidenţiază crearea aplicaţiilor bazate pe resurse reutilizabile pentru tehnologia limbii române (RRTLR). În particular, elaborarea unui sistem adaptabil de instruire asistată de calculator pentru morfologia limbii române, crearea unui generator de dicţionare, modernizarea sau crearea unor motoare noi de căutare Web, care să utilizeze sinonimele şi formele flexionate ale cuvântului. Soluţionarea acestor probleme depinde de structura, integritatea, corectitudinea şi instrumentarul de gestionare a RRTLR. Nu în ultimul rând, valoarea reală a aplicaţiilor lingvistice va depinde şi de volumul achiziţiilor de resurse.

Volumul mare de informaţie nu permite operatorului să verifice, în mod vizual, corectitudinea informaţiei. Achiziţionarea automată, realizată prin intermediul programelor a căror funcţionare a fost verificată, ar spori şi veridicitatea datelor. Urmărind acest scop ne-am propus elaborarea unui set de programe, care contribuie la automatizarea procesului de populare a bazei de date (BD) a RRTLR. Din acest set fac parte următoarele componente: flexionare (statică şi dinamică), prefixare, popularea bazei de date, control al corectitudinii şi integrităţii.

2. *Automatizarea achiziţionării RRTLR. Popularea BD*

2.1 Metoda statică de flexionare

Metoda se bazează pe clasificarea descrisă de A. Lombard şi C. Gâdei (Lombard&Gâdei, 1981) şi programele de flexionare propuse de S. Cojocaru (S. Cojocaru, 1997). Algoritmul utilizează o gramatică de flexionare care formalizează

procesul de realizare a alternanţelor şi de concatenare a seturilor de terminații. Pentru limba română această gramatică include 866 reguli şi 320 seturi de terminații. Această metodă a contribuit substanțial la acumularea resurselor lingvistice, fiind aplicată pentru flexionarea a circa 30.000 de cuvinte-leme.

2.2 Metoda dinamică de flexionare

În cadrul metodei nu se utilizează liste concrete de cuvinte, dar se încearcă să se calculeze paradigma de flexionare pentru fiecare cuvânt aparte, utilizând clasificări asemănătoare cu cele descrise de A. Lombard şi C. Gâdei (Lombard&Gâdei, 1981). Algoritmul a fost verificat pe câteva mii de cuvinte, care nu au fost incluse în clasificarea statică. De asemenea, au fost depistate unele iregularități (3% din mulțimea de cuvinte flexionate). S-a elaborat un set de programe care în regim semi-automat generează toate formele flexionate. În anumite cazuri se solicită implicarea utilizatorului pentru selectarea paradigmei potrivite.

2.3 Popularea bazei de date a RRTLN

Informația verificată, obținută ca rezultat al programelor de flexionare, se utilizează la completarea bazei de date a RRTLN. În scopul automatizării procesului de populare s-a elaborat o structură unică pentru intrările bazei; totodată fişierele cu informația necesară au fost convertite în acest format (Ciubotaru et al., 2006). Programele de populare generează un fişier log care informează asupra informației incluse şi a erorilor comise. Pentru a introduce în bază o informație nouă, utilizând algoritmii elaborați în cadrul corectorului RomSP, s-a elaborat un set de programe care în regim semi-automat generează toate formele flexionate pentru a fi incluse în bază. Programele se execută în regim *wizard* şi necesită implicarea unui lingvist expert.

2.4 Popularea BD prin prefixare

Derivarea prin prefixare este unul dintre procedeele principale de formare a cuvintelor în limba română. Unele caracteristici cantitative pentru prefixele *ne-, des-, re-* au fost stabilite în (Petic, 2007a).

Pentru simularea procesului de prefixare automată s-au definit şi analizat unele reguli care ar permite extragerea automată a formațiilor analizabile cu prefixe simple şi compuse. Drept sursă pentru extragerea formațiilor analizabile cu prefixe simple au servit resursele lingvistice reutilizabile elaborate anterior (Ciubotaru et al., 2006), care conțin nu doar reprezentarea grafică a cuvântului, ci consemnează şi ce fel de parte de vorbire are cuvântul. Utilizând această informație şi ținând cont de particularitățile prefixelor şi cele ale derivatelor lor, s-a elaborat un algoritm de extragere automată a formațiilor analizabile cu prefixe simple şi compuse (Petic, 2007b).

Pentru inventarul dicționarului morfologic al limbii române (Lombard&Gâdei, 1981) doar cu ajutorul prefixelor *ne-* şi *re-* s-au obținut 17.274 cuvinte noi, ceea ce reprezintă aproximativ 60% din lexiconul inițial. Prin aplicarea programelor de flexionare (Cojocaru, 1997) s-au generat apoi, din acestea, 511.280 forme flexionate.

3. Integritatea şi corectitudinea bazei de date (BD)

Pe parcursul creării şi completării bazei de date, în mod inevitabil pot apărea erori. Aceasta se explică parţial prin volumul mare de informaţie (în prezent sunt incluse circa un milion de intrări), dar şi prin erori provenite din alte cauze:

- datorate surselor lexicografice utilizate (formatul electronic),

- comise în programele de procesare a informaţiei lexicografice,

- comise de către operatorii umani.

Acest aspect ne impune să elaborăm metode pentru a verifica corectitudinea şi integritatea bazei de date. Desigur, sunt binevenite metodele de verificare automată. Din păcate, aceste verificări nu pot fi efectuate integral în mod automat. Adesea rămâne o informaţie care poate fi suspectată ca incorectă şi care trebuie selectată şi verificată de un expert filolog. Este de dorit ca informaţia ce poate fi suspectată ca incorectă să fie cât mai redusă posibil.

3.1 Motoare de vizualizare

Au fost realizate cinci motoare care verifică corectitudinea şi posibilităţile de utilizare a informaţiei incluse în BD: vizualizarea informaţiei morfologice, a formelor flexionate, a sinonimelor, a traducerilor în limbile engleză şi rusă. Fiecare motor constă din două componente. Prima reprezintă un şablon de căutare. Cu ajutorul acestui şablon se poate solicita vizualizarea unui singur cuvânt, sau a unui set de cuvinte specificând o expresie regulată în care caracterul "?" semnifică un simbol arbitrar, iar "*" – un şir arbitrar (posibil vid). Şablonul mai are cinci butoane care permit selectarea literelor cu diacritice, în caz că acest lucru nu-l permite tastatura. A doua componentă comună tuturor motoarelor poate fi numită "paginator" şi gestionează distribuirea informaţiei din BD pe pagini (5 blocuri pe o pagină). Astfel este posibilă selectarea paginii cu numărul dat, trecerea la pagina următoare sau anterioară etc. Motorul pentru vizualizarea informaţiei morfologice se adresează tabelului *word_flexies* şi va arătă toate atributele morfologice prezente în tabel pentru fiecare din cuvintele specificate.

Motorul pentru vizualizarea formelor flexionate ale cuvântului foloseşte tabelului *words* cu un cuvânt-lemă concret, selectează şi afişează toate formele flexionate ale acestui cuvânt. Motoarele pentru sinonime, traduceri (engleză, rusă) funcţionează analog folosind tabelele respective.

3.2 Integritatea bazei de date. Metode formale

Crearea bazei de date a RRTLN este un proces dificil. Datorită surselor de erori menţionate mai sus, putem presupune că fiecare câmp al BD ar putea conţine erori. Este imposibil de efectuat verificarea integrală şi corectarea tuturor erorilor în regim automat. De exemplu, nici un program de verificare nu poate decide, fără intervenţia unui expert filolog, dacă un cuvânt dat este sinonim sau reprezintă traducerea altui cuvânt. Pot fi elaborate programe (Cojocaru et al., 2006) care rezolvă parţial această problemă. Pentru început se propun tehnici formale de verificare a validităţii structurii BD. Aceste tehnici au fost formulate folosind semantica şi interdependenţele câmpurilor BD şi a tabelelor. Toate câmpurile BD se clasează în patru categorii:

– care conțin o reprezentare textuală a cuvintelor (română, rusă, engleză),
– care conțin referințe către alte câmpuri din tabele, de exemplu, numere ce codifică cuvintele în tabelul sinonimelor,
– care conțin atribute morfologice, sau alt tip de atribute,
– care conțin reprezentarea textuală (descifrarea) a atributelor; astfel de câmpuri se întâlnesc doar în tabelele auxiliare.

Pentru fiecare câmp în tabel se specifică categoria şi metoda de verificare. De exemplu, câmpul *part_code* din tabelul *words* (şi toate tabelele de categoria 3) trebuie să conțină doar numere – coduri ale părților de vorbire din tabelul *parts_of_speech* – şi de aceea pot avea numai valori întregi de la 1 (codul pentru verb) până la 10 (codul pentru conjuncție). Câmpurile *flexy_word* din tabelul *word_flexies* (şi toate tabelele de categoria 1) admit doar litere româneşti.

3.3 Integritatea bazei de date. Verificarea cuvintelor

Următoarea metodă formală a fost aplicată la verificarea cuvintelor. Pentru cuvintele în limba română a fost utilizat corectorul RomSP (Colesnicov, 1995), care operează cu o listă de cuvinte deja testată de elaboratorii şi utilizatorii acestui produs. Am utilizat de asemenea corectoarele de texte MS Office pentru limbile română, rusă şi engleză. Cuvintele din limba română neacceptate de ambele corectoare au fost marcate ca fiind în mod special susceptibile de a fi greşite. Verificările ulterioare au arătat că majoritatea lor sunt greşite.

O metodă efectivă de verificare a fost utilizarea *n*-gramelor (părți de cuvinte ce conțin exact *n* litere, $n > 2$). Cuvintele care conțin *n*-grame mai puțin frecvente se consideră a fi suspicioase şi au fost verificate ulterior de experți.

3.4 Integritatea bazei de date. Verificarea atributelor

Toate câmpurile de categoria 3 pot fi verificate formal simplu: ele trebuie să conțină doar referințe la tabelele auxiliare. Corespondența câmpurilor de categoria 3 şi 4 poate fi verificată parțial utilizând intervale de valori pentru atribute. Tabelele auxiliare pot fi verificate vizual deoarece sunt scurte. O altă metodă de verificare constă în căutarea codurilor atributelor care se folosesc rar sau nici nu se folosesc în tabelele de bază.

3.5 Verificarea multiplicărilor

În tabelul *words* câmpul *prim_word_code* este unic. Informația respectivă constă din cuvânt (în forma textuală), partea lui de vorbire şi domeniul de utilizare. Aceste date sunt verificate pentru a asigura unicitatea completării bazei de date. Repetarea unor combinații demonstrează că nu sunt corecte programele de populare. În acest caz putem vizualiza fişierele erorilor pentru aceste combinații. Forma textuală a cuvintelor nu ne permite să determinăm domeniul lor de utilizare. În acest caz, în câmpul respectiv întotdeauna este stabilită valoarea 1 (se indică domeniul general de utilizare). Astfel, putem verifica unicitatea perechilor „formă textuală a cuvântului – parte de vorbire".

De asemenea, se pot verifica câmpurile şi tabelele care folosesc cuvinte cu domenii specifice de utilizare. S-au depistat şi cazuri când informaţia pentru sinonime şi traduceri era eronată. Mai mult decât atât, am verificat cuvintele din tabelul *words* din punct de vedere a unicităţii formei textuale a cuvântului, chiar ignorând partea de vorbire. În limba română adjectivele pot coincide cu adverbele, iar substantivele – cu adjectivele, dar aceste cazuri sunt relativ rare. Această situaţie se deosebeşte de limba engleză unde, de regulă, acelaşi cuvânt poate fi atât verb cât şi substantiv. Această verificare a permis de asemenea detectarea unor erori. Unicitatea înregistrărilor în tabelele *word_flexies, word_synonyms, word_translations, word_engl* şi *word_rus* este verificată în procesul populării bazei de date. Testarea programelor de populare a bazei de date poate fi efectuată după completarea bazei de date.

3.6 Verificarea statistică a flexionărilor

Verificarea statistică a procesului de flexionare ne-a permis să depistăm un şir de erori pentru cazurile când numărul formelor flexionate depăşea numărul admisibil pentru o anumită parte de vorbire, de exemplu, 35, 39, sau 40 pentru verb. Devierea de la aceste repere numerice ne indică posibile erori.

S-au cercetat derivatele existente ale cuvintele din tabelul *words*. Am observat, de exemplu, că există verbe care conţin 160 de forme flexionate. Numărul suspect de forme pentru unele cuvinte ne permite să depistăm şi să corectăm anumite erori. De exemplu, analizând situaţia cu verbele, am ajuns la concluzia că, la etapa de proiectare, nu au fost luate în consideraţie unele detalii ale gramaticii limbii române.

3.7 Utilizarea dicţionarelor paralele

Utilizarea dicţionarelor paralele s-a dovedit a fi o metodă utilă în PLN (Tufiş&Barbu, 2002). În cazul nostru, acestea au fost traducerile în limba rusă. Limba rusă, ca şi limba română, are un grad înalt de flexionare. Au fost analizate verbe, adjective şi adverbe. Aceste părţi de vorbire în limba rusă au terminaţii tipice, care, mai mult sau mai puţin, depind de partea de vorbire.

De exemplu, verificările pentru cuvintele care nu sunt verbe, dar traducerile lor în limba rusă conţin terminaţii verbale, au depistat 4.119 astfel de cuvinte. Majoritatea erau cuvinte corecte, dar s-au găsit şi erori. De asemenea, s-au verificat cuvintele care nu sunt adjective, dar traducerile lor în limba rusă conţin terminaţiile adjectivale. Astfel de cuvinte n-au fost depistate. Verificările pentru cuvintele care nu sunt adverbe, dar traducerile lor în limba rusă conţin terminaţiile adverbiale au generat prea multe cuvinte, 18.974. Acest număr a putut fi micşorat eliminând toate verbele, adjectivele şi substantivele. În acest mod au mai fost depistate o serie de erori.

De asemenea, au fost clasificate suspecte şi examinate suplimentar cazurile când partea de vorbire nu corespundea celei aşteptate.

3.8 Completitudinea bazei de date

Pentru verificarea completitudinii bazei de date a fost propus următorul test. Având lista cuvintelor limbii române obţinută dintr-o sursă alternativă este posibilă sortarea şi compararea ei cu lista sortată a cuvintelor din tabelele *words* sau *word_flexies*. Tabelul

word_flexies ar putea fi utilizat dacă sursa alternativă admite formele flexionate ale cuvintelor. Cu ajutorul acestui procedeu au fost găsite cuvinte care nu existau în BD.

4. Sistem adaptabil de instruire asistată de calculator pentru morfologia limbii române

La etapa inițială a fost proiectat şi realizat un sistem de instruire asistată de calculator pentru morfologia limbii române, bazat pe o arhitectură ierarhică orientată spre trei categorii de utilizatori (Boian et al., 2006a; Boian et al., 2006b). Au fost elaborate metode de expunere a materialului, structura lecțiilor, metode de aplicare a elementelor multimedia. La etapa actuală se produce acumularea cantitativă a materialelor pentru cursuri, se aplică metodele elaborate la realizarea acestor cursuri, se implementează utilizarea resurselor lingvistice reutilizabile. Informația din resurse se expune în cadrul cursurilor cu ajutorul unor module PHP care asigură vizualizarea informației morfologice solicitate.

5. Motor de căutare Web adaptat la particularitățile limbii române

S-a elaborat un motor de căutare Web (MC) cu posibilități de implicare a unor particularități ale morfologiei limbii române. Drept bază pentru implementarea lui a servit unul din cele mai populare motoare cu cod deschis *mnoGoSearch* (www.mnogosearch.org), care suportă conectarea unei baze de date preconstruite a formelor flexionate. El oferă două posibilități de flexionare automată a cuvintelor din interpelare: a) dicționare Ispell b) lista cuvintelor şi a tuturor flexiunilor acestora. Dicționarele Ispell sunt utilizate în corectoarele de texte. În urma efectuării unor experimente s-a stabilit, că cel mai eficient şi corect este să generăm lista cu flexiuni pe care urmează s-o integrăm în MC. Eficient, fiindcă flexiunile cuvântului nu trebuie generate în momentul căutării prin aplicarea regulilor din fişierul cu afixe. Corect, fiindcă putem folosi algoritmi specifici unei limbi pentru generarea flexiunilor, şi nu suntem limitați de puterea de exprimare a formatului Ispell. În plus, putem exclude flexiunile ce se utilizează rar pentru optimizarea procesului de căutare.

S-a folosit lexiconul computațional pentru limba română care conține circa 1 milion de cuvinte (derivate din aproximativ 60.000 de cuvinte-lemă) (Cojocaru, 1997). Tehnic vorbind, a fost creat un tabel din două coloane [cuvânt, flexiune]. Cu ajutorul mecanismului SQLWordForms apărut în mnoGoSearch în ianuarie 2006, fiecare cuvânt din interpelare este înlocuit prin lista flexiunilor lui. Ambele coloane ale tabelului sunt indexate, de aceea procesul de extindere a interpelării este foarte rapid.

A fost implementată căutarea pe site-ul Consiliului Național pentru Acreditare şi Atestare (www.cnaa.acad.md), care conține în jur de 2.7GB texte în format PDF.

Pentru a oferi o acoperire mai mare, semnele diacritice româneşti nu se iau în considerație la căutare. Cu alte cuvinte, semnele Ş, Ă, Î, Ț, Â se echivalează cu S, A, I, T, A. La indexare, semnele diacritice sunt eliminate. Caracterul Open Source al MC *mnoGoSearch* ne-a permis să efectuăm modificări în codul sursă pentru a echivala

semnele Î şi Â. De exemplu, dacă vom căuta *câmp*, vom găsi şi paginile ce conţin *cîmp*, şi invers.

Extinderea interpelării prin includerea formelor flexionate măreşte numărul de rezultate. De exemplu, cuvântul *teorie* a fost întâlnit de 149 ori, iar formele flexionate ale acestui cuvânt au fost întâlnite de 2.262 ori. Au fost găsite 64 de pagini în care se întâlneşte cuvântul *teorie* şi 423 de pagini în care au fost întâlnite formele flexionate ale lui.

Mulţumiri. Lucrarea este efectuată în cadrul proiectului INTAS Ref. Nr. 05-104-7633 şi grantului 06-411-03-01P, Consiliul Suprem pentru Ştiinţă şi Dezvoltare Tehnologică.

Referinţe bibliografice

E.Boian, C.Ciubotaru, S. Cojocaru, G.Magariu, T.Verlan, Iu. Rogojin (2006b). Sistem de instruire asistată de calculator pentru morfologia limbii Române, Lucrările atelierului „Resurse lingvistice şi instrumente pentru prelucrarea limbii Române”, Iaşi, noiembrie 2006, pp. 135-139.

E. Boian, C. Ciubotaru, G. Magariu, T. Verlan, O. Calughin (2006a). Sistem de instruire asistată de calculator pentru morfologia limbii române, Proceedings of The XIV Conference On Applied And Industrial Mathematics, dedicated to the 60th anniversary of the foundation of the Faculty of Mathematics and Computer Science of Moldova State University, satellite conference of ICM2006, Chisinau, Republic of Moldova, August 17-19, 2006, pp. 353-357.

C. Ciubotaru, S.Cojocaru, E.Boian, A.Colesnicov, L.Malahova, V.Demidova, O. Burlaca (2006). Resurse Lingvistice Reutilizabile. Lucrările atelierului „Resurse lingvistice şi instrumente pentru prelucrarea limbii Române”, Iaşi, noiembrie 2006, pp. 75-79.

S. Cojocaru (1997). Romanian Lexicon: Tools, Implementation, Usage. In: Dan Tufis, Poul Andersen (eds.). Recent Advances in Romanian Language Technology. ISBN 973–27–0626–0, Editura Academiei, I, pp. 107–114.

S. Cojocaru, A. Colesnicov, L. Malahova (2006). Integrity and correctness checking of a lexical database. Computer Science Journal of Moldova, vol.14, Nr.1(40), pp. 138-151.

A. Colesnicov (1995). The Romanian spelling checker ROMSP: the project overview. Computer Science Journal of Moldova, v. 3, Nr. 1(7), pp. 40–54.

A. Lombard, C. Gâdei (1981). Dictionnaire morphologique de la langue roumaine, Editura Academiei, Bucureşti.

M. Petic (2007b). Automatic extraction of the analysable formations with simple prefixes. Proceedings of the Second International Conference of Young Scientists Computer Science and Engineering-2007, Lvov, october 2007, pp 215-217.

D. Tufiş and A.M. Barbu (2002). Revealing Translator's Knowledge: Statistical Methods in Constructing Practical Translation Lexicons for Language and Speech Processing. International Journal of Speech Technology, pp. 199–209.

O ALTĂ METODĂ DE RESTABILIRE A SEMNELOR DIACRITICE

VICTORIA BOBICEV

Universitatea Tehnică a Moldovei

vika@rol.md

Rezumat

În lucrarea de față este prezentată aplicarea algoritmului de comprimare a textelor PPM (prediction by partial matching – prezicere prin potrivire parțială) la rezolvarea problemei restabilirii semnelor diacritice în textele româneşti. Metoda propusă nu are nevoie de resurse suplimentare în afară unui volum de texte corecte. Avantajul metodei este viteza sporită şi precizia medie de 97-99% în funcție de textele folosite la experimente. Algoritmul este realizat în Perl.

1. Introducere

În procesarea textului acuratețea rezultatelor depinde considerabil de „calitatea" textului procesat. Una din caracteristicile importante ale calității textului pentru multe limbi rămâne prezența sau absența semnelor diacritice. În absența lor sistemele de procesare nu pot obține rezultate acceptabile, de aceea este necesară includerea unui modul specializat de restabilire a diacriticelor.

Problema restabilirii semnelor diacritice a fost abordată de mai mulți cercetători. În (Mihalcea & Nastase, 2002) a fost prezentat un tabel cu lista limbilor europene ce se confruntă cu această problema. În majoritatea cazurilor, ca şi în cazurile verificării şi corectării automate a cuvintelor cu erori ortografice, restabilirea dicriticelor poate fi efectuată cu ajutorul unui dicționar suficient de mare. Cum este menționat în (Mihalcea & Nastase, 2002), metodele bazate pe dicționare au o acuratețe de peste 90% (Yarowsky, 1994) (Galicia-Haro et al., 1999). Inconveniente sunt, în primul rând, indispobibilitatea unui astfel de dicționar şi, în al doilea rând, ambiguitatea alegerii între variantele de cuvinte corecte date de vocabular. Pentru selectarea variantei potrivite au fost propuse diferite metode statistice.

În (El-Beze et al., 1994) a fost folosit un lanț Markov în baza secvențelor din trei cuvinte şi precizia obținută a fost în jur de 99% pentru textele franceze. O metodă bazată pe secvențe de cuvinte vecine a fost prezentată în (Nagy & Sabourin, 1998). În (Yarowsky, 1999) au fost prezentate mai multe metode, majoritatea având la bază alegerea candidatului potrivit dintr-o submulțime a dicționarului sugerată de definirea de vecinătăți.

Limba română conține 5 caractere diacritice: ă, â, î, ş şi ț. Cazul literelor â şi î este special în limba română: deşi pronunția lor este identică, folosirea lor este guvernată de reguli bazate pe poziția lor în cuvânt. Textele, pe care s-au făcut experimentele, folosesc în principal ortografia românească veche, care nu includea „â".

În (Tufiş & Chițu, 1999) este prezentată statistica cuvintelor cu semne diacritice în texte din diferite domenii. Aproximativ 75-78% de cuvinte din text pot fi restaurate în mod determinist. Cuvintele lipsite de semne diacritice reprezintă 55-58% din text,

aproximativ 20% sunt cuvinte care nu există în varianta fără diacritice, ele putând fi corectate cu ajutorul unui dicţionar, restul de 20-25% fiind cuvinte ambigue, care necesită prelucrare specială.

Pentru limba română au fost propuse două metode de restabilire a diacriticelor. Prima a fost descrisă de (Tufiş & Chiţu, 1999) şi se bazează pe un analizator morfologic. Prima etapă de analiză morfologică constă în consultarea dicţionarului în scopul obţinerii lemei şi a codului morfologic. Paralel cu adnotarea automată se face şi corectarea cuvintelor cu semnul diacritic pierdut. Alegerea cuvântului potrivit se efectua pe baza contextului şi a caracteristicilor morfologice folosind lanţuri Markov.

O metodă originală a fost descrisă în (Mihalcea & Năstase, 2002). Metoda lor nu rezolvă problema la nivel de cuvînt ci la nivelul literelor. În limba română există patru grupe de litere ambigue a-ă-â, i-î, t-ţ, s-ş. Decizia se face pe baza contextului, mai exact – cinci litere precedente şi cinci litere următoare. A fost folosită implementarea TiMBL (Daelemans et al., 2003) ca şi metoda de învăţare automată (*machine learning method*). Precizia obţinută este de 98-99% de semne diacritice restabilite corect la nivel de literă.

Metoda această este elegantă, rapidă şi de bună calitate, însă la utilizarea ei am întâlnit o problemă. Nu am reuşit să instalăm sistemul TiMBL pe Windows nici cu un emulator de Unix.

În lucrarea de faţă este prezentată o altă metodă de restabilire a semnelor diacritice la nivel de literă bazată pe model statistic de comprimare a textelor PPM (*prediction by partial matching* – prezicere prin potrivire parţială).

2. PPM

Cercetătorii au observat că modelele create pentru comprimarea textului sunt foarte asemănătoare cu modelele create pentru prelucrarea lui. Comprimarea micşorează cerinţa de memorie pentru păstrarea fişierelor în format electronic şi timpului necesar pentru transmiterea datelor. Metodele cele mai bune au la bază codificarea statistică (Moffat et al., 1995) (Witten et al., 1987). În comprimarea statistică fiecărui simbol îi este atribuit un cod bazat pe probabilitatea sa în text. Simbolurile cu probabilitatea mare obţin coduri scurte, simbolurile cu probabilitatea mică au coduri mai lungi.

Legătura între probabilitatea şi lungimea codului este descrisă în teorema lui Shannon referitor la codificarea sursei (Shannon, 1948), în care este demonstrat faptul că simbolul cu probabilitatea ***p*** în cazul optimal este prezentat cu ***-log p*** biţi, unde ***log p*** este logaritm în baza 2 a probabilităţii simbolului.

$$H = -\sum_{i} p(i) \log p(i) \qquad (1)$$

Valoarea aceasta este numită entropia mesajului. În cadrul modelării sunt estimate probabilităţile simbolurilor textului codificat. În baza probabilităţilor aflate se calculează entropia care este o caracteristică a modelului statistic şi arată cât de adecvat modelul reprezintă probabilităţile reale ale simbolurilor în text.

O proprietate interesantă a modelelor de comprimare este posibilitatea de a se adapta la textul comprimat. Modelul adaptabil de comprimare a textelor PPM a atras interesul cercetătorilor din domeniul procesării limbajului natural (Teahan, 1998).

Metoda PPM clasică este o metodă de comprimare pe baza contextului limitat (*finite-context modelling*), care estimează probabilitatea simbolurilor pe baza contextului, adică simbolurile precedente în text. Inițial, metoda a fost propusă în (Cleary & Witten, 1984), mai apoi fiind modificată şi optimizată. În (Moffat, 1990) a fost descrisă o implementare a algoritmului optimizat PPMC, care este considerat cel mai bun algoritm de comprimare a textelor (Teahan, 1998).

PPM prezintă o variantă de amestecare (*blending*) a probabilităților, când probabilitățile obținute în baza contextelor cu lungime diferită sunt unite într-o probabilitate comună. În cazul general, probabilitatea amestecată p(c_k) a simbolului c_k poate fi calculată ca:

$$p(c_k) = \sum_{i=1}^{o} \lambda_i \, p'(c_k \mid c_{k-i} \ldots c_{k-1}) \qquad (2)$$

unde $p'(c_k \mid c_{k-i} \ldots c_{k-1})$ este probabilitatea simbolului curent, determinat de model pe baza contextul $c_{k-i} \ldots c_{k-1}$, începând cu contextul maximal *o*; λ_i este coeficientul de normalizare:

$$\sum_{i=1}^{o} \lambda_i = 1 \qquad (3)$$

În PPM amestecarea contextelor cu lungime diferită se efectuează prin mecanismul '*escape probability*'. Calculul probabilității simbolului începe considerând cel mai lung context şi trecând la contextul mai scurt în cazul în care contextul dat nu a fost întâlnit. '*Escape probability*' este atribuită contextului care nu a fost întâlnit când sistemul abandonează contextul dat şi trece la un context mai scurt. În cazul în care nici simbolul însuşi nu a fost întâlnit, este utilizat contextul -1. '*Escape probability*' poate fi calculată prin diferite metode. În metoda implementată în lucrarea de față este folosită metoda C (Moffat, 1988).

Contextul maximal de la care începe calculul probabilității se numeşte ordinul modelului. În (Teahan, 1998) a fost analizată influența ordinului modelului la calitatea modelului şi a fost observat faptul că ordinul egal cu 5 simboluri precedente este optimal pentru comprimarea textelor.

3. Aplicarea PPM pentru restabilirea semnelor diacritice

În (Teahan, 1998) a fost propusă aplicarea metodei PPM pentru corectarea textelor la nivel de literă. Dacă considerăm că acele cuvinte care ar trebui sa aibă semne diacritice şi nu le au sunt scrise greşit, atunci putem folosi algoritmul de corectare a textelor pentru corectarea erorilor de tipul dat. Pentru corectarea automată a erorilor din text este folosit un tabel numit ‚tabel de înlocuiri', în care se înscriu fragmentele de text greşite şi variantele posibile de înlocuire pentru obținerea textului corect. De exemplu, în textul scanat este frecventă litera ‚d', obținută greşit din ‚cl' sau ‚ci'. Pentru selectarea variantei corecte dintre toate variantele posibile se calculează probabilitatea tuturor variantelor şi se alege cea cu probabilitatea maximă. Probabilitatea este calculată în baza modelului statistic antrenat pe texte corecte.

În cazul restabilirii semnelor diacritice tabelul de înlocuiri este simplu; el conține numai patru litere (***a, i, t, s***) şi înlocuirile posibile sunt respectiv (***ă, î, ţ, ş***). Pentru verificarea necesității înlocuirii este calculată entropia fragmentului textului pentru varianta fără semn diacritic şi cu acesta. Varianta cu entropia minimală este considerată corectă. În continuare este prezentat algoritmul de restabilire a semnelor diacritice în baza modelului PPM. Modelul este creat adaptiv folosind un volum de texte cu semne diacritice. Probabilitățile fragmentelor în procesul corectării se calculează static, fără adaptarea la acestea.

ALGORITMUL de restabilire a semnelor diacritice cu ajutorul modelului PPM:

Pentru fiecare literă a textului c_i:

dacă c_i este literă ambiguă (***a, i, t, s***):

calculăm entropia E_i a segmentului de text în care se află c_i;

înlocuim c_i cu litera cu semn diacritic c'_i;

calculăm entropia E'_i a aceluiaşi segment de text;

dacă $E'_i < E_i$ atunci înlocuirea este definitivă;

altfel, rămâne litera inițială c_i.

În scopul găsirii metodei cele mai potrivite au fost examinate câteva variante. Prima, cea mai simplă, se aplică asupra unui simbol ambiguu (t, s, a, i): se calculează entropia sa, cât şi cea a simbolului alternativ (ţ, ş, ă, î) şi, în cazul în care entropia simbolului alternativ este mai mică, se execută înlocuirea simbolului.

În a doua metodă se consideră că, dacă simbolul curent din text este ambiguu, se identifică cuvântul în care simbolul s-a găsit şi se calculează entropia întregului cuvînt, după care se defineşte entropia cuvântului cu varianta alternativă a simbolului. Iarăşi se compară entropiile calculate pentru variantele alternative şi se alege varianta cu entropia minimă.

A treia metodă se deosebeşte de a doua prin faptul că, în locul entropiei cuvântului, se calculează entropia unui fragment de text format dintr-un număr de simboluri egal cu $2n+1$. În centrul acestui fragment se află simbolul ambiguu, la dreapta şi la stânga lui se iau n simboluri vecine, indiferent de tipul lor. Figura 1 exemplifică un fragment de text pentru care se calculează entropia.

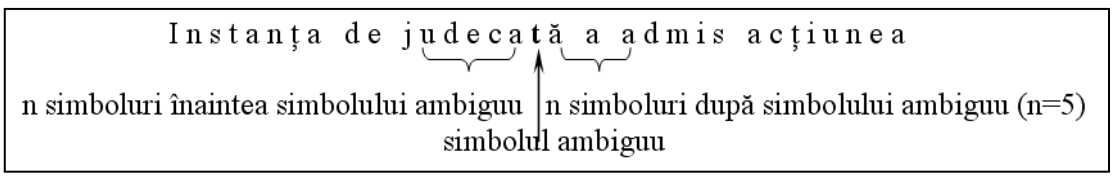


Figura 1. Exemplu de fragment care se foloseşte pentru calculul probabilității.

Fiecare din cele trei metode descrise poate fi folosită cu modelele PPM de ordin diferit: PPM de ordinul 3; PPM de ordinul 4; PPM de ordinul 5 ş.a.m.d. Au fost executate experimente cu modele de ordin diferit cu scopul găsirii modelului cel mai productiv.

4. Experimente

Pentru primul set de experimente a fost folosit corpusul de hotărâri ale Curții de Apel şi Recurs a Republicii Moldova[42], care conțin semne diacritice şi care sunt scrise cu litera ‚î' în toate cazurile (deci, fără litera „â"), ceea ce permite lucrul cu doar patru perechi de litere. Corpusul constă din 880 documente ce conțin 789.520 cuvinte (4.511.057 caractere). 800 documente au fost folosite pentru învățare şi 80 – pentru testare. Înainte de testare, semnele diacritice au fost extrase din datele de test.

Performanța este calculată ca numărul semnelor diacritice restabilite corect în raport cu numărul literelor ambigue.

Tabelele 1-3 prezintă rezultatele verificării fiecărei din cele trei metode descrise mai sus cu modelele PPM de ordinul 4, 5, 6, 7 şi 8, pentru fiecare pereche de litere ambigue.

Tabelul 1: Rezultatele folosirii PPM după prima metodă (literă)

ordinul PPM	**litere ambigue**			
	t – ț	s – ş	i - î	a – ă
4	97,7%	96,4%	97,1%	96,4%
5	97,9%	96%	97,5%	97%
6	98%	96,4%	97,8%	97,2%
7	98,3%	96,5%	98,3%	97,5%
8	**98,6%**	**96,6%**	**98,5%**	**98,3%**

Tabelul 2: Rezultatele folosirii PPM după a doua metodă (cuvânt)

ordinul PPM	**litere ambigue**			
	t – ț	s – ş	i – î	a – ă
4	**99,5%**	94,4%	99,6%	99%
5	99,4%	95%	99,7%	**99,3%**
6	99,3%	96%	99,7%	99%
7	99,2%	96,1%	99,8%	99%
8	99,2%	**96,7%**	**99,8%**	99%

Tabelul 3: Rezultatele folosirii PPM după a treia metodă (fragment)

ordinul PPM	**litere ambigue**			
	t – ț	s – ş	i - î	a – ă
4	**99,8%**	94,2%	99,8%	99%

[42] http://moldova.wjin.net

5	99,7%	94,8%	99,8%	**99,5%**
6	99,76%	96,2%	99,8%	99%
7	99,63%	96,9%	99,9%	99%
8	99,57%	**96,9%**	**99,9%**	99%

Din rezultatele obținute prezentate în tabele putem conclude că lungimea contextului nu influențează în mod substanțial calitatea restabilirii semnelor diacritice. Numai pentru prima metodă se vede dependența clară a calității restabilirii de lungimea contextului. În majoritatea cazurilor, calitatea metodei în baza fragmentelor este cea mai bună.

În general, rezultatele metodei sunt satisfăcătoare. Pentru literele 'a', 'i' şi 't' sunt restabilite corect mai mult de 99% de semne diacritice. Numai pentru litera 's' procentul maximal de semne diacritice restabilite este 96,9%, ceea ce înseamnă că fiecare a 25-a literă 'ş' va fi restabilită greşit. Litera aceasta este cea mai rară din cele patru litere ambigue, ceea înditruieşte ipoteza că o mărire a corpusului de învățare va conduce la rezultate mai bune.

Trebuie menționat şi faptul că, corpusul de hotărâri fiind relativ mic, este posibil ca din acest motiv mărirea lungimii contextului să nu ducă la îmbunătățirea calității restabilirii. Încercând să îmbunătățim rezultatele obținute am repetat toate experimentele efectuate folosind un corpus mai mare, cu semne diacritice. Este vorba de articolele arhivei revistei electronice „România literară”[43]. Corpusul conține 6.339 de documente (7.743.969 cuvinte, respectiv 48.648 093 litere). O căutare rapidă confirmă faptul că în acest corpus apare litera ‚â’, dar nu în toate documentele. Compărând rata literelor ‚â’ şi ‚î’ am hotărât să ignorăm literele ‚â’. Tabelele 4-6 prezintă rezultatele experimentelor în baza corpusului „România literară”.

Tabelul 4: Rezultatele folosirii PPM după prima metodă (literă)

ordinul PPM	litere ambigue			
	t – ț	s – ş	i – î	a – ă
4	**97%**	**93,8%**	95,8%	**92%**
5	97,2%	92,7%	95,8%	91,7%
6	96,9%	91,6%	96,1%	92,2%
7	96,9%	91%	96,2%	92%
8	96,6%	90,9%	**96,3%**	91,9%

[43] http://www.romlit.ro

Tabelul 5: Rezultatele folosirii PPM după a doua metodă (cuvânt)

ordinul PPM	litere ambigue			
	t – ț	s – ş	i – î	a – ă
4	98,4%	96%	99,5%	94%
5	98,5%	95,9%	99,5%	94%
6	98,5%	96%	99,5%	94,6%
7	98,5%	96%	99,5%	94,5%
8	**98,6%**	**96%**	99,5%	**94,6%**

Tabelul 6: Rezultatele folosirii PPM după a treia metodă (fragment)

ordinul PPM	litere ambigue			
	t – ț	s – ş	i – î	a – ă
4	99,3%	96,1%	99,7%	93%
5	99,4%	**96,2%**	99,7%	93,8%
6	**99,5%**	96,1%	99,7%	94,1%
7	99,4%	96,1%	99,7%	94,1%
8	99,4%	96,1%	99,7%	**94,2%**

Din tabele este clar că pentru al doilea corpus, ca şi pentru primul, lungimea contextului nu joacă un rol important. Pe baza faptului că pentru comprimare contextul optimal a fost stabilit egal cu 5, am considerat un context de această lungime. Din cele trei metode testate cele mai bune rezultate au fost obținute în cazul metodei ce utilizează un fragment de text.

Compărând cu corpusul precedent, observăm că pentru al doilea corpus rezultatele sunt mai puțin reuşite. Aceasta se explică prin faptul că textele „României literare” sunt mai variate decât textele corpusului de hotărâri. Astfel apar mai multe cazuri când modelul greşeşte la restabilirea semnelor diacritice. Rezultatul cel mai slab este pentru litera 'a' – 94%, ceea ce înseamnă că eroarea apare în fiecare a 20-a literă 'a'. Litera 'ă', de exemplu, apare în multe cazuri la sfârşitul cuvintelor nearticulate şi se schimbă în 'a' dacă cuvântul este articulat. Este evident că modelul în baza literelor nu poate trata corect cazul când cuvântul este articulat. În afară de aceasta, în corpusul din „România literară” apar şi litere 'â', ignorate de noi, ceea ce, de asemenea, are o oarece influență negativă asupra rezultatelor.

Am decis totuşi, să introducem litera „â” în procesul de restabilire. În scopul acesta am ales 600 documente din „România literară” ce cuprind cu un număr mai mare de litere „â” pentru învățare şi 30 de documente – pentru testare. A fost folosit modelul cu contextul maximal egal cu 5 şi metoda pe baza fragmentului cu lungimea de 11 litere. Literele „â” şi „ă” în documentele de testare au fost înlocuite cu „a”. La etapa testării,

programul compara trei variante posibile: „a”, „ă” şi „â” pentru fiecare literă „a” întâlnită în text. Procentul total al literelor „a”, „ă” şi „â” corect definite de program a fost egal cu 96,13%.

Compărând rezultatele obținute cu lucrările din domeniul dat, putem spune că modelul PPM obține rezultate comparabile cu cele prezentate de alți autori. Tabelul 7 prezintă compararea directă a celor mai reuşite rezultate ale metodei propuse şi ale metodei prezentate în (Mihalcea & Nastase, 2002) în baza corpusului „România literară”.

Tabelul 7: Compararea directă a celor mai reuşite rezultate ale metodei propuse şi ale metodei prezentate în (Mihalcea & Nastase, 2002) în baza corpusului „România literară”.

	t – ț	s – ş	i – î	a – ă
Mihalcea & Nastase	98,75%	99,07%	99,69%	96,14%
Metoda propusă	99,5%	96,2%	99,7%	96,13%

Dacă comparăm calitatea restabilirii semnelor diacritice pentru litere diferite, observăm că cel mai bine se restabilesc semnele diacritice pentru literele 'i' şi 't', indiferent de tipul corpusului. Problemele apar la literele 's' şi 'a'. Aceasta arată că 'î' şi 'ț' apar într-un număr limitat de contexte concrete, iar 'ş' şi 'ă' sunt folosite mai variat şi este mai greu de hotărât când este nevoie de inserat semnul diacritic.

Trebuie luat în considerație faptul că textele folosite în experimente conțin nume proprii, citate din alte limbi, abrevieri etc., ceea ce afectează calitatea rezultatului obținut. De exemplu, într-un text litera ‘ş’ apare într-un nume propriu german care se repetă de mai multe ori.

Ultima problemă pe care am rezolvat-o este detectarea dependenței calității restabilirii semnelor diacritice de volumul corpusului de învățare. În textul românesc literele cu semne diacritice sunt cu mult mai rare decât perechile lor fără semne diacritice. De exemplu, din toate literele 't'-'ț', litera 't' se întâlneşte în 88% de cazuri, litera 'ț' - numai în 12%. Respectiv, dacă semnele diacritice sunt eliminate, în 88% de cazuri, litera este corectă. În tabelul 8 este prezentată precizia, în corelanță de mărimea textului de învățare, măsurat în pagini.

Tabelul 8: Dependența procentului semnelor diacritice restabilite corect de mărimea corpusului de învățare.

mărimea corpusului de învățare (pagini)	procentului semnelor diacritice restabilite corect	
	metoda II (cuvânt)	metoda III (fragment)
4000	98,7	99,3
3000	98,7	99,3
2000	98,7	99,3
1000	98,7	99,3

500	98,4	99,2
300	97,5	99,1
200	96,7	98,6
100	96,5	98,2
50	**95,2**	**97,2**
30	84,5	86,5
20	84,1	86,2
10	82	84,5
5	81,2	82,5
4	79,9	81,8
3	79,7	81,2
2	79,2	79,3
1	77,5	79,1

Din tabel se observă că un rezultat mai bun decât ignorarea totală a semnelor diacritice se obține după un proces de învățare în baza a 50 de pagini (în tabel cifrele acestea sunt subliniate). Pentru obținerea unor rezultate satisfăcătoare sunt de ajuns 500 de pagini de text. Mărirea volumului corpusului peste 1000 de pagini practic nu măreşte precizia.

5. Concluzie

În lucrarea de față este descrisă o metodă de restabilire a semnelor diacritice în textele româneşti. Metoda propusă are anumite avantaje față de metodele deja existente. Metoda dată are nevoie de un volum comparativ mic de texte corecte şi nu apelează la alte resurse lexicale sau produse soft. Încă un avantaj al metodei este viteza procesării textului care este aproximativ de patru pagini pe secundă. Algoritmul este realizat în limbajul Perl. Calitatea restabilirii semnelor diacritice este 99,3% la nivel de literă.

Există unele modalități de îmbunătățire a calității metodei date. Depistarea numelor proprii care nu sunt supuse regularităților statistice generale ale limbii poate ajuta la sporirea acurateței. Ca şi pentru orice metodă statistică, textele de învățare joacă un rol crucial şi învățarea pe texte similare cu cele procesate sporeşte considerabil calitatea corectării textelor.

Mulțumiri. Autorii sunt recunoscători recenzenților anonimi pentru atenție şi numeroasele corectări şi recomandări, ce au ajutat la îmbunătățirea lucrării.

Referințe bibliografice

Cleary J.G., Witten I.H., (1984). Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications, Vol. 32(4), p.396-402.

Daelemans, W., Zavrel, J., Van Der Sloot, K., Van Den Bosch, A., (2003). TiMBL: Tilburg memory based learner, version 5.0 reference guide. ILK Technical Report ILK 03-10, Unirversity of Antwerp, p.56.

El-Beze, M., Merialdo, B., Rozeron, B., Derouault, A., (1994). Accentuation automatique des textes par des metodes probabilistes. Techniques et sciences informatique, 16(6), p.797-815.

Galicia-Haro, S. N., Bolshakov, I. A., Gelbukh , A. F., (1999). A simple Spanish part of speech tagger for detection and correction of accentuation error. In Proceedings of the Second International Workshop on Text, Speech and Dialogue, Plzen, Czech Republic, p. 219-222.

Mihalcea, R., Nastase, V., (2002). O metodă automată pentru inserarea diacriticelor în texte. În "Limba Româna în Societatea Informationala - Societatea Cunoasterii", D. Tufis and F. G. Filip Editors, Academia Româna, Ed Expert, Bucharest, p. 191-206.

Moffat, A., (1988). A note on the PPM data compression algorithm, Research Report 88/7, Department of Computer Science, University of Melbourne, Parkville, Victoria, Australia.

Moffat, A., (1990). Implementing the PPM data compression scheme. IEEE Transactions on Communications, Vol. 38, No. 11, pp. 1917-1921.

Moffat, A., Neal, R., Witten, I. H., (1995). Arithmetic coding revisited. Proceedings DCC'95, edited by Storer, J.A. & Cohn, M., IEEE Computer Society Press, pp. 202-211.

Nagy, G.N., Sabourin, M., (1998). Signes diacritiques: perdus et retrouvés. In Actes du Colloque International Francophone sur l'Ecrit et le Document CIFED, Quebec, Canada, pp. 404-412.

Shannon, C.E., (1948). A mathematical theory of communication. Reprinted with corrections from The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656.

Teahan, W. J., (1998). *Modelling English Text*. PhD thesis, University of Waikato, p. 243.

Tufiş, D., Chițu, A., (1999). Automatic Insertion of Diacritics in Romanian Texts. In Proceedings of the 5th International Workshop on Computational Lexicography COMPLEX, Pecs, Ungaria, pp. 185-194.

Witten, I.H., Neal, R.M., Cleary, J.G., (1987). Arithmetic coding for data compression. Communications of the ACM. 30(6), pp. 520-540.

Yarowsky, D., (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM, pages 88-95.

Yarowsky, D., (1999). Corpus-based techniques for restoring accents in Spanish and French texts. In Natural Language Processing Using Very Large Corpora. Kluwer Academics Publisher, pp. 99-120.

CAPITOLUL 4

MODELARE LINGVISTICĂ

O PROPUNERE DE ANALIZĂ MORFOLOGICĂ BAZATĂ PE PARADIGMELE NOMINALE

NADIA LUIZA DINCĂ

Institutul de Cercetare pentru Inteligența Artificială al Academiei Române

hnadia_luiza@hotmail.com

Rezumat

În mod tradițional, paradigmele sunt utilizate pentru a gestiona limbaje cu un sistem flexional complex.

Acest studiu propune o aplicare la morfologia limbii române a unei metode analitice relativ uzuale, folosind paradigmele nominale ca modalitate de stabilire a unei relații între membrii unei perechi morfonologice. Am considerat forma substantivului de Nominativ-Acuzativ, plural, nearticulat drept restricție şi, în acelaşi timp, identificator de paradigmă, am categorizat terminațiile de flexiune şi am impus operațiile *scad şir*, *adun şir*, *alternanțe fonetice* în interiorul regulilor lexicale şi al relațiilor morfologice.

Rezultatul este o propunere de analiză gramaticală, organizată în jurul conceptului de *paradigmă nominală*.

1. Introducere

Redundanțele existente la nivelul datelor lingvistice pot fi utilizate pentru a acoperi generalizări pertinente, în condițiile în care datele sunt reprezentate prin paradigme gramaticale. Studiul nostru propune o abordare analitică a limbii române utilizând paradigmele nominale ca modalitate de stabilire a unei relații între membrii unei perechi morfonologice.

Proiectul este organizat în trei etape:

a. **de cercetare**, în care am generat toate formele flexionare pentru un număr de aproximativ 500 de substantive de toate genurile, dupa metoda mersului invers, de la forma flexionată la cea bază: <<G.D, plural, articulat> <N.Ac., plural, articulat> <N.Ac., plural, nearticulat> <G.D., singular, articulat> <N.Ac., singular, articulat> <N.Ac., singular, nearticulat>>.

b. **de analiză**, în care am clasificat terminațiile, am numit restricțiile de Nominativ-Acuzativ, plural, nearticulat ca fiind identificatorii paradigmelor, am impus operațiile de *scad şir*, *adun şir*, *alternanțe fonetice* în interiorul regulilor lexicale şi al relațiilor morfologice. Rezultatul a fost crearea unor paradigme nominale ce folosesc o bază de terminații, una de restricții şi operația de unificare de şiruri pentru analiza morfologică.

c. **de evaluare** a paradigmelor create anterior, etapă în care am considerat aproximativ 200 de substantive în diferite forme flexionare, am identificat paradigma lor proprie şi am parcurs ordinea operațiilor indicate pe liniile acestora.

Raţiunea pentru care am preferat conceptul de *paradigmă* celui de *regulă* rezidă în considerarea paradigmei ca un construct în care relaţionează module şi sub-module, conform regulilor lexicale şi relaţiilor morfologice. Un exemplu de modul component este *obiectul operaţie*, definit prin date de tipul *terminaţie* sau *restricţie* şi prin metoda *modifică şir de caractere*, împărţită, la rândul ei, în operaţiile *scad şir*, *adun şir*, *alternanţe*. În schimb, conceptul de regulă permite ierarhizarea entităţilor componente, dar nu şi gruparea lor ca obiecte complexe.

Acest studiu continuă cu prezentarea paşilor algoritmului de analiză bazată pe paradigmă nominală prin considerarea unor substantive diferite ca informaţie flexionară, după care în următoarea secţiune vom detalia definirea formală a paradigmei. Concluzia punctează rezultatele evaluării analizei gramaticale bazate pe conceptul de *paradigmă*, iar partea finală a lucrării prezintă direcţiile viitoare de lucru.

2. Definirea formală a paradigmei

2.1 Algoritmul de deflexionare bazat pe paradigma nominală

Lucrarea a fost construită în jurul unui algoritm care are la bază o listă bine definită de reguli. Fiecare regulă conţine date exacte, de tipul *clase de terminaţii* şi *de restricţii*, dar şi operaţii, ordonate după anumite criterii, denumite sugestiv prin „scad şir", „adun şir", „alternanţe". Aceste reguli sunt realizate după principiul reprezentării bazate pe obiecte, adică prin proiectarea unor entităţi de sine stătătoare, complexe, care deţin şi parte de interfaţă, precum şi de implementare.

Procedura de deflexionare consideră următorii paşi (Figura 1):

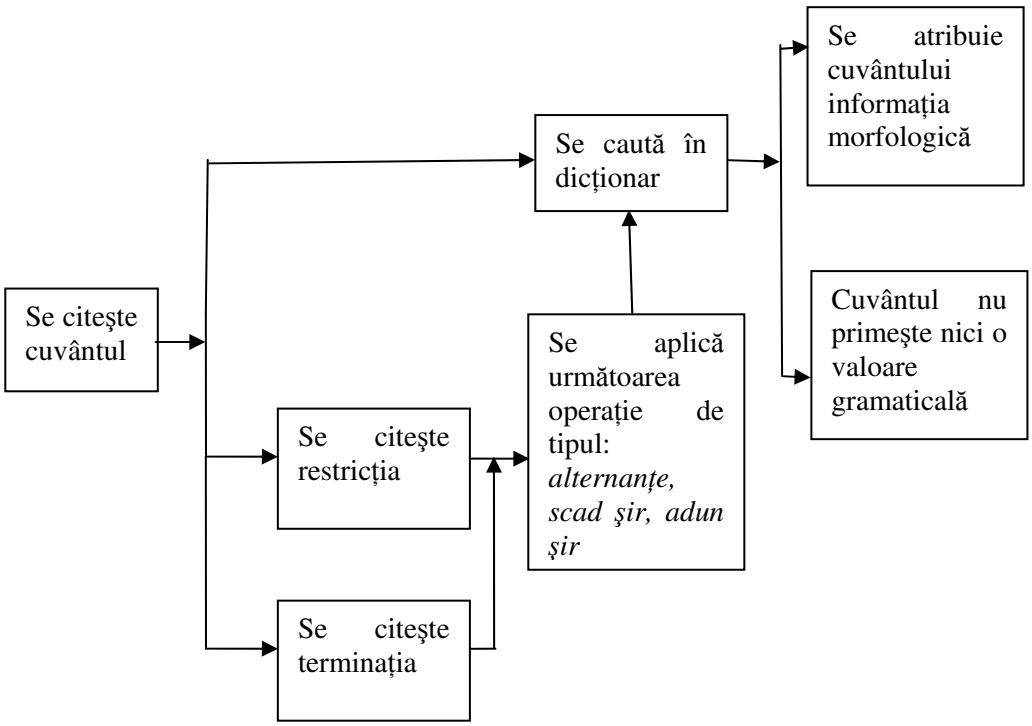


Figura 1: Paşii procesului de deflexionare

1. Se citeşte cuvântul şi se caută în dicţionar. Dacă acest cuvânt e în forma bază, atunci i se atribuie valoarea morfologică asociată.
2. Altfel, se separă şirul de caractere ce alcătuiesc cuvântul în terminaţii de până la maximum 4 caractere considerate de la ultimul spre primul, iar acestea sunt căutate în baza de reguli începand cu terminaţia-restricţie pentru Nominativ-Acuzativ. Dacă restricţia este găsită, atunci se consideră că programul de deflexionare se găseşte implicit în pasul *Nominativ-Acuzativ, plural, nearticulat* şi că pasul următor este realizarea primei operaţii specificate în coloanele formelor de Genitiv-Dativ, singular, articulat (*alternanţe, scad şir, adun şir*). Pentru substantiv am considerat terminaţie şirul de 4 litere din poziţie finală, citit recursiv ca subşir de 4 litere, de 3, de 2 şi, respectiv, de o literă.
3. Dacă şirul de intrare nu subsumează subşirul specificat de restricţie, atunci se citeşte terminaţia, se caută în celelalte clase de terminaţii corespunzătoare cazurilor Genitiv-Dativ, plural sau singular, Nominativ-Acuzativ, plural, urmând să se aplice succesiv operaţiile care fac parte din regulă.
4. Dacă, după realizarea tuturor operaţiilor impuse de fiecare regulă în parte, rezultatul este găsirea unei forme bază în dicţionar, atunci se opreşte deflexionarea şi se validează regula parcursă. Forma derivată, cea pentru care s-a creat deflexionarea, primeşte categoria gramaticală a numelui comun şi toată informaţia morfologică din câmpul pentru care s-a identificat terminaţia.

5. Altfel, dacă rezultatul deflexionării nu este o intrare de dicționar, atunci nu este validată nici regula aplicată, iar forma derivată nu primește nici o valoare morfologică. Urmează reluarea căutării în alt câmp de restricție, parcurgându-se aceiași pași, dar în cadrul altei reguli din listă până când se găsește o regulă care în final este validată.

Pentru exemplificare, fie așadar cuvântul băieți, reprezentat prin paradigma nominală de mai jos. Aceasta este specifică pentru toate numele comune având același identificator „ieți", dar și pentru cuvintele având terminațiile *lor* și *i* prezente în regulă, înaintea restricției.

(nume_băieți, băieți: [nume comun, masculin, restricție NAc pl neart #ieți],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +i -i ț<t, ie<ia –i +ului -ui -ul])

Potrivit algoritmului de deflexionare, primul pas este căutarea cuvântului în dicționar. Nu este găsit și se continuă cu al doilea pas: verificarea restricției, altfel spus a terminației de Nominativ-Acuzativ, plural, nearticulat, considerată nucleul paradigmei nominale. Citim astfel terminația formată din ultimele 4 litere din cuvântul *băieți*, iar programul identifică restricția din specificația de șir *ieți*. Se trece la prima operație care urmează citirii terminației de N-Ac, plural, nearticulat, respectiv la *alternanțele fonetice*. Paradigma cuvântului de intrare afișează două alternanțe prin care se obțin consecutiv formele: *băieti, băiati*. Regula lexicală pentru Genitiv-Dativ, singular, articulat prevede încă două operații de actualizat, *scad șir*, prin care se obține forma *băiat*, respectiv *adun șir*, pentru forma *băiatului*. Algoritmul continuă cu operația de scădere de șir, prin care se produce specificația de Nominativ-Acuzativ, singular, articulat: *băiatul*. Ultimul pas este indicat de regula lexicală pentru Nominativ-Acuzativ, singular, nearticulat și constă din nou în scădere de șir, creându-se forma *băiat*. Cuvântul este găsit în dicționar, drept pentru care paradigma este validată, iar cuvântul derivat *băieți* primește informația asociată regulii lexicale pe care a deschis-o prin unificarea terminației cu restricția.

Să luăm însă un alt exemplu, în care unificarea restricției să nu se realizeze: *munților*. Reluând algoritmul, primul pas este căutarea în dicționar pentru a verifica dacă lexemul de intrare este în forma lui bază. Nu este cazul, astfel încât parcurgem al doilea pas, citirea terminației de restricție. Nici acest pas nu este validat și înaintăm către pasul trei, citirea terminației cuvântului de intrare în maniera recursivă: *ilor*, *lor*, *or*, *r* (subșirurile de 4,3,2 și respectiv, un caracter). Programul recunoaște terminația *lor*, existentă în paradigma numelui comun *munților* ca specificație de șir pentru Genitiv-Dativ, plural, articulat, consideră acest pas ca fiind implicit și avansează spre prima operație indicată de regula lexicală Nominativ-Acuzativ, plural, articulat:

(nume_munților, munților: [nume comun, masculin, restricție NAc pl neart #nți],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +i -i ț<t, i<e +lui -lui +le -le])

Formele create prin relațiile morfologice și regulile lexicale indicate de Nominativ-Acuzativ, plural, articulat sunt, pe rând: *munți*, *munții*. În pasul pentru Nominativ-Acuzativ, plural, nearticulat există două proceduri: prima este operația de scădere de șir, cu rezultatul *munți*, a doua este verificarea restricției. Este validată subsumarea formei de restricție de către șirul de intrare și programul avansează la regula lexicală pentru

Genitiv-Dativ, singular, articulat. Pe rând, alternanţele produc: *munti*, *munte*, apoi prin adunare de şir rezultă forma: *muntelui*. Penultima regulă lexicală specificată de Nominativ-Acuzativ, singular, articulat figurează o scădere şi o adunare de şir, obţinându-se formele: *munte*, *muntele*. Ultima regulă lexicală implică, de asemenea, două proceduri: scăderea şirului, cu rezultatul *munte*, respectiv căutarea în dicţionar şi găsirea intrării lexicale *munte*. Paradigma este validată, iar cuvântul *munţilor* primeşte valorile morfologice asociate regulii lexicale pe care a deschis-o.

Să considerăm acum forma *merii* şi să urmărim în ce manieră sunt refăcuţi paşii algoritmului de deflexionare bazat pe paradigmă nominală:

Pasul 1: Se caută în dicţionar cuvântul *merii*.

Pasul 1.1.: Nu se găseşte şi se trece la pasul 2.

Pasul 2: Se verifică restricţia de Nominativ-Acuzativ, plural, nearticulat, confruntându-se cu lista restricţiilor existente în toate paradigmele nominale.

Pasul 2.1.: Nu se găseşte şi se trece la pasul 3.

Pasul 3: Se verifică dacă subşirurile *erii*, *rii*, *ii*, *i* se află pe lista terminaţiilor înregistrate în toate paradigmele nominale.

Pasul 3.1.: Subşirul *erii* nu există în lista terminaţiilor considerate intrări implicite pentru algoritm şi se trece la subşirul *rii.*

Pasul 3.2.: Nici acest subşir nu este validat, astfel încât se trece la subşirul *ii*.

Pasul 3.3.: Se identifică subşirul *ii* ca terminaţie de Nominativ-Acuzativ, plural, articulat şi se trece la prima operaţie indicată de regula lexicală următoare (Nominativ-Acuzativ, plural, nearticulat).

Pasul 4: Prin scădere de şir se obţine forma *meri*. Se repetă pasul 2.

Pasul 4.1.: Restricţia este validată şi se citeşte următoarea paradigmă:

(nume_merii, merii: [nume comun, masculin, restricţie NAc pl neart #eri],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +i -i e<ă -i +ului -ui -ul])

Pasul 5: Se parcurge regula lexicală Genitiv-Dativ, singular, articulat, în ordinea:

Pasul 5.1.: alternanţe fonetice: e<ă; Se obţine forma *mări*.

Pasul 5.2.: scad şir: -i; Se obţine forma *măr*.

Pasul 5.3.: adun şir: +ului. Se obţine forma *mărului*.

Pasul 6: Regula lexicală pentru Nominativ-Acuzativ, singular, articulat este formalizată prin operaţia de scădere de şir (*-ui*), cu rezultatul dat de specificaţia de şir *mărul.*

Pasul 7: Regula lexicală de Nominativ-Acuzativ, singular, nearticulat include operaţia de scădere de şir (*ul*), obţinându-se forma *măr*.

Pasul 8: Se caută în dicţionar cuvântul *măr*.

Pasul 8.1.: Este găsită intrarea lexicală *măr*, paradigma numelui comun masculin este validată, iar cuvântul *merii* primeşte informaţia gramaticală de Nominativ-Acuzativ, plural, articulat deoarece acest câmp a fost activat prin pasul 3.3.

Algoritmul de deflexionare bazat pe paradigme nominale este identic pentru toate tipurile de substantive comune: masculine, feminine, neutre. În fapt, aceiaşi paşi sunt urmăriţi şi pentru obţinerea formei bază în cazul adjectivelor, pronumelor demonstrative, nehotărâte, numeralului ordinal, articolului nehotărât.

2.2 Paradigma substantivului

Paradigma substantivului este exprimată prin relaţiile morfologice şi regulile lexicale de tipul *substantiv [Tipul, Şir: Proprietăţi, RL, S]*, unde Proprietăţile pot include ecuaţii de şir, iar RL şi S sunt liste ale numelor regulilor lexicale şi, respectiv, ale specificaţiilor de şir.

În exemplul următor sunt descrise paradigmele pentru substantivele masculine cu terminaţia *uri*, pentru substantivele feminine cu terminaţia *şti* şi pentru substantivele neutre cu terminaţia *oare*, toate cele trei terminaţii fiind subşiruri pentru formele de Nominativ-Acuzativ, plural, nearticulat. Prima linie completează tipul substantivului, categoria genului şi precizează terminaţia de restricţie. A doua linie ordonează regulile lexicale prin care se vor genera formele flexionate, iar ultima porneşte de la terminaţia de G.D. plural, articulat, continuă cu operaţiile de *scad şir*, *adun şir*, *alternanţe fonetice* până la ultimul câmp completat.

(nume_iepuri, iepuri: [nume comun, masculin, restricţie NAc pl neart #uri],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +i -i i<e +lui -lui +le -le])

(nume_poveşti, poveşti: [nume comun, feminin, restricţie NAc pl neart #şti],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +le -le şt<st -i +ei -ei +ea -a])

(nume_picioare, picioare : [nume comun, neutru, restricţie NAc pl neart #oare],
[GD pl art NAc pl art NAc pl neart GD sg art NAc sg art NAc sg neart]
[#lor -lor +le -le oa<o –e +ului -ui -ul])

Figura 2: Exemple de paradigme nominale

Relaţia de subsumare pe care se bazează descrierea paradigmelor defineşte o ordine parţială asupra specificaţiilor de şir, astfel încât o specificaţie S subsumează o alta S' dacă toate instanţele bază din S' sunt, de asemenea, instanţe în S. Pentru cele trei paradigme anterioare, unificarea şirurilor se produce deoarece, pe de o parte, subşirul *uri* este inclus în şirul *iepuri*, *şti* în *poveşti*, respectiv *oare* în *picioare*, iar, pe de altă parte, subsumarea este condiţionată, în realizarea ei, de regula lexicală prin care se specifică restricţia de Nominativ-Acuzativ, plural, nearticulat.

Exemplele de mai sus validează paradigmele nominale nu doar pentru substantivele ce figurează ca intrare lexicală, ci grupează toate numele comune având aceeaşi terminaţie de restricţie. Criteriul de unificare a restricţiei se respectă pentru toate paradigmele create în etapa de analiză a proiectului. Graful din figura 2 ilustrează o reţea de noduri în care terminaţiile **ete**, **etre** şi **pe** au acelaşi comportament flexionar pentru paradigma numelui comun feminin, în timp ce nucleul unifică trei segmente morfonologice distincte:

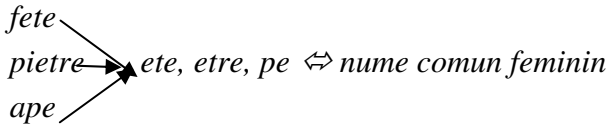


Figura 3: Nucleul morfonologic *ete, etre, pe* şi categoria realizată prin flexiune

Dacă o formă anume, d.e. *fetele*, e dată ca intrare, programul încearcă să identifice o categorie potrivită pe baza analogiei cu şirul ţintă, folosindu-se de şirurile bază deja stocate. Nucleul va fi activat de forma *fetele* deoarece, prin operaţia *scad şir*, este eliminat şirul *le*, găsit ca terminaţie în baza de date morfologice, şi se citeşte restricţia *ete*, cea care identifică, de fapt, categoria substantivului comun feminin, reprezentat prin forma bază *fată* :

Intrare: fetele
<scad şir>: [-le] fete
<verific restricţia>: citesc terminaţia de nucleu [ete]
<adun şir>: [+i] fetei
<scad şir>: [-ei] fet
<adun şir>: [+a] feta
<alternanţe fonetice>: [e/a] fata
<scad şir>: [-a] fat
<adun şir>: [+ă] fată
Ieşire: validez ete ⇔ *nume comun feminin*

Figura 4: Paşii de analiză a formei *fete*

Restricţia-nucleu este activată complet atunci când este în întregime conţinută în şirul de intrare. În exemplul *zilele*, este activat nucleul *le*, pe care programul îl găseşte în baza de date ca restricţie (Figura 4), pe când în exemplul *fetele*, şirul *le*, citit din nou ca restricţie, va genera o formă nevalidă:

Intrare: zilele
<scad şir>: [-le] zile
<verific restricţia>: citesc terminaţia ***le***
<*adun şir****>****: [+i] zilei*
<scad şir>: [-lei] zi
<adun şir>: [+ua] ziua
<scad şir>: [-ua] zi
Ieşire: validez ***le*** ⇔ *nume comun feminin pentru forma* ***zilele***

Figura 5: Identificarea şirului *le* ca restricţie pentru forma *zilele*

Intrare: fetele
<verific restricţia>: citesc terminaţia ***le***
<*adun şir****>****: [+i] *fetelei*
*<scad şir>: [-lei] *fete*
*<adun şir>: [+ua] *feteua*
*<scad şir>: [-ua] *fete*
Ieşire: nu validez ***le*** *⇔ nume comun feminin pentru forma* ***fetele***

Figura 6: Activarea parţială a nucleului pentru forma *fetele*

3. Concluzii

Studiul nostru se înscrie pe linia tradiţională a morfologiei paradigmatice, însă aduce nou conceptul de restricţie văzută ca identificator al paradigmei nominale.

Evaluarea acestei abordări s-a realizat prin analiza unor forme substantivale arbitrare, bază ori flexionate. Am urmărit paşii algoritmului şi am verificat dacă, după realizarea ultimului pas, se obţine intrarea de dicţionar potrivită. Validarea paradigmei parcurse a antrenat, de asemenea, şi atribuirea unor valori morfologice pentru cuvântul introdus în mecanismul deflexionării.

În 90% din situaţii, algoritmul a condus la o analiză morfologică validă, iar cazurile ce au înregistrat eroare au fost rafinate, pentru a nu mai intra în conflict cu alte restricţii deja existente în baza de date. S-au reparcurs paşii de analiză, iar deflexionarea a fost corectă, cuvântul primind valori corespunzatoare morfologic.

4. Direcţii viitoare de lucru

În continuarea acestui studiu ne propunem să proiectăm paşii de analiză a verbului, urmărind, de asemenea, conceptul de *paradigmă*.

Paradigma verbului este puţin mai complicată decât cea nominală din cauza formelor verbale numeroase, diversificate în funcţie de mod, timp, persoană şi număr. Ca şi în cazul paradigmei nominale, Proprietăţile includ ecuaţii de şir, iar regulile lexicale şi specificaţiile de şir descriu formele verbale tranzitive de la verbul intrare, conjugat, la cel existent în dicţionar la infinitiv, formă scurtă.

Schema logică de reprezentare a algoritmului de obţinere a formei verbale bază coincide cu schema logică iniţiată pentru substantiv, în pasul iniţial, de căutare în dicţionar, respectiv în pasul final, de validare ori respingere a paradigmei parcurse. Identificatorul de paradigmă va fi, de data aceasta forma verbală de infinitiv, iar baza de date morfologice va fi organizată pentru a include grupe de terminaţii, linii de alternanţe şi restricţii de unificare. Operaţiile ce se aplică în interiorul regulilor lexicale sunt, din nou, cele de *scădere* şi de *adunare de şir*, respectiv de *alternanţe fonetice*. Validarea paradigmei parcurse se va realiza numai în condiţiile regăsirii în dicţionar a formei lexicale obţinute în finalul analizei.

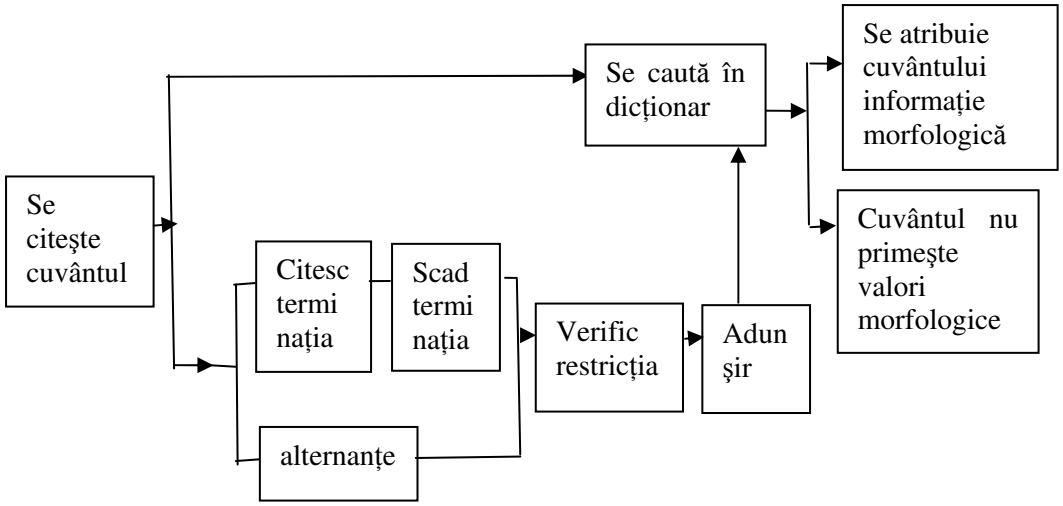


Figura 7: Schema logică de analiză a verbului

Referințe bibliografice[44]

Calder, J. (1989). Paradigmatic morphology. *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, pages: 58 – 65, University of Manchester, Institute of Science and Technology, Manchester, UK.

Koktová, E.(1985). Towards a new type of morphemic analysis. *Proceedings of the second conference on European chapter of the Association for Computational Linguistics*, Geneva, pages: 179 - 186.

Koskenniemi, K. (1983). Two-level morphology: a general computational model for word-form recognition and production. *Technical Report 11,* Department of General Linguistics, University of Helsinki..

Nerbonne, J. (1993). Feature-based allomorphy. *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, Columbus, pages: 140 -147.

Pirrelli, V., Federici, S. (1994). "Derivational" paradigms in morphonology. *Proceedings of the 15th conference on Computational linguistics*, Kyoto, pages 230 – 240.

Tufiş, D., Popescu, O. (1991). A unified management and processing of word-forms, idioms and analytical compounds. in Jurgen Kunze and Dorothy Reinman (eds.) *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*, Berlin, pages 95 – 100.

Tufiş, D.(1989). It would be much easier if WENT were GOED. *Proceedings of the fourth conference on European chapter of the Association for Computational*

[44] Motivul pentru care în interiorul lucrării nu se regăsesc trimiteri la referințele bibliografice este dat de faptul că studiile enumerate au constituit baza teoretică a formării mele, bază continuată firesc de o aplicare personală la morfologia limbii române a conținuturilor parcurse.

Linguistics, pages 145-152, University of Manchester, Institute of Science and Technology, Manchester, UK.

INDEX DE AUTORI