# Aspecte de Dezvoltare a Analizatorului de Text Nestructurat

Petic M.
Catedra de Matematică și Informatică,
Universitatea de Stat Alecu Russo, Bălți,
str. Pușkin 38, Bălți, 3100, Republica Moldova,
Laboratorul Sisteme de Programare
Institutul de Matematică și Informatică al AȘM
str. Academiei 5A, Chișinău, Republica Moldova
petic.mircea@gmail.com

Osoian E.
Catedra de Matematică și Informatică,
Universitatea de Stat Alecu Russo, Bălți,
str. Pușkin 38, Bălți, 3100, Republica Moldova

**Abstract – the article presents our approach in the elaboration of the system for processing unstructured text data in order to create a structured data output as computer linguistics resources using a lexicon of markers. First, a description of the research on the proposed topic, as well as its relation to the national and international level research is presented, being followed by the depiction of a useful to this particular research functionality - PoS Tagger for Romanian. A special section is dedicated to the algorithm to be used to elaborate our system. Finally, we describe several ways of marker lexicon completion by means of derivation.**

***Termeni cheie*** **- lingvistică computațională, tehnologii informaționale, servicii Web, text nestructurat, etichete lingvistice.**

## I. Introducere

Evoluția tehnologiilor informaționale, a rețelelor și comunicațiilor, a implicat generarea și acumularea unor volume enorme de informație textuală greu de procesat, nestructurată explicit în sens computațional, inaccesibilă sistemelor software rigide. Necesitatea critică de interpretare, categorizare, corelare și reutilizare automatizată a acestor informații a făcut specialiștii interesați să folosească cuvintele-cheie, părțile de vorbire, relațiile cunoscute (de antonimie, sinonimie etc.) dintre unele cuvinte, algoritmi euristici, pentru a extrage date relevante din texte. Într-un așa mod, pot fi detectate, interpretate și structurate informații de ordin social, cultural, de securitate, și altele.

Unul din obiectivele proiectului nostru este aplicarea mijloacelor de procesare a textelor existente cu scopul de a detecta efectiv aceste aspecte. Proiectul reprezintă un sistem software întitulat SoFTcrates, care va fi bazat pe macanisme și practici PLN contemporane.

Primind date nestructurate la intrare, ca urmare a procesării, va produce resurse lingvistice computaționale structurate, permițînd creșterea potențialului de intrerpretare a textelor create de om într-un mod sistematic și pentru mai multe scopuri, printre care: analiza contextuală a activităților utilizatorilor de soft într-un context statistic, detectare de sensuri, atitudini, emoții, aspecte de securitate.

Studiul va fi axat pe instrumente de procesare a textelor umane în limba română.

Articolul este format din cîteva părți: secțiunea 2 descrie problematica subiectului și rezultatele obținute în lume în adresarea acesteia, în special cele mai noi practici; secțiunea 3 este dedicată intrumentului PoS Tagger pentru limba română; secțiunea 4 conține descrierea algoritmului de obținere a textelor structurate pentru aplicația SOFTcrates; secțiunea 5 vizează detalii de completare a etichetelor lexicale tematice.

## II. Tendințe Actuale în PLN

Majoritatea studiilor efectuate și a aplicațiilor de procesare a limbajului natural sînt realizate în limba engleză. Din acest motiv Comisia Europeană a inițiat un număr de proiecte în suportul tehnologizării altor limbi europene decît engleza. Politica de promovare a multilingvismului prin intermediul tehnologiilor informaționale a derivat în inițierea programelor Framework Programme 7 și HORIZON 2020.

La efectuarea unui studiu sumar al soluțiilor PLN existente, devine evident faptul că majoritatea acestora depind de resursele computaționale lingvistice existente. Cu cît aceste resurse sunt mai voluminoase și coerente, cu atît mai coerente și sigure sînt rezultatele procesărilor.

Limba română treptat devine tot mai semnificativă în ceea ce ține de materiale științifico-practice din domeniul tehnologiilor informaționale. Respectiv, crește necesitatea automatizării și grăbirii acumulării de resurse computaționale lingvistice românești.

Soluționarea oricărei probleme netriviale, luînd în considerare complexitatea și tangențele cu mai multe domenii de cunoaștere, se cere a fi una complexă. Iar soluționarea problemei lingvisticii computaționale, în particular a textelor poetice, trebuie abordată prin antrenarea de metode lingvistice în combinație cu metode computațional-informaționale.

Studierea detaliată a istoricului problematicii și a situației actuale în lingvisticile română, rusă, italiană, spaniolă, franceză, sîrbă, a dus la ideea utilizării resurselor de anotare morfo-sintactică existente.

## III. Studiul Serviciilor Web Lingvistice Publice Pentru Limba Română

Numeroase intrumente de procesare a limbajului natural au fost create în România, cu scopul de a satisface necesitățile cercetătorilor locali, a utilizatorilor de soluții PLN, progresînd în baza adoptării celor mai bune practicii în domeniu. Mai jos este prezentată o descriere succintă a produsului PoS Tagger pentru limba română1 creat la Universitatea Alexandru Ioan Cuza din Iași.

Aplicația implementează un model de procesare a limbajului natural în baza extracției de informație bazate pe reguli și pe metode statistice. Regulile sînt folosite ca constrîngeri împotriva unor ambiguități dificile și sînt realizate cu ajutorul aplicației Graphical Grammar Studio, un produs software open-source, care facilitează identificarea și crearea legăturilor între token-uri și îmbinări de cuvinte care pot fi adnotate la fel ca părțile de vorbire atomare.

PoS Tagger, la fel ca majoritatea utilităților enumerate mai jos, sînt publicate în calitate de servicii web, descrise în baza specificației WDSL și vehiculate cu ajutorul protocolului SOAP [1].

## IV. Algoritmul

Algoritmul exemplificat mai jos utilizează serviciile "Romanian Part of Speech Tagger" pentru elaborarea dicționarului de semnificații poetice ale culorilor.

Primul pas în crearea dicționarului ar fi stabilirea concordanțelor cromatice - extragerea fragmentelor de text (versuri) care conțin o etichetă, un cuvînt care exprimă un spectru cromatic.

Liniile vor fi grupate și apreciate după numărul de cuvinte-cheie. Este evidentă utilizarea automatizării mecanismelor de lucru, pentru care este necesară dispunerea de text poetic în formă electronică text și de lista de cuvinte-cheie care să exprime setul cromatic. Rezultatul procesării va fi un fișier cu conținutul format din linii cu formula:

`nr1 nr2 verse`

unde `nr1` reprezintă numărul de ordine al liniei din textul poetic, `nr2` – numărul primei apariții a cuvîntului-cheie în vers și `verse` – textul versului .

Procedura normală a procesării este următoarea (vezi fig.1):

- Citire a unei linii de text într-o variabilă;
- Căutarea vreunui cuvînt care ar avea vreun corespondent în setul de culori;
- În cazul localizării măcar a unui termen cromatic - scrierea liniei în fișierul de ieșire.

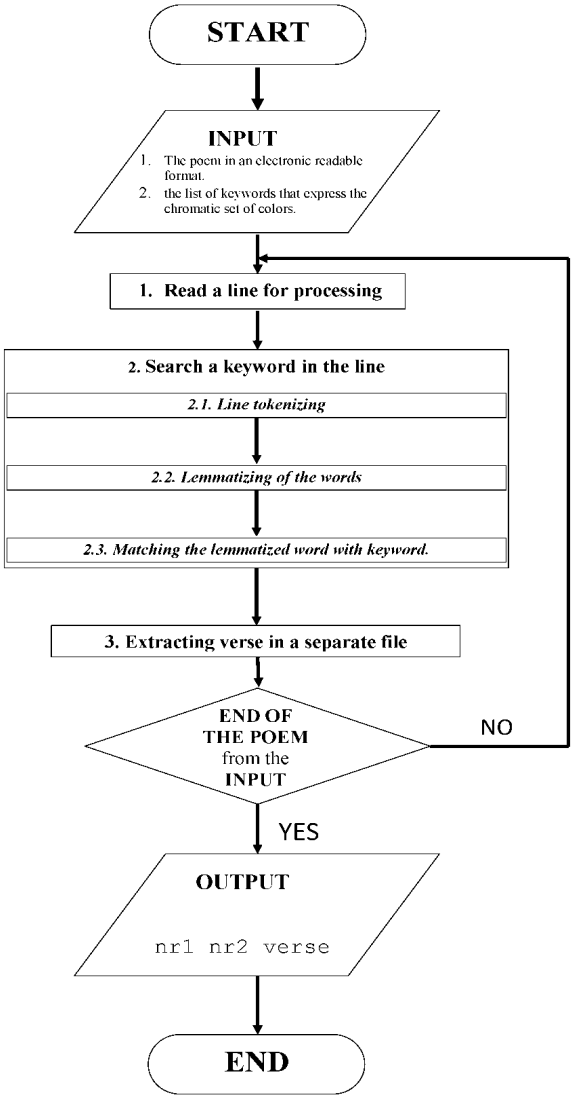


Fig.1 Algoritmul de extragere a versurilor cu etichete indicate.

Mai jos este prezentată o descriere mai detaliată a realizării pașilor enumerați.

**Citirea liniei de procesat** se realizează prin proceduri standard de citire din fișier static local sau remote sau dinamic de la consolă, componentă input sau orice alt canal informațional.

**Căutarea prin vers** a unui cuvînt cheie este realizată prin cîteva operații:

1. ***Divizarea versului*** pe termeni;
2. ***Lematizarea*** fiecărui cuvînt (extragerea lemei);
3. ***Compararea*** cuvîntului-lemă cu cuvintele-cheie și identificarea legăturilor.

Divizarea versului pe termeni se efectuează simplu prin funcții standard de prelucrarea a șirurilor de caractere.

Lematizarea este un proces mult mai complex, cel mai complicat și important din în treg algoritmul. Rezolvarea este simplificată prin consumul de servicii facilitatoare (PoS

[1] http://nlptools.infoiasi.ro/WebPosRo/ - Simionescu Radu, UAIC Romanian Part of Speech Tagger, 2011

Tagger for Romanian[2]), creat în cadrul grupului ştiinţific NLP din Facultatea de Informatică a Universităţii "A. I. Cuza".

Serviciul web prestat întoarce un text anotat, lematizat şi structurat ca răspuns la o cerere împreună cu textul nenormalizat. Interacţiunea cu PoS Tagger este uşor automatizabilă datorită uşurinţei de integrare cu alte sisteme software prin protocoale şi limbaje de specificaţii (SOAP, WSDL) [2].

Procesul de comparare a şirurilor de caractere va fi realizat la rîndul său cu ajutorul unor biblioteci de manipulare complexă a şirurilor de caractere.

**Extragerea versului ales într-un fişier aparte** se efectuează la fel prin funcţii standard, pentru fiecare cuvînt-cheie se va crea un fişier aparte. Chiar cu nivelul de precizie înalt al automatizării (circa 96.6%) detectării expresiilor cromatice, este necesară etapa de analiză şi validare a rezultatelor de către operatori umani. Această prelucrare este prestată de lingvişti profesionişti.

## V. Completarea Automată a Lexiconului de Etichete

Particularităţile mecanismelor de derivare morfologică (? morfologiei derivative) ajută la extinderea resurselor lexicale fără informaţie semantică. Mai mult, există mecanisme similare pentru mai multe limbi europene (engleză, franceză, spaniolă, română) sau chiar slavone (rusă, ucraineană ş.a.). Abordările şi mecanismele prezentate în lucrare au fost studiate pe exemple din limba română, dar nu sunt limitate la lexicul acesteia.

### *A. Substituţia afixelor*

Idea a fost o inspiraţie din morfologia sîrbă(?Serbia)[3], unde derivativele generate în baza unei rădăcini au sensuri previzibile, de exemplu modificarea genului în substituţie de prefix e. g., munci*tor* ↔ munci*toare*, modificare de conotaţie în substituţie de prefix e. g., *ante*belic ↔ *post*belic.

Substituţia afixelor este un proces larg răspîndit printre diferitele limbi ale lumii, (spaniolă: amortizar-amortizable, franceză: revoir-prevoir, rusă: прочитать-дочитать etc.

Într-un caz general, pentru substituţie de sufixe, fie $x_1$ un cuvînt de forma $x_1=\omega\alpha_1$, unde $\alpha_1$ e sufix. După substituţia $\alpha_1 \rightarrow \alpha_2$ se va obţine cuvîntul $x_2=\omega\alpha_2$, ex., corig*enţă*-corig*ent*. Îm cazul substituţiei de prefixe, fie $x_1$ un cuvînt de forma $x_1=\alpha_1\omega$, unde $\alpha_1$ este un prefix. După substituţia $\alpha_1 \rightarrow \alpha_2$ se va obţine cuvîntul $x_2=\alpha_2\omega$, unde $x_2$ este derivativul obţinut, ex., *în*chide-*des*chide [4].

Din regulile de mai sus a fost conceput un algoritm de examinare a cuvintelor în lexic şi a substituţiei de afixe în cazurile care corespund categoriilor stabilite de regulile de mai sus.

### *B. Modelele formale*

Modelele formale ale regulilor de derivare reprezintă baza de generare a cuvintelor derivate cu un nivel înalt de certitudine. O abordare similară în morfologia derivativă se întîlneşte în limba franceză [5]. Dar în timp ce sistemul de derivare franceză funţionează regulat cu doar 3 sufixe (-able,-ite,-is (er)), pentru română studiul se extinde pentru la 3 prefixe (ne-, re-, in-/im-) şi 2 suffixe (-re,-iza).

***Reguli de prefixare:***

*re-* $[\omega]_{inf} \rightarrow [re\ [\omega]_{inf}]_{inf}$

*ne-* $[\omega'\beta]_{adj} \rightarrow [ne\ [\omega'\beta]_{adj}]_{adj}$

$\beta \in$ {*-tor, -bil, -os, -at, -it, -ut,-ind, -înd* }

*in-/im-*$=\gamma$ $[\omega'\beta]_{adj} \rightarrow [\gamma[\omega'\beta]_{adj}]_{adj}$ $\beta \in$ {*-bil, -ent, -ant*}

***Reguli de sufixare:***

*-re* $[\omega]_{inf} \rightarrow [[\omega]_{inf}\ re]_{subst}$

-iza $[\omega'\beta\alpha]_{adj} \rightarrow [[\omega'\beta]_{adj}\ iza]_{inf}$

### *C. Proiecţia derivativelor*

Proiecţia derivativelor reprezintă procedeul de formare de cuvinte prin prefixare de termeni sufixaţi cu rădăcină comună. Potrivit cercetătorilor spanioli, verbul speniol *amortizar* poate fi derivat cu prefixul *des-* obţinînd *desamortizar*. La fel, *amortizar* poate fi derivat cu sufixele *–cion* and *–able*. Şi derivativul din prefixul *des-* poate fi la rîndul său derivat cu sufixele *–cion* and *–able*. Ipoteza constă în faptul că derivativele pot moşteni sau proiecta derivative cu sufixele rădăcinii prefixarea căreia a fost realizată[6].

Generalizînd cele menţionate, facem concluzia că este posibil de prezentat în mod formal mecanismul derivaţional pentru limba română, astfel încît acesta să poată fi la necesitate reutilizat. Vom considera în limba română cuvîntul $\omega$, $\alpha$ - prefixul, şi $\beta$ - sufixul acestuia. În cazul dat, următoarea relaţie are loc [4]:

$$(\omega \rightarrow \alpha\omega) \wedge (\omega \rightarrow \omega\beta) \Rightarrow (\omega \rightarrow \alpha\omega\beta),$$

spre exemplu, (a lucra → a *pre*lucra) ∧ (a lucra → lucr(a)*ător*) ⇒ (a lucra → *pre*lucr(a)*ător*);

$$(\omega \rightarrow \alpha\omega) \wedge (\omega \rightarrow \alpha\omega\beta) \Rightarrow (\omega \rightarrow \omega\beta),$$

spre exemplu, (a capitula→ recapitula) ∧ (a capitula →recapitulaţie) ⇒ (a capitula→capitulaţie)

$$(\omega \rightarrow \alpha\omega\beta) \wedge (\omega \rightarrow \omega\beta) \Rightarrow (\omega \rightarrow \alpha\omega),$$

spre exemplu, (a centraliza → *des*centraliz*ator*) ∧ (a centraliza → centraliz*ator*) ⇒ (a centraliza → *des*centraliza);

Examinînd cuvintele din lexic şi verificîndu-le în baza relaţiilor de mai sus, a fost elaborat un algoritm original care să genereze derivative prin proiecţie de afixe.

### *D. Constrîngeri derivaţionale*

Acolo unde nu există vreun model determinat universal, conform căruia să fie generate derivativele, este necesar de stabilit unele stări iniţiale în baza regulilor numite constrîngeri derivative. Constrîngerile derivaţionale comune sunt: părţile de vorbire, inflexiunile de număr, gen, caz, literele care preced/succed sufixele. Constrîngerile derivaţionale, astfel, reprezintă scheme cu parametri, care reduc rădăcinile şi afixele claselor cu scopul formării de derivative. Ex. funcţiile de forma:

f: {cuv, pdv, mod, sla, fgw, mvca} → derivative

unde *cuv* este cuvîntul de derivat, *pdv* – partea de vorbire a *cuv*, *mod* – modelul de derivare, *sla* – setul de cuvinte pentru ataşarea afixelor, *gfc* – grupul de flexiune(?flection) al *cuv*, *mavc* – modificări şi alternări de vocale şi consoane[4].

[2] http://nlptools.infoiasi.ro/WebPosRo/ - Simionescu Radu, UAIC Romanian Part of Speech Tagger, 2011

Ca urmare a examinării regulilor de mai sus, a fost creat un algoritm de generare a derivativelor prin aplicare de constrîngeri de derivare[7].

Drept exemple de generare de derivative prin constrîngeri pot servi cazurile de mai jos, generare automată

Ca un exemplu de generarea derivatelor prin constrîngeri poate servi derivarea automată a cuvintelor cu prefixul *des-* și sufixele *-bil* și *-ime*.

*f*: {*a spinteca*, *verb*, des<verb>, ...*s*..., V14, evitarea dublării consoanei } → *de*(*s*)spinteca.

*f*: {*a programa*, *verb*, <verb>bil-itate, ...*a*..., V201, ... } → programa*bil*itate

*f*: {*crud*, *adjectiv*, des<adjectiv>, ..., A3, alternanța consonantică d - z } → cru(d)z*ime*.

Iată de ce constrîngerile necesare pentru procesul de generarea automată, nu depind de tipul afixului, dar la fel de valoarea prefixului sau sufixului, mai mult ca atît, fiecare limbă își are particularitățile proprii in procesul de derivare.

## VI. Concluzii

Sistemul SoFTcrates va prezenta o contribuţie substanţială în domeniul produselor soft de procesare a limbajului natural, ca produs de sinteză, intermediere și perfecționare a unor instrumente elaborate anterior. Prin urmare, aceasta va permite o utilizare mai coerentă a mecanismelor inteligente descrise, va facilita accesul la ele a unui grup mai mare de utilizatori, va contribui la formarea de noi competenţe privind utilizarea soluțiilor din acest domeniu. Și, nu în ultimul rînd, va facilita valorificarea unor noi direcţii de cercetare de către tinerii specialiști în produse software de acest gen prin intermediul informării, stimulării interesului și facilitării accesului studenților la problemele, resursele și mecanismele ce fac parte din domeniul procesării limbajului natural.

Implementarea sistemului soft va implica un suport reciproc, colaborare și schimb de experienţă între mai multe grupuri de specialiști, atît în țară cît și din afară, pentru a obține mai multe idei, informații și tehnologii lingvistice pentru atingerea scopului proiectului. Instrumentul va fi util diferitor utilizatori, dar în special filologilor în lucrul lor cu textele literare în scopuri de cercetare.

## Contribuții

Articolul este scris ca parte a proiectului " Elaborarea unui sistem de prelucrare a textelor cu structură neomogenă" susținut de Consiliul Suprem pentru Ştiinţă şi Dezvoltare Tehnologică al A.Ş.M.

## Bibliografie

[1] Simionescu R. Hybrid POS Tagger. In: Proceedings of "Language Resources and Tools with Industrial Applications" Workshop (Eurolan 2011 summerschool), 2011.

[2] Petic M., Raciula L., Computer Based Identification of Lines with Romanian Chromatic Words from Poems, In: Electrotechnic and Computer Systems Journal, № 13 (89), 2014, Section Systems of Artificial Intelligence, 2014, Odessa, pp. 114-119

[3] Duško V., Krstev C. Derivational Morphology in a E-Dictionary of Serbian. In: Zygmunt Vetulani (ed.), Proceedings of the 2nd Language & Technology Conference. Poznan, Poland, 2005, p. 139-143.

[4] Petic M. Lexical derivation approaches for functional extention of computational linguistic resources. In: Proceedings of the 8th International Conference "Linguistic Resources and Tools for processing of the Romanian language" 8-9 december 2011, 26-27 april 2012. Bucharest, Editura Universitatii "Alexandru Ioan Cuza" Iasi, pp. 29-38

[5] Fiammetta N., Dal G. GéDériF: Automatic generation and analysis of morphologically constructed lexical resources. Second International Conference on Language Resources and Evaluation (LREC). Athens, Greece, May 31 – June 2, 2000, p. 1447–1454.

[6] Santana O., Perez J., Carreras F. and Rodriges G. Suffixal and Prefixal Morpholexical Relationships of Spanish, Lecture Notes in Artificial Intelligence, Ed. Springer-Verlag, 2004, pp. 407-418.

[7] Petic. M. Computational linguistic resources interoperabilit in automatic lexical derivation. In: Proceedings of the International Conference "Human, Computer and Communication" HCC2013 28-29 May 2013. Lviv, Ukraine, pp. 25-28.