Generator semi-automat de fişiere de antrenament pentru Conditional Random Fields, cu etichetare morfologică pentru limba română

Radu Răzvan SLĂVESCU, Marcel BUIAN, Adrian GROZA
Universitatea Tehnică din Cluj-Napoca
{Radu.Razvan.Slavescu, Marcel.Buian, Adrian.Groza}@cs.utcluj.ro

Ioana BĂRBĂNȚAN
Recognos Romania
Ioana.Barbantan@recognos.ro

***Abstract* — Se prezintă un sistem care să asiste generarea unui fişier de intrare pentru implementarea CRF++ a modelului Conditional Random Fields. CRF++ are nevoie de un astfel de fişier pentru a construi un model care să permită detectarea opiniilor şi a aspectelor de interes pentru utilizatorii dintr-un anumit domeniu. Generatorul este implementat sub forma unui editor intuitiv şi flexibil, capabil să ofere mai multe facilităţi avansate. Astfel, se oferă posibilitatea de etichetare morfologică pentru limba română cu un nivel de precizie ridicat, chiar şi în cazul scrierii fără diacritice, prin dezvoltarea unei implementări Hidden Markov Model existente. De asemenea, este posibilă etichetarea automată a unor coloane, pe baza importului datelor din fişiere. Se permite clonarea propoziţiilor pe bază de sinonime, ceea ce permite mărirea rapidă a setului de date de antrenament pentru CRF++. Se demonstrează calitatea soluţiei prin măsurători cantitative de acurateţe şi prin prezentarea unor exemple ilustrative de rezultate.**

***Index Terms* — Conditional Random Fields, feature detection, Hidden Markov Model, opinion mining, Part-of-Speech Tagging for Romanian**

I. INTRODUCERE

Apariţia siteurilor în care utilizatorii îşi pot împărtăşi opiniile referitoare la experienţele avute cu un anumit produs, serviciu etc. precum şi a blogurilor a dus la disponibilitatea unui volum de informaţii care în urmă cu un deceniu sau două ar fi părut de neconceput. Analiza acestora capătă în ultima vreme un interes tot mai mare din partea comunităţii ştiinţifice. Scopul urmărit este de a depista care sunt opiniile referitoare la un anumit subiect, dar şi care din caracteristicile ("features") acestuia au captat atenţia diverşilor utilizatori. Apare deci necesitatea de a dezvolta modele care să detecteze opiniile publicului cu privire la diferite subiecte, exprimate în diferite limbi (de exemplu, română), la diferite nivele de granularitate, încât să poată oferi o percepţie corectă asupra acestora.

O soluţie la această problemă este bazată pe modelul Conditional Random Fields (CRF). Aşa cum se arată în [1], acesta poate fi folosit pentru detectarea cu bune performanţe a trăsăturilor diferitelor produse menţionate în recenziile utilizatorilor. Pe baza unor fişiere de antrenament etichetate manual, se generează un model care ulterior se poate folosi la etichetarea unor texte noi. Acurateţea abordării depinde puternic de dimensiunile şi calitatea fişierelor de intrare ale programului care implementează modelul CRF. Construirea şi modificarea acestor intrări este însă o sarcină laborioasă şi mare consumatoare de timp.

Elaborarea articolului a fost susţinută prin Proiectul de Cooperare Bilaterală România-Moldova intitulat "ASDEC: Argumentare Structurată pentru Decizii cu Constrângeri Normative" şi prin proiectul PN-II Cecuri de Inovare al UEFSCDI România pentru susţinerea inovării in Intreprinderi Mici şi Mijocii intitulat "LELA-Sistem de Recomandare Colaborativă în Domeniul Turistic folosind tehnologii din Semantic Web şi analiza de texte în limba română". Adrian Groza a fost susţinut prin Proiectul Intern UTCN intitulat "Green VANETS".

Lucrarea de faţă prezintă un generator semi-automat de fişiere de antrenament menit să asiste agentul uman la această sarcină, inclusiv prin etichetarea morfologică (Part-Of-Speech tagging) automată a cuvintelor din textele de analizat. Informaţia referitoare la partea de vorbire în care poate fi încadrat un cuvânt este importantă pentru sarcina generală a detectării trăsăturilor la care se referă opiniile exprimate [1]. Problema se complică în cazul limbii române, unde este dificil de găsit un program open source performant de etichetare morfologică. Soluţia dezvoltată pentru etichetarea morfologică a cuvintelor din română reprezintă una din contribuţile prezentului articol.

Pe lângă etichetarea morfologică a recenziilor, sistemul poate eticheta automat câmpurile din anumite coloane ale intrării CRF pe baza importării datelor dintr-un set de fişiere existente. Această facilitate este utilă pentru a permite transmiterea către CRF a unor informaţii de natură semantică, cum ar fi apartenenţa unui cuvânt la o anumită categorie (de exemplu, "Chişinău"“ este un "oraş"). Generatorul permite totodată clonarea propoziţiilor pe bază de relaţii de sinonimie între cuvinte, ceea ce are drept rezultat mărirea rapidă a setului de date de antrenament pentru CRF. După ştiinţa noastră, un astfel de generator nu este disponibil în momentul de faţă; soluţiile adoptate pentru implementarea unui generator având facilităţile descrise anterior constituie principala contribuţie a acestui articol. Cu excepţia părţii de etichtare morfologică, funcţionalităţile oferite sunt independente de limba în care sunt scrise propoziţiile investigate.

Lucrarea este organizată în felul următor. Secţiunea II schiţează o soluţie bazată pe CRF pentru problema detectării trasăturilor produselor menţionate în recenzii. Secţiunea III prezintă un editor care permite generarea automatizată a fişierelor de intrare pentru CRF. Câteva exemple ilustrative sunt prezentate în Secţiunea IV. Secţiunea V prezintă concluziile şi posibilele dezvoltări.

II. DETECTARE DE TRĂSĂTURI CU CRF

Lucrarea [1] propune o metodă de determinare a opiniilor legate de un anumit produs la nivelul caracteristicilor sale ("features"), cum ar fi opiniile privitoare la calitatea imaginii produse de un aparat foto mai degrabă decât la aparat în întregul său. Această metodă are la bază modelul CRF. Pentru problema etichetării unei secvențe W de cuvinte, putem privi un CRF ca o distribuție de probabilitate condiționată asupra unei secvențe T de etichete, notată p(t|w). Problema care se dorește a fi rezolvată este găsirea lui $\text{argmax}_t p(t|w)$. Subliniem că secvența T nu este formată neapărat din etichete morfologice; de exemplu, ea poate fi o secvență de etichete "EADFB" și "O" după cum cuvântul de etichetat reprezintă sau nu o noțiune legată de domeniul "Eating and Drinking (EAD)". Din motive de spațiu, vom îndruma cititorul spre [2] pentru o tratare detaliată a CRF.

Metoda din [1] se concentrează pe opinii scrise în limba engleză. Pentru experimentele necesare adaptării sale la limba română, s-a ales la CRF++, o implementare open source a CRF[1]. Aceasta permite definirea la un nivel mai înalt a funcțiilor implicate în calcularea lui p(t|w), ceea ce a permis o flexibilitate mai mare. CRF++ are nevoie de un fișier de intrare cu secvențe de cuvinte gata etichetate, într-un format standard, pe baza căruia va construi un model folosit ulterior la etichetarea altor secvențe de cuvinte. Întregul flux este rezumat în Figura 1 și constă din 2 faze: antrenare (TRAIN), în care se generează un model pe baza exemplelor furnizate, și exploatare (TEST), în care modelul e folosit la etichetarea unor propoziții noi.

În faza de antrenare, se pornește de la un fișier cu o structură fixă pe n coloane, care conține un set de propziții etichetate. Un exemplu de astfel de fișier este dat în Tabelul I. Pe fiecare linie se găsește câte un cuvânt din propozițiile de antrenament, urmat de un set de informații care ar putea fi utile în generarea etichetelor care ne interesează. Astfel, prima coloană conține cuvintele propriu-zise, iar următoarea conține etichetele morfologice corespunzătoare. Coloana 3 marchează faptul că "Toulouse" este o instanță a lui EAD (clasă care grupează locurile în care se mănâncă și bea); ultima coloană afirmă că "ambianța" este o trăsătură a unui loc de tip "EAD" care este de interes pentru un anumit autor de recenzii. Această ultimă coloană conține de fapt etichetele pe care CRF++ le învață și cu care el va eticheta textele în faza de exploatare, în scopul indicării altor trăsături ale obiectelor de tip EAD care au fost socotite interesante de către autorii recenziilor.

TABELUL I. EXEMPLU DE FIȘIER DE INTRARE PENTRU CRF++

La	PRP	O	O
restaurantul	SCO	O	O
Toulouse	SPR	EADB	O
ambianta	SCO	O	EADFB
este	VRB	O	O
una	PNE	O	O
detasata	ADJ	O	O

Pe lângă acest fișier, în faza de antrenare este necesar și un fișier template care specifică legăturile posibile pe care CRF++ ar trebui să le ia în calcul la generarea modelului. O astfel de legătură ar putea fi descrisă astfel: "în cazul în care cuvântul curent este un substantiv comun (SCO), iar cuvântul precedent este un substantiv propriu (SPR) din categoria EAD, atunci cuvântul curent trebuie luat în calcul la generarea modelului privind EAD".

În faza de testare, modelul construit în prima fază este furnizat CRF++ împreună cu un fișier de intrare cu (n-1) coloane. Fișierul de intrare are o structură identică cu cea a fișierului de propoziții furnizat în faza de antrenare, cu singura deosebire că acum ultima coloana nu există, ea urmând să fie generată de către CRF. Restul coloanelor fișierului de intrare trebuie de asemenea furnizate CRF++.

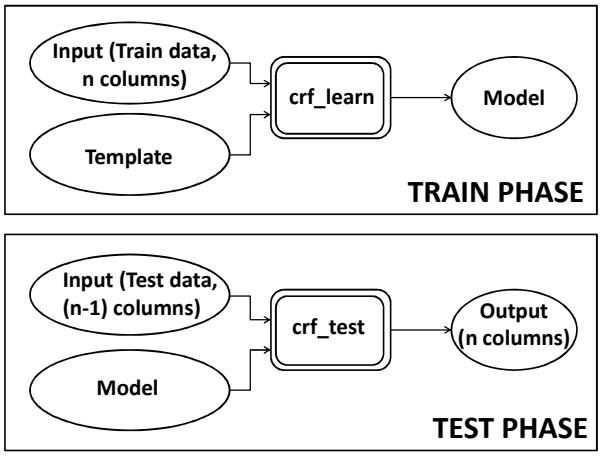

Figura 1. Rolul fișierelor de intrare pentru CRF

III. EDITAREA AUTOMATIZATĂ A FIȘIERELOR DE INTRARE CRF

Generatorul, a cărui interfață este prezentată în Figura 2, asistă utilizatorul la producerea fișierelor de intrare pentru ambele faze. Secțiunea de față descrie principalele sale funcționalități.

O primă problemă pe care acest generator trebuie să o rezolve este determinarea automată a părții de vorbire corespunzătoare fiecărui cuvânt al unei propoziții (propoziția se introduce în câmpul de sus al ecranului "Parse text", apoi se apasă butonul Parse; vezi Figura 2).

O abordare posibilă ar fi folosirea Taggerului dezvoltat la Universitatea "Alexandru Ioan Cuza" din Iași [3]. Acesta combină un set de reguli și un model statistic (modelul entropiei maximale [4]) pentru a genera etichetele morfologice și este expus sub formă de serviciu web. Întrucât proiectul nostru se dorea să nu depindă de implementări proprietare sau de servicii web externe, am decis să apelăm la o altă soluție.

Ea pornește de la implementarea existentă a unui POS tagger pentru limba română[2] bazat pe Hidden Markov Model [5]. Acest sistem se bazează pe un lexicon și un set de trigrame (secvențe de 3 etichete morfologice consecutive prezente în diferite propoziții). Este capabil să lucreze atât pe bază de digrame cât și pe bază de trigrame. În primul caz, o stare a modelului Markov este constituită dintr-o singură etichetă morfologică și aceasta este produsă ca rezultat în momentul în care se atinge starea

[1] Disponibilă la http://crfpp.googlecode.com/svn/trunk/doc/index.html

[2] http://romanian-pos-tager.googlecode.com/svn/trunk

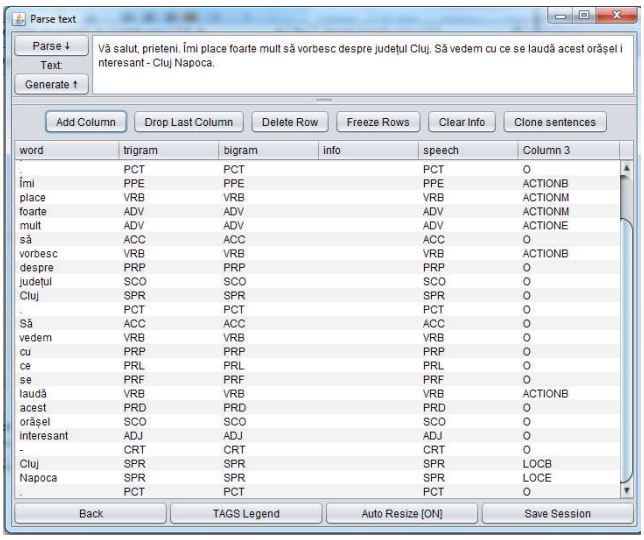

Figura 2. Interfața generatorului

respectivă. În cel de-al doilea, o stare este reprezentată de o pereche de etichete morfologice consecutive; rezultatul corespunzător unei stări este dat de eticheta aflată pe poziția 3 în trigrama care are pe pozițiile 1 și 2 etichetele ce constituie starea curentă.

Date fiind un text, sub forma unei secvențe W de cuvinte se dorește producerea unei secvențe T de etichete morfologice corespunzătoare, astfel încât să se maximizeze probabilitatea p(T|W), ceea ce, folosind regula lui Bayes, revine la maximizarea valorii produsului p(T)*p(W|T). Pentru cazul digramelor, p(T) se calculează ca un produs al unor termeni de tip $p(t_i|t_{i-1})$. Aici, $p(t_i|t_{i-1})$ se obține ca raport între frecvența unei digrame și cea a primului ei element. Termenul p(W|T) este un produs de factori de tip $p(w_i|t_i)$, cu $p(w_i|t_i)$ calculat ca raportul dintre frecvența cazurilor în care cuvântul w_i are eticheta t_i și cea a digramelor care îl au pe t_i pe poziția a doua. Pentru cazul trigramelor formulele sunt analoage. Având aceste valori pentru probabilități, calculul secvenței celei mai probabile de etichete se face cu algoritmul lui Viterbi [6].

După rezolvarea unor probleme minore legate de implementarea originală, au fost importate în lexicon un număr de intrări obținute din diferite bloguri studiate de noi, precum și o parte din lexiconul oferit de [3]. În același timp, au fost ajustate valorile frecvențelor de apariție ale diferitelor intrări în lexicon, respectiv ale diferitelor trigrame. Dimensiunea lexiconului a crescut de la circa 14.000 de intrări la aproximativ 800.000 de intrări (incluzând aici diferite forme flexionare, nume proprii etc.). Pentru problema lipsei diacriticelor, s-a implementat o căutare flexibilă, astfel încât cuvintele negăsite în lexicon sunt modificate succesiv ca să conțină diacriticele necesare (de exemplu, dacă nu se va găsi în lexicon cuvântul "las", se va căuta "lăs", apoi "laș" etc.).

Pentru robustețe, am folosit atât etichetarea pe bază de trigrame, cât și de digrame, cu evidențierea eventualelor erori. Coloana "speech" conține de fapt eticheta morfologică finală. Coloana "info" semnalează existența unei neconcordanțe între rezultatele celor două metode sau dacă vreuna din ele a produs un rezultat invalid.

Importarea de date din fișiere existente este și ea posibilă (butonul "Add column"), oferind un mecanism de implementare a unei semantici simple pentru cuvinte. Ca exemplu, să considerăm problema detectării polarității cuvintelor. O resursă pentru limba română extrem de utilă în cazul nostru a fost [7]. După filtrarea cuvintelor componente și adăugarea diferitelor forme flexionare, am obținut un fișier de cuvinte cu polaritate pozitivă, respectiv negativă. Sistemul prezentat în această lucrare este capabil să genereze o coloană de etichete pentru cuvintele din propozițiile analizate pe baza apartenenței acestora la fișierul respectiv (de exemplu, se poate adăuga automat eticheta "POSB" ("Positive-Begin") în dreptul cuvântului "minunat", cu condiția ca acest cuvânt să apară în fișierul de cuvinte pozitive și să se fi decis importarea acestuia). Pe imaginea din Figura 2, se observă, în ultima coloană, astfel de etichetări: etichetele "LOCB" și "LOCE" ("Location-Begin", respectiv "Location-End") semnalează faptul că perechea de substantive proprii "Cluj Napoca" formează de fapt numele unui loc ("Location"). Sufixele "B", "E", "M" sunt adăugate automat și semnalează începutul, mijlocul și sfârșitul entității. Într-o coloană adăugată se poate importa un singur fișier; setul de etichete din coloană este însă arbitrar de mare și este stabilit de utilizator. De exemplu, putem avea un fișier numit Polar.txt, cu 6 linii, care conține adjective cu polarități pozitive, respectiv negative, precedate de etichetele POS, respectiv NEG. Dacă in acest fișier avem liniile următoare: POS; bun; gustos; (linie goală); NEG; rău; și îl importăm în coloana 3, atunci, ori de câte ori într-un text apare cuvântul "bun", "gustos", sau "rău", în coloana 3 se va trece în dreptul lui eticheta "POSB", "POSB", respectiv "NEGB".

Clonarea propozițiilor (butonul "Clone Sentences") pe baza unui model existent și a unui set de sinonime simplifică mult procesul de construire a unui fișier de antrenament suficient de cuprinzător. Procesul are la bază un set de sinonime și o propoziție model, de exemplu "Ana prețuiește cărțile". Dacă avem setul de sinonime "prețuiește-apreciază", "cărțile-volumele" (într-un fișier text sau o bază de date MySql), sistemul va genera propozițiile-clonă "Ana apreciază cărțile", "Ana prețuiește volumele", "Ana apreciază volumele", ceea ce ar putea fi util la îmbunătățirea performanțelor modelului CRF.

Interfața grafică permite utilizatorului corectarea manuală a valorii oricărui câmp din tabel. Generatorul implementează de asemenea conceptul de sesiuni, oferind posibilitatea gestionării mai multor fișiere de intrare.

IV. REZULTATE OBȚINUTE

Modulul de etichetare morfologică al aplicației a fost testat pe un corpus adnotat manual, obținut dintr-o selecție de bloguri publice din domeniul turistic. Tabelul II prezintă un fragment din acest corpus, care constă din propozițiile folosite ca intrare, cu un cuvânt pe rând (coloana 1), apoi eticheta morfologică stabilită manual ("standardul de aur") și apoi cea propusă de program. Precizia obținută, calculată pe baza concordanțelor între etichetele corecte și cele produse de acest modul a fost de aproximativ 95% (față de circa 70% folosind codul de la care s-a pornit proiectul).

Etichetarea a dat rezultate bune chiar și în cazul textelor scrise fără diacritice. Ca exemplu, fraza "Ana se scoală apoi se duce la școală" este etichetată corect chiar și când este scrisă fără diacritice și există o ambiguitate între

TABELUL II. CONCORDANȚA ETICHETELOR STABILITE MANUAL ŞI AUTOMAT

Cuvânt	**Manual**	**Program**
La	PRP	PRP
Trattoria	SCO	SCO
Pinetta	SPR	SPR
pizza	SCO	SCO
este	VRB	VRB
delicioasa	ADJ	ADJ
.	PCT	PCT
…	…	…
In	PRP	PRP
Cluj	SPR	SPR
soarele	SCO	SCO
straluceste	VRB	VRB
inca	ADV	ADV
de	PRP	PRP
dimineata	*ADV*	*SCO*
.	PCT	PCT

"scoala" în sens de "scoală"/VRB, respectiv de "şcoală"/SCO. Precizia etichetării în acest caz a fost cu aproximativ 1.5 procente mai mică decât în primul caz. Evaluarea s-a făcut după aceeaşi procedură, pe acelaşi corpus, însă cu diacriticele substituite (ț/t etc.).

Pentru a ilustra aplicabilitatea generatorului de fişiere, considerăm următorul exemplu simplu. Pornim de la un fişier etichetat morfologic de către taggerul nostru şi de la un set existent de fişiere cu informaţii semantice. Acestea din urmă constau din fapte precum "Trattoria Pinetta" este o instanță a lui EAD (de unde etichetele EADB/EADE) sau că "pizza" este o proprietate a instanțelor EAD (de unde eticheta EADFB: EAD feature). Un cuvânt C etichetat cu EADFB indică relația "x are C", de exemplu "x are pizza" sau "x are ambianță". Modelul construit de CRF++ pe baza fişierului generat de programul nostru va servi la etichetarea altor texte; de pildă, pentru fraza "Am fost la restaurantul Baraka unde atmosfera ne-a surprins iar bresaola a fost excelentă", se produce ieşirea prezentată în Tabelul III. CRF++ a etichetat cuvântul "bresaola" cu "EADFB", deşi acest cuvânt nu figura în fişierul de antrenament. Singura informație din lexicon despre cuvântul "bresaola" este că e un substantiv comun. Modelul construit de CRF++ depistează că "bresaola" este un EADF. Aşadar fişierul produs de generator e folosit la construirea unui model capabil să descopere noi astfel de relații. Generarea rapidă de fişiere de antrenament pentru CRF++ ajută la creşterea acurateții modelelor construite de acesta.

TABELUL III. UN EXEMPLU DE FIŞIER DE IEŞIRE LA APELAREA LUI CRF_TEST

Am	VAU	O	O
Fost	VPA	O	O
la	PRP	O	O
resturantul	SCO	EADB	O
Baraka	SPR	EADB	O
unde	ADV	O	O
atmosfera	SCO	O	EADFB
ne	PPE	O	O
-	CRT	O	O
a	VAU	O	O
surprins	VPA	O	O
iar	CCO	O	O
bresaola	SCO	O	**EADFB**
a	VAU	O	O
fost	VPA	O	O
excelenta	ADJ	O	O

V. CONCLUZII SI DEZVOLTĂRI ULTERIOARE

S-a prezentat un sistem care să ajute utilizatorii la generarea de fişiere de intrare pentru CRF++. Acesta este capabil să eticheteze morfologic propozițiile conținând opiniile utilizatorilor cu o precizie suficient de mare încât să nu impieteze asupra modelului CRF generat. S-a propus o soluție de tratare a diacriticelor, astfel încât etichetarea morfologică să aibă o precizie suficientă chiar şi în lipsa acestora. Sistemul poate completa anumite coloane în mod automat, pe baza importului datelor din fişiere existente, ceea ce ajută la implementarea unor informații simple de natură semantică. De asemenea, generatorul poate clona propozițiile de intrare pe bază de sinonime, permițând mărirea rapidă a setului de date de antrenament al CRF++.

Ca dezvoltare, vom evalua performanța programului pe un corpus extins, precum şi pe corpusuri de texte din domenii particuare, care e posibil sa aibă valori specifice domeniului pentru frecvențele cuvintelor şi trigramelor, diferite de cele folosite de noi pentru testare. Dorim de asemenea implementarea funcției de modificare direct din editor a fişierelor lexicon şi trigramă, spre a permite îmbunătățirea progresivă a performanței taggerului pe măsura etichetării de noi texte de către utilizatori.

ACKNOWLEDGMENTS

Autorii mulțumesc recenzorilor pentru comentarii şi studentei UTCN Lidia Corde pentru contribuția la etichetarea corpusului şi la construirea modelului CRF.

REFERENCES

[1] L. Qi, L. Chen, “Comparison of model-based learning methods for feature-level opinion mining’, in *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, Lyon, France,* pp. 265–273, *2011*.

[2] C. Sutton, A. McCallum, “An introduction to conditional random fields”, in *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 267-373, 2012.

[3] R. Simionescu, “Graphical grammar studio as a constraint grammar solution for part of speech tagging”, in *Proc. of ConsILR Conference*, Bucureşti, Romania, 2012.

[4] A. Ratnaparkhi, “A maximum entropy model for part-ofspeech tagging”, in *Proc. of the Conference on Empirical Methods in Natural Language Processing*. University of Pennsylvania, Philadelphia, PA, pp. 133–142, 1996.

[5] L. E. Baum, T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains”, in *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[6] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”, in *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[7] V. Bobicev, V. Maxim, T. Prodan, N. Burciu, V. Anghelus, “Emotions in words: Developing a multilingual WordNet-Affect”, in *Computational Linguistics and Intelligent Text Processing*, LNCS vol. 6008, Springer Berlin Heidelberg, pp. 375–384, 2010.