Revista Română de Interacţiune Om-Calculator 7 (4) 2014, 273-288

Identificarea entităţilor cu nume

Liviu Sebastian Matei[*], Ştefan Trăuşan Matu[*,**]

[*]Universitatea Politehnica din Bucureşti
Splaiul Independenţei 313, 060042, Bucureşti
[**]Institutul de Cercetări în Inteligența Artificială
Calea 13 Septembrie 13, 050711, Bucureşti
E-mail: liviumat@gmail.com, stefan.trausan@cs.pub.ro

Rezumat. Identificarea entităţilor cu nume într-un text reprezintă un subiect important pentru prelucrarea limbajului natural. În cadrul acestui articol se descrie o metodă nouă de detectare a entităţilor cu nume care încearcă să îmbunătăţească rezultatele obţinute cu detectorul de entităţi cu nume de la Stanford. Modelul prezentat va folosi clasificatorul Bayesian naiv pentru a determina şi clasifica entităţile cu nume.

Cuvinte cheie: entităţi cu nume, Bayes, învăţare automată, prelucrarea limbajului natural

1. Introducere

Identificarea entităţilor cu nume reprezintă un subiect important în prelucrarea limbajului natural având multiple aplicabilităţi: de la motoare de cautare (unde se poate acorda o pondere mai mare entităţilor cu nume), la rezumarea automată a textor (unde rezumarea se poate efectua în jurul entităţilor cu nume), la răspunsuri automate la întrebari, etc. Există multiple aplicaţii care încearcă să determine entităţile cu nume în texte dintre care amintim: Stanford NLP (Stanford NLP group website), Apache OpenNLP (OpenNLP website), NTLK – Natural Language Processing Toolkit (NTLK website), o bibliotecă scrisă în Python pentru prelucrarea limbajului natural, Gate – General Arhitecture for Text Extraction (Gate website), LingPipe (LingPipe website), Alchemy API (Alchemy API website), OpenCalais (OpenCalais), etc. În acest articol vom prezenta o metoda nouă de identificare a entităţilor cu nume în texte. Entităţile cu nume pot fi de mai multe tipuri după cum vom vedea în continuare în acest articol, dar sistemul pe care îl construim este axat pe entităţi cu nume de tipul organizaţie şi nume de persoane, putând fi uşor extins şi la alte tipuri de entităţi cu nume. Identificarea entităţilor cu nume reprezintă unul dintre subiectele importante din cadrul prelucrării limbajului având numeroase aplicabilităţi.

În prezenta secțiune, introductivă, se vor evidenția o serie de informații generale despre entitățile cu nume: ce reprezintă, rolul lor în contextul prelucrării limbajului natural, ce tehnici de detectare a entităților cu nume există şi nu în ultimul rând o scurtă trecere în revistă a principalelor pachete software folosite pentru determinarea entităților cu nume. Cea de-a doua secţiune va descrie soluția propusă în acest articol de identificare a entităților cu nume. Se va prezenta arhitectura sistemului, modul în care funcționează, ce date de intrare sunt folosite şi respectiv ce rezultate va furniza. Ultima secțiune va fi dedicată concluziilor şi posibilelor dezvoltări ulterioare. Trebuie menționat că întreaga analiză se va realiza pe texte în limba engleză. Atât pachetele software care vor fi prezentate cât şi modelul nostru funcționează pe limba engleză dar poate fi extins şi la alte limbi.

1.1 Aspecte generale privind entitățile cu nume

Determinarea entităților cu nume reprezintă un subiect important în prelucrarea limbajului natural. El constă din identificarea şi clasificarea entităților cu nume. Astfel, dându-se următoarea propoziție: "John and Alice live in Bucharest. They work at Microsoft" (John şi Alice trăiesc în Bucureşti. Ei lucrează la Microsoft), am vrea să extragem cuvintele 'John', 'Alice', 'Microsoft' şi să le putem clasifica. Am vrea să putem detecta în mod automat că 'John' şi 'Alice' sunt nume de persoane iar 'Microsoft' este numele unei companii ('Bucharest' reprezintă de asemenea o entitate cu nume dar este de tipul 'localitate' motiv pentru care nu face obiectul acestei lucrări – reamintim faptul că în această lucrare ne vom referi la entitățile cu nume de tipul nume de persoane şi organizații). Deoarece entitățile cu nume reprezintă elemente centrale ale limbajului natural, identificarea şi clasificarea lor ne ajută mult în înțelegerea textului.

Numeroase cercetari s-au desfăşurat în zona identificării entităților cu nume (Finkel, 2007; Dekang, 2009; Ronan Collobert, 2011; R. K. Ando, 2005; T. Kudoh, 2000) folosind diferite abordări. Seturi variate de date de test au fost folosite pentru a calcula precizia, acoperirea şi performanța metodelor propuse. Unele abordări folosesc diferite tehnici de învățare automată precum modelul Markov ascuns (Stamp, 2004), modelul Markov condiționat (CMM) (Berger, Pietra, & Pietra, 1996), câmpuri aleatoare condiționale (CRF) (Lafferty, McCallum, & Pereira, 2001), clasificatorul Bayesian (Rocha), maşini cu suport vectorial (SVM) (Cristianini & Shawe-Taylor, 2000) pe când alte abordări folosesc diferite seturi de date

adnotate sau şabloane pentru a identifica entitățile cu nume. În general toate aceste abordări folosesc tehnici avansate de prelucrare a limbajului natural precum etichetarea cu părți de vorbire, detectarea propozițiilor, etc.

1.2 Aplicații software pentru identificarea entităților cu nume

În cele ce urmează voi prezenta pe scurt două dintre cele mai cunoscute pachete software de prelucrare a limbajului natural: Apache OpenNLP şi biblioteca de prelucrare a limbajului natural de la Stanford. Ambele biblioteci sunt dezvoltate folosind limbajul de programare JAVA.

Potrivit (OpenNLP website) Apache OpenNLP este "*o unealtă care foloseşte învățarea automată pentru procesarea limbajului natural în texte*". Practic OpenNLP conține un set de API-uri care pot fi uşor integrate într-o aplicație. Aceste API-uri realizează sarcinile uzuale ale unei unelte de prelucrare a limbajului natural: detectarea propozițiilor, despărțirea în cuvinte, recunoaşterea entităților cu nume, clasificarea textelor (putem avea de exemplu texte despre fizică sau despre chimie care pot fi clasificate automat). Pentru realizarea clasificării se folosesc o serie de date de antrenare după care se aplică algoritmul bazat pe entropia maximă pentru a determina clasificarea noului text. De asemenea, OpenNLP mai oferă sprijin şi pentru adnotarea textelor cu părțile de vorbire. Asemenea tehnicilor prezentate anterior şi în cazul acesta, OpenNLP foloseşte un dicționar care conține pentru fiecare cuvânt toate părțile de vorbire pe care acel cuvânt poate să le aibă.

Un alt pachet software foarte popular este reprezentat de biblioteca de la Stanford. Dintr-o perspectivă funcțională, pachetul de la Stanford este similar cu OpenNLP, realizând şi el operațiile uzuale de prelucrare a limbajului natural precum: despărțirea textului în cuvinte, asocierea cuvintelor cu partea de vorbire, extragerea propozițiilor din texte, etc. Conform site-ului web asociat proiectului, pachetul Stanford NLP conține: "*un sistem competitiv de identificare a coreferințelor; un model nou de determinare a parților de vorbire în texte; un model de parsare probabilistic foarte performant; un model competitiv de recunoaştere a entităților cu nume din domeniul biologiei; cât şi algoritmi de procesare pentru limba arabă, chineză şi germană* " (http://nlp.stanford.edu).

Asemeni lui OpenNLP şi Stanford NLP este o bibliotecă uşor de folosit. Folosirea lui Stanford NLP presupune în primul rând inițializarea unei benzi de asamblare de procesare a limbajului natural cu o listă de funcționalități

necesare în aplicație precum: extragerea cuvintelor, extragerea propozițiilor, identificarea entităților cu nume etc. Ulterior tot ce trebuie făcut este să se treacă textul care se dorește să fie analizat cu Stanford NLP prin banda de asamblare.

O analiză comparativă a extragerii entităților cu nume de tipul persoană, localitate şi organizație folosind uneltele Stanford NLP (http://nlp.stanford.edu), Apache OpenNLP (OpenNLP website), Alchemy API (Alchemy API website) şi OpenCalais (OpenCalais) este realizată în lucrarea 'Comparison of Named Entity Recognition tools for raw OCR text' (Rodriquez, Bryant, Blanke, & Luszczynska, 2012). Extragerea entităților cu nume este analizată din punctul de vedere al preciziei, acoperirii (recall-ului) şi al factorului F_1. Autorii au folosit pe post de date două colecții de texte adnotate manual şi au calculat factorii menționați anterior. Rezultatele obținute de catre Rodriquez, Bryant, Blanke şi Luszczynska sunt următoarele (considerăm doar datele obținute în cazul extragerii entităților cu nume, deoarece precizia OCR-urlui considerată de asemenea în acel articol nu face obiectul analizei noastre).

Tabelul 1 – Setul 1 de date de test

Sistem NLP	Precizie	Acoperire	Factorul F_1
Alchemy API	63%	38%	48%
OpenCalais	69%	30%	42%
Open NLP	53%	13%	21%
Stanford NER	60%	61%	60%

Tabelul 2 - Setul 2 de date de test

Sistem NLP	Precizie	Acoperire	Factorul F_1
Alchemy API	43%	44%	44%
OpenCalais	46%	39%	42%
Open NLP	33%	25%	28%
Stanford NER	35%	60%	44%

După cum se poate observa în tabelele anterioare Stanford NER oferă în medie cele mai bune rezultate (considerând media celor trei factori măsurați: precizie, acoperire, F_1). Totuşi dacă analizăm în detaliu rezultatele prezentate de (Rodriquez, Bryant, Blanke, & Luszczynska, 2012) vom observa că un capitol la care Stanford NER obține rezultate mai slabe este în cazul numelor de organizații unde pentru primul set de date avem o precizie de 12% şi un acoperire de 23% fața de Alchemy (precizie de 50% , acoperire de 23%) şi respectiv Open NLP (precizie de 20% , acoperire de

8%). După cum se poate observa mai ales în cazul lui Stanford NER (dar şi al celorlalte unelte) rezultate mai slabe au fost obținute în cazul identificării entităților cu nume de tipul organizații (precizie de 12% fată de media de 60%, acoperire de 23% fată de o medie de 61%). Metoda pe care o vom prezenta în secțiunile următoare va încerca să îmbunătățească acest aspect pentru Stanford NLP.

Folosind pachetele sofware menționate anterior - atât Stanford NLP cât şi Apache OpenNLP - putem identifica şi clasifica entități cu nume. S-au folosit diferite variante de date de intrare pentru a evalua cele două biblioteci. În timpul testelor s-au identificat situații în care se pot observa limitările lor. De exemplu să considerăm următorul exemplu: 'John works at Microsoft'. Stanford NLP identifică corect 'John' ca fiind numele unei persoane şi respectiv 'Microsoft' ca fiind numele unei organizații. Dar dacă încercăm să înlocuim 'Microsoft' cu 'Oracle' biblioteca eşuează în a identifica 'Oracle' ca un nume valid de organizație. Astfel în acest studiu, bazându-ne pe aceste limitări, vom încerca să dezvoltăm un sistem de detectare a entităților cu nume bazat pe Stanford NLP. În următoarea secțiune vom prezenta această abordare.

2. Abordarea noastră

După cum a fost menționat în secțiunea anterioară, sunt situații în care atât Stanford NLP cât şi Apache OpenNLP nu identifică entitățile cu nume. În acest caz vom propune un model de identificare al entităților cu nume bazat pe clasificatorul Bayesian naiv pentru entitățile cu nume de tipul nume de persoane şi organizație – dar el poate fi extins usor şi la alte tipuri de entități cu nume. Acesta este folosit pentru a acoperi unele dintre cazurile în care biblioteca Stanford NLP nu identifică unele entități cu nume. Soluția care va fi prezentată în cele ce urmează poate fi extinsă în continuare pentru a creşte precizia calculului. Ea realizează ambele obiective propuse de un sistem de identificare a entităților cu nume:

- Identifica cuvintele care reprezintă entități de tipul organizație şi nume de persoane
- Clasificarea lor ca organizații sau nume de persoane

În general există două moduri de a identifica şi clasifica entitățile cu nume în texte:

- entități cu nume deja cunoscute - recunoaştem entitățile cu nume bazându-ne pe cunoştințe sau clasificări anterioare (de exemplu: când IBM,

Microsoft, Google apar singure într-un context pot fi considerate nume de companii, ceea ce ne ajuta să le identificăm imediat cu compania pe care o referă).

- pe baza contextului - uitându-ne la context putem deduce că un anumit cuvânt reprezintă o entitate cu nume. De exemplu dacă întâlnim într-un text următoarea propoziție: "I work at Google" (Eu lucrez la Google). Chiar dacă nu este cunoscut a priori faptul că Google reprezintă o entitate cu nume de tipul 'organizație' (companie), bazându-ne pe context putem deduce acest lucru datorită secvenței care precedă entitatea cu nume. Câteodată este posibil să clasificăm o entitate cu nume doar inspectând contextul în care apare deoarece tipul entității cu nume este dependent de context.

Un caz aparte îl reprezintă entitățile cu nume compuse, de exemplu: New York, Google Play, Microsft SQL Server, etc. În acest caz fie avem o listă de entități cu nume deja cunoscută şi identificarea se face imediat, fie pe baza contextului, prin calculul de probabilități.

După cum a fost menționat şi anterior, aplicația foloseste clasificatorul Bayesian naiv pentru a detecta şi clasifica entitățile cu nume - acest lucru se întâmplă în modulul 'Sistem de identificare a entităților cu nume'. El foloseşte trei surse de date pentru a antrena modelul folosit:

- Text adnotat manual - furnizat sistemului de identificare a entităților cu nume.
- Text adnotat cu biblioteca Stanford NER - în acest caz folosim text care nu este adnotat şi pe care îl clasificăm cu Stanford NER. Rezultatul obținut este folosit ulterior pentru a antrena noul model.
- Set de reguli - definim un set de reguli în format XML.

Arhitectura sistemului propus în această lucrare este descrisă în Figura 1.

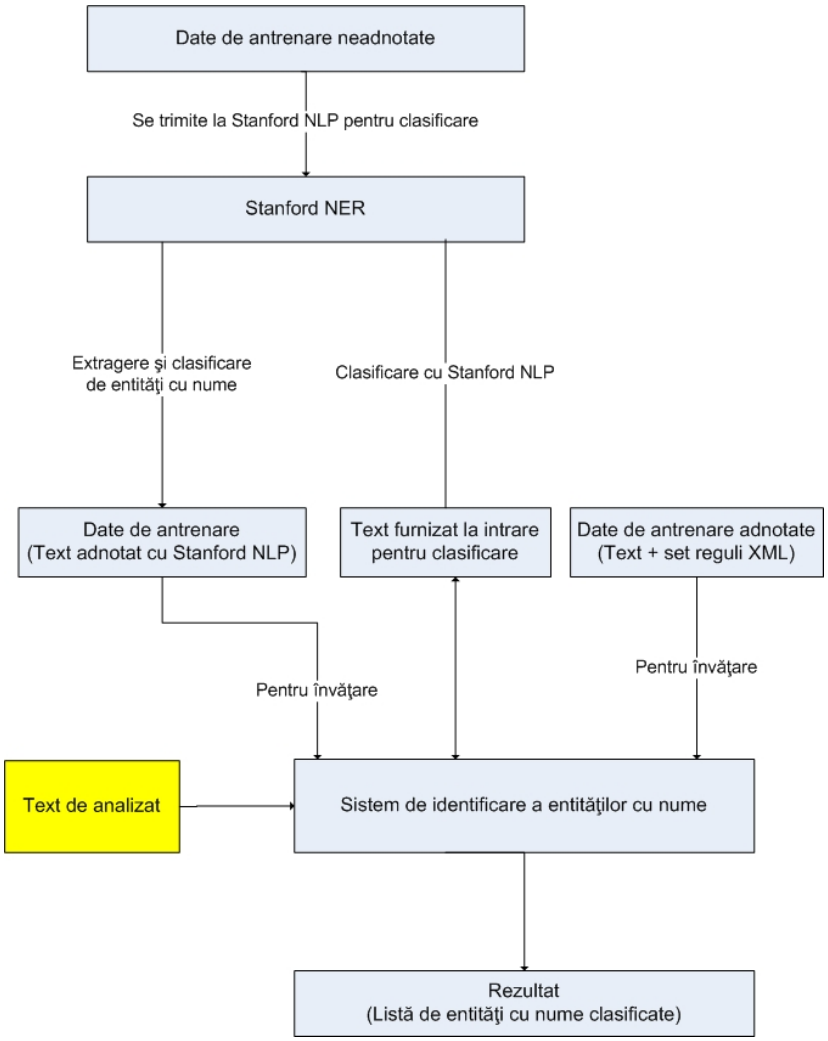

Figura 1 Arhitectura sistemului

Fişierele XML descriu diferite reguli care sunt luate în considerare în modelul de antrenare. Astfel pentru fiecare tip de clasificare – momentan sunt suportate doar tipurile ’nume de persoane’ şi ’nume de organizaţii’ – definim câte un director cu fişierele XML corespunzătoare acelei clasificări. Modelul pe care îl propunem se bazează mult pe acest set de reguli şi poate

fi uşor extins. XML-ul defineşte un format ce descrie relaţiile din text. De exemplu, putem avea următoarea regulă în XML:

```
<rule>
        <verb>work</verb>
        <preposition>at,for</preposition>
        <trust>80%</trust>
</rule>
```

XML-ul anterior descrie un model în care verbul "work" urmat de una dintre prepoziţiile "at" sau "for" precedă o entitate cu nume de tipul organizaţie, cu o încredere de 80%. Această valoare poate fi calculată folosind un text adnotat pe baza căruia putem să extragem probabilităţile unor cuvinte.

Pe baza teoremei lui Bayes putem afirma (Rocha):

$p(C|F_1, F_2 \ldots F_n) = p(C) * p(F_1, F_2 \ldots F_n \mid C) / p(F_1, F_2 \ldots F_n)$, unde C

reprezintă clasificarea iar F_1, F_2 ... F_n reprezintă factori care determină clasificarea.

Deoarece $p(F_1, F_2 \ldots F_n)$ este invariant în functie de clasă şi deoarece ne interesează doar clasificarea - nu şi procentul exact, putem ignora această probabilitate. Astfel putem spune:

$p(C|F_1, F_2 \ldots F_n) \sim p(C) * p(F_1, F_2 \ldots F_n \mid C) = p(C) * p(F_1 \mid C) * p(F_2 \ldots F_n \mid C, F_1) = p(C) * p(F_1 \mid C) * p(F_2 \mid C, F_1) * p(F_3 \ldots F_n \mid C, F_1, F_2) = p(C) * p(F_1 \mid C) * p(F_2 \mid C, F_1) * p(F_3 \mid C, F_1, F_2) * \ldots * p(F_n \mid C, F_1 \ldots F_{n-1})$

În expresia anterioară vom face urmatoarea aproximare: vom considera factorii care determină clasificarea ca fiind independenţi între ei – această simplificare este numită *presupunerea condiţional independentă a modelului Naiv Bayesian* şi are rolul de a simplifica calculul. În cazul nostru precizia nu este afectată deoarece factorii consideraţi sunt atribute ale textului reprezentat sub forma ’pungă de cuvinte’. Astfel putem spune:

$$p(C|F_1,F_2 \ldots F_n) \sim p(C) * p(F_1|C) * p(F_2|C) * p(F_3|C) * \ldots * p(F_n|C) \quad (1)$$

Această formulă se găseşte în spatele modulului 'Sistem de identificare a entităţilor cu nume'. Când încercăm să clasificăm un cuvânt ca fiind o entitate cu nume, înlocuim în formula anterioară clasa cu tipul clasificării. Astfel putem avea următoarele clasificări:

NOT-NE - în acest caz cuvântul curent nu este o entitate cu nume

NE-ORG - în acest caz cuvântul curent este identificat ca o entitate cu nume de tipul 'organizaţie'.

NE-PN - cuvântul curent este clasificat ca o entitate cu nume de tipul 'nume de persoană'

NE-LOC - reprezintă o entitate cu nume de tipul 'loc' (nume de localitate, nume de țară)

Clasificarea C va lua pe rând valorile fiecareia dintre clasele menționate anterior, şi pe baza valorilor pe care le obținem vom fi capabili să clasificăm fiecare cuvânt. Acum, că ştim ce valori poate să ia C, analizăm factorii F_1, F_2, ... F_n. Aceste valori reprezintă diferite atribute definite pe baza contextului în care apare cuvântul. Modelul poate fi uşor extins pentru a defini noi reguli. În prezent folosim următorii factori: valorea celor două cuvinte care preced cuvântul analizat, valorile următoarelor două cuvinte şi partea de vorbire a celor două cuvinte adiacente cuvântului analizat. Toți aceşti factori intervin în formula (1). Asfel putem reformula relația (1): dându-se cuvântul 'w' pe care dorim să îl clasificăm în termenii identificării entităților cu nume, care se gaseşte în secvența w_{p1} w_{p2} w w_{n1} w_{n2}, putem considera următoarea relație bazată pe probabilități:

$$p(C(w) \mid w_{p1}w_{p2}, w_{p2}, w_{n1}, w_{n1}w_{n2}, POS(w_{p2})\ POS(w_{n1})\ ) = P(C(w)) * P(w_{p1}w_{p2} \mid C(w)) * P(w_{p1} \mid C(w)) * P(w_{n1} \mid C(w)) * P(w_{n1}w_{n2} \mid C(w)) * P(POS(w_{p1}) * POS(w_{n1}) \mid C(w)) \quad (2)$$

unde:

C(w) = clasificarea cuvântului 'w' care este subiectul analizei;

$p(C(w) \mid w_{p1}w_{p2}, w_{p2}, w_{n1}, w_{n1}w_{n2}, POS(w_{p2})\ POS(w_{n1}))$ = probabilitatea clasificării cuvântului 'w' în clasa C;

w_{p1} = cuvântul aflat la distanță de un cuvânt de 'w' care este subiectul analizei - între 'w' si 'w_{p1}' se găseşte doar 'w_{p2}';

w_{p2} = cuvântul care precedă cuvântul 'w';

w_{n1} = cuvântul poziționat după 'w';

w_{n2} = cuvântul aflat la o distanță de un cuvânt de 'w'. Între 'w' şi 'w_{n2}' se găseşte numai cuvântul 'w_{n1}';

$POS(w_{n1})$ = partea de vorbire asociată cuvântului w_{n1};

$POS(w_{p1})$ = partea de vorbire asociată cuvântului w_{p1}.

De fiecare dată când vrem să determinăm dacă un cuvânt este o entitate cu nume (şi ce tip de entitate cu nume este) calculăm probabilităţile asociate fiecărei clasificări. Probabilitatea cu valoarea cea mai mare va indica clasificarea cuvântului. În cazul în care se obtin două sau mai multe probabilităţi egale cuvântul nu poate fi clasificat. În relaţia (2) observăm că valoarea probabilităţii pe care vrem să o calculăm depinde de $P(C_w)$ care reprezintă probabilitatea a priori a clasei C_w. Deoarece această probabilitate depinde mult de tipul textului analizat – în unele tipuri de articole entităţile cu nume apar mai des decât în altele, calculul unei valori exacte pentru o astfel de probabilitate este dificil de realizat. Astfel, vom face un compromis şi vom considera un model similar celui folosit de clasificatorul Bayesian pentru determinarea mesajelor poştei electronice nedorite (în care probabilitatea a priori de clasificare a unui email ca spam sau nu este de 50%). Folosim această abordare în cadrul sistemului nostru de identificare a entităţilor cu nume şi atribuim probabilităţi egale tuturor clasificărilor efectuate a priori. În lipsa cunoaşterii unor valori precise dorim să ignorăm această probabilitate în procesul de clasificare – lucru realizat prin considerarea de probabilităţi egale:

$$P(Cw=NOT\text{-}NE)=P(Cw=NE\text{-}ORG)=P(Cw=NE\text{-}PN)=P(Cw=NOT\text{-}LOC)=25\%$$

Ştiind că aceşti factori sunt aceeaşi pentru toate clasificările, ei pot fi ignoraţi astfel încât relaţia devine:

$$p(C(w) \mid w_{p1}w_{p2}, w_{p1}, w_{n1}, w_{n1}w_{n2}, POS(w_{p1}) POS(w_{n1})) = P(w_{p1}w_{p2} \mid C(w)) * P(w_{p1} \mid C(w)) * P(w_{n1} \mid C(w)) * P(w_{n1}w_{n2} \mid C(w)) * P(POS(w_{p1})) * POS(w_{n1}) \mid C(w)) \quad (3)$$

Formula (3) reprezintă varianta finală a formulei de calcul a probabilităţilor. Desigur această formulă poate fi extinsă cu alte probabilităţi (corespunzătoare unor noi atribute), de exemplu se pot lua în calcul mai multe cuvinte sau mai multe criterii (de exemplu se poate considera poziţia unui cuvânt în text deoarece sunt şanse mai mari ca o entitate cu nume să apară la începutul unei propoziţii). O data identificată o nouă probabilitate care poate fi luată în considerare tot ce trebuie să facem este să o introducem în formula (3) şi să o înmulţim cu celelalte probabilităţi. Este important de menţionat că pentru a calcula această nouă valoare a probabilităţii fie avem valoarea furnizată direct de către utilizator, fie este determinată pe baza datelor de antrenament. În cazul în care una dintre probabilitaţi nu poate fi calculată (de exemplu cuvântul pe care vrem să îl clasificăm este primul din propoziţie şi astfel $P(w_{p1}w_{p2} \mid C(w))$ nu poate fi calculat), acea probabilitate este ignorată.

După cum a fost menționat anterior, probabilitățile sunt calculate folosind datele de antrenament adnotate, date de antrenament obținute din text neadnotat prin rularea pachetului Stanford NLP asupra sa şi o serie de fişiere XML cu reguli. În datele de antrenament adnotate se găsesc propoziții în care fiecare cuvânt este adnotat cu partea sa de vorbire şi cu clasificarea sa (cuvânt normal sau entitate cu nume, caz în care tipul entității cu nume este specificat de asemenea). Pe baza acestor clasificări putem calcula probabilitățile.

O propoziție adnotată arată în felul următor:

```
John [substantiv, NE-PN] works [verb, NOT-NE] for[prepoziție,
     NOT-NE] Google [substantiv, NE-ORG].
```

În textul care va fi supus analizei este posibil să avem propoziții similare celei anterioare, asfel încât prezența verbului 'work' urmat de prepoziția 'for' determină o probabilitate ridicată ca următorul cuvânt să fie o entitate cu nume - această probabilitate poate fi reprezentată de expresia P('work' , 'for' | NEORG).

Datele de antrenament obținute cu pachetul Stanford NLP sunt similare cu datele deja adnotate prezentate în paragraful anterior. Singura diferență este data de modul în care este realizată adnotarea, proces descris în Figura 2.

Practic modulul din Figura 2 primeşte la intrare un text care este trimis la Stanford NLP pentru a clasifica cuvintele din punctul de vedere al identificării entităților cu nume. După aceea folosim un modul cu rol de adaptor care foloseşte ca intrare rezultatul furnizat de Stanford NLP şi îl transformă în formatul convenit în modelul nostru. În acest moment textul furnizat la intrarea modulului este exact în forma în care se gaseşte şi textul adnotat manual descris anterior.

După cum a fost descris anterior (şi afişat în Figura 1) cea de-a treia sursă de antrenare constă din fişiere XML cu reguli. Toate regulile descriu şabloane de cuvinte care pot să apară în text. În momentul de față oferim suport doar pentru şabloane care specifică în mod explicit cuvintele, dar această abordare poate fi schimbată pentru a oferi suport şi pentru părțile de vorbire - a se vedea secțiunea 'Concluzii'.

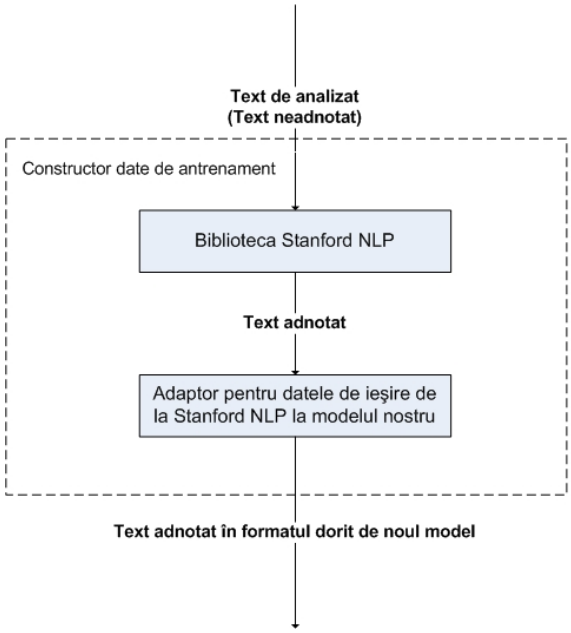

Figura 2- Constructor de date de antrenare

În momentul în care vrem să realizăm identificarea entităților cu nume într-o propoziție, primul lucru pe care îl realizăm este împărțirea textului în propoziții. După acest prim pas vom lua fiecare cuvânt şi pe baza formulei specificate în relația (3) vom calcula clasificarea. Este important de menționat că acest mecanism se aplică doar în cazul cuvintelor care nu au fost identificate şi clasificate cu biblioteca Stanford NER.

3. Rezultate

În această secțiune vom prezenta rezultatele care au fost obținute folosind metoda descrisă anterior de detectare a entităților cu nume. Pe post de date de antrenare s-au folosit următoarele:

- date adnotate manual - un set de 150 articole Reuters care au fost adnotate
- date adnotate cu biblioteca Stanford NLP - un set de 21.500 de articole Reuters au fost adnotate
- reguli XML - 20 de reguli au fost reprezentate în format XML. Aceste reguli descriu şabloane des întâlnite în texte. Fiecare dintre aceste reguli are asociată o probabilitate care reprezintă 'încrederea' în acea regulă:

- "work at" ; "work for". - sunt în general folosite înainte de entități cu nume de tipul organizație
- "live in" ; "travel to". - folosite înainte de entități cu nume de tipul localitate.
- "declares that" - este folosit înaintea unor entități cu nume de tipul persoană

În continuare, vom considera diverse exemple pentru a evalua modelul prezentat. Unele exemple vor evidenția slăbiciuni ale bibliotecii Stanford NLP – entități cu nume care nu sunt identificate, caz în care vom specifica regulile şi/sau exemplele care au ajutat modelul nostru să identifice şi sa clasifice corect, îmbunătățind astfel modulul de identificare al entităților cu nume din Stanford NLP.

Astfel să considerăm următoarele exemple:

```
1. John is going to work for Adobe.  (John va lucra pentru
Adobe.)
```

În acest caz Stanford NLP nu identifică în mod corect Adobe ca un nume de companie, identifică doar 'John' ca nume de persoană. Aplicând modelul nostru obținem un procent de 87% în favoarea clasificării ca entitate cu nume de tipul organizație. Acest lucru se întâmplă doarece avem o regula definită în fisierul XML care specifică cu o încredere de 80% că după "work for" avem o entitate cu nume de tipul organizație.

```
2.  Adobe's stock options incresed by 10%.    (Opțiunile de
acțiuni  Adobe au crescut cu 10%.)
```

În acest caz de asemenea "Adobe" nu este detectat ca o entitate cu nume dar în cazul exemplelor "Adobe Corp.'s stock options increased by 5%" or "Microsoft's stock options increased by 5%" , Stanford NER va clasifica corect "Adobe Corp." şi respectiv "Microsoft Corp." ca entități cu nume.

```
3. Perry joins Adobe from Microsoft, where she played a
leadership role ...  (Corp., 2011)  (Perry se alătură Adobe
de la Microsoft, )
```

În exemplul din (Corp., 2011) folosind biblioteca Stanford NLP obținem clasificări pentru "Perry" - care este clasificat în mod corect ca 'persoană' şi pentru 'Microsoft' care este clasificat de asemenea în mod corect ca o organizație. În acest caz folosind exemple de tipul: "Rosenfeld joined Salomon in 1979 ..." şi "Gottlieb joined Lorimar as senior vice president in 1985." (Reuters, 1997) din setul de date de antrenare obținem o probabilitate de 58% ca 'Adobe' să fie clasificat ca o entitate cu nume.

Pentru a evalua metoda prezentată s-a folosit colecția de 21.500 de articole Reuters care contine 56 de entități cu nume de tipul organizație şi respectiv 267 de entități cu nume de tipul nume de persoane. Pe acest set de date s-a rulat întai biblioteca Stanford NLP după care s-a rulat modelul prezentat în această lucrare şi s-a calculat precizia, acoperirea şi factorul F_1 .

Factorul F_1 este definit astfel:

F_1 = (2*precizie*acoperire) / (precizie+acoperire).

Tabelul 4 – Rezultate entități cu nume de tipul organizație

Metoda considerata	Precizie	Acoperire	Factorul F_1
Stanford NLP	24%	32%	27%
Metoda prezentată în această lucrare	29%	32%	30%

Tabelul 5 – Rezultate entități cu nume de tipul nume de persoane

Metoda considerata	Precizie	Acoperire	Factorul F_1
Stanford NLP	76%	71%	73%
Metoda prezentată în această lucrare	78%	71%	74%

După cum se observă din cele două tabele noua metodă aduce îmbunătățiri în ceea ce priveste precizia.

4. Concluzii

În această lucrare am prezentat o abordare nouă, folosită pentru identificarea entităților cu nume de tipul organizație şi nume de persoane care se bazează pe biblioteca Stanford NLP. Practic folosim clasificatorul Bayesian naiv pentru a detecta mai multe entități cu nume. Modelul pe care îl prezentăm creşte în precizie dacă furnizăm mai multe date de antrenament cu condiția să dispunem de date de test de calitate – această creştere se va realiza până la un anumit prag începând de la care creşterile devin nesimnificative. Este important de menționat că o modificare greşită în fişierul XML cu reguli poate duce la o scădere a preciziei deoarece anumite cuvinte vor fi clasificate greşit.

Metoda prezentată anterior are şi anumite limitări. Datorită faptului că folosim o metodă probabilistică este foarte important să avem date de antrenare cât mai bune după cum a fost menționat şi anterior. O altă limitare

poate să apară deoarece nu aplicăm nici un fel de validare asupra entităților cu nume determinate (modelul prezentat nu foloseşte o bază da date cu entități cu nume care să poată fi folosită pentru validare). De exemplu putem avea o propoziție de tipul: "I work for Unknown" (Lucrez pentru Necunoscut). Modelul nostru probabilistic va clasifica "Unknown" ca numele unei organizații deşi nu putem spune cu siguranță daca această clasificare este validă.

Această metodă poate fi şi îmbunătățită. Astfel în prezent aplicația noastră este construită peste Stanford NLP dar o soluție ar fi să se folosească pentru clasificare şi alte biblioteci similare precum Apache NLP sau GATE.

Anterior a fost menționat fişierul XML care conține reguli ce sunt ulterior aplicate în procesul de detecție. O abordare interesantă ar fi să formulăm problema invers: dându-se texte adnotate se doreşte construția fişierului XML cu reguli. Astfel am putea construi o serie de reguli care să abstractizeze textele adnotate.

Referințe

Alchemy API website. Preluat pe Septembrie 21, 2014, din Alchemy API website: http://www.alchemyapi.com/

Berger, A., Pietra, V., & Pietra, S. (1996). *A maximum entropy approach to natural language processing.* Computational Linguistics (MIT Press) .

Chen, E. *Blog post regarding conditional random fields*. Preluat pe Decembrie 15, 2013, din http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

Corp., A. (2011, Septembrie 12). *Adobe Appoints Two New Vice Presidents.* Preluat pe 15 Septembrie, 2014, din http://www.adobe.com/aboutadobe/pressroom/pressreleases/201109/AdobeAppointsIlgandPe

CRF-NER. Preluat pe Decembrie 17, 2013, din Stanford NLP toolkit: http://nlp.stanford.edu/software/CRF-NER.shtml

Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.

Dekang Lin, X. W. (2009). Phrase Clustering for Discriminative Learning. *4th International Joint Conference on Natural Language Processing of the AFNLP*, (pp. 1030-1038).

Finkel, J. R. (2007, Martie 9). Named Entity Recognition and the Stanford NER Software. Stanford University.

Finkey Jenny Rose, G. T. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Annual Meeting on Association for*

Computational Linguistics.

Gate website. Preluat pe Septembrie 2014, 22, de la Gate website: https://gate.ac.uk/

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, (pp. 282–289).

LingPipe website. Preluat pe Septembrie 21, 2014, de la LingPipe website: http://alias-i.com/lingpipe/

NTLK website. Preluat pe Septembrie 22, 2014, de la NTLK website: http://www.nltk.org/

OpenCalais. Preluat pe Septembrie 21, 2014, de la OpenCalais: http://www.opencalais.com/

OpenNLP website. Preluat pe Decembrie 15, 2013, de la http://opennlp.apache.org/

R. K. Ando, T. Z. (2005). A framework for learning predictive structures from multiple tasks. *Journal of Machine Learning Research 6* , 1817–1953.

Reuters. (1997, Septembrie 26). Reuters-21578 text categorization test collection.

Rocha, A. d. *Naive Bayes classifier.* Preluat pe Februarie 9, 2014, from http://www.ic.unicamp.br/~rocha/teaching/2011s2/mc906/aulas/naive-bayes-classifier.pdf

Rodriquez, K. J., Bryant, M., Blanke, T., & Luszczynska, M. (2012). Comparison of Named Entity Recognition tools for raw OCR text. *Proceedings of KONVENS 2012 (LThist 2012 workshop).* Viena.

Ronan Collobert, J. W. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 2461-2505.

Smith, A. (2003). Markov Chain Monte Carlo Simulation Made Simple. New York.

Stamp, M. (2004). *A revealing introduction to hidden Markov models.* Department of Computer Science San Jose State University.

Stanford NLP group website. Preluat pe Decembrie 15, 2013, de la Stanford NLP group website: http://nlp.stanford.edu/

T. Kudoh, Y. M. (2000). Use of support vector learning for chunk identification. In Conference on Natural Language Learning (CoNLL) and Second Learning Language in Logic Workshop, pp. 142–144.

V.Kanaka Durga, M. R. (2012). Accurate Spam Mail Detection using Bayesian Algorithm. *International Journal of Advanced Research in Computer Engineering & Technology*, Volume 1, Issue 4, pp. 402-406.