

Unsupervised Lexical Acquisition for Part of Speech Tagging

Dan Tufiş, Elena Irimia, Radu Ion, Alexandru Ceauşu

Research Institute for Artificial Intelligence, Romanian Academy

13, “Calea 13 Septembrie”, Bucharest 050711, Romania

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

Abstract

It is known that POS tagging is not very accurate for unknown words (words which the POS tagger has not seen in the training corpora). Thus, a first step to improve the tagging accuracy would be to extend the coverage of the tagger’s learned lexicon. It turns out that, through the use of a simple procedure, one can extend this lexicon without using additional, hard to obtain, hand-validated training corpora. The basic idea consists of merely adding new words along with their (correct) POS tags to the lexicon and trying to estimate the lexical distribution of these words according to similar ambiguity classes already present in the lexicon. We present a method of automatically acquire high quality POS tagging lexicons based on morphologic analysis and generation. Currently, this procedure works on Romanian for which we have a required paradigmatic generation procedure but the architecture remains general in the sense that given the appropriate substitutes for the morphological generator and POS tagger, one should obtain similar results.

1. Introduction

Part of speech (POS) tagging is known to be accurate. For English, experimental results show accuracies starting with 96% using various tag sets and training corpora sizes (Brill, 1996; Ratnaparkhi, 1998; Brants, 2000). In the case of Hidden Markov Models (HMM) POS tagging, what really makes the difference, is the accuracy obtained on unknown words (words that the tagger has not seen in the training phase) because for these, the tagging algorithm does not have the choice option on its possible POS tags. As a result, the algorithm has to predict the word’s correct POS tag based on whatever information is available: word features such as affixes or capitalization and the history of tagging (already assigned POS tags) up the current tagging point.

To diminish the guessing effect on the unknown words tagging, one could consider at least two options: either improve the guessing routine capacity of predicting the right POS of the unknown word or, keep on updating the tagger lexicon with as many word forms (along with their probable tags) as possible.

The first option is limited in the sense that most of the effective heuristics presented in the literature have local scope (e.g. suffix and capitalization information) and, besides the extra processing time to perform ending analysis, usually, they over-generalize on the possible ambiguity class of the unknown word, inducing spurious ambiguities. Most often than not, the adherents of this approach assume that the (n-gram) optimization process would filter out the unlikely tags in a guessed ambiguity class. Dermatos & Kokkinakis (1995) assume that the unknown words have POS ambiguity classes (and probability distributions) similar to the rare known words (in their experiments they considered hapax legomena words only). The same assumption is adopted by Ratnaparkhi (1998). The major advocated advantage of this method is that in arbitrary texts one would always

find unknown words, irrespective of the tagger lexicon coverage.

The second option, although presumably more accurate and much faster, has a drastic limitation: it requires larger and larger POS annotated training corpora.

Our approach is a kind of reconciling the two methods, using essentially a strategy of the first type (but more informed, as will be discussed below) yielding an additional probabilistic lexicon, followed, when the size of this lexicon became significant, by an automatic rebuilding of the language model of the tagger. In this phase the additional probabilistic lexicon is merged with the tagger probabilistic lexicon, thus bringing our method closer to the second way of dealing with the unknown words.

As in the Dermatos & Kokkinakis (1995) or Ratnaparkhi (1998), the main problems in dealing with unknown words remain deciding on the possible POS tags and determining their probability distribution without having occurrence contextual data for all possible interpretations. Our work follows the same hypothesis concerning the distributional similarity between the possible interpretations of the unknown words and those of rare words. The distinct difference, however, is that, in our approach, the list of possible POS tags is computed based on more informed knowledge sources.

2. Preliminaries

Having chosen our method of improving the tagger’s accuracy on unknown words, we now arrive at the discussion on how to populate the lexicon with new entries without having reliable POS tagged corpora to train the tagger.

The fundamental equation of a trigram HMM POS tagger states that the best state sequence $t_1 \dots t_T$ of the HMM is given by:

$$\arg \max_{t_1 \dots t_T} \left[\prod_{i=1}^T P(t_i | t_{i-2}, t_{i-1}) P(w_i | t_i) \right]$$

for a sequence of T words, $w_1 \dots w_T$ where $t_1 \dots t_T$ is the best POS tags assignment of the given word sequence. Thus, the selection of the optimal POS tag at position i , is simultaneously considering the trigram probability containing t_i and the lexical probability of w_i , both of them in the context of (usually) the T-word sentence. Estimating the lexical probability of w_i is based on evidence provided by the counts extracted from the training tagged corpora (Brants, 2000). By collecting frequency counts for each pair $\langle w, t_j \rangle$, one obtains a list of word forms w , each of them associated with a set of tag-frequency pairs $\{ \langle t_1 f_1 \rangle \dots \langle t_k f_k \rangle \}$ called the *lexical distribution* of w (each tag t_i is paired with the frequency f_i with which it labeled w in the training data). The set of tags a word w has been associated with represents its *ambiguity class* while a word associated with its lexical distribution represents an entry in the *lexicon* of the POS tagger.

Extending the tagger's lexicon with new entries, without having larger training corpora, comes to being able to provide for the new words their ambiguity classes along with associated lexical distributions. The first and the simplest choice, is to assign a frequency equal to 1 to each tag from the ambiguity class, thus obtaining a uniform distribution and hoping that the trigram model from the above equation is able to correctly choose the correct tag since all $P(w | t_i)$ are equal. A second, more appealing solution is to assign the frequencies from a similar ambiguity class of a rare word in the training corpora.

In languages with productive inflectional morphology, several forms of the same lemma may not be seen in a training corpus and such word forms, when occurring in a new text, would be dealt with as unknown words. Thus, unknown words may be genuinely unknown to the tagger or represent new inflected forms in the same paradigmatic family with already known words. The procedure in section 3 takes care of extending the tagger lexicon with the entire paradigmatic family of a previously unseen word, so that on successive text tagging only genuinely unknown words are added to the tagger unigram lexicon.

Here we must observe that rare words tend to have smaller ambiguity classes than the frequent words. This is due to the fact that homograph occurrence is extremely rare for the low frequency words, so, their ambiguity is essentially an intra-category ambiguity (ICA), a matter of distinguishing among various values for the attributes specific to a grammatical class. In (Tufiş, 1999) we argued that, for highly inflectional languages, the hard to solve ambiguities are ICAs and, that the main grammar category (part of speech) for the homonyms is easily predictable in context. The paradigmatic approach to the lexicon acquisition, described in the next section, addresses this very issue.

To verify the claim that rare words have smaller ambiguity classes, we made an experiment on our hand validated, POS annotated English-Romanian parallel corpus (roughly, one million words per language). Thus, we have extracted all the words from the Romanian text, sorted them in descending order by their occurrence frequency and assigned each word in this list its ambiguity class extracted from the corpus. Figure 1 depicts a plot of frequency ranks (X axis) with POS ambiguities (Y axis) for Romanian. One can observe a clear decrease of the POS ambiguity as the frequency rank increases (obviously, frequency itself decreases).

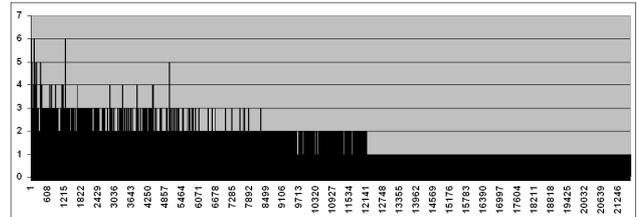


Figure 1: POS ambiguity decrease with word frequency

In what follows, we will describe an automatic procedure for generating ambiguity classes for unknown words as they are dealt with by one of our POS taggers TTL (Ion, 2007). In fact, our procedure will generate whole inflected sets of word forms along with their proper POS tags beginning from a few POS tagged word forms with the same lemma and the same part of speech. These sets are ready to populate the tagger's lexicon thus contributing to the definition of the ambiguity classes for the unknown words.

3. The lexical acquisition procedure

The procedure takes into account only open class words because these words are the likely candidates for unknown words in POS tagging (it is assumed that closed class words are already stored in the tagger's lexicon along with their complete ambiguity classes). The main body of the lexical acquisition procedure performs the following steps:

1. given a new text, run it through TTL in MSD mode (which performs POS tagging¹ and lemmatization) and obtain a list of unknown words (which TTL marks as such) along with their lemmas and POS tags;
2. from the list in step 1, extract and group word forms with the same lemma and the same part of speech into separate sets, called (*partial*) *paradigmatic families*;
3. for each set in step 2, deduce the inflectional paradigm for the respective (*partial*) *paradigmatic*

¹ TTL uses two tagsets: the Morpho-Syntactic Descriptor (MSD) tagset, a set of POS tags that encode various languages morpho-syntactic attributes for each POS (see <http://nl.ijs.si/ME/V2/msd/> for details) and a reduced tagset (CTAG). The two tagsets are not independent and they observe the tiered tagging philosophy (Tufiş, 1999). The implicit tagset is MSD, which although gives a slightly lower precision than CTAG, is much more informative.

family and apply the ROG morphological generator to get the complete paradigmatic family of the corresponding lemma along with their generated POS tags;

4. for each generated word form, extract all its corresponding POS tags acquired at step 3 and construct the ambiguity class of the word form.

The algorithm takes into consideration only the complete sets of inflected word forms that have been unambiguously generated from a single paradigm thus ensuring that the output of the morphological generator is correct. The next section details the 3rd step from the aforementioned algorithm.

4. Paradigm generation

Paradigm generation relies on the paradigmatic morphology and its implementation for Romanian (Tufiş, 1989) as well as on the ROG paradigmatic morphological generator (Irimia, 2007). The term *paradigm* will be used with two distinct meanings: **1)** the complete set of inflected word forms of a given lemma with a specific part of speech (for instance, English noun “car” has two members: “car” (singular) and “cars” (plural)); **2)** the formalized representation of the inflection mechanism. Throughout the article, we will mark the two meanings with subscripts 1 and 2 for disambiguation.

According to the paradigmatic morphology framework, a word form is composed of a *root* and an *ending* that in turn may contain a derivational suffix and an inflectional termination. For instance, nouns *ceasornicar* (a person who repairs watches) and *cărbunar* (a person who produces coal) both inflect according to one and the same paradigm₂: nominative masculine suffix 1 (\$nomsuff1). Thus, *ceasornicar* is formed from the morphological root *ceasornic* (“watch”) and the agentive derivational suffix *ar*. Similarly, *cărbunar* has the root *cărbun* (*cărbune* is the English “coal”) and the same derivational suffix. In nominative/accusative, singular, definite, both word forms are composed by adding the inflectional suffix *ul* such that we have *ceasornic+ar+ul* and *cărbun+ar+ul* and so on for the entire paradigm₁.

The information associated with the root is stored in a dictionary (lexical repository) entry corresponding to the lemma of the corresponding root. Such an entry has the following structure:

POS

@lemma

root_1 root_2 ... root_k associated_paradigm1

root_k+1 ...root_j associated_paradigm2

With a new word, the root must be determined along with information associated with it and this is the task of ROG. The root alternation phenomenon (2 roots for the same inflectional paradigm₂ of a noun, 2 to 7 roots for the same inflectional paradigm₂ of a verb) is quite frequent in Romanian. The multiple roots of a verb are very difficult to identify automatically (it is never the case that all the different roots are to be found in a specific text).

Fortunately, we possess a rich database of such kind of verbs with all their possible roots. So, it's not very likely that a new verb will manifest the root alternation phenomenon. For nouns, the identification of the two roots in case of root alternation is quite easy (the two roots have fixed roles - one is used for the nominative form of the singular and the other for all the plural forms and for the genitive/dative form of the singular – and the presence of two representative forms in a text is very likely). Also, there are some regular, but very frequent alternations (such as infixed diphthongs alternations oa->o, ea->e or final consonant alternation d->z, t->ț) which help in determining the lemma of an unknown word.

The information associated with the ending is stored in ROPMORPH, a file containing a complete inventory of the Romanian paradigms₂ for verbs, nouns and adjectives. A paradigm₂ is a collection of endings (or suffixes) each of them associated with the morphological information characterizing a word form. An entry in this file has the structure of a tree, with the *endings* occupying the leaves position, the morphological information in the intermediate position, and the paradigm₂ identifier in the root. Some important information that comes with both the roots and the endings is, in case of root alternation, the specification of what root combines with what ending. For such cases, the leaves of most of the trees in ROPMORPH have one supplementary attribute ALT, which specifies the alternate root that combines with the ending in that leaf. The value of the ALT attribute is a number that represents the position of the root in the list of the alternate roots: *root_1 root_2 ... root_k*.

By combining the information supplied in the dictionary (or identified by ROG) with the one in ROPMORPH, we can identify all the legal endings (and the associated root restrictions) that can be concatenated with the root or roots of a specific lemma, to obtain correct inflected word forms. Exploiting the structure of an entry in ROPMORPH, we can easily generate the family of all the inflected forms for a lemma (= paradigm₁), if we can identify the root(s) and the correct paradigm₂(s). The information we find on the way from the root to a leaf when we generate a word form helps in associating the proper MSD tag.

Accordingly, the real difficult problem to be solved for a group of new words (not in the ROG dictionary) with the same lemma and part of speech is the correct identification of the root(s) and associated paradigm(s).

4.1. Paradigm identification for a group of word forms with the same lemma and part of speech

To start with an example, we choose the noun lemma *carte* (English “book”) and the word forms group: *carte*, *cărți*, *cărții*. One can observe a change in the word form's root:

- *carte*: root: *cart*, suffix: *e*, morphological features: feminine, nominative, singular, indefinite;

- *cărți*: root: *cărț*, suffix: *e*, morphological features: feminine, nominative, plural, indefinite or feminine, genitive/dative, singular, indefinite;
- *cărții*: root: *cărț*, suffix: *ii*, morphological features: feminine, genitive/dative, singular, definite.

From TTL, ROG knows the lemma of the group, *carte* and the morphological features of every word form encoded according to the MSD tagset: *carte*: Ncfsrn, *cărți*: Ncfprn, *cărții*: Ncfs0y. ROG has to identify the root(s) of the word forms group *cart*, *cărț* and the paradigm(s) that match all the word forms in the group. In order to do that, we conceptualize the word forms group as a list of word forms w_i each having lemma l and some specific MSD tag M_i :

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

From the file ROPMORPH we computed, a list L_2 of all possible suffixes (s) in Romanian, together with their associated MSD tags (M) and paradigms₂ identifiers (p). Thus, for each w_i we can find the set S_i of all the triplets $(s_j, M_j, p_j) \subset L_2$, s_j is a suffix of w_i . At this stage, for every inflected form in the list L_1 , we extract the set of all paradigms to which the word form can be associated by its suffix/suffixes. The search for possible suffixes is restricted to the morphological features that match the MSD tag of the specific form (i.e. for the word form *cărții* with the associated MSD tag Ncfs0y, we take into account only those branches of the paradigmatic tree that contain all of the following features: **Noun**, **common**, **feminine**, **singular**, **oblique** (genitive/dative) and **definite** (**y**)). The cases when the suffix is NULL are also taken into account.

In order to identify the root(s) of the set $\{w_1, \dots, w_n\}$, for each w_i , we compute the set

$$R_{w_i} = \{w_i - s_1 = r_1, \dots, w_i - s_k = r_k\}$$

This means that for each inflected form, we extract a list of roots by eliminating the suffix specific to every applicable paradigm₂. For every root in every R_{w_i} , we save the associated information: the suffix, the MSD tag and the corresponding paradigm₂ (in other words, a triplet in L_2).

We look up the root(s) common to all the forms in the group that have compatible paradigmatic information. So we compute the set R of roots r that has the following properties:

$$r \in \bigcap_{i=1}^n R_{w_i}$$

and the paradigmatic information p associated to r is the same in every R_{w_i} .

If R has exactly one element r , then the group does not manifest root alternation and r is the root of the whole paradigmatic family. If R has more than one element, we may have encountered a root alternating word or a derivative word (i.e. nouns formed by the suffixation of the verb, augmentative or diminutive words). These kinds of words have particular paradigms documented in the paradigmatic morphology and so they can be correctly characterized by two different roots and two different paradigms. For the lexical suffixes that change the grammar category of the word (*privi+tor+ul*), as suggested before, the selected root is the one with the category compliant with the MSD assigned by the tagger and, therefore, no ambiguity is present. For the preserving category lexical suffixes (such as augmentatives or diminutives), in principle, there might be considered two legal roots. For instance, the word *copilașul* (the small child) may have the following two correct interpretations: i) *copil+aș+ul* – \$nomdimasc2 or ii) *copilaș+ul* – \$nomasc8. However, for the sake of morpho-syntactic tagging, as the grammatical suffix carries contextual relevant information (e.g. gender), the selected root will always be considered at the grammatical suffix boundary².

If R has no element, we analyze again all the R_{w_i} and extract the pairs of roots (r_a, r_b) that are elements of different R_{w_i} but share the same paradigmatic information p . This is a clear case of root alternation and can be completely solved for nouns and adjectives but, usually, is much more difficult to deal with it for irregular verbs (the likelihood that all relevant forms of the verb, containing the alternate roots, co-occur in the text is very low).

During the process of root(s) identification, the paradigmatic information p is maintained and sometimes used (in case of root alternation). When the process is finished, every root in the R set is associated with the correct paradigm₂ p . If the number of available word forms is not enough to uniquely identify the correct paradigm₂, we have to deal with a set of possible paradigms₂ PAR . In this case, we generate all the possible word forms for all the paradigms₂ in PAR and use an in house library that extracts information from the Google™ search engine to count, for every word form in every paradigm₁, the number of occurrences on the web. Discarding word forms which appear less than 10 times, every paradigm₁ with the identifier $p_i \in PAR$ is scored with respect to the sum of the web occurrences of its word forms and the highest scored paradigm₂ p is chosen.

² For instance, the word *căsoiului* (of the big house) could be arguably interpreted as *casa* (feminine noun) + *oi* (augmentative) + *ului* (masculine, oblique case, definite form). Any modifier of the word *căsoiului* should have the masculine gender, in spite of the semantic gender of the word (which is feminine).

4.2 Paradigm identification for a group containing a single word form

The possible paradigms₂ for a new word are predicted based on the similarities between the ending of its presumed lemma and the endings for the lemmas in the dictionary. Similarly, we predict the roots looking at the ending of the roots of the lemmas that we already identified as similar with our lemma. For instance, searching for lemmas having the same ending as the word *fanzin* (inexistent in the ROG dictionary), one gets the following results:

```

bazin    noun    zin    $nomneul
magazin  noun    zin    $nomneul
mezin    noun    zin    $nommasc8
muezin   noun    zin    $nommasc8
sarazin  noun    zin    $nommasc8

```

Thus, *fanzin* may be inflected according to two paradigms₂: \$nomneul or \$nommasc8. All the found examples show regular behaviour, with the root identical to the lemma form. Hence, we set *fanzin* as root and generate the paradigms₁ for \$nomneul and \$nommasc8 respectively. After the possible wordforms have been generated according to the two paradigms₂, we apply the Google filtering method from the previous section to select only one paradigm₁. Thus, for this example, given the much more Google evidence, the winning model will be the \$nomneul paradigm₂.

5. Evaluation

Currently, the lexicon of the TTL tagger contains over 800,000 entries and it was built starting from a lexicon containing 450,000 hand validated entries by application of the procedure described in the previous sections on large and clean fiction and journalistic corpora. Each entry contains a word form along with its lemma and POS tag (MSD). With such a large word form lexicon, extracted from carefully edited texts, TTL has little chance to encounter unknown words. However, when dealing with other registers and less carefully edited texts (such as web data) the frequency of unknown words proves to be significant (almost 2%) and as such, not only a source for propagating tagging errors, but also a valuable source of word form lexicon extension.

We have randomly collected 6 Romanian texts belonging to different domains from the Internet totaling approximately 9.5K tokens and computed statistics on the number of tokens and out of these, on the number of unknown words. For the latter, manually analyzed, we are also interested in the POS tagging and lemmatization accuracy. The results of the experiment are summarized in Tables 1 and 2.

	Tokens	Unknown
Philosophical	1922 / 880	26 / 24
Comp. Sci.	1018 / 488	26 / 22
Medical	2601 / 1002	106 / 73

Religious	1312 / 540	10 / 10
Journalistic	1080 / 527	2 / 2
Encyclopedic	1559 / 737	12 / 12
TOTAL	9492 / 4174	182 / 143

Table 1: The proportion of unknown words from 6 randomly gathered Romanian texts

	Spelling Errors	POS Errors	Lemma Errors
Philosophical	0	2	4
Comp. Sci.	8	5	9
Medical	1	3	6
Religious	1	1	2
Journalistic	0	0	0
Encyclopedic	4	5	6
TOTAL	14 (9.79%)	16 (11.18%)	27 (18.88%)

Table 2: Spelling, POS and lemmatization errors for unknown words

In Table 1, in **Tokens** and **Unknown** columns, the first figure is the number of tokens and the second one (separated by '/') is the number of unique tokens. The **Unknown** column presents the number of word forms that were not seen in the training data (corpora + 800,000 word form lexicon) and from these, the **Spelling Errors** column from Table 2 counts the number of spelling errors. The last two columns of Table 2 count the POS tag and lemmatizing errors of the unknown words. Thus the POS accuracy on unknown words is $100 - 11.18 = 88.82\%$ and the lemmatization accuracy on the same word form set is 81.12% . Obviously, these numbers are affected by the spelling errors, which produced both POS tagging and lemmatization errors.

We ended up with 143 unknown word forms from which, 14 were incorrectly spelled and as such generated more than half of the lemmatization errors (14 from 27) and also almost half of the tagging errors (7 from 16). We grouped all 143 word forms by lemma and part of speech (step 2 of the lexical acquisition procedure). Each group was fed into the ROG morphological analyzer which first identified the paradigm₂ of the group and then generated the full set of inflected word forms according to it (step 3). The incorrectly spelled/POS tagged/lemmatized word forms, all together 33 word forms, were classified in single token classes for which ROG was not able to select any paradigms₂. For the remaining 110 word forms, we identified groups containing one, two or three word forms (104 such groups). Table 3 shows the number of groups for which ROG has been able to uniquely identify the proper paradigm₂ (**Unambiguous** column, 78 groups) and the number of groups for which this was not possible (**Ambiguous** column, 26 groups). Column **1 wf** displays the number of groups containing only one word form and column **>1 wf** shows the number of groups with more

than one word form.

	Unambiguous		Ambiguous	
	1 wf	>1 wf	1 wf	>1 wf
Nouns	23	2	10	1
Adjectives	53	0	5	9
Verbs	0	0	1	0

Table 3: Paradigm₂ identification statistics

Out of 104 word form groups, 78 (75%) were assigned a unique paradigm₂. For each such group, ROG generated 21 word forms for adjectives and 10 word forms for nouns (including the seed word forms) resulting in an addition of $25 \times 10 + 53 \times 21 = 1363$ new word forms to the TTL lexicon. An example of a complete paradigm₁ is given below (seed word forms are marked with '>>>'):

minicalculatorul	Ncmsry	\$nomneul
minicalculatorului	Ncmsoy	\$nomneul
minicalculatorul	Ncmsvy	\$nomneul
minicalculatorule	Ncmsvy	\$nomneul
>>minicalculatoare	Ncfp-n	\$nomneul
minicalculatoarele	Ncfpry	\$nomneul
minicalculatoarelor	Ncfpoy	\$nomneul
minicalculatoarele	Ncfpvy	\$nomneul
minicalculatoarelor	Ncfpvy	\$nomneul

With respect to the ambiguous paradigm₂ filtering, out of 26 word form groups with the same lemma and part of speech, the Google filter was able to validate only one paradigm₂ in 9 cases resulting in an addition of 97 word forms to the TTL lexicon. For the rest of 17 groups, after manual inspection, we concluded that the correct paradigm₂ was identified in all cases but that the Google filter was not able to successfully select it from the rest of paradigms₂ matching seed word forms.

6. Conclusions

We have described an automatic procedure for generating new POS tagging lexicon entries beginning from unknown words encountered when POS tagging new texts. The method relies on a morphological generator that given a group of word forms with the same lemma and part of speech, generates the whole paradigm₁ of the group along with POS tag information for each generated word form. Given several such paradigms₁, one can construct intra-categorical ambiguity classes (ICA) for every generated word form and, inferring lexical distributions similar to other ICAs for rare words, one can update the POS tagger lexicon with new entries without having to train the tagger on POS tagged and validated corpora.

Until now, in order to eliminate the risk of erroneous entries, the new entries were added to the tagger lexicon in a supervised way. However, given that Google filtering (although time consuming) eliminates most incorrectly spelled/POS tagged/lemmatized word forms, and that the

few potentially surviving erroneous word entries are almost harmless for the accuracy of the tagger in processing clean texts, we plan to incorporate this module as a background processing tool to the TTL web service (Tufiş et al., 2008). In the first phase the unsupervised automatic lexicon update will consider only the case of unambiguous paradigm₂ identification. The Google filtering for ambiguous paradigm selection will remain an off-line supervised operation as it is a time consuming procedure. At a later stage, we plan to eliminate Google filtering transferring its function to a Markov model trained on Romanian word forms.

7. References

- Brants, T. (2000). TnT – A Statistical Part-Of-Speech Tagger. In *Proceedings of the 6th Applied NLP Conference ANLP-2000*. Seattle, WA, pp 224–231.
- Brill, E. (1996). A Simple Rule-Based Part-Of-Speech Tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*. Trento, Italy, pp 152–155.
- Dermatas, E., Kokkinakis G. (1995). Automatic Stochastic Tagging of Natural Texts. *Computational Linguistics*, vol. 21, Number 2, pp. 137–164
- Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. PhD thesis (in Romanian). Romanian Academy, Bucharest.
- Irimia, E. (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*. Poznań, Poland.
- Ratnaparkhi, A. (1998). Maximum Entropy Models for Natural Language Ambiguity Resolution. PhD thesis. University of Pennsylvania, Philadelphia, PA.
- 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, UK.
- 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., Ceaşu, A., Ştefănescu, D. (2008). RACAI's Linguistic Web Services. In this volume.