Prototype-based Active Learning for Lemmatization

Walter Daelemans
Centre for Dutch Language and Speech (CNTS)
University of Antwerp
Antwerp, Belgium
walter.daelemans@ua.ac.be

Hendrik J. Groenewald
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
handre.groenewald@nwu.ac.za

Gerhard B. van Huyssteen
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
gvhuyssteen@csir.co.za

Abstract

Annotation of training data for machine learning is often a laborious and costly process. In Active Learning (AL), criteria are investigated that allow ordering the unannotated data in such a way that those instances potentially contributing most to the speed of learning can be annotated first. Within this context we explore a new approach that focuses on prototypicality as a criterion for the selection of instances to act as training data in order to optimize prediction accuracy. In parallel with the prototype-based active classification (PBAC) approach of Cebron & Berthold (2009), we investigate whether the basic PBAC assumption rings true for linguistic data. The NLP task we address is lemmatization, the reduction of inflected word forms to their base-form. We operationalize prototypicality as features (i.e. word frequency and word length) of the already available training data items, and combine this with a measure of uncertainty (entropy). We show that the selection of less prototypical instances first, provides performance that is better than when data is randomly selected or when state of the art AL methods are used. We argue that this improvement is possible due to the fact that language processing tasks have highly disjunctive instance spaces, as there are often few regularities and many irregularities.

Keywords

Active Learning; Prototype Theory; Lemmatization; Afrikaans.

1. Introduction

Supervised machine learning techniques are still superior to unsupervised machine learning techniques for many NLP tasks. However, annotation of training data is often a laborious and costly process. In Active Learning (AL) [1,2] criteria are investigated that allow ordering the unannotated data in such a way that those instances potentially contributing most to the speed of learning can be annotated first. Rather than relying on random samples to act as instances in training data, AL entails the selection of instances to act as training data in order to optimize prediction accuracy. Ideally, AL leads to the creation of a supervised learning classifier at a fraction of the annotation effort needed when selecting new training items to be annotated randomly, and without loss in accuracy. Also in domain adaptation, AL has been proposed [3] as a feasible approach. Research on AL centers around the development and comparison of different approaches that could be used to order the available unannotated examples in such a way that those selected first are the ones most informative for learning. Another research area is the design of suitable stopping criteria.

In this paper, we explore a new approach that focuses on prototypicality (i.e. the degree to which some examples are better, more representative examples of a category than others) as a criterion for ordering the data. In parallel with the prototype-based active classification (PBAC) approach of Cebron & Berthold [4], we investigate whether the basic PBAC assumption rings true for linguistic data. We operationalize prototypicality on the basis of different features of the already available training data items and show that the selection of less prototypical instances first provides performance that is better than when data is randomly selected or when state of the art AL methods are used (i.e. a committee-based entropy method). The NLP task we address is lemmatization, the reduction of inflected word forms to their base-form.

In Section 2 work related to Prototype Theory, Active Learning, and lemmatization is discussed. Section 3 outlines our approach, Section 4 describes our experiments on Afrikaans lemmatization and the remainder of the paper discusses these results.

2. Related work

2.1 Prototype Theory

In studies on human cognition, prototypes have been studied for many years [5]. Prototype effects are especially prevalent in language structure and language usage, and have been a central topic in the Cognitive Linguistics paradigm [6,7]. It is widely accepted that language structures (including lexical items) show prototype effects: some instances are better examples than others. For example, with regard to plural formation the {s} morpheme (like in *tables*) could be considered more prototypical than the {en} morpheme (as in *oxen*); likewise, the lexical item *chair* would probably be considered to be a more proto-

International Conference RANLP 2009 - Borovets, Bulgaria, pages 65–70

typical English lexical item than, say, *cache*. In both these cases frequency plays a central role in determining which example is more prototypical than the other (e.g. *chair* is more frequently used than *cache*). When working with linguistic data for natural language processing (NLP) purposes, we can therefore safely assume that some examples in a data set (of unknown examples) are better examples, or more representative, than others.

Various interdependent physiological, referential, statistical and/or psychological factors determine prototypicality [7]; in our current research we focus on frequency (i.e. most commonly or productively used; cf. the statistical hypothesis in Prototype Theory), and size/length (i.e. prototypical concepts are often represented by shorter words; cf. insights on basic-level effects in Prototype Theory [6]). Other factors include membership function (i.e. centrality and salience within a family resemblance model), activation time (i.e. time to process/classify/identify), association/chaining (i.e. the link between form, function, and meaning), conventionalization (i.e. how well-known a word is), acquisition (i.e. more prototypical items are learned first), etc. [6]). These factors are not considered in our current research, but could also be operationalized in future work.

2.2 Active Learning

Predominant approaches to AL include uncertainty-based sampling [8], Support Vector Machine methods [9], and query-by-committee [2]. The latter is a popular method in Active Learning, where entropy computed on the basis of a committee of classifiers is used as a criterion for sample selection.

Recently, Cebron & Berthold [4] presented a novel approach to AL, which they call prototype-based active classification (PBAC). In their algorithm a new, labeled prototype is added in each learning iteration to fine-tune the classification of the datasets; prototypical (i.e. representative) examples are selected first, and examples at the classification boundary (i.e. less prototypical examples) are only selected/focused on automatically when it becomes necessary. In all their experiments, they only use non-linguistic data.

In the PBAC algorithm the relative importance of each data point is calculated as a combination of (a) its representativeness of the data set as a whole (i.e. based on density estimates on the unannotated data), and (b) the uncertainty of a classifier to assign a class to it (i.e. measured as entropy – based on the annotated data – that is inversely related to the voting confidence of the classifier). These two measures are then combined as a new data selection criterion (i.e. the uncertainty distribution), which is calculated as the weighted sum of the representativeness (called potentials) and the classification uncertainty, to the extent that "the remaining potential on the data point still prevents unrepresentative samples from being chosen", which "helps to prevent selection of rare or borderline cases"; see Mazzoni et al. [10] for the detrimental effects of choosing irrelevant data points in AL). Thus, the uncertainty distribution is being used to choose prototypical examples for classification, an approach which promises, with regard to non-linguistic data, to outperform AL with random initialization and closest-to-boundary selection; the algorithm also proved to be stable, and reaches levels of accuracy close to the final one after only a few iterations.

One central aspect of Prototype Theory that is important for the PBAC algorithm is the radial model of categorization. Lakoff [6] makes it clear that "the center, or prototype, of the category is predictable. And while the non-central members are not predictable from the central members, they are 'motivated' by it, in the sense that they bear family resemblances to it." Hence, in the PBAC algorithm, the value of the radius of the neighborhood (i.e. a positive constant defining a neighborhood) is one of the parameters that determine the performance of the algorithm. Cebron & Berthold [4] found that a "...larger radius seems to be beneficial in the first iterations, whereas a small radius leads to more regions with high potential. This causes more exploration and leads to a more detailed (but slower) exploration of the datasets, which proves beneficial in later iterations".

In their conclusion, Cebron & Berthold [4] indicate that future work could include tuning the parameters of the PBAC algorithm to a specific problem; in this research, we look at a specific linguistic problem, viz. lemmatization.

2.3 Lemmatization

AL has been applied to a large range of natural language processing (NLP) tasks, including document classification ([9], Part-of-Speech tagging [11], and parse selection [12]. To our knowledge, no literature has been published on using AL in the development of lemmatizers.

Lemmatization is a common NLP task for most languages, and can simply be defined as "a normalisation step on textual data, where all inflected forms of a lexical word are reduced to its common headword-form, i.e. lemma" [13]. For example, the grouping of the inflected forms *swim*, *swimming* and *swam* under the base-form *swim* is seen as an instance of lemmatization. The last part of this definition applies to this project, as the emphasis is on recovering the base-form from the inflected form of the word. The base-form or lemma is the simplest form of a word as it would appear as headword in a dictionary.

Our experiments in this paper deal specifically with lemmatization for Afrikaans. Inflection is a productive, but rather simple (in comparison to languages like Spanish or Finnish) morphological process in Afrikaans, with nine basic categories of inflection, viz. plural, diminutive, comparative, superlative, partitive genitive, infinitive, past tense, participle, and attributive. The *-e* suffix is by far the most frequent affix, occurring across many of these inflec-

tional categories [14]. We could therefore predict that words ending on *-e* would be prototypical examples of inflected words in Afrikaans.

3. Using prototypes in AL

3.1 Assumptions

The broad aim of our research is to investigate whether the basic PBAC assumption rings true for linguistic data. Note that we don't implement the PBAC algorithm directly; our implementation was developed in parallel with the research of Cebron & Berthold [4], and is merely related to and compatible with the broad approach taken in the PBAC algorithm. As such, our research can be seen as a contribution towards a better understanding of the representativeness parameter in the PBAC approach.

A central assumption of the PBAC approach is that less prototypical cases (i.e. peripheral cases or exceptions) are not important for machine classification, since they do not contribute much information to the construction of a global model. This contrasts directly with our view that, with regard to linguistic data, less prototypical instances are in actual fact important for learning. Our view is based on two grounds: firstly, it is generally accepted in Cognitive Linguistics [6] that outliers contribute as much to the construction of cognitive models as central members (see Section 2.2); hence the interest that Cognitive Linguistics takes in studying not only prototypical instances, but also those peripheral, less prototypical instances of language usage [15]. Secondly, in memory-based language processing [16] it has been argued, on the basis of comparative machine learning experiments on natural language processing data, that exceptions are crucial for obtaining high generalization accuracy. It therefore seems as if the assumption of the PBAC approach is at odds with what is widely believed about natural language data.

For purposes of this paper, our assumption is that long words (e.g. *manifestations*) are less prototypical than short words (e.g. *chair*); likewise, we assume that low frequency words (e.g. *cache*) are less prototypical than high frequency words (e.g. *chair*). (See 2.1 above for a motivation of these assumptions.)

3.2 Hypothesis

Based on our above stated point of view, we hypothesize that less prototypical linguistic examples should provide better results quicker in AL; this is in contrast with the PBAC approach that would predict that more prototypical instances should provide better results quicker.

Our basic hypothesis is that, contrary to what is expected from the PBAC approach, adding less prototypical instances to a baseline classifier (seeded with randomly selected data) at the start of the learning process has a bigger impact than adding prototypical instances (specifically with regard to linguistic data). The reason for this is that less prototypical instances (in our case long, low frequency words, such as *manifestations*) contribute new information to the classifier, and are therefore more informative for learning than prototypical instances (i.e. short, high frequency words, such as *chair*).

Similarly, words with high entropy should provide more information to the construction of a classification model than words with low entropy. Hence, using long, low frequency words with high entropy should provide better results in AL.

3.3 Approach

Recall that in the PBAC approach the criterion for data selection is calculated as the weighted sum of (1) the representativeness, and (2) the classification uncertainty.

With regard to (1), we must determine for a certain dataset and/or a certain task what factors determine the representativeness (i.e. prototypicality) of category members. These factors must then be operationalized as density estimates on the unannotated data so that it could be used to select the data point with the highest estimate as prototype. Such operationalizations could be implemented as features of the training instances, or otherwise as organizational principles of the data set. In our current research, our experiments are geared towards the exploration of word frequency and word length as density estimates in the task of lemmatization (see Section 3).

With regard to (2), we follow the PBAC approach by using entropy as an indicator of the degree of uncertainty or disagreement among different classifiers to assign a class to it, where entropy is inversely related to the voting confidence of the classifier. High entropy therefore indicates higher levels of uncertainty. Words with high entropy are believed to be less prototypical and should therefore be beneficial at the start of the learning process. Entropy correlates with exploitation in the PBAC approach.

Our criterion for data selection, called the combination distribution (CD), is calculated as a weighted combination of word frequency, word length and entropy. The formula for the combination distribution is as follows:

$$CD = w_1(Entropy) + w_2(WL) + w_3(WF) \quad (1)$$

where w_1, w_2 *and* w_3 indicate the different weights.

4. Experiments

4.1 Setup

For purposes of our experiments, we want to construct a model of the data where we can (a) distinguish between more or less prototypical instances; and (b) select different subsets of the data for AL purposes. In this way, we want to explore, for the task of lemmatization, word frequency and word length as parameters of representativeness, and entropy as an indication of classifier uncertainty. In our experiments, we operationalize these three factors in the following way.

Concerning word frequency, the data set (see 4.2) is ordered based on frequency counts for the words in the data

set, which were calculated on the basis of frequency counts (on types) obtained from an Afrikaans corpus containing more than 160 million tokens [17]. It should be noted that, if frequency is viewed as a density estimate of the distribution of instances in memory, it means that such a distribution will have a high density with regard to low frequency words, which could be viewed as a large group of similar training instances (i.e. in the core of a radial representation). High frequency words (which are less frequent in the data set) will appear, on the other hand, outside the boundaries of this core. Since words appearing inside the boundaries are deemed to be more prototypical than words that fall outside of the boundaries, this representation of the word frequency is so to speak the inverse of a commonsense representation; if viewed as a density estimate, low frequency words are therefore actually more prototypical than high frequency words.

The same argument is also valid when considering word length as a feature. Longer words are less prototypical than shorter words, since the data set contains larger numbers of short words grouped together, than longer words. Short words will appear inside the boundaries and are therefore viewed as more prototypical than longer words. Word length was calculated by counting the number of characters comprising each word in the data set.

Entropy is calculated on the basis of a class distribution obtained from a committee of three different classifiers, each using a different machine learning algorithm. The algorithms that were used are the default TiMBL implementations of IB1, IGTree and IB2 [18], where *k*=1. *k* refers to the *k*-nearest distances, rather than the *k*-nearest neighbors. This means the class distribution may contain several instances, despite *k* having a value of 1. The class distribution therefore consists of all the classes of the instances contained within a distance of 1 from the classified instance, as indicated by the committee of classifiers. The formula used for the calculation of the entropy of a word is shown in Equation 1:

$$Entropy(w) = -\sum_{i=1}^{n} p(w_i)\log p(w_i) \qquad (2)$$

where n is the number of classes in the distribution and $p(w_i)$ is the proportional number of a particular class relative to the total number of classes in the class distribution output by the committee.

4.2 Data

Data for the Afrikaans lemmatiser was constructed by extracting word-forms that contain substrings that correspond to inflectional affixes (at the surface level) from an Afrikaans lexicon, together with an equal number of instances where lemma and word-form are equal. The extraction yielded 72,226 instances, which were manually lemmatized as training data. Each instance consisted of 20 features (letters of the word-form as separate features).

271 classes were automatically derived by means of a comparison based on the longest common substring of the extracted word-forms and their manually provided lemmas. The classes indicate the transformation that a word-form must undergo in order to obtain its linguistically correct lemma, specifying the character string to be removed, the relative position of the operation (i.e. L (left), R (right) and M (middle)), and the replacement string. If a word-form and its lemma are identical, the class awarded will be "0", denoting the word should be left in the same form. This annotation scheme yields classes like those in the third column of Table 1. The classifiers are not prohibited from predicting impossible classes (e.g. "Lge>" is not a valid class for the word bote, since the word does not containing the string "ge").

Table 1. Inflected words with their lemmas and classes as found in the Afrikaans training data

Word-form	Lemma	Class
"geel" 'yellow'	"geel" 'yellow'	0
"geslaap" 'slept'	"slaap" 'sleep"	Lge>
"hondjie" 'puppy'	"hond" 'dog'	Rjie>
"bote" 'ships'	"boot" 'ship'	Rte>ot

4.3 Implementation

In all our experiments, we use a *k*-Nearest Neighbor (*k*-NN) approach as learner (i.e. memory-based learning). In this approach, classification of a new instance is based on local extrapolation from memorized similar instances. We employed the standard *k*-NN algorithm, IB1, with default algorithmic parameter settings as implemented in the TiMBL software package [18].This package also contains implementations of IGTree (a decision tree based approximation of *k*-NN, and IB2 (a variation of IB1 in which only instances misclassified with the current contents of memory are added to that memory). These variations were used in computing the entropy measure (see 4.1 above).

Experiments were performed by using the entire data set, consisting of 72,226 words, where every word is a single instance in the training data. 10-fold cross validation was used throughout the evaluation process.

We started by training the system with a seed memory (10% of the data set) containing randomly-selected instances. We then arranged the remaining instances in the training data set according to the parameters to be evaluated (i.e. word frequency (WF), word length (WL), and entropy, as well as using the combination distribution (CD) described in 3.3 above). In each case the instances were added both in a high-to-low and in a low-to-high order to the learner in sets of 6,500 instances.

We are interested in obtaining a learning curve with a steeper gradient than that of the baseline experiment (indicated as "Random" in Figure 1) in order to show that our

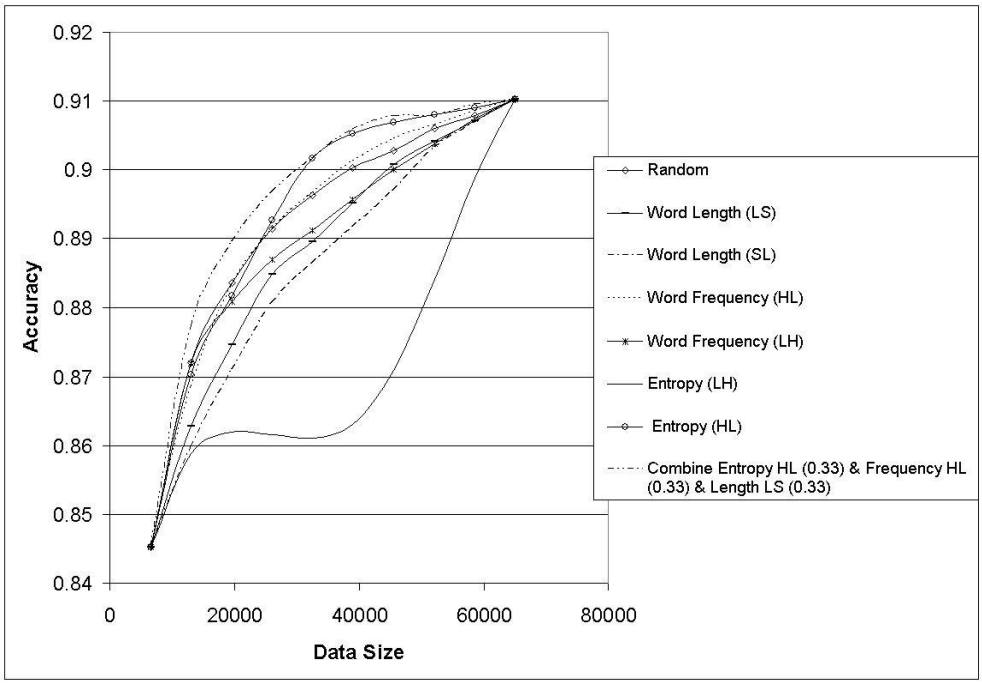

Figure 1. Learning curves

data selection method performs better than a random selection. We also want to compare our method with a standard state of the art approach in AL, which we consider to be the committee-based entropy method (indicated as "Entropy (HL)" in Figure 1).

5. Results

The learning curves for the various parameters, the combination distribution, as well as a random distribution (the base-line experiment) are indicated in Figure 1. Using the same set of randomly selected instances for computing the accuracy obtained in the first fold of every 10-fold cross-validation experiment results in the learning curves of all experiments starting at the same point on the graph. For the calculation of the combination distribution (see Equation 1), we also experimented with different weight values, but found that the best combination distribution curve was obtained with equal weight values.

Figure 1 shows that adding unprototypical words to the seed memory at the start of the learning process clearly outperforms the experiments where prototypical words were added first. This is true for both the evaluated parameters and can be observed by comparing the unprototypical learning curves (e.g. Word Frequency [Low to High] and Word Length [Long to Short] with the prototypical curves (e.g. Word Frequency [High to Low] and [Word Length Short to Long]). (With regard to the learning curves representing word frequency, refer to 4.1 for an explanation of why [High to Low] is indicated as better than [Low to High] in Figure 1.) Another finding from Figure 1 is that the combination distribution (CD) with equal weights yields a steeper learning curve than any of the other individual parameters, including that of the committee-based entropy method.

Even though the gains of this approach seem small at first, the significance of our results is appreciated when considering the difference in the number of training instances required by each of the distributions to reach a certain accuracy figure. The combination distribution, for example, requires 19,864 instances to achieve an accuracy of 0.89, compared to the 24,656 instances required by the random distribution to achieve the same accuracy. In this case it means that 4,792 less instances are needed when using the combination distribution, representing a significant saving in terms of the annotation effort.

6. Discussion

Entropy computed on the basis of a committee of classifiers is a popular method for selecting instances in AL (see 2.2 above). Our results indicate that the performance of this method can be improved by combining entropy with other parameters of representativeness, selected on the basis of Prototype Theory. This approach also improves notably upon the random baseline. However, contrary to intuition and to results for AL in other areas than language processing, it is the selection of less prototypical instances first that provides the best improvement, both for word frequency and word length. A possible explanation for this is that language processing tasks have highly disjunc-

tive instance spaces, as there are often few regularities and many irregularities, and pockets of exceptions [16]. Starting from a random seeding may already provide sufficient structure (as there is little structure of the instance space), and in such a case, finding the boundary cases is as least as important as finding the central cases of classes. A meta-learning analysis in which the prototype-based selection approach is investigated for a larger range of language processing tasks with class systems of different complexities could shed more light on this issue.

Our research shows in any case that a prototype-based selection approach indeed improves upon a committee-based and random baseline approach, but not necessarily the way expected in the PBAC approach.

7. Conclusion

In this paper we have shown that a prototype-based selection strategy for AL improves upon both random baseline and entropy-based committee approaches. Interestingly, for our language processing problem, prototypicality works not in the way expected and documented in other research (more prototypical instances first is better than less prototypical first), but exactly the other way round. A possible explanation for this is the lack of structure found in instance spaces of language processing problems, which typically show large disjunctivity.

Future work includes the investigation of more natural language processing tasks with more operationalizations of prototypicality to investigate whether our findings indeed point to a different superior selection strategy for language processing tasks than for other types of problems. Another aspect to be investigated is the interaction of this approach with possible stopping criteria for AL.

8. Acknowledgments

Van Huyssteen is jointly affiliated with the Human Language Technologies Research Group, Meraka Institute, CSIR, Pretoria, South Africa. Support by the CSIR is hereby acknowledged.

We would like to extend our gratitude to JA Pienaar, who was involved in the initial conceptualization of this project.

Part of this research was made possible through a research grant by the South African National Research Foundation (GUN: 65462).

9. References

[1] Cohn, D.A., Ghahramani, Z. & Jordan, M.I. 1996. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*. 4: 129-145.

[2] Freund, Y., Seung, H.S., Shamir, E. & Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Machine Learning*. 28: 133-168.

[3] Chan, Y. & Ng, H. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Conference of the Association for Computational Linguistics*. pp. 49-56.

[4] Cebron, N. & Berthold, M.R. 2009. Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*. 18: 283-299.

[5] Rosch, E. & Lloyd, B.B. (eds.). 1978. *Cognition and categorization*. Hillsdale: Lawrence Erlbaum.

[6] Lakoff, G. 1987. *Women, Fire, and Dangerous Things*. Chicago: University of Chicago Press.

[7] Geeraerts, D. 2006. *Words and other wonders: papers on lexical and semantic topics*. Berlin: Walter de Gruyter.

[8] Hwa, R. 2000. Sample selection for statistical grammar induction. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*. pp. 45–52.

[9] Schohn, G. & Cohn, D. 2000. Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann. pp. 839-846.

[10] Mazzoni, D., Wagstaff, K.L. & Burl, M. 2006. Active learning with irrelevant examples. In *Proceedings of the 17th European Conference on Machine Learning*. pp. 695-702.

[11] Engelson, S. & Dagan, I. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of the 34th Meeting of the Association for Computational Natural Language Learning*. San Fransico: Morgan Kaufmann. pp. 319-326.

[12] Baldridge, J. & Osborne, M. 2004. Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain. pp. 9-16.

[13] Erjavec, T. & Džeroski, S. 2004. Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*. 18(1): 17-40.

[14] Groenewald, H.J. & Van Huyssteen, G.B. 2008. Outomatiese Lemma-identifisering vir Afrikaans [Automatic Lemmatisation for Afrikaans]. *Literator*. 29(1): 65-91.

[15] Langacker, R.W. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford: Oxford University Press.

[16] Daelemans, W. & Van den Bosch, A. 2005. *Memory-based language processing*. Cambridge: Cambridge University Press.

[17] Pharos Dictionaries. 2007. *Media24 Corpus*. Media24: Cape Town.

[18] Daelemans, W., Zavrel, J., Van der Sloot, K. & Van den Bosch, A. 2004. *TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide*. ILK Technical Report 04-02. Tilburg: University of Tilburg.