AUTOMATIC TEXT SEGMENTATION OF AFRIKAANS USING MEMORY-BASED LEARNING

Martin J Puttkammer & Gerhard B van Huyssteen

Centre for Text Technology (CTexT), North-West University,
Potchefstroom, 2531, South Africa

{martin.puttkammer; gerhard.vanhuyssteen}@nwu.ac.za

Abstract

A text segmentor for the identification of sentences, named entities, words, abbreviations and punctuation in Afrikaans texts is described in this paper. The task is viewed as an integrated annotation process, and a memory-based classifier is hence trained to perform the task. Compared to baseline results for other languages, the classifier performs quite well (overall *f*-score of 97.79% on the full tag set), especially in consideration of the relatively small training data set used. The paper concludes with directions for future research.

1 Introduction

Text pre-processing is a very basic though important process in almost all text-based human language technology (HLT) applications (such as spelling and grammar checkers, information retrieval systems, etc.) [1]. Pre-processing tasks such as sentencisation (i.e. the identification of sentence boundaries) and tokenisation (i.e. the identification of words and punctuation) are generally assumed quite trivial (e.g. [2], [3]), while the recognition of multiword expressions (idioms, fixed expressions and phrasal verbs) and other specialised units (such as named entities, abbreviations, email addresses, dates, etc.) are seen as more challenging (see for example various annual challenges and shared tasks at major conferences, such as the Conference on Computational Natural Language Learning (CoNLL), and the Message Understanding Conference (MUC)). Hence, these different sub-processes are mostly treated as distinct, ad hoc processes in HLT applications, with different modules and different techniques used for different tasks: for example, a rather simple rule-based module would be used for sentencisation (e.g. [4]), while a more advanced statistical-based module would be used to handle the task of named-entity recognition (NER; e.g. [5]).

Since no research has been done on the development of modules for text pre-processing in Afrikaans, the aim of this research is to develop such modules for Afrikaans. Although Afrikaans does not necessarily pose any unique structural problems with regard to text pre-processing (i.e. the orthographical structure of Afrikaans is similar to languages such as English, Dutch, or German), Afrikaans is generally considered to be a so-called "resource-scarce language" (i.e. very few and relatively small corpora and other resources exist for Afrikaans). In as such, the development of text pre-processing modules could be interesting from a more general methodological perspective, especially with regard to statistical approaches. Furthermore, contrary to the general practice sketched above, we take a somewhat different stance in this research when we depart from the following two presuppositions:

(a) We see text pre-processing as an integrated process, consisting of various interconnected sub-processes to identify text segments such as sentences, words, punctuation, and various named entities (e.g. proper names, dates, numerical expressions, etc.) in running text. Following [1], we identify three basic, sub-processes, viz. sentencisation, tokenisation, and named-entity recognition, and we use the term *Text Segmentation* as an overarching, generic descriptor for these various interrelated sub-processes. The aim of this research is therefore rather to develop a single text segmentor for Afrikaans, instead of various different modules for text pre-processing, as stated earlier. Insofar we know, this is a novel approach to text pre-processing.

(b) We also view text segmentation as a text-enriching annotation process, where tags are assigned to the different text segments through a process of disambiguation (e.g. resolving the ambiguity of full-stops, capital letters, and white spaces, while also assigning tags to unambiguous elements). For example, instead of producing a list of tokens as output for the sub-process of tokenisation, text segments are rather tagged as words, punctuation marks, abbreviations, named entities, etc. In other words, a sentence like *My name is Lucky.* could be tagged as follow:

> *My*<WORD; SENTENCE BEGIN> *name*<WORD; SENTENCE MIDDLE> *is*<WORD; SENTENCE MIDDLE> *Lucky*<NAMED ENTITY; SENTENCE END> .<PUNCTUATION; SENTENCE END>

Although this approach in itself is not necessarily novel or innovative, we will indicate that our tag set is linguistically and theoretically well-motivated, and that it supports our first presupposition well.

Subsequently, in the next section we will give an elaborate discussion of our tag set, since this is fundamental to the approach we take. In section 3 we present our solution and methodology, while section 4 discusses the results of our experiments. We conclude with a critical evaluation and ideas for further research.

2 Tag Set

Since text segmentation is viewed as an annotation process in this research, we need to construct a tag set for tagging the different text elements; in this case specifically sentences, words, abbreviations, punctuation, and named entities. In constructing our tag set, we followed the EAGLES guidelines [6] for Part-of-Speech (POS) tags, and aligned it with the POS tag set for Afrikaans as explicated by [7]. This ensures, inter alia, the reusability and portability of the tags across languages (through the use of so-called intermediate tags – included in the tables), as well as the possibility to expand the tag set

Main	Sub	Specific	Position	Intermediate
W			SB	W001
			SM	W002
			SE	W003
	E		SB	W101
			SM	W102
			SE	W103
A			SB	A001
			SM	A002
			SE	A003
	T		SB	A101
			SM	A102
			SE	A103
P			SM	P002
			SE	P003
	P	L	SB	P111
			SM	P112
		R	SM	P122
			SE	P123

Table 1: Tags for WORD, ABBREVIATION, AND PUNCTUATION

in future to a larger, or more specific tag set. The tags are structured to comprise of four "slots" (viz. (1) main category, (2) sub-category, (3) specific category, and (4) sentence position), which allows us to specify the level of granularity when annotating a text. For example, one can decide to tag the word *Lucky* in the example above "coarsely" as a NAMED ENTITY, or otherwise very specifically as a NAMED ENTITY of the kind NAME of a PERSON, occurring at the SENTENCE END.

Our tag set comprises of 51 tags, structured into four main categories, namely WORD, ABBREVIATION, PUNCTUATION and NAMED ENTITY; in [7], definitions, and elaborate structural descriptions and linguistic motivations for each of these are presented. These main categories are all further particularised in sub-categories and specific categories, while the relative sentence position of the specific text segment is indicated in the last "slot" (i.e. whether the text segment appears at the beginning, in the middle, or at the end of a sentence).

Table 1 gives a summary of the main categories WORD, ABBREVIATION and PUNCTUATION. In the main category WORD, there is a differentiation between COMMON WORD<W> (*ek, stoel*) and ENCLITIC FORM<WEV> (*'n, hy's*), while the main category ABBREVIATION distinguishes between COMMON ABBREVIATION<A> (*a.g.v., bl.*) and TITLE<AT> (*mnr., me.*). The latter category is introduced for purposes of NER, since titles mostly form part of a person's name (e.g. *Mr. Lucky Luke* is a named entity). The main category PUNCTUATION includes the sub-categories PUNCTUATION<P> (.,;), PUNCTUATION LEFT PARENTHESIS<PLP> ({, [) and PUNCTUATION RIGHT PARENTHESIS<PRP> (},]).

The tags we devised for the main category NAMED ENTITY are based by and large on the NER task of the Message Understanding Conferences (MUCs). These conferences produced a scheme for named-entity annotation [9], and offered developers the challenge to evaluate systems for English on the same data. The MUC NER tasks consist of three main categories (with sub-categories) of named entities to be recognised, viz. ENTITY NAME (specifically PERSON, LOCATION, and ORGANIZATION), TEMPORAL EXPRESSION (specifically DATE, TIME, and DURATION), and NUMBER EXPRESSION (specifically MONEY, MEASURE, PERCENTAGE, and CARDINAL NUMBER) [9]. Subsequently, the CoNLL-2002 [10] and CoNLL-2003 [11] NER tasks were also based on this scheme; however, their main categories were PERSON, LOCATION, and ORGANIZATION, with various sub-categories and specific categories under each [12]. They also introduced a fourth main category, MISCELLANEOUS, which is used for named entities such as religions, languages, and wars, as well as for derived forms of named entities (e.g. *African*). For our tag set for NAMED ENTITIES<B> we opted not to include all the finer grained categories of CoNLL, although these categories can be added easily as sub-categories or specific categories, according to specific needs.

Main	Sub	Specific	Position	Intermediate
B	N	P	SB	B111
			SM	B112
			SE	B113
		L	SB	B121
			SM	B122
			SE	B123
		B	SB	B131
			SM	B132
			SE	B133
		A	SB	B141
			SM	B142
			SE	B143

Table 2: Tags for NAME

In Table 2, the sub-category NAME<N> is divided into specific categories PERSON<P>, LOCATION<L> and ORGANISATION/PRODUCT<B> (i.e. extending MUC's category to also include product names, since product names are most often also business names, e.g. *Ford, Nederburg*, etc.). We also include MISCELLANEOUS NAME<A> to accommodate all other names not belonging to these specific categories.

MUC-7's main category TEMPORAL EXPRESSION is in our tag set subsumed under the sub-category NUMBER EXPRESSION<S>, as tokens belonging to this group usually contain digits written in a specific format or some unit of measure. Therefore, as can be seen in Table 3, we distinguish DATE<D>, TIME<T> (which includes MUC-7's DURATION category), PERCENTAGE<P> and MONEY<G> as specific catego-

ries of the sub-category NUMBER EXPRESSION. Tokens containing digits, but not belonging to one of the four specific categories (e.g. measures, cardinal numbers, telephone numbers, identity numbers, etc.) are classified as MISCELLANEOUS DIGIT<A>.

Main	Sub	Specific	Position	Intermediate
B	S	D	SB	B211
			SM	B212
			SE	B213
		T	SB	B221
			SM	B222
			SE	B223
		P	SB	B231
			SM	B232
			SE	B233
		G	SB	B241
			SM	B242
			SE	B243
		A	SB	B251
			SM	B252
			SE	B253

Table 3: Tags for NUMBER EXPRESSION

Although web addresses and e-mail addresses were not included in the above-mentioned shared tasks of MUC and CoNLL, we specifically wanted to recognise these addresses for purposes of information extraction and spam filtering. We therefore include a sub-category INTERNET ADDRESS<I>, with specific categories WEB ADDRESS<W> and E-MAIL ADDRESS<E>, as represented in Table 4.

Main	Sub	Specific	Position	Intermediate
B	I	W	SB	B301
			SM	B302
			SE	B303
		E	SB	B401
			SM	B402
			SE	B403

Table 4: Tags for INTERNET ADDRESS

3 Implementation: Afrikaans Text Segmentor

Following general trends in natural language processing (NLP) with regard to annotation tasks such as POS tagging and NER, we choose a machine learning approach for this annotation process, and more specifically, a memory-based approach. The rationale for this is simply based on the satisfactory results that were obtained by researchers in other, similar annotation tasks (e.g. [5], [13]). Since not all resources available for other languages (e.g. large annotated corpora), the challenge here is to find ways to obtain satisfactory results, given limited resources.

Before a classifier can be trained, however, a few preparatory steps are necessary. These steps, including the step where the classifier is trained, are described next.

3.1 Step 1: Pre-processing

In order to ensure uniformity of the text, the input text is first submitted to a range of replacement processes. These include, amongst other, replacement of all smart quotes with straight quotes, normalising all different forms of the indefinite article (*'n*) to the same form, converting all text to UTF-8 encoding, replacing all white spaces with paragraph marks, etc.

3.2 Step 2: Feature Assignment

The input required by memory-based learners is a sequence of instances (i.e. the tokens), described by a set of features (e.g. whether a word is written with a capital letter, whether it appears in a certain lexicon, etc.) [14]. To automatically assign such features to the data, we use two types of processes: *rules* (i.e. pattern matching with regular expressions), and *lookup* in a variety of lexicons. The outputs of these processes are stored as feature attributes, where the values of the various attributes are numerical: "1" if the token contains the feature, and "-1" if the token does not contain the feature. In total, 32 feature attributes are identified for each token, derived via the processes described next.

The *rules* used are so-called expert rules (or grammar rules), based on regular patterns in the language and text structure. Since this process requires considerable time and precision from the expert, it is limited to the recognition of unknown/novel abbreviations, number expressions, and internet addresses, as well as the identification of word-case characteristics from the specific string (i.e. whether a string starts with a capital letter or not, etc.). A total of 37 regular expressions are implemented in this rule section.

With regard to *lexicon lookup*, one can distinguish between two types of lexica, namely a *common lexicon* (i.e. a comprehensive list of common words, normally written in lower case, including lexemes, inflected forms, derivations, compounds, etc.), and *specialist lexica* (a.k.a. gazetteers – i.e. lists of abbreviations, acronyms, place names, business- and product names, titles, etc.). It is self-evident that the more comprehensive these lists are, the higher the recall and precision of any system will be, even more so in the case of named-entity recognisers (see [15], [16], amongst others).

In this Afrikaans text segmentor, the common lexicon of the *Afrikaans Spelling Checker 3.0* of the North-West University, comprising 314,706 common words, is employed. The fifteen gazetteers we use, consisting of 84,618 words, are also based on specialised sections of the *Afrikaans Spelling Checker 3.0* lexicon, but was divided

into specific categories by hand, and thereafter expanded where necessary and/or possible.

After assigning feature attributes to each token through the processes described above, each token, together with its three preceding and three following tokens (including all their feature attributes) are stored as a single training instance. In other words, each training instance consists of 7 tokens, each with 32 feature attributes; each training instance thus has 231 features (i.e. 7 tokens, plus 7x32 attributes). Lastly, the data is stored in the format required by the machine learner, and subsequently passed on to the machine learner as training data.

3.3 Step 3: Training the Classifier

The classifier is developed using the Tilburg Memory-Based Learner (TiMBL; [17]), a learner in which a variety of learning algorithms are used (including specifically the *k*-nearest neighbour algorithm). The classifier is trained with training data sets, which are explicitly stored in memory. These stored instances are then used to classify new instances based on similar cases (i.e. the *k*-nearest neighbours).

For development data, we compiled a corpus of 1,607 sentences of Internet texts, comprising 40,906 words, of which 3,068 are named entities. The corpus is loosely based on the stratum of the International Corpus of English (Great Britain), and consists of formal texts (e.g. scientific articles, newspapers, etc.) and informal texts (e.g. chat room conversations, personal emails, etc.). It should be noted that this is a relatively small corpus, compared for instance to the German corpus used in CoNLL-2003, which consisted of 12,705 sentences, 206,931 tokens, and 11,851 named entities [11]. Our corpus was semi-automatically annotated with the tags described in Section 2, after which it was thoroughly verified by an external linguist.

Next, the data is randomised (on sentence level) and divided into ten different 90-10% parts, with a view to do tenfold cross validation. Following the methodology of [18] and [19], the 90% parts are used as training data and the 10% parts as test data. Experiments are executed ten times (once for each 90%-10% pair), and results based on the mean score achieved by each classifier on all ten folds are reported. These experiments are done to determine the parameter settings for the best classifier, which will be evaluated on another dataset (see 4 below).

We experimented with four of TiMBL's parameter settings, namely three algorithm types, distance metrics, feature-weighting possibilities and the number of nearest neighbours (*k*=2-7). All possible permutations are used, and 2,700 classifiers are subsequently trained to determine the best parameter settings for the task at hand. The classifier that achieved the highest *f*-score (97.25%) on the complete tag set used the IB1-algorithm (i.e. the classic instance-based algorithm) with the distance metrics set on *M* ("Modified value difference metric"), a feature-weighting possibility *sv* ("Shared Variance"), and a *k*-value of 7. This classifier is used in subsequent evaluations.

4 Evaluation Results

For evaluation purposes, we compiled an evaluation set comprising 4,010 tokens, of which 378 are named entities. This evaluation set consists of texts from three newspapers during the month of November 2006, and includes general news articles, articles from the sport, automobile and financial sections, as well as letters/comments from readers. Both the complete tag set and a simplified tag set (comprising of the main categories, as well as sentence position) have been used. Results on the complete tag set give an indication of the system's capability to differentiate between specific categories, while results on the simplified tag set reflect the system's capability to differentiate between the main categories only. Three metrics are used to evaluate the system, namely precision (number of correct tags over total number of tags predicted), recall (number of correct tags over total number of tags in data) and f-score (harmonic mean of precision and recall).

4.1 Overall Results

Table 5 provides a summary of the overall results of the best classifiers with the complete tag set (51 tags) and the simplified tag set (12 tags).

	Precision	**Recall**	**F-Score**
Complete tag set	97.78%	97.80%	97.79%
Simplified tag set	98.80%	99.20%	99.00%

Table 5: Overall Results

4.2 Sentence Recognition

In the development phase, 120 classifiers (from a total of 2,700) obtained an *f*-score of 100% (with 100% precision and 100% recall) on the task of sentence recognition. On the evaluation set the classifier once again achieves an *f*-score of 100%. This is not surprising, since the task of sentence recognition is generally considered not a difficult one [1]. However, it proves that the way we conceptualised the tag set (i.e. integrating the relative sentence positions in the tags for the various categories) is unproblematic, and that the feature attributes we assign to tokens are effective for detecting sentence boundaries.

4.3 Word, Abbreviation and Punctuation Recognition

Table 6 provides a summary of the results of the best classifier on the main categories WORD, ABBREVIATION, and PUNCTUATION, as well as on their various sub-categories. Overall, these results are on par with baseline results for other languages (cf. [4]), while there is room for improvement in the recognition of abbreviations. This could be done, for example, by continuously expanding the list of abbreviations (to incorporate newly formed abbreviations), as well as to refine the regular expressions that identify probable new abbreviations.

	Precision	**Recall**	**F-Score**
WORD	99.17%	99.58%	99.37%
ENCLITIC FORM	98.80%	89.13%	93.71%
ABBREVIATION	92.86%	61.90%	74.29%
TITLE	90.00%	94.74%	92.31%
PUNCTUATION	97.59%	100%	98.78%
PUNCTUATION LEFT PAR.	100%	100%	100%
PUNCTUATION RIGHT PAR.	92.86%	100%	96.30%

Table 6: Results for WORD, ABBREVIATION and PUNCTUATION

4.4 Named-Entity Recognition

Table 7 provides a summary of the results of the best classifier on NER, using the complete named-entity tag set (33 tags) and a simplified named-entity tag sets (3 tags, viz. NAME, NUMBER EXPRESSION, and INTERNET ADDRESS).

	Precision	Recall	F-Score
Complete tag set	86.76%	85.15%	85.94%
Simplified tag set	97.84%	96.02%	96.92%

Table 7: Results for NAMED ENTITY

If one consider baseline results for the same task in other comparable languages (e.g. in CoNLL-2003, the best Precision obtained for English was 88.99%, and for German 83,87%, using five tags; [11]), our system performs surprisingly well – especially if one takes into cognisance the large differences in training data sizes (mentioned above), as well as the fact that POS data was available in the CoNLL-2003 task.

The relative success of our system could probably be ascribed to the fact that we have access to a very large common lexicon (314,706 common words), as well as extensive and well-structured gazetteers (consisting of 84,618 words). For other languages without such resources, the task will definitely be more challenging. This once again proves the value of comprehensive and specialized lexica as some of the most important resources to be developed for resource-scarce languages.

Yet another contributing factor to the performance of the system could be the large number of features (i.e. 231) that we use, especially in the context of memory-based learning. By increasing the instance space in this way, one succeeds to create more detail around a certain query point in the instance space, thus increasing the chance of a better classification. It would be worth the while to experiment in future with even more features, such as increasing the context to, say, five or six tokens on both sides.

In order to identify specific problems and areas for future improvement, we derived a confusion matrix (given in Table 8) from the test data (n=4,010). From this confusion matrix, it becomes clear that gazetteers need to be refined/expanded further, since tokens appear in the data that are not included in the gazetteers. For example, the large number of person names classified as OTHER NAME<BNA> can be ascribed to the restricted number of indigenous person names in the gazetteers (e.g. *Lebogo, Bosasa*), since these gazetteers mainly contain Afrikaans names and only a limited number of well-known or high profile names from other South African languages. Along the same lines, person names classified as COMMON WORD<W> is encountered when unknown person names appear at the beginning of a sentence (e.g. *Schabir, Wolela*, etc.).

An interesting case is the classification of the preposition *van* (i.e. a common word) as OTHER NAME<BNA>, in the case where *van* appeared between two named entities (*Eleine Roets van Pretoria*). In addition, various tokens that belong to the category OTHER NAMES are erroneously classified as WORD; compare for example the *van* in *Direkteur-Generaal van Grondsake* and the *en* in *Brixton Moord en Roof Orkes*. These problems with regard to prepositions and conjunctions (such as *van* and *en*) will most probably only be resolved when more training data is used.

5 Conclusion

In this paper, we have shown that text pre-processing tasks such as sentencisation, tokenisation, and named-entity recognition, can be successfully handled as an integrated annotation process (what we called here text segmentation). For purposes of this annotation process, we presented a comprehensive tag set that could be transported to other languages, and that could be implemented on various levels of granularity.

We have also shown that, for a resource-scarce language like Afrikaans, one could achieve satisfactory results by only using comprehensive and well-structured lexica, as well as a few expert rules for the training of a classifier. Our overall results (*f*-score of 97.79% when the complete set of tags is used), as well as our results on NER specifically (*f*-score of 85.94% on the complete named-entity tag set), compare well to baseline results for other languages.

Future work includes experiments with other machine-learning algorithms, as well as combination techniques where different classifiers could be combined in various ways to improve results. We will also need to expand the training data, in order to achieve better performance.

Since named entities (specifically from the category NAME) belong to a so-called "open category" (it is expanded productively), it is inevitably also of importance that the gazetteers are constantly updated [20].

References

[1] W. Daelemans, and H. Strik, (eds.), "Het Nederlands in de taal- en spraaktechnologie: prioriteiten voor basisvoorzieningen," [Dutch in language and speech technology: priorities for basic provisions]. *Final report*, (01/07/2002). Dutch Language Union, 2002. [Web:] taalunieversum.nl/taal/technologie/docs/-daelemans-strik.pdf [Date: 2005/12/04].

[2] G. Grefenstette, and P. Tapanainen, "What is a Word, What is a Sentence? Problems of Tokenization," *Third Conference on Computational Lexicography and Text Research*, COMPLEX-94, Budapest, pp. 79–87, 1994 [Web:] http://iling.torreingenieria.unam.mx/lecturasprohibidas/mltt-004.pdf [Date: 31-10-2005].

[3] T. Yamashita, and Y. Matsumoto, "Language Independent Morphological Analysis," *Proceedings of the 6th Applied Natural Language Processing Conference*, pp. 232–238, 2000. [Web:] http://acl.eldoc.ub.rug.nl/mirror/A/A00/A00-1032.pdf [Date: 31-10-2005].

[4] G. Grefenstette, "Tokenization," *Syntactic Wordclass Tagging*," H. van Halteren, (ed.), Kluwer Academic Publishers, Dordrecht, pp. 117-133, 1999.

[5] F. De Meulder, and W. Daelemans, "Memory-Based Named Entity Recognition using Unannotated Data," *Proceedings of CoNLL-2003*, ACL, Edmonton, pp. 208–211, 2003.

[6] EAGLES, "The EAGLES extension to ISO 9126," 1995 [Web:] http://www.issco.unige.ch/ewg95/node15.html [Date: 2002/10/20].

	BNB	BNL	BNP	BIW	BIE	BNA	BSD	BST	BSP	BSG	BSA	P	A	W
BNB	1	1	1			1								
BNL		26				3								1
BNP			127			**22**								**10**
BIW				2										
BIE					2									
BNA	3	1	**5**			91								4
BSD			2				26							
BST								1						
BSP									11					
BSG										5	1			
BSA											34			
P												389		
A		1	1			4							34	1
W			1			1						**9**	1	3182

Table 8: Confusion matrix of 4,010 tokens

[7] S. Pilon, "Outomatiese Afrikaanse Woordsoortetikettering," [Automatic Afrikaans Part-of-Speech Tagging], *MA thesis*, North-West University, Potchefstroom, 2005.

[8] M.J. Puttkammer, "Outomatiese Afrikaanse Tekseenheididentifisering," [Automatic Afrikaans Tokenisation], *MA thesis*, North-West University, Potchefstroom, 2006.

[9] N. Chinchor, E. Brown, L. Ferro, and P. Robinson, "1999 Named Entity Recognition Task Definition," *Unpublished Technical Report*, MITRE and SAIC, 1999.

[10] E.F. Tjong Kim Sang, "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition," *Proceedings of the Conference on Natural Language Learning (CoNLL-2002)*, Association for Computational Linguistics (ACL), Taipei, pp. 155-158, 2002.

[11] E.F. Tjong Kim Sang, and F. De Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," *Proceedings of the Conference on Natural Language Learning (CoNLL-2003)*, W. Daelemans, and M. Osborne, (eds), Association for Computational Linguistics (ACL), Edmonton, pp. 142-147, 2003.

[12] F. De Meulder, "CONLL-2003: List of tags with associated categories of names," 2003. [Web:] http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt [Date: 2006/10/25].

[13] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis, "MBT: A Memory-Based Part of Speech Tagger-Generator," *Proceedings of the Fourth Workshop on Very Large Corpora*, ACL, Copenhagen, pp. 14-27, 1996 [Web:] http://acl.ldc.upenn.edu/W/W96/W96-0102.pdf [Date: 31-10-2005].

[14] D. Aha, D. Kibler, and M.K. Albert, "Instance-Based Learning Algorithms," *Machine Learning*, vol.6, no.1, pp. 37-66, 1991.

[15] S.H. Baluja, V. Mittal, and R. Sukthankar, "Applying Machine Learning for High-Performance Named-Entity Extraction," *Computational Intelligence*. vol.16, no. 4, pp. 586-596, 2000.

[16] C. Seon, Y. Ko, J. Kim, and J. Seo, "Named Entity Recognition Using Machine Learning Methods and Pattern-Selection Rules," *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, Tokyo, pp. 229–236, 2001

[17] W. Daelemans, J. Zavrel, K. Van der Sloot, and A. van den Bosch, "TiMBL: Tilburg Memory Based Learner, Version 5.0, Reference Guide," *Technical Report*, ILK, 03-10. Tilburg University, Tilburg, 2003.

[18] E.F. Tjong Kim Sang, "Memory-based named entity recognition," *Proceedings of CoNLL-2002*, ACL, Taipei, pp. 203-206, 2002.

[19] L. Lin, T. Tsai, W. Chou, K. Wu, T. Sung, and W. Hsu, "A Maximum Entropy Approach to Biomedical Named Entity Recognition," *Proceedings of the 4th workshop on data mining in bioinformatics*, pp. 56-61, 2004

[20] O. Uryupina, "Semi-Supervised Learning of Geographical Gazetteers from the Internet," *Proceedings of the HLT-NAACL 2003, Workshop on Analysis of Geographic References*, Association for Computational Linguistics (ACL), Alberta, pp. 18-25, 2003.