Comparing Data-Driven Learning Algorithms for PoS Tagging of Swedish

Beáta Megyesi
Centre for Speech Technology
Department of Speech, Music and Hearing
Royal Institute of Technology
SE-100 44, Stockholm, Sweden
bea@speech.kth.se

Abstract

The aim of this study is a systematic evaluation and comparison of four state-of-the-art data-driven learning algorithms applied to part of speech tagging of Swedish. The algorithms included in this study are Hidden Markov Model, Maximum Entropy, Memory-Based Learning, and Transformation-Based Learning. The systems are evaluated from several aspects. Both the effects of tag set and the effects of the size of training data are examined. The accuracy is calculated as well as the error rate for known and unknown tokens. The results show differences between the approaches due to the different linguistic information built into the systems.

1 Introduction

In the last decade several machine learning algorithms have been developed and applied to various natural language processing (NLP) tasks. One of the most popular approaches to test the data-driven methods has been morphosyntactic disambiguation of running texts, also called Part-of-Speech (PoS) tagging. One of the reasons is the need of PoS annotated texts in natural language processing systems and applications. Another reason is that benchmark data, i.e. correctly annotated texts, are available for several languages, making the training and test procedure easily feasible.

The data-driven PoS taggers used in this study are claimed to be both language- and tagset-independent and easily applicable to new languages, given a set of correctly annotated training corpora. According to the literature, each tagger has been tested for English with an average accuracy of between 95% and 97%. However, the manner in which the taggers are evaluated by the researchers differs greatly, which makes it difficult to compare the performance of the systems.

Several recent studies report comparisons of data-driven PoS taggers. However, the purpose of these studies has been primarily to attain higher tagging performance by means of different system combinations.

Brill & Wu (1998) trained statistical unigram and trigram, Maximum Entropy and Transformation-Based learning on the English Penn Treebank Wall Street Journal Corpus consisting of 1.1 million words. They showed that the Maximum Entropy framework as it was implemented by Ratnaparkhi (1996) achieved the highest accuracy in total and in the annotation of ambiguous and unknown words.

Van Halteren et al. (1998) included Maximum Entropy, Memory-Based, Statistical trigram, and Transformation-Based approaches in the ensemble of classifiers and trained on 80% of the LOB corpus consisting of 931062 tokens. Similarly to the results given by Brill & Wu (1998), the Maximum Entropy framework achieved highest accuracy.

In contrast to the previous studies, Zavrel and Daelemans (2000) report that a statistical trigram approach, TNT (Brants, 2000) gave the best result over the Maximum Entropy, Memory-Based, and Transformation-Based approaches when trained on 90% and tested on 10% of the Spoken Dutch Corpus consisting of 5,000, 10,000 and 20,000 tokens respectively.

However, the goal of these studies was not a systematic evaluation and comparison of the classifiers.

De Pauw and Daelemans (2000) describe a systematic comparison between the Maximum Entropy framework and Memory-Based Learning performed on the LOB-corpus. They report that the overall tagging accuracy of the methods

are similar, although Maximum Entropy succeeds better in tagging unknown words. Furthermore, they pointed out that the "differences in accuracy can be attributed largely to differences in information sources used, rather than to algorithm bias".

2 Comparison of four PoS taggers

The purpose of the study is to evaluate four widespread learning algorithms in a systematic way. The aim is to find the advantages and drawbacks of the methods and to describe the type of errors they make, the effects of the tag set size, and the effect of the size of training material. Since English is a widely studied language and has received significant attention from computational linguists, it seems appropriate to evaluate the taggers on a different language; especially, as the taggers are said to be language-independent.

The experiments described in this paper are based on well-known algorithms that have implementations for the PoS tagging approach. Common to these taggers is that they are claimed to be language- and tagset-independent, easy to apply to new domains, languages and tag sets and available to the public. Each approach will be briefly described below.

2.1 Taggers

MEMORY-BASED LEARNING (MB), described by Daelemans et al. (1996), is a case-based approach where new items are classified on the basis of similarities to the earlier examples stored in memory during learning. In this study, decision tree induction, called IG-TREE, was re-implemented for Swedish by Berthelsen, based on the description given in Zavrel, et al. (1999)[1]. Here, an instance is represented by a vector where the elements are the different features of the instance. Information gain is used to determine at each node in the tree which feature should be used to create new branches. The implementation of the system used in this study contains information about the focus word, the preceding and following word forms, the two preceding (and already disambiguated) tags and the one following (still ambiguous) tag for known words. For unknown words, information about capitalization, the presence of a hyphen or a numeral feature, the preceding tag, the focus word, the ambiguous right tag and the last three letters occurring in the word is used.

The MAXIMUM ENTROPY (ME) framework, called MXPOST, is described by Ratnaparkhi (1996). It is a probabilistic classification-based approach based on a Maximum Entropy model where contextual information is represented as binary features that are used simultaneously in order to predict the PoS tag. The default binary features include the current word, the following and preceding two words and the preceding two tags. For rare and unknown words the first and last four characters are included in the features, as well as information about whether the word contains uppercase characters, hyphens or numbers. The tagger uses a beam search in order to find the most probable sequence of tags. For known words it generates the possible tags, and for unknown words it generates all tags in the tag set. The tag sequence with the highest probability is chosen.

TRANSFORMATION-BASED LEARNING (TBL), developed by Brill (1995), is a rule-based approach that learns by detecting errors. It begins with an unannotated text that is labeled by an initial-state annotator in a heuristic fashion. Known words (according to a lexicon) are annotated with their most frequent tag while unknown words receive an initial tag (e.g. the most frequently occurring tag in the corpus). Then, an ordered list of rules learned during training is applied deterministically to change the tags of the words according to their contexts. TBL uses a context of three preceding and following words and/or tags of the focus word. Unknown words are first assumed to be nouns and handled by prefix and suffix analysis by looking at the first/last one to four letters, capitalization feature and adjacent word co-occurrence.

TRIGRAMS'N'TAGS (TNT) is a statistical approach, developed by Brants (2000). The tagger is a trigram Hidden Markov Model and uses the Viterbi algorithm with beam search for fast processing. The states represent tags and the transition probabilities depend on pairs of tags.

[1] The reimplementation of the Swedish tagger was necessary because it was not available on the ILK web page (http://ilk.kub.nl/software.html). The results given by the re-implemented tagger are comparable to the results reported by (Daelemans et al., 1996)

The system uses maximum likelihood probabilities derived from the relative frequencies. The main smoothing technique implemented by default is linear interpolation. Unknown words are handled by suffix analysis, i.e. up to the last ten letters of the word. Additionally, information about capitalization is included as default.

Since the main goal is to evaluate the systems as they are available, all systems are used with the default settings according to their documentation. The taggers were retrained on the Swedish training data that will be described next.

2.2 Data

Swedish belongs to the Scandinavian, North Germanic family of the Germanic branch of Indo-European languages. It is morphologically richer than for example English. Nouns in general have two gender distinction. The genders are marked mainly by articles, adjectives, anaphoric pronouns and in plural endings. As in English, nouns can appear with or without articles. There are, however, definite and indefinite articles that agree with the head noun in gender, number and definiteness. Furthermore, adjectives have gender, definiteness and plurality markers. Thus, in a noun phrase both articles and adjectives agree in number, gender and definiteness with the head noun. Also, compound nouns are frequent and productive. Verbs lack markers for person or number of the subject but retain tense including complex tense forms. From a syntactic point of view, Swedish has subject-verb-object (SVO) order in independent declarative sentences, as well as in subordinate clauses, similar to English. However, in subordinate clauses the sentence adverbs normally precede the finite verb and the perfect auxiliary can be omitted.

All experiments presented in this paper were run on the second version of Stockholm-Umeå Corpus (SUC) (Ejerhed et al., 1992). The corpus is balanced, consisting of over one million PoS tagged words taken from different text genres in Swedish. The corpus used in this study is annotated with a Swedish version of PAROLE tags[2]. The tag set consists of totally 139 tags and encodes part-of-speech as well as morphological features.

The corpus was randomly divided into ten approximately equal parts, sentence by sentence, in order to get subsets containing different genres.

2.3 Evaluation

There are many ways in which evaluation can proceed. The type and the size of the tag set, the training and test data are all factors that have an effect on the results. Therefore, the evaluation of the learning methods is carried out from three different aspects.

First, the accuracy of each classifier is determined using the entire tag set (of 139 different tags), and one part (10%) of the SUC corpus as training data. The reason for the small size of the training data is that two of the learning algorithms are very time-consuming.

Secondly, the effect of training on different sizes of tag sets is examined. This is done because different NLP applications require annotation of various explicitness, e.g. full morphological analysis is not always needed.

Thirdly, the size of the training corpus is varied from one thousand up to one million tokens for each algorithm in order to find out how the size of the learning data influences the error rate for each approach.

For a fair comparison of the methods, each algorithm was trained in each experiment on the same part of the SUC corpus to build four classifiers. Then, in all cases, each classifier was evaluated on the same test set. In all the experiments, the training and the test set were disjoint and the test sets included unknown words. When tagging, the classifiers are allowed to assign exactly one tag to each token in the test.

The systems are evaluated from several perspectives. First, the overall tagging accuracy is computed by calculating the percentage of correctly assigned tags (given by the output of each classifier) in the test set compared to the correctly annotated benchmark.

$$Accuracy = \frac{Number\,of\,correctly\,tagged\,tokens}{Total\,number\,of\,tokens} \quad (1)$$

Additionally, since unknown tokens are more difficult to process than known tokens for which the possible tags are available from the lexicon

[2]Thanks to Britt Hartmann at the Department of Linguistics, Stockholm University for making the second version of SUC with PAROLE tags available.

during learning, separate accuracy is given for known and unknown words. Furthermore, for some of the experiments, the types of errors will be described but due to the sparse space, the recall and precision rates cannot be given for each category.

At the starting point of this study, the aim was to use 10-fold cross validation as the evaluation method for the experiments, but unfortunately that proved to be impracticable because of the long learning time for two of the approaches (see Section 3.4). Therefore, most of the experiments were accomplished on 100k tokens for learning and 100k for test.

3 Results

3.1 System performance

In order to find out the average accuracy of the systems, each algorithm was trained on 7388 sentences, including 115862 tokens and 24572 types. The corpus used for testing consists of totally 7464 sentences, 117685 tokens, 24492 types, of which 85.23% are known and 14.77% are unknown words.

The baseline performance is 77.37% and is obtained on the test data by selecting the PoS tag that is most frequently associated with the current word.

The results, given in Table 1, show that all systems outperformed the baseline, but the performance of the taggers is significantly lower than is reported for English.

The statistical trigram approach, TNT, has the highest overall accuracy and also succeeds best in the annotations of known and unknown words. The MXPOST tagger (ME) shows high performance because of the high precision of the annotation of unknown words. The Transformation-Based learner (TBL) manages to disambiguate known words but succeeds poorly on unknown words compared to the other systems. The memory-based method (MB) is slightly better than TBL because of its better success in the annotation of unknown words.

The analysis of the errors made by each classifier on individual tags shows that all systems failed most frequently on words belonging to the open classes (nouns, verbs, and adjectives), as could be predicted, since these categories are morphologically more complex and 95% of the unknown words belong to these classes. Among the closed categories, verbal particles, prepositions and adverbs are often confused, as well as determiners and pronouns. These belong to well-known ambiguity classes in Swedish.

Table 1: The tagging accuracy for all the words, and the accuracy of known and unknown words are given for each classifier. Training and test set are disjoint, consisting of 100k tokens, respectively. Tag Set includes 139 tags.

ACCURACY	MB	ME	TBL	TNT
TOTAL %	89.28	91.20	89.06	**93.55**
KNOWN %	92.85	93.34	94.35	**95.50**
UNKNOWN %	68.65	78.85	58.52	**82.29**

Table 2: The tagging accuracy for all the words, for known and unknown words are given for each classifier when tagging and testing on the original tag set but not considering the correctness of the subtags in the evaluation.

ACCURACY	MB	ME	TBL	TNT
TOTAL %	92.28	93.49	92.39	**95.31**
KNOWN %	94.69	94.72	95.63	**96.53**
UNKNOWN %	78.37	86.39	73.70	**88.24**

The difference in the type of errors among the classifiers lies in the frequency of the type of confusions. For example, by looking at the most frequent types of fault each system makes by counting precision and recall for each category, we found that TBL and MB more often make mistakes in the morphological analysis of categories (e.g. plural is often annotated as singular) while ME and TNT more frequently confuse PoS categories among ambiguity classes. This fact can be utilized for improvement of accuracy by using the large tag set marking inflectional properties of a word in training and tagging but not considering the correctness of the morphological tags in the evaluation. The results are shown in Table 2.

By comparing the results from Table 1 and 2, it is clear that accuracy is improved by the removal of the morphological tags in the evaluation. This can be useful for applications in

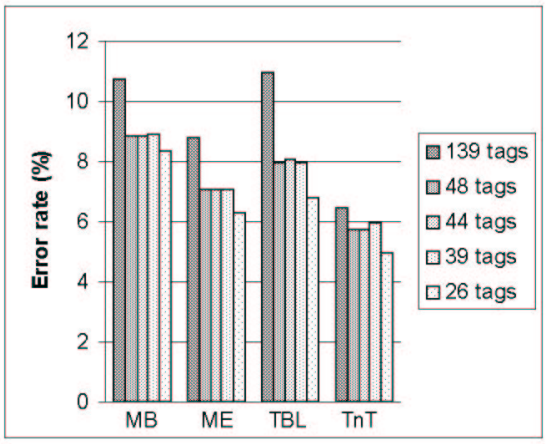

Figure 1: The error rate for each classifier when training on 139, 48, 44, 39 and 26 tags.

which high performance is required but morphological analysis is not needed.

3.2 Training on different size of tag sets

In some NLP applications, a tag set with complete morphological tags is not needed. One example is parsing where the PoS tag, and in some cases a few morphological tags for handling agreement, is enough. Therefore, the original tag set was mapped into smaller tag sets. The goal was to construct tag sets that can be useful for different applications. The most reduced tag set consists of the 26 PoS tags used in the second version of SUC. The PoS tags include some subcategorization information. There are, for example, distinctions between common and proper nouns, ordinal and cardinal numbers, possessive, possessive wh-, and personal/indefinite pronouns. The other tag sets, consisting of 39, 44 and 48 tags respectively are all subsets of the original PAROLE version. The tags were mapped together in categories designed for parsing. The difference between the smaller subsets is in the types of morphological tags that are included in order to be able to handle agreement in NPs.

As is shown in Figure 1, the number of errors made by each algorithm is higher when using a large tag set. This is not surprising since the classification is more difficult when the system has to choose from many categories. However, the amount with which the error rate decreases with a smaller tag set differs from system to system. By decreasing the size of the tag set from 139 to 26 tags, the error rate decreases by 38% for TBL, 29% for ME, and 23% for MB and TNT. Thus, TBL and ME seem to be more sensitive to the size of tag set than MB and TNT. Furthermore, when considering the results from training on between 39 and 48 tags (i.e. the distance between the amount of tags is small), the system performances show rather similar results. In the case of TNT, the error rate when training on 39 tags even increases with 3% compared to when training is done on 44 tags. Here, the type of information the tags bear seems to be important. Thus, the size of the tag set as well as the type of information are crucial factors for system performance.

Next, the results from training on different sizes of training corpora will be described.

3.3 Training on different sizes of data

In order to examine how the size of training data influences the performance of the classifiers, each algorithm was trained ten times on the same data set of various sizes from one thousand to one million tokens: 1k, 2k, 5k, 10k, 20k, 50k, 100k, 200k, 500k and 1000k tokens, respectively. Then, the same test set was annotated by each classifier.

Unfortunately, the results for ME and TBL when training on one million words cannot be reported in this paper. The learning algorithms will still be occupied after the deadline for the final version of this manuscript. The results will be published on the author's web page http://www.speech.kth.se/~bea/research.html as soon as the learners have finished their struggle.

The total error rate, i.e. the percentage of erroneous tags, is shown in Figure 2 for each classifier. It is not surprising that as the size of the training data increases, the error rate decreases. This is partly due to the fact that the number of unknown words is significantly smaller in large training data as shown in the first and second columns of Table 3.

There are some remarkable differences between the systems in their sensitivity to the size of the training data. ME shows most sensitivity, i.e. when increasing the training data from one thousand to five hundred thousand tokens, the error rate decreases by 88%. TBL, on the other hand, shows less sensitivity, the error rate is decreasing by 50%. This is due to the fact

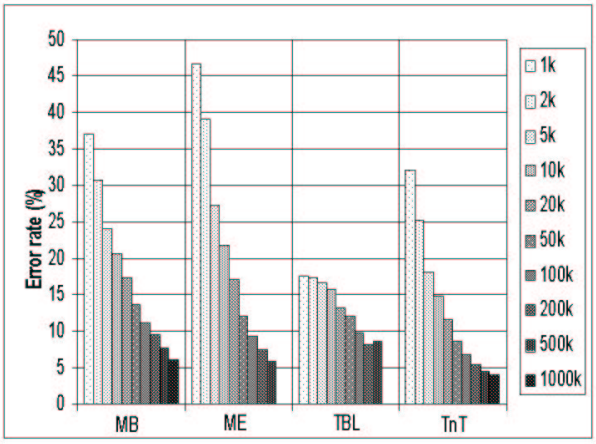

Figure 2: Error rates when training on 1000 to 1 million tokens, totally ten training corpora of various sizes, seen as ten columns for each classifier.

Table 3: The size of the training data, the percentage of unknown tokens in the test set, and the error rate, i.e. the percentage of wrongly annotated unknown words for each system.

TOKEN (K)	UNKNOWN (%)	MB	ME	TBL large	TBL small	TNT
1	50.2	63.9	60.9	**25.5**	66.5	54.8
2	44.2	56.4	54.5	**28.1**	69.7	46.3
5	37.2	49.2	42.1	**30.8**	63.9	37.3
10	32.8	45.3	36.7	**29.5**	58.6	31.6
20	27.8	40.3	31.7	29.0	50.7	**26.6**
50	20.8	35.2	25.5	32.8	44.5	**21.9**
100	15.6	32.4	21.5	32.5	34.2	**18.4**
200	11.9	30.2	19.6	26.6	27.1	**15.9**
500	8.1	28.2	18.8	32.0	32.5	**14.1**
1000	5.9	28.5	*	*	*	12.7

that there is a possibility to use a large lexicon listing all possible tags for a word that the tagger can choose from during tagging. Such a possibility is not available in the current public implementations of the other systems. The usage of the large lexicon decreases the error rate when training is performed on a small corpus only. Thus, TBL is useful when the user does not have access to a large correctly annotated corpus but only to a large lexicon. TBL can be put to use in the development of large corpora by applying a boot-strapping procedure.

Since PoS tagging has two main purposes, namely the annotation of unknown words and the disambiguation of the known words according to their context, the error rates are presented separately for unknown and known tokens as well. Additionally, because a large lexicon may not be available in some cases or for some languages, results for TBL are also given when tagging is performed on the basis of a small lexicon derived from the training data only.

The error rates for the annotation of unknown words is shown in columns 3-7 in Table 3. For MB, ME, TBL with a small lexicon and TNT, larger training data improves the overall accuracy. The error rate decreases by 75% for TNT, 70% for ME, 56% for MB and 51% for TBL with a small lexicon when increasing the training data from one thousand tokens to five hundred thousand tokens.

TNT conquers TBL as well as the other systems in the correct annotation of unknown words when the size of the training corpus is 20k tokens or more. The success of TNT can be explained by the way the system handles unknown words; The tag probabilities are set on the basis of the word endings, i.e. suffix analysis up to ten final characters of a word. The other systems do not look at as many characters, which has an effect on the results because inflectional and agglutinative languages with a complex morphological structure have suffix combinations longer than four characters.

The annotation of known words, i.e. the morphological disambiguation, seems to be an easier task for all the systems included in this study as is shown in Table 4. Here, TNT shows the lowest error rates in all the experiments. The ME approach succeeds poorly when training on a small corpus compared to the other three systems. In the case of large training corpora (100k and above), the differences between the systems converge due to the similar linguistic information implemented in the systems. For example, all systems use a lexicon listing possible tags for a token derived from the training data and contextual information in order to disambiguate the target word. The differences between the systems can be explained by the various window sizes implemented in the systems.

Table 4: The size of the training data, the percentage of known tokens in the test set, and the error rate, i.e. the percentage of wrongly annotated known words for each system.

TOKEN (K)	KNOWN (%)	MB	ME	TBL large	TBL small	TNT
1	49.8	10.1	32.2	9.4	11.0	**9.1**
2	55.8	10.2	25.2	8.6	10.2	**8.3**
5	62.8	9.2	18.5	8.1	9.2	**7.0**
10	67.2	8.7	14.7	8.9	9.2	**6.7**
20	72.2	8.3	11.4	7.0	7.8	**5.9**
50	79.2	7.9	8.4	6.6	6.9	**5.2**
100	84.4	7.2	7.1	5.7	5.7	**4.6**
200	88.1	6.7	5.8	5.8	5.0	**4.1**
500	92.9	5.9	4.7	6.7	4.3	**3.7**
1000	94.1	4.6	*	*	*	3.5

3.4 Time for learning and test

Another approach that has to be mentioned when comparing algorithms is the time it takes to learn from the training data and to test new texts. There are significant differences in the learning time between the systems[3]. TNT is able to learn from 100k tokens within one second and manages to tag a text containing the same amount of data in three seconds. MB is also fast in both training and tagging, i.e. the learning and annotation are carried out within a minute. ME and TBL, on the other hand, are time-consuming. Training on 100k words takes approximately one day for both systems.

When large training corpora are used (200k tokens or above) the training time can be expressed in a few seconds for TNT, in a minute for MB and in weeks or months for ME and TBL depending on the size of the corpus. TBL is slower in training than ME, but as fast as MB in tagging.

4 Discussion and future work

This paper has given a comparison of four state-of-the-art data-driven PoS taggers applied to Swedish. Although the performance of the taggers is lower than has been reported for English, the accuracy and the effect of training size are in some cases comparable to the results for English.

In the case of TNT, tagging accuracy is high for known tokens even with a small amount of training data just as has been reported for English by Brants (2000).

Also, the performance of MB, ME and TBL described in this study are comparable to the results given by van Halteren, et al. (1998) since both studies were performed on the same size of training data consisting of 100k tokens. In both cases, highest performance is achieved by ME, followed by MB and TBL.

The result that ME is generally better in the annotation of unknown words than MB is also supported by the study on the systematic evaluation of MB and ME (De Pauw & Daelemans, 2000).

In Zavrel & Daelemans, (2000) the same algorithms were used as in this paper but the algorithms were trained on Dutch training data of small size: 5k, 10k, and 20k tokens. In their experiments, TNT achieved the highest accuracy while in our experiment TBL showed highest performance on small training data. A possible explanation could be that a small lexicon, including the words from the training data only, was used when tagging with TBL without using any additional lexicon available for Dutch.

Future work includes the optimization of the taggers to better fit Swedish. For example, in TNT, several smoothing techniques are available that were not tested in this study. In the case of TBL and ME, the maximum length of the first/last characters should be increased in order to improve the system performance on unknown words since Swedish has a more complex morphological structure – suffixes are often longer than four characters.

Additionally, in order to further improve tagging accuracy, one could determine the best combinations of the approaches by constructing ensembles of classifiers.

It would be very interesting to evaluate the taggers in a systematic way on other languages belonging to different language types; in particular agglutinative and inflective languages with complex morphological structure and/or free word order. The linguistic information included in the systems seems in many cases to be optimized for English and other Germanic

[3]In this study, the experiments were run on a Pentium III, 800 MHz computer running Linux.

or perhaps Romance languages, but not for, for example, Uralic or Turkic languages. The task of data-driven PoS taggers is not completed until we have systems with high performance for the various language types.

5 Conclusion

In this study, four state-of-the-art data-driven algorithms have been compared based on PoS tagging of Swedish texts. The effects of the size of the tag set and the size of the training data have been examined by a systematic evaluation of the systems. The results show interesting differences between the classifiers. The TRIGRAMS'N'TAGS (TNT) approach has the highest overall accuracy, succeeds best in the annotation of known as well as unknown words, and is also fastest in both training and tagging. TRANSFORMATION-BASED LEARNING (TBL) has high performance on small training data, hence can be used as an aid when building large corpora by using a boot-strapping procedure. MXPOST (ME) has a high error rate in the annotation of known tokens when training on small corpora, but succeeds similarly to the other approaches when training is performed on large training data. MEMORY-BASED LEARNING (MB) is fast in both training and test and succeeds well in morphological disambiguation.

Acknowledgements

Many thanks go to my supervisor Rolf Carlson for his endless support and encouragement, and to the anonymous reviewers for their helpful suggestions and comments. Thanks also to Sheri Hunnicutt and Jens Edlund for reading and comments. Last, but not least, I would like to thank all the researchers who created the taggers used in this study for the excellent systems they have developed and making them available to the public.

References

T. Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference.* Seattle, Washington, USA.

E. Brill. 1994. Some Advances in Rule-Based Part of Speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94).* Seattle, Washington.

E. Brill and J. Wu. 1998. Classifier Combination for Improved Lexical Combination. In *Proceedings of the 17th International Conference on Computational Linguistics (ACL-98).* Montreal, Canada.

W. Daelemans, J. Zavrel, P. Berck, and S.E. Gillis. 1996. MBT: a Memory-Based Part of Speech Tagger-Generator. In *Proceedings of Fourth Workshop on Very Large Corpora (VLC-96).* pp. 14-27. Copenhagen, Denmark.

G. De Pauw, W. Daelemans. 2000. The Role of Algorithm Bias vs Information Source in Learning Algorithms for Morphosyntactic Disambiguation. In *Proceedings of Computational Natural Language Learning* pp. 19-24. Lisbon, Portugal.

E. Ejerhed, G. Källgren, O. Wennstedt, and M. Åström. 1992. *The Linguistic Annotation System of the Stockholm-UmeåProject.* Department of General Linguistics, University of Umeå.

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-96).* Philadelphia, PA, USA.

H. van Halteren, J. Zavrel, and W. Daelemans. 1998. Improving Data-Driven Wordclass Tagging by System Combination. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL-98).* Montreal, Canada.

H. van Halteren. 1999. *Syntactic Wordclass Tagging.* Kluwer Academic Publishers. Dordrecht, The Netherlands.

J. Zavrel, and W. Daelemans. 1999. Recent Advances in Memory-Based Part-of-Speech Tagging. In *Proceedings of the VI Simposio Internacional de Comunicacion Social.* pp. 590-597. Santiago de Cuba.

J. Zavrel, and W. Daelemans. 2000. Bootsrapping a Tagged Corpus trough Combination of Existing Heterogeneous Taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000).* pp. 17-20. Athens, Greece.