In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 19–24, Lisbon, Portugal, 2000.

The Role of Algorithm Bias vs Information Source in Learning Algorithms for Morphosyntactic Disambiguation

Guy De Pauw and **Walter Daelemans**
CNTS - Language Technology Group
UIA - University of Antwerp
Universiteitsplein 1, 2610 Antwerpen, Belgium
{depauw, daelem}@uia.ua.ac.be

Abstract

Morphosyntactic Disambiguation (Part of Speech tagging) is a useful benchmark problem for system comparison because it is typical for a large class of Natural Language Processing (NLP) problems that can be defined as *disambiguation in local context.* This paper adds to the literature on the systematic and objective evaluation of different methods to automatically learn this type of disambiguation problem. We systematically compare two inductive learning approaches to tagging: MXPOST (based on maximum entropy modeling) and MBT (based on memory-based learning). We investigate the effect of different sources of information on accuracy when comparing the two approaches under the same conditions. Results indicate that earlier observed differences in accuracy can be attributed largely to differences in information sources used, rather than to algorithm bias.

1 Comparing Taggers

Morphosyntactic Disambiguation (Part of Speech tagging) is concerned with assigning morpho-syntactic categories (tags) to words in a sentence, typically by employing a complex interaction of contextual and lexical clues to trigger the correct disambiguation. As a contextual clue, we might for instance assume that it is unlikely that a verb will follow an article. As a lexical (morphological) clue, we might assign a word like *better* the tag *comparative* if we notice that its suffix is *er*.

POS tagging is a useful first step in text analysis, but also a prototypical benchmark task for the type of disambiguation problems which is paramount in natural language processing: assigning one of a set of possible labels to a linguistic object given different information sources derived from the linguistic context. Techniques working well in the area of POS tagging may also work well in a large range of other NLP problems such as word sense disambiguation and discourse segmentation, when reliable annotated corpora providing good predictive information sources for these problems become available.

Finding the information sources relevant for solving a particular task, and optimally integrating them with an inductive model in a disambiguator has been the basic idea of most of the recent empirical research on this type of NLP problems and part of speech tagging[1] in particular.

It is unfortunate, however, that this line of research most often refrains from investigating the role of each component proper, so that it is not always clear whether differences in accuracy are due to inherent bias in the learning algorithms used, or to different sources of information used by the algorithms.

This paper expands on an empirical comparison (van Halteren et al., 1998) in which TRIGRAM tagging, BRILL tagging, MAXIMUM ENTROPY and MEMORY BASED tagging were compared on the LOB corpus. We will provide a more detailed and systematic comparison between MAXIMUM ENTROPY MODELING (Ratnaparkhi, 1996) and MEMORY BASED LEARNING (Daelemans et al., 1996) for morpho-syntactic disambiguation and we investigate whether earlier observed differences in tagging accuracy can be attributed to algorithm bias, information source issues or both.

[1] See van Halteren (ed.) (1999) for a comprehensive overview of work on morphosyntactic disambiguation, including empirical approaches.

After a brief introduction of the 2 algorithms used in the comparison (Section 2), we will outline the experimental setup in Section 3. Next we compare both algorithms on respectively typical MBT-features (Section 4) and typical MXPOST-features (Section 5), followed by a brief error analysis and some concluding remarks.

2 Algorithms and Implementation

In this Section, we provide a short description of the two learning methods we used and their associated implementations.

2.1 Memory-Based Learning

Memory-Based Learning is based on the assumption that new problems are solved by direct reference to stored experiences of previously solved problems, instead of by reference to rules or other knowledge structures extracted from those experiences (Stanfill and Waltz, 1986). A memory-based (case-based) approach to tagging has been investigated in Cardie (1994) and Daelemans et al. (1996).

Implementation

For our experiments we have used TIMBL[2] (Daelemans et al., 1999a). TIMBL includes a number of algorithmic variants and parameters. The base model (IB1) defines the distance between a test item and each memory item as the number of features for which they have a different value. Information gain can be introduced (IB1-IG) to weigh the cost of a feature value mismatch. The heuristic approximation of computationally expensive pure MBL variants, (IGTREE), creates an oblivious decision tree with features as tests, ordered according to information gain of features. The number of nearest neighbors that are taken into account for extrapolation, can be determined with the parameter K.

For typical symbolic (nominal) features, values are not ordered. In the previous variants, mismatches between values are all interpreted as equally important, regardless of how similar (in terms of classification behavior) the values are. We adopted the *modified value difference metric* (MVDM) to assign a different distance between each pair of values of the same feature.

[2]TIMBL is available from: http://ilk.kub.nl/

For more references and information about these algorithms we refer to Daelemans et al. (1999a).

2.2 Maximum Entropy

In this classification-based approach, diverse sources of information are combined in an exponential statistical model that computes weights (parameters) for all features by iteratively maximizing the likelihood of the training data. The binary features act as constraints for the model. The general idea of maximum entropy modeling is to construct a model that meets these constraints but is otherwise as uniform as possible. A good introduction to the paradigm of maximum entropy can be found in Berger et al. (1996).

MXPOST (Ratnaparkhi, 1996) applied maximum Entropy learning to the tagging problem. The binary features of the statistical model are defined on the linguistic context of the word to be disambiguated (two positions to the left, two positions to the right) given the tag of the word. Information sources used include the words themselves, the tag of the previous words, and for unknown words: prefix letters, suffix letters, and information about whether a word contains a number, an upcase character, or a hyphen. These are the primitive information sources which are combined during feature generation.

In tagging an unseen sentence, a beam search is used to find the sequence of tags with the highest probability, using binary features extracted from the context to predict the most probable tags for each word.

Implementation

For our experiments, we used MACCENT, an implementation of maximum entropy modeling that allows symbolic features as input.[3] The package takes care of the translation of symbolic values to binary feature vectors, and implements the iterative scaling approach to find the probabilistic model. The only parameters that are available in the current version are the maximum number of iterations and a value frequency threshold which is set to 2 by default (values occurring only once are not taken into account).

[3]Details on how to obtain MACCENT can be found on: http://www.cs.kuleuven.ac.be/~ldh/

3 Experimental Setup

We have set up the experiments in such a way that neither tagger is given an unfair advantage over the other. The output of the actual taggers (MBT and MXPOST) is not suitable to study the proper effect of the relevant issues of information source and algorithmic parameterisation, since different information sources are used for each tagger. Therefore the taggers need to be *emulated* using symbolic learners and a preprocessing front-end to translate the corpus data into feature value vectors.

The tagging experiments were performed on the LOB-corpus (Johansson et al, 1986). The corpus was divided into 3 partitions: an 80% training partition, consisting of 931.062 words, and two 10% partitions: the VALIDATION SET (114.479 words) and the TEST SET (115.101 words) on which the learning algorithms were evaluated.

The comparison was done in both directions: we compared both systems using information sources as described in Daelemans et al. (1996) as well as those described in Ratnaparkhi (1996).

Corpus Preprocessing

Since the implementations of both learning algorithms take propositional data as their input (feature-value pairs), it is necessary to translate the corpora into this format first. This can be done in a fairly straightforward manner, as is illustrated in Tables 1 and 2 for the sentence *She looked him up and down.*

word	d	f	a	value
She	*	PP3A	VBD-VBN	PP3A
looked	PP3A	VBD-VBN	PP30	VBD
him	VBD	PP30	RP-IN	PP30
up	PP30	RP-IN	CC	RP
and	RP	CC	RP	CC
down	CC	RP	SPER	RP
.	RP	SPER	*	SPER

Table 1: Contextual features

The disambiguation of known words is usually based on contextual features. A word is considered to be known when it has an ambiguous tag (henceforth *ambitag*) attributed to it in the LEXICON, which is compiled in the same way as for the MBT-tagger (Daelemans et al., 1996). A lexicon entry like *telephone* for example carries the ambitag *NN-VB*, meaning that it was observed in the training data as a noun or a verb and that it has more often been observed as a noun (frequency being expressed by order). Surrounding context for the focus word (*f*) are disambiguated tags (*d*) on the left-hand side and ambiguous tags (*a*) on the right-hand side.

In order to avoid the unrealistic situation that all disambiguated tags assigned to the left context of the target word are correct, we simulated a realistic situation by tagging the validation and test set with a trained memory-based or maximum entropy tagger (trained on the training set), and using the tags predicted by this tagger as left context tags.

word	p	s	s	s	c	h
She	S	S	h	e	T	F
looked	l	k	e	d	F	F
him	h	h	i	m	F	F
up	u	*	u	p	F	F
and	a	a	n	d	F	F
down	d	o	w	n	F	F
.	.	*	*	.	F	F

Table 2: Morphological features

Unknown words need more specific word-form information to trigger the correct disambiguation. Prefix-letters (*p*), suffix-letters (*s*), the occurrence of a hyphen (*h*) or a capital (*c*) are all considered to be relevant features for the disambiguation of unknown words.

4 Using MBT-type features

This section describes tagging experiments for both algorithms using features as described in Daelemans et al. (1996). A large number of experiments were done to find the most suitable feature selection for each algorithm, the most relevant results of which are presented here.

Validation Phase

In the validation phase, both learning algorithms iteratively exhaust different feature combinations on the VALIDATION SET, as well as learner-specific parameterisations. For each algorithm, we try all feature combinations that hardware restrictions allow: we confined ourselves to a context of maximum 6 surrounding

Known Words %	f	df	fa	dfa	ddfaa	dddfaaa
TIMBL IGTREE	92.5	95.1	95.9	97.2	**97.2**	97.2
TIMBL IB1	92.5	95.1	95.9	97.2	**97.4**	97.3
TIMBL IB1 K=5	92.5	95.1	95.6	93.8	96.4	**97.0**
TIMBL IB1 K=10	92.5	95.1	95.6	93.4	93.7	**96.1**
TIMBL MVDM	92.5	95.1	95.9	**97.4**	97.4	97.2
TIMBL MVDM K=5	92.5	95.1	95.2	**97.5**	97.5	97.4
TIMBL MVDM K=10	92.5	95.1	94.9	97.5	**97.5**	97.3
MACCENT	92.5	94.5	95.8	97.5	**97.6**	97.4
Unknown Words %	ddaap	ddaas	ddaaps	ddaapss	ddaapsscn	ddaapsshcn
TIMBL IGTREE	42.1	65.9	65.2	65.8	68.6	**70.0**
TIMBL IB1	53.8	63.7	66.3	68.3	68.8	**70.7**
TIMBL IB1 K=5	54.2	61.6	66.7	71.4	72.5	**74.3**
TIMBL IB1 K=10	49.5	55.3	64.2	68.4	70.3	**72.7**
TIMBL MVDM	58.1	72.0	70.9	**75.1**	71.0	73.3
TIMBL MVDM K=5	61.2	72.0	75.6	**79.7**	75.5	77.6
TIMBL MVDM K=10	61.7	72.7	76.0	**79.7**	77.1	77.9
MACCENT	61.8	67.0	74.8	**78.6**	75.3	77.0

Table 3: Validation Phase Results

tags or less, since we already noticed performance degradation for both systems when using a context of more than 5 surrounding tags. For unknown words, we have to discern between 2 different tuning phases. First, we find the optimal contextual feature set, next the optimal morphological features, presupposing both types of features operate independently.

We investigate seven of the variations of Memory-Based Learning available in TIMBL (see Daelemans et al. (1999b) for details) and one instantiation of maccent, since the current version does not implement many variations.

A summary of the most relevant results of the validation phase can be found in Table 3. The result of the absolute optimal feature set for each algorithm is indicated in bold. For some contexts, we observe a big difference between IGTREE and IB1-IG and IB1-MVDM. For unknown words, the abstraction made by the IGTREE-algorithm seems to be quite harmful compared to the true lazy learning of the other variants (see Daelemans et al. (1999b) for a possible explanation for this type of behaviour).

Of all algorithms, Maximum Entropy has the highest tagging accuracy for known words, outperforming TIMBL-algorithms however by only a very small margin. The overall optimal context for the algorithms turned out to be *dfa* and *ddfaa* respectively, while enlarging the context on either side of the focus word resulted in a lower tagging accuracy.

Overall, we noticed a tendency for TIMBL to perform better when the information source is rather limited (i.e. when few features are used), while MACCENT seems more robust when dealing with a more elaborate feature space.

Test Phase

The Test Phase of the experiment consists of running the optimised subalgorithm paired with the optimal feature set on the test set. TIMBL, augmented with the Modified Value Difference Metric and k set to 5, was used to disambiguate known words with a *dfa* feature value, unknown words with the features *ddaapss*. MACCENT used the same features for unknown words, but used more elaborate features (*ddfaa*) to disambiguate known words. The results of the optimised algorithms on the test set can be found in Table 4.

	TIMBL	MACCENT
Known Words	97.6	97.7
Unknown Words	77.3	78.2
Total	97.2	97.2
Sentence	62.7	63.5

Table 4: Test results with MBT features

Overall tagging accuracy is similar for both algorithms, indicating that for the overall tagging problem, the careful selection of optimal information sources in a validation phase, has a bigger influence on accuracy than inherent properties or bias of the two learning algorithms

Algorithm	Accuracy (%) on test set
IGTREE K=1	94.3
TIMBL MVDM K=5	92.8
Maccent	94.3
Maccent Beam(n=5)	94.3

Table 5: Test results with MXPOST features

tested.

Beam Search

Note that MACCENT does not include the beam search over N highest probability tag sequence candidates at sentence level, which is part of the MXPOST tagger (but not part of maximum entropy-based learning proper; it could be combined with MBL as well). To make sure that this omission does not affect maximum entropy learning adversely for this task, we implemented the beam search, and compared the results with the condition in which the most probable tag is used, for different beam sizes and different amounts of training data. The differences in accuracy were statistically not significant (beam search even turned out to be significantly worse for small training sets). The beam search very rarely changes the probability order suggested by MACCENT, and when it does, the number of times the suggested change is correct is about equal to the number of times the change is wrong. This is in contrast with the results of Ratnaparkhi (1996), and will be investigated further in future research.

5 Using MXPOST-type features

In order to complete the systematic comparison, we compared maximum entropy (again using the MACCENT implementation) with MBL when using the features suggested in (Ratnaparkhi, 1996). Due to the computational expense of the iterative scaling method that is inherent to maximum entropy learning, it was not tractable to incorporate an extensive validation phase for feature selection or algorithmic variant selection. We simply took the features suggested in that paper, and 2 different settings for our MBL implementation, IGTREE and MVDM K=5, the latter being the optimal algorithm for the previous experiments. The results on the test set are shown in Table 5.

Beam search

Notice that again, the sentence level beam search does not add significantly to accuracy. Also note that the results report in Table 5 differ significantly from those reported for MXPOST in (van Halteren et al., 1998). The difference in tagging accuracy is most likely due to the problematic translation of MXPOST's binary features to nominal features. This involves creating instances with a fixed number of features (not just the *active* features for the instance as is the case in MXPOST), resulting in a bigger, less manageable instance space. When IGTREE compresses the elaborate instance space, we consequently notice a significant improvement over a MVDM approach.

6 Error Analysis

The following table contains some more detailed information about the distribution of the errors[4]:

	Known	Unknown
Both wrong - same tag	1384	335
Both Wrong - different tag	117	130
Only MACCENT Wrong	1008	181
Only TIMBL Wrong	1103	193

In 87% of the cases where both algorithms are wrong, they assign the same tag to a word. This indicates that about 55% of the errors can either be attributed to a general shortcoming present in both algorithms or to an inadequate information source. We can also state that 97.8% of the time, the two algorithms agree on which tag to assign to a word (even though they both agree on the wrong tag 1.7% of the time).

We also observed the same (erroneous) tagging behavior in both algorithms for lower-frequency tags, the interchanging of noun tags and adjective tags, past tense tags and past participle tags and the like.

Another issue is the information value of the ambitag. We have observed several cases where the correct tag was not in the distribution specified by the ambitag, which has substantial information value. In our test set, this is the case for 1235 words (not considering unknown words). 553 times, neither algorithm finds the correct tag. Differences can be observed in the

[4]The error analysis described in this Section, is based on the first set of experiments in which MBT-features were used to disambiguate the test set.

way the algorithms deal with the information value of the ambitag, with Maximum Entropy exhibiting a more conservative approach with respect to the distribution suggested by the ambitag, more reluctant to *break free* from the ambitag. It only finds the correct part-of-speech tag 507 times, whereas TiMBL performs better at 594 correct tags. There is a downside to this: sometimes the correct tag *is* featured in the ambitag, but the algorithm breaks free from the ambitag nevertheless. This happens to TiMBL 267 times, and 288 times to MACCENT.

In any case, the construction of the ambitag seems to be a problematic issue that needs to be resolved, since its problematic nature accounts for almost 40% of all tagging errors. This is especially a problem for MBT as it relies on ambitags in its representation.

7 Concluding Remarks

A systematic comparison between two state-of-the-art tagging systems (maximum entropy and memory-based learning) was presented. By carefully controlling the information sources available to the learning algorithms when used as a tagger generator, we were able to show that, although there certainly are differences between the inherent bias of the algorithms, these differences account for less variability in tagging accuracy than suggested in previous comparisons (e.g. van Halteren et al. (1998)).

Even though overall tagging accuracy of both learning algorithms turns out to be very similar, differences *can* be observed in terms of accuracy on known and unknown words separately, but also in the differences in the (erroneous) tagging behaviour the two learning algorithms exhibit.

Furthermore, evidence can be found that given the same information source, different learning algorithms, and also different instantiations of the same learning algorithm, yield small, but significant differences in tagging accuracy. This may be in line with theoretical work by Roth (1998);Roth (1999) in which both maximum entropy modeling and memory-based learning (among other learning algorithms) are shown to search for a decision surface which is a linear function in the feature space. The results put forward in this paper support the claim that, although the linear separator found can be different for different learning algorithms, the feature space used is more important.

We also showed that which information sources, algorithmic parameters, and even algorithm variants are optimal depends on a complex interaction of learning algorithm, task, and data set, and should accordingly be decided upon by cross-validation.

References

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1).

C. Cardie. 1994. *Domain Specific Knowledge Acquisition for Conceptual Sentence Analysis*. Ph.D. thesis, University of Massachusets, Amherst, MA.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In E. Ejerhed and I. Dagan, editors, *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT.

W. Daelemans, A. Van den Bosch, and J. Zavrel. 1999a. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 1999b. TiMBL: Tilburg Memory Based Learner, version 2.0, reference manual. Technical Report ILK-9901, ILK, Tilburg University.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proc. of the Conference on Empirical Methods in Natural Language Processing, May 17-18, 1996, University of Pennsylvania.*

Dan Roth. 1998. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 806–813, Menlo Park, July 26–30. AAAI Press.

Dan Roth. 1999. Learning in natural language. In *Proceedings of the 16th Joint Conference on Artificial Intelligence.*

C. Stanfill and D. Waltz. 1986. Toward memory-based reasoning. *Communications of the* ACM, 29(12):1213–1228, December.

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, Montr'eal, Quebec, Canada*, pages 491–497, Montreal, Canada, August 10-14.

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