Recent Advances in Memory-Based Part-of-Speech Tagging

Jakub Zavrel & Walter Daelemans*
ILK / Computational Linguistics
Tilburg University
P.O. Box 90153, NL-5000 LE, Tilburg, The Netherlands
URL: http://ilk.kub.nl
{zavrel,walter}@kub.nl

Abstract

Memory-based learning algorithms are lazy learners. Examples of a task are stored in memory and processing is largely postponed to the time when new instances of the task need to be solved. This is then done by extrapolating directly from those remembered instances which are most similar to the present ones. Using memory-based learning for Part-of-Speech tagging has a number of advantages over traditional statistical POS taggers: (i) there is no need for an additional smoothing component for sparse data, (ii) even low-frequent or exceptional patterns can contribute to generalization, (iii) the use of a weighted similarity metric allows for an easy integration of different information sources, and (iv) both development time and processing speed are very fast (in the order of hours and thousands of words/sec, respectively). In recent work, we have applied the Memory-Based tagger (MBT) to a number of different languages and corpora (English, Dutch, Czech, Swedish, and Spanish). Furthermore, we have performed a controlled experimental comparison of MBT with several other POS tagging algorithms.

1 Introduction

In Part-of-Speech (POS) tagging, the problem is to assign to each word in a sentence the most appropriate morphosyntactic category from among those listed in the lexicon, given the context. Annotating a text with POS tags is useful for many subsequent manipulations of the text. First, the tags provide a useful abstraction from the actual words themselves if we want to process all words that belong to a certain class in some special way (e.g. extract all the nouns from a text). Second, the tagger provides a superficial degree of disambiguation which might either be beneficial for following levels of processing (such as e.g. parsing) or useful in itself (e.g. the same word with different tags might have different pronunciations or different meanings).

The general solution of the POS tagging problem requires full understanding of the sentence, but fortunately a fairly accurate solution can be reached by training a system on the patterns of tag usage in a large annotated corpus. The earliest statistical approaches (Church, 1988; DeRose, 1988) which make use of Hidden Markov Models (HMM) and related techniques have focused on building probabilistic models of tag transition sequences in sentences. Although these systems have achieved a reasonable level of

*This research was performed in the context of the "Induction of Linguistic Knowledge" research programme, partially supported by the Foundation for Language Speech and Logic (TSL), funded by the Netherlands Organization for Scientific Research (NWO).

performance (usually around 95 to 96 % correct[1]), they tend to suffer from a number of problems. The most notable among these are i) the problems associated with sparse data and ii) the limits on the types of information that they can take into account.

The *sparse data* problem is the fact that events that have not been observed in the training data get a probability of zero, and hence cannot be dealt with in the test data. A related problem is the fact that the test data also contains words that are not in the lexicon (which is usualy constructed from the training data). To deal with both types of sparse data, a practical tagger must use some sort of *smoothing* strategy (see Chen and Goodman (1996) for an overview) to estimate the probabilities for unseen events, and a separate guesser for the lexical probabilities of unknown words (Weischedel et al., 1993).

The second main problem with HMM-type taggers is that the features of the context are represented as states in the model and hence the incorporation of richer feature-sets of the context will lead to an explosion of the number of states, leading to an even more severe version of the sparse data problem.

In recent years it has been shown (see e.g. Ratnaparkhi (1996)) that the remaining 4 % or so of errors can be reduced considerably when richer models of the context are used. In these approaches, the POS tagger is usualy seen as a classifier, rather than as a model of the sequence structure of sentences. The context can then be represented in terms of a rich set of features (e.g. surrounding words, tags, and word-form features such as suffixes and prefixes). The construal of the POS tagging task as such a classification problem allows one to use many existing machine learning algorithms.

In our own work, we have advocated the use of Memory-Based Learning (MBL) techniques for POS tagging (Daelemans et al., 1996), and for classification tasks in Natural Language Processing in general (Daelemans et al., 1998). MBL provides a solution to both the sparse data problem, via an implicit similarity-based smoothing scheme, and the challenges of a rich feature set, via automatic feature-weighting. In this paper we will first review the basic techniques of Memory-Based Learning (Section 2). Next, in Section 3, we describe the architecture of the tagger. Section 4 reports experimental results of the application of our Memory-Based tagger (MBT) to a number of different languages and corpora (English, Dutch, Czech, Swedish, and Spanish). For English, we have performed a controlled experimental comparison of MBT with several other POS tagging algorithms (rule-based, HMM, and maximum entropy). In Section 5 we present an analysis of the strengths and weaknesses of the MBL approach to POS tagging. Finally, in Section 6, we conclude.

2 Memory-Based Learning

Memory-based learning is founded on the hypothesis that performance in cognitive tasks (in our case language processing) is founded on reasoning on the basis of similarity of new situations to *stored representations of earlier experiences*, rather than on the application of *mental rules* abstracted from earlier experiences (as in rule induction and rule-based processing).

An MBL system[2] contains two components: a *learning component* which is memory-based, and which is sometimes called 'lazy' as memory storage is done without abstraction or restructuring, and a *performance component* which does similarity-based classification. During classification, a previously unseen test example is presented to the system. Its similarity to all examples in memory is computed using a *similarity metric*, and the

[1] when tested on the same corpus that training was performed on, and depending on the type of corpus and tagset.

[2] TiMBL, a software package implementing several variants of memory-based learning algorithms, is is freely available for research purposes from the ILK web pages; consult URL http://ilk.kub.nl. Here you can also find demo's of the MBT tagger.

category of the most similar example(s) is used as a basis for extrapolating the category of the test example. We will now outline the functioning of the IB1-IG and IGTREE algorithms.

2.1 Weighted MBL: IB1-IG

IB1-IG (Daelemans and Van den Bosch, 1992) is a memory-based learning algorithm that builds a data base of instances (the *instance base* or case base) during learning. An instance consists of a fixed-length vector of n feature-value pairs, and an information field containing the classification of that particular feature-value vector. After the instance base is built, new (test) instances are classified by matching them to all instances in the instance base, and by calculating with each match the *distance* between the new instance X and the memory instance Y. In IB1-IG, the distance metric is a weighted sum of the distances per feature. The distance for a particular feature is zero when the values of both instances for this feature are equal, and one otherwise. Because not all features are of equal importance their contributions are weighted in the total summed distance. The weight for a feature is its Information Gain (Quinlan, 1993), a measure of how much information it contributes to our knowledge of the correct class label. The Information Gain of feature f is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature.

The possibility of automatically determining the relevance of features implies that many different and possibly irrelevant features can be added to the feature set. The weighting factors make the integration of diverse sources of information, with differing degrees of relevance to the task, relatively painless.

2.2 Optimized weighted MBL: IGTREE

Because the search for the nearest neighbors in IB1-IG is computationaly expensive, and POS tagging needs to be very fast, we use a decision tree approximation to the search. This algorithm is called IGTREE (Daelemans, Van den Bosch, and Weijters, 1997). In IGTREE the instance memory is restructured in such a way that it contains the same information as before, but in a compressed decision tree structure. Information gain is used to determine the order in which instance feature values are added as arcs to the tree, so that, during testing, search can be restricted to matching a test instance to those memory instances that have the same feature value as the test instance at the feature with the highest weight. Instead of indexing all memory instances only once on this feature, the instance memory can then be optimized further by examining the second most important feature, followed by the third most important feature, etc. A considerable compression is obtained as similar instances share partial paths. Furthermore, it is not necessary to fully store an instance as a path when only a few feature values of the instance make the instance classification unique.

Processing an unknown input involves traversing the tree (i.e., matching all feature-values of the test instance with arcs in the order of the overall feature information gain), and either retrieving a classification when a leaf is reached (i.e., an exact match was found), or using the most probable classification on the last matching non-terminal node if an exact match fails.

In sum, it can be said that the IGTREE approach chooses to invest more time in organizing the instance base using information gain and compression, to obtain considerably simplified and faster processing during classification, as compared to IB1-IG. The generalization accuracy of IGTREE is usualy comparable or slightly lower to that of IB1-IG.

word	case representation				
	D	D	F	A	T
Pierre	=	=	np	np	np
Vinken	=	np	np	,	np
,	np	np	,	cd	,
61	np	,	cd	nns	cd
years	,	cd	nns	jj-np	nns
old	cd	nns	jj-np	,	jj

Table 1: Example of instances of the POS learning task (known words case base). Instances represent fixed-sized snapshots of a focus (an ambiguous tag), surrounded by a left and right context (of disambiguated tags on the left, and ambiguous tags on the right).

3 MBT: Memory-Based Part-of-Speech Tagging

The MBT tagger (Daelemans et al., 1996) takes an annotated corpus as input, and produces a lexicon and memory-based POS tagger as output. In this section we describe the architecture of the tagger.

The construction of a POS tagger for a specific corpus is achieved in the following way. Given an annotated corpus, three data structures are automatically extracted: a *lexicon*, associating words to ambiguity classes of tags as evidenced in the training corpus, a case base for *known words* (words occurring in the lexicon), and a case base for *unknown words*. Case Bases are compressed using IGTREE for efficiency.

During tagging, each word in the text to be tagged is looked up in the lexicon. If it is found, its lexical representation is retrieved and its context is determined, and the resulting pattern is disambiguated using extrapolation from nearest neighbors in the known words case base. When a word is not found in the lexicon, its lexical representation is computed on the basis of its form, its context is determined, and the resulting pattern is disambiguated using extrapolation from nearest neighbors in the unknown words case base. In each case, the output is a best guess of the category for the word in its current context.

The cases are represented by a variety of features, whose relevance is automatically determined by the Information Gain weights. The reason for the split into known and unknown words is that for known words, the ambiguity class of the focus word turns out to be the most important feature, and is therefore found at the top of the IGTREE. However, for unknown words we do not known the ambiguity class, and hence we would get a mismatch at the highest level of the tree. In the separate unknown words classifier, we proceed directly to the context and word-form features. Below we will use the following notation for the features. As we go from left to right, we can assume that the words to the left of the word to be tagged have been disambiguated already. These tags are denoted with 'D', the position of the (ambiguity class) of the focus word is given by 'F', and the ambiguous tags to the right are denoted by 'A'. Features referring to particular word forms are denoted as 'W'. Further, there are a number of features referring to the parts of the word form: its suffix letters 'S', prefix letters 'P', a capitalization feature 'C', the presence of a hyphen 'H', and the presence of numerals 'N'. For training cases 'T' denotes the correct target.

Table 1 and 2 display example instances from the known words and the unknown words case bases respectively.

word	case representation						
	P	D	A	S	S	S	T
Pierre	P	=	np	r	r	e	np
Vinken	V	np	,	k	e	n	np
61	6	,	nns	=	6	1	cd
years	y	cd	jj-np	a	r	s	nns
old	o	nns	,	o	l	d	jj

Table 2: Example of instances of the POS learning task (unknown words case base). Instances represent 'morphological' information about the focus word (first letter and the three last letters), surrounded by a left and right context (of one disambiguated tags on the left, and one ambiguous tag on the right).

4 Experiments

In the first paper on MBT (Daelemans et al., 1996), we trained it on the English Wall Street Journal corpus (ACL/DCI version), tagged with the Penn Treebank tagset (Marcus, Santorini, and Marcinkiewicz, 1993). For the known words we used 'DDFA' features and for the unknwon words 'PDFASSS'[3]. The results are reiterated in Table 3. Since then we have experimented with a number of different languages and corpora, and we have gradually increased the richness of our feature set. For our experiments on Dutch we used the WOTAN annotated Eindhoven corpus (Berghmans, 1995), with the same feature set as for the WSJ, attaining very competitive results (for more details, see Daelemans, Zavrel, and Berck (1996)). For the experiments on Czech we used an annotated corpus of newspaper texts obtained from the Institute for the Czech Language, Prague.[4]. Again the features were the same as on the WSJ corpus. For Spanish, the CRATER Multi-Lingual Aligned Corpus was used. In this case the known words case base was constucted using 'DDFWAA' features; i.e. in addition to the ambiguity class of the focus word, information was also provided about its identity (only for the 100 most frequent words). For unknown words, we used the 'CHNDFASSS' feature set. For Swedish, the Stockholm Umea Corpus (SUC) was used. Tuned on a held-out portion of the training data, we found slightly better performance for the known feature set DDWFWA (with word form features for the directly neighboring words) than for the 'DDFWAA' used for Spanish. We also found that performance measured solely on the unknown words, rose from 77.3% to 81.0% if the unknown words cases were constructed only from words that had five or less occurrences in the training set. The results of all these experiments are summarized in Table 3. As a practical remark, it should be noted that the whole cycle of feature-validation training and testing is very fast, and was usualy completed in about 8 hours of work for all of the taggers described above.

4.1 A Comparison of MBT with Alternative Tagging Methods

The results in the previous section by themselves are difficult to interpret in terms of comparison to other tagging approaches. Therefore, we also conducted some experiments, comparing a number of alternative tagging methods (R: rule-based (Brill, 1994), T: trigram (Steetskamp, 1995), and E: maximum entropy (Ratnaparkhi, 1996)) on the same corpus, the tagged LOB corpus (Johansson, 1986). This work is described in more detail in (van Halteren, Zavrel, and Daelemans, 1998). Each of these taggers uses different

[3]The F in the unknown words pattern only indicates the position of the focus, it is not included as a feature in the actual pattern.

[4]Thanks go to Prof. Jiří Kraus of the Czech Academy of Sciences for permission to use this corpus.

Language	Tag-set size	# Words × 1000 train	test	% Correct test words
English - WSJ	44	2000	200	96.4
English - LOB	170	931	115	97.0
Dutch	13	611	100	95.7
Czech	42	495	100	93.6
Spanish	484	711	89	97.8
Swedish	23	1156	11	95.6

Table 3: Results for the POS task for different languages/corpora. The size of the tag-set used, the size of train and test set and the generalization accuracy (combines known and unknown) are given. All taggers use the IGTREE algorithm. Details of the used corpora can be found in the main text.

Tagger	accuracy (%)
T	96.1
R	96.5
MBT	97.0
E	97.4

Table 4: Accuracy of different taggers (T: trigram, R: Rule-Based Learner, MBT: Memory-Based, E: Maximum Entropy) on the LOB corpus.

features of the text to be tagged, and each has a completely different representation of the language model. Due to lack of space we will not go into detailed descriptions of these systems here. The training set consists of 80% of the data (931062 tokens), constructed by taking the first eight utterances of every ten. 10 % was used as a validation set to tune the individual taggers. The results are given on the test set, which consists of the remaining 10% (115101 tokens).

The results, given in Table 4, show that MBT performs at state-of-the-art levels, providing better generalization accuracy than two widely-used methods (trigram tagging and transformation-based tagging), which is remarkable given the minimal language engineering involved and the computational efficiency of the method (both in training and testing). The E tagger performs significantly better, which is due, in our opinion, to the fact that Maximum Entropy weighting is better able to deal with the dependecies in the rich feature-set. However, E uses a slightly more elaborate feature set than MBT, and a preliminary comparison of learning algorithms on the data from Ratnaparkhi (1996) resulted in a close tie (IGTREE 95.5 % correct vs. Maximum Entropy 95.4 % correct). Moreover, compared to MBT, E is very slow in training.

An interesting side-result for high accuracy tagging is the fact that in van Halteren, Zavrel, and Daelemans (1998), an error reduction of 19 % (to 97.9 % accuracy) was achieved over the best tagger (E) by a combination of the results from all four taggers.

5 Discussion

In contrast to explicitly probabilistic methods, there is no need for an additional smoothing component for sparse data in MBL, as this is already embodied in the similarity-based extrapolation itself (Zavrel and Daelemans, 1997). The use of the weighted similarity metric allows for an easy integration of different information sources (e.g. context tags, words, morphology, spelling etc.) with no clear a-priori ordering. Moreover, the fact that only one parameter is needed per feature (i.e. its IG weight) makes MBL more robust to overfitting than approaches which use very large numbers of parameters. The down side of this robustness is that the feature-weighting capabilitities are quite rough: i) each feature is weighted in isolation, so that no specific weights are assigned to interesting feature interactions, and the weight estimate of conjunctions of redundant features tends to be too large, and ii) there is no separate weight for specific values of a feature.

A second advantage of MBL, when compared to both probabilistic and other 'eager' machine learning approaches, is that in MBL all information is stored in memory, and even low-frequent or exceptional events are available, and useful for accurate generalization (Daelemans, Van den Bosch, and Zavrel, 1999 to appear). At present this is not entirely made use of in the tagger, because we use the IGTREE approximation of MBL nearest neighbor search. In the TiMBL package, however, we have implemented several optimizations of MBL search, and we hope to that these will turn out to be fast enough to enable us to use ib1-ig in future work, without a too large loss of speed.

Finding a good balance in the accuracy-speed trade-off is an important issue for MBT, as at present this clearly is an important practical advantage of our system: both development time and processing speed are very fast (in the order of hours and thousands of words/sec, respectively).

6 Conclusion

We have presented additional evidence that the Memory-Based approach to Part-of-Speech tagging quickly yields very fast and highly accurate taggers for a variety of languages and corpora.

Acknowledgments

We thank Peter Berck, the members of the ILK group in Tilburg, and the members of the CNTS group in Antwerp for their input and discussions.

References

Berghmans, J. 1995. Wotan - een probabilistische grammatikale tagger voor het Nederlands. Master's thesis, TOSCA Research Group, University of Nijmegen, Nijmegen, The Netherlands.

Brill, E. 1994. Some advances in transformation-based part-of-speech tagging. In *Proceedings AAAI'94*.

Chen, S.F. and J. Goodman. 1996. An empirical study of smoothing techniques for language modelling. In *Proc. of the 34th Annual Meeting of the ACL, Santa Cruz, CA*. ACL, June.

Church, K. W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of Second Applied NLP (ACL).*

Daelemans, W. and A. Van den Bosch. 1992. Generalisation performance of backpropagation learning on a syllabification task. In M. F. J. Drossaers and A. Nijholt, editors, *Proc. of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede. Twente University.

Daelemans, W., A. Van den Bosch, and A. Weijters. 1997. IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

Daelemans, W., A. Van den Bosch, and J. Zavrel. 1999, to appear. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning.*

Daelemans, W., A. Van den Bosch, J. Zavrel, J. Veenstra, S. Buchholz, and G. J. Busser. 1998. Rapid development of NLP modules with Memory-Based Learning. In *Proceedings of ELSNET in Wonderland, March, 1998*, pages 105–113. ELSNET.

Daelemans, W., J. Zavrel, and P. Berck. 1996. Part-of-speech tagging for dutch with MBT, a memory-based tagger generator. In K. van der Meer, editor, *Informatiewetenschap 1996, Wetenschappelijke bijdrage aan de Vierde Interdisciplinaire Onderzoeksconferentie Informatiewetenchap*, pages 33–40, The Netherlands. TU Delft.

Daelemans, W., 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.

DeRose, S. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14:31–39.

Johansson, S. 1986. *The tagged LOB Corpus: User's Manual.* Bergen, Norway: Norwegian Computing Centre for the Humanities.

Marcus, M., B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Quinlan, J.R. 1993. C4.5*: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A. 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.*

Steetskamp, R. 1995. An implementation of a probabilistic tagger. Master's thesis, TOSCA Research Group, University of Nijmegen, Nijmegen, The Netherlands.

van Halteren, H., J. Zavrel, and W. Daelemans. 1998. Improving data-driven wordclass tagging by system combination. In *Proceedings of COLING-ACL'98*, pages 491–497, Montreal, Canada, August 10-14.

Weischedel, R., M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382.

Zavrel, J. and W. Daelemans. 1997. Memory-based learning: Using similarity for smoothing. In *Proc. of 35th annual meeting of the ACL*, Madrid.