Memory-Based Word Sense Disambiguation

Optimising word disambiguation experts for SENSEVAL

Jorn Veenstra, Antal van den Bosch, Sabine Buchholz, Walter Daelemans and Jakub Zavrel

Tilburg University

Abstract

We describe a memory-based classification architecture for word sense disambiguation and our experience with its application to the SENSEVAL evaluation task. In a memory-based approach, selecting the correct sense of a word in a new context is achieved by finding the closest match to stored examples of this task. Advantages of the approach include (i) fast development time for classifiers, (ii) easy and elegant automatic integration of information sources, (iii) use of all available data, and (iv) relatively high accuracy without language engineering.

1 Introduction

In this paper we describe a memory-based approach to *word sense disambiguation* (WSD) as defined in the SENSEVAL[1] task: the association of a word in context with its contextually appropriate sense tag. For this task our WSD method is trained on POS-tagged corpus examples and selected information from dictionary entries as provided by SENSEVAL. We believe that this approach is promising because it is completely automatic – it only relies on the availability of some annotated examples for each sense, and not on human linguistic or lexicographic intuitions – and is therefore easily adaptable and portable, as we have seen in its application to the SENSEVAL task.

Memory-Based Learning (MBL) is a classification-based, supervised learning approach. To solve the WSD task in this framework, it has to be formulated as a classification task: given a set of feature values describing the context in which the word appears and any other relevant information as input, a *classifier* has to select the appropriate output class from a finite number of a priori given possibilities. In our approach we construct a distinct classifier for each word to be disambiguated. This classifier can be seen as a type of word-expert (Berleant 1995), and might be constructed with any supervised learning algorithm.

The distinguishing property of memory-based learning as a classification-based supervised learning method is that it does not abstract from the training data the way e.g. decision tree learning, rule induction, or neural network machine learning methods do. MBL keeps all training data in memory (in an efficient way), and only abstracts at classification time (i.e. it is a *lazy* learning method instead of the more common *eager or greedy* learning approaches).

[1] SENSEVAL is a project set up to evaluate Word Sense Disambiguation systems, for more information see http://www.itri.brighton.ac.uk/events/senseval/

Since the early nineties, we have been advocating *memory-based learning* as a methodology in language engineering (Daelemans 1995, Daelemans, Van den Bosch, Zavrel, Veenstra, Buchholz and Busser 1998a). The memory-based algorithms discussed in this paper have been successfully applied to a large range of Natural Language Processing tasks: hyphenation and syllabification (Daelemans and Van den Bosch 1992); assignment of word stress (Daelemans, Gillis and Durieux 1994); grapheme-to-phoneme conversion (Daelemans and Van den Bosch 1996); diminutive formation (Daelemans, Berck and Gillis 1997); morphological analysis (Van den Bosch, Daelemans and Weijters 1996); part of speech tagging (Daelemans, Zavrel, Berck and Gillis 1996); PP-attachment (Zavrel, Daelemans and Veenstra 1997); NP chunking (partial parsing) (Veenstra 1998); and subcategorization frame learning (Buchholz 1998).

In the remainder of this paper, we briefly describe memory-based learning, discuss the setup of our memory-based classification architecture for word sense disambiguation, and show the figures of generalization accuracy on the SENSEVAL data both for cross-validation on the training data, and for the final run on the evaluation data. We will also briefly discuss relations with alternative memory-based and supervised-learning approaches to word sense disambiguation.

2 Memory-Based Learning

Memory-Based Learning keeps all training data in memory and only abstracts at classification time by extrapolating a class from the most similar item(s) in memory (i.e. it is a *lazy* learning method instead of the more common *eager* learning approaches). In recent work (Daelemans, Van den Bosch and Zavrel 1999) we have shown that for typical natural language processing tasks, this lazy learning approach is at an advantage because it "remembers" exceptional, low-frequency cases which are nevertheless useful to extrapolate from. Eager learning methods "forget" information, because of their pruning and frequency-based abstraction methods. Moreover, the automatic feature weighting in the similarity metric of a memory-based learner makes the approach well-suited for domains with large numbers of features from heterogeneous sources, as it embodies a smoothing-by-similarity method when data is sparse (Zavrel and Daelemans 1997). For our experiments we have used TiMBL[2], an MBL software package developed in our group (Daelemans, Zavrel, Van der Sloot and Van den Bosch 1998b). TiMBL includes the following variants of MBL:

IB1: The distance between a test item and each memory item is defined as the number of features for which they have a different value (overlap metric).

IB1-IG: In most cases, not all features are equally relevant for solving the task; this variant uses information gain (an information-theoretic notion measuring the reduction of uncertainty about the class to be predicted when knowing the value of a feature) to weight the cost of a feature value mismatch during comparison.

IB1-MVDM: For typical symbolic (nominal) features, values are not ordered. In the previous variants, mismatches between values are all interpreted as equal-

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

ly important, regardless of how similar (in terms of classification behaviour) the values are. We adopted the *modified value difference metric* to assign a different distance between each pair of values of the same feature.

MVDM-IG: MVDM with IG weighting.

IGTREE: In this variant, an oblivious decision tree is created with features as tests, and ordered according to information gain of features, as a heuristic approximation of the computationally more expensive pure MBL variants.

For more references and information about these algorithms we refer to Daelemans et al. (1998b) and Daelemans et al. (1999).

3 System Architecture and Experiments

For the WSD task, we train classifiers for each word to be sense-tagged (or in those cases where the SENSEVAL task requires it, for a word/POS-tag combination). To settle on an optimal memory-based learning algorithm variant (i.e. IB1, IB1-IG, IB1-MVDM, or IGTREE) and the number of nearest neighbours (the k parameter), as well as different possible feature construction settings (see below), ten-fold cross-validation is used: the training data is split into ten equal parts, and each part in turn is used as a test set, with the remaining nine parts as training set. All sensible parameter settings, algorithm variants, and feature construction settings are tested, and those settings giving the best results in the cross-validation are used to construct the final classifier, this time based on all available training data. This classifier is then later tested on the SENSEVAL test cases for that word or word/POS-tag combination.

3.1 Feature Extraction

The architecture should be suited for WSD in general, and this can include various types of distinctions ranging from rough senses that correspond to a particular POS tag, to very fine distinctions for which semantic inferences need to be drawn from the surrounding text. The 36 words (or rather word/POS-tag pairs) and their senses in the SENSEVAL task supposedly embody many such different types of disambiguations. Since we do not know beforehand what features will be useful for each particular word and its senses, and because we believe to have a classifier which can automatically assess feature relevance, we have chosen to include a number of different information sources in the representation for each case. All information is taken from the dictionary entries in the HECTOR dictionary (Atkins 1993), and the corpus files, both of which have been labeled with Part of Speech from the Penn Treebank tag set (Marcus, Santorini and Marcinkiewicz 1993) using MBT, our own Memory-Based Tagger (Daelemans and Van den Bosch 1996). We did not use any further information such as external lexicons or thesauri.

The sentences in the corpus files contain sense-tagged examples of the word in context. For example:

```
800002 An image of earnest Greenery is almost tangible.
Eighteen years ago she lost one of her six children in an
<tag_"532675">accident</> on Stratford Road, a tragedy which
has become a pawn in the pitiless point-scoring of small-
town vindictiveness.
```

The dictionary contains a number of fields for each sense, some of which (i.e. the 'ex' (example) and 'idi' (idiom) fields) are similar to the corpus examples. These underwent the same treatment as the corpus examples: these cases were used to extract both context features (directly neighboring words and POS-tags, as described in section 3.1.1), and keyword features (informative words from a wide neighborhood; see section 3.1.2). The only other field from the dictionary that we used is the 'def' field, which gives a definition for a sense. As the 'def' field often does not contain the word of interest at all, these were only used to help the selection of keywords for the cases which did contain the word in a context. During the cross-validation, the examples which originated from the dictionary were always kept in the training portion of the data to have a better estimate of the generalization error, because the system is unlikely ever to be tested on data that resemble dictionary data. Note that for both dictionary and corpus examples, we took the sense-tag that it was labeled with as a literal atom, and did not take into account the hierarchical sense/sub-sense structure of the category labels. All cases that were labeled as errors or omissions (i.e. the 999997 and 999998 tags) were discarded. Disjunctions were split into (two) separate cases.

3.1.1 Context Features

We used the word form and the Part-of-Speech (POS) tag of the word of interest (which we shall further refer to as the *focus word*) and the surrounding positions as features. After some initial experiments, the size of the window was set to two words to both the left and the right. These features were always recorded, even when they had very low frequency values. This gives the following representation for the example given above:

```
800002,in,IN,an,DT,accident,NN,on,IN,Stratford
```

3.1.2 Keyword Features

Often the direct context cannot distinguish between two senses, either because it is too generic, or because it has not been seen before in the training data. In such cases it is useful to look at a larger context (e.g. the whole text snippet that comes with the example) to guess the semantics from its content words. As there is a large number of possible content words, and each sentence contains a different number of them, it is somewhat difficult to represent all of them in the fixed-length feature-value vector that is required by the learning algorithm. Hence we choose to use only a limited set of "informative" words, which we will call the *keywords*. The

method is essentially the same as in the work of Ng and lee (1996), and extracts a number of keywords per sense. These keywords are then used as binary features, which take the value 1 if the word is present in the example, and the value 0 if it is not. A word is a keyword for a sense if it obeys the following three properties:

M1 the word occurs in more than $M1$ percent of the cases with the sense; a high value of $M1$ thus restricts the keywords to those that are very specific for a particular sense.

M2 the word occurs at least $M2$ times in the corpus; a high value of $M2$ thus eliminates low-frequent keywords.

M3 only the $M3$ most frequently occurring keywords for a sense are extracted, this restricts somewhat the number of keywords that are extracted for very frequent senses.

The above statistics for the keywords are computed based on i) sentences in the corpus file and ii) the 'ex' and 'idi' sentences in the dictionary file. The parameters can be manipulated from no keywords at all to including almost all words as keywords. We have also tried to use a simpler criterion, i.e. the Information Gain of a word, but Ng and Lee's scheme was found to give better results. For M1=0.8, M2=5 and M3=5 (a rather restrictive setting), the following keywords were found in one of the train/test splits of the dataset for for the different senses of the word 'accident':

```
538889 plaintiffs
538889 operate
538889 refusing
532675 west
532675 Howard
538895 sickness
```

3.1.3 Definition Features

In addition to the keywords that passed the above selection, we use all the open class words (nouns, adjectives, adverbs and verbs) in the 'def' field in the dictionary entry as features. Comparable to the keyword feature the definition word feature has the value '1' if it occurs in the test sentence else it has the value '0'. The 'def' field is only used for this purpose, and is not converted to a training case.

After the addition of both types of keywords, a complete case for the continuing example will look as follows:

```
800002,in,IN,an,DT,accident,NN,on,IN,Stratford,NNP,0,0,...
...0,0,0,0,0,0,0,1,0,0,...,0,0,532675
```

4 Post-processing

Some senses are restricted to specific multi-word expressions. A good example are the multi-word expressions like "elastic band" or "brass band". If we ever have to disambiguate "band" in a sentence where the previous word is "elastic" or "brass", we can be quite sure that the correct sense tag is the one given in the dictionary for the multi-word entry. Although the classifier has access to this type of information by looking at the direct context and the form of the focus word, it can still make errors on this type of pattern.

We have therefore implemented an additional component, acting as a postprocessor to the memory-based learner. To this end, we first extracted a list of simple patterns from various parts of the dictionary entries. For example:

```
band    dance-band      532782
band    dance band      532782
bet     in the betting  520894
shake   handshake       516772
shake   shake off       504585
```

These patterns are then used to re-consider the sense-tags assigned by the classifier. If the multi-word expression occurs in the test data, the according sense tag is assigned to the test case. The patterns were extracted from the dictionary entries automatically and underwent some manual editing (based only on the cross-validation results on the training data). For example, the phrasal verb pattern "seize on or upon" is split into two pattern "seize on" and "seize upon".

The pattern matcher in the form just described is already able to correct some of the misclassifications of the memory-based learner. However, it might also introduce new errors, as we discovered when testing it on part of the labeled training data. These errors are often due to over-generalizations of the patterns. A nice example is "band saw" which is supposed to be a pattern for a multi-word noun, but which also matches if "band" is the subject of the verb "saw". At the moment, we try to minimize the errors by the pattern matcher by removing all patterns that introduce new errors when applied to the output that the memory-based learner produced in 10-fold cross-validation experiments on the SENSEVAL training data.

Table 1 shows the improvement in the score that the post-processor has on the words it affects. This is measured on the 10-fold cross-validation on the training set.

5 Results

In this section we present the results we obtained with the optimal choice of metrics and feature construction parameters found with 10-fold cross validation on the training data, and the results on the evaluation data, as measured by the SENSEVAL coordination team. For comparison we also provide the baseline results (on the training data), obtained by always choosing the most frequent sense. Our submis-

band	0.45
bet-n	0.91
bet-v	1.49
bitter	3.47
bother	1.02
brilliant	0.45
excess	0.40
float-a	4.76
knee	0.23
modest	0.29
promise-n	0.17
scrap-n	11.11

Table 1: Effect (gain in % accuracy) of the post-processor.

sion was one of the four submissions to SENSEVAL that scored above this baseline, all four were statictical systems.

Table 2 shows the results per word. The applied algorithm and metric are indicated in the metric column; the value of k in the third column; the values of $M1$, $M2$ and $M3$ in the next column; the accuracy with the optimal settings can be found in the 'tr.opt' column; and the accuracy obtained with the default setting (M1=0.8, M2=5, M3=5; the default suggested by Ng and Lee (1996) and algorithm (IB1-MVDM, k=1, no weighting) is given in the column 'tr.def'. The three rightmost columns give the scores on the evaluation data, measured by the fine-grained, medium, and coarse standard respectively. As we can see from Table 2 the optimization yields a much higher score than the default for some words, the average improvement being 14.8%.

For a general overview of SENSEVAL, and a description of the other participating systems we would like to refer to the special issue of Computers and the Humanities on SENSEVAL (Kilgarriff forthcoming).

6 Conclusion

We have presented a Memory-Based architecture for word sense disambiguation that does not require any hand-crafted linguistic knowledge, but only annotated training examples. Because for the present SENSEVAL task, dictionary information was present, we made use of this as well, and it was easily accommodated in the learning algorithm. In future work we would like to determine what requirements our method has with respect to the amount of training data, and whether it could also feed on dictionary information only, when there should not be an abundant source of labeled training examples.

We believe that Memory-based learning is well-suited to domains such as WSD, where large numbers of features and sparseness of data interact to make life

word	metric	k	M1-M2-M3	basel.	tr.def	tr.opt	ev.f	ev.m	ev.c
accident	MVDM	3	0.3-3-3	67.0	81.4	90.2	92.9	95.4	98.1
amaze	IB1-IG	1	1.0-500-0	57.9	99.7	100	97.1	97.1	97.1
band	IGTREE	-	0.5-7-4	73.0	85.4	88.8	88.6	88.6	88.6
behaviour	MVDM-IG	9	0.3-5-5	95.9	94.9	96.7	96.4	96.4	96.4
bet-n	MVDM-GR	1	0.0-5-100	25.5	56.7	71.1	65.7	72.6	75.5
bet-v	IB1-IG	3	0.7-3-3	37.3	64.3	88.6	76.9	77.8	81.2
bitter	MVDM-IG	5	0.5-5-100	30.6	57.6	59.1	65.8	66.4	66.4
bother	MVDM-IG	3	0.2-5-100	45.6	72.8	83.6	85.2	87.1	87.1
brilliant	MVDM-IG	1	0.6-2-100	47.3	57.5	58.8	54.6	62.0	62.0
bury	MVDM-IG	3	0.5-5-100	32.4	35.9	46.2	50.2	51.0	51.7
calculate	IB1-IG	7	0.7-3-3	72.0	79.2	83.2	90.4	90.8	90.8
consume	IGTREE	-	0.7-5-5	37.5	32.9	58.8	37.3	43.8	49.7
derive	MVDM	5	0.0-2-100	42.9	63.9	67.3	65.0	66.1	66.8
excess	MVDM-IG	5	0.5-1-1	29.1	82.6	89.3	84.4	86.3	88.2
float-a	IGTREE	-	0.3-3-3	61.9	57.0	73.5	57.4	57.4	57.4
float-n	MVDM-IG	1	0.8-5-5	41.3	50.8	70.2	64.0	65.3	68.0
float-v	IGTREE	-	0.4-2-100	21.0	34.2	44.0	35.4	40.6	44.1
generous	MVDM	15	0.6-5-100	32.5	44.8	49.3	51.5	51.5	51.5
giant-a	IGTREE	-	1.0-500-0	93.1	92.8	94.1	97.9	99.5	100
giant-n	MVDM-IG	5	0.2-5-100	49.4	77.2	82.6	78.8	85.6	97.5
invade	MB1-IG	3	0.1-10-1	37.5	48.0	62.7	52.7	59.2	62.3
knee	MVDM-IG	5	0.0-5-100	42.8	70.3	81.4	79.3	81.8	84.1
modest	MVDM-IG	9	0.0-5-100	58.8	61.1	67.1	70.7	72.8	75.2
onion	IB1	1	0.8-5-5	92.3	90.0	96.7	80.4	80.4	80.4
promise-n	MVDM-IG	5	0.2-5-100	59.2	63.6	75.3	77.0	83.2	91.2
promise-v	IB1-IG	3	0.5-5-10	67.4	85.6	89.8	86.2	87.1	87.9
sack-n	MVDM-IG	1	0.3-3-3	44.3	75.0	90.8	84.1	84.1	84.1
sack-v	IB1	9	1.0-500-0	98.9	97.8	98.9	97.8	97.8	97.8
sanction	MVDM-IG	1	0.5-3-3	55.2	74.9	87.4	86.3	86.3	86.3
scrap-n	IB1	1	0.4-5-100	37.0	58.3	68.3	68.6	83.3	86.5
scrap-v	IGTREE	-	0.7-3-3	90.0	88.3	91.7	85.5	97.8	97.8
seize	IGTREE	-	0.5-5-100	27.0	57.1	68.0	59.1	59.1	63.7
shake	MVDM-IG	7	0.2-5-100	24.7	71.5	73.3	68.0	68.5	69.4
shirt	IGTREE	-	0.7-5-5	56.9	83.7	91.2	84.4	91.8	96.7
slight	IB1-IG	1	0.3-3-3	66.8	92.7	93.0	93.1	93.3	93.6
wooden	IGTREE	-	0.5-1-1	95.3	97.3	98.4	94.4	94.9	94.9

Table 2: The best scoring metrics and parameter settings found after 10-fold cross-validation on the training set (see text). The scores are the baseline, the default and optimal settings on the training set (average of 10-fold cross-validation), and the fine-grained, medium and coarse scores on the evaluation set respectively. The scores on the evaluation set were computed by the SENSEVAL coordinators.

difficult for many other (e.g. probabilistic) machine-learning methods, and where nonetheless even very infrequent or exceptional information may prove to be essential for good performance. To determine whether this belief is well-founded, however, we must conduct an extensive error-analysis, since, together with Ng and Lee (1996), this work presents the first excursion of MBL techniques into WSD territory.

In particular, it seems that combining different kinds of information sources in one case-representation may have the effect that the relevance of redundant or highly correlated features is overestimated in the metric, because the feature-weights are determined independently of one another. An interesting alternative seems to be to train a number of different classifiers (one per information source) and combine these using a second level classifier (Wilks and Stevenson 1998).

Although the work presented here is similar to many other supervised learning approaches, and in particular to the Exemplar-based method used by Ng and Lee (1996) (which is essentially IB1-MVDM with k=1), the original aspect of the work presented in this paper, lies in the fact that we have used a cross-validation step per word to determine the optimal parameter-setting, yielding an estimated performance improvement of 14.8 % over their default setting. Moreover, we have used a representation for the direct context that is more simple than that used by Ng and Lee (1996).

Concluding, we can say that the method presented in this paper achieves a relatively high accuracy (see Table 2) with very simple means, and in very fast development time (approximately 2 person months were used to develop the entire architecture, train it and test it, given the availability of TiMBL, the MBL software package).

Acknowledgements

This research was done in the context of the "Induction of Linguistic Knowledge" research programme, partially supported by the Foundation for Language Speech and Logic (TSL), which is funded by the Netherlands Organization for Scientific Research (NWO). We would like to thank the organisers of SENSEVAL.

References

Atkins, S.(1993), Tools for computer-aided lexicography: the HECTOR project, *Papers in Computational Lexicography, COMPLEX'93*, Budapest.

Berleant, D.(1995), Engineering word-experts for word disambiguation, *Natural Language Engineering* pp. 339–362.

Buchholz, S.(1998), Distinguishing complements from adjuncts using memory-based learning, *Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*.

Daelemans, W.(1995), Memory-based lexical acquisition and processing, *in* P. Steffens (ed.), *Machine Translation and the Lexicon*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, pp. 85–98.

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

Daelemans, W. and Van den Bosch, A.(1996), Language-independent data-oriented grapheme-to-phoneme conversion, *in* J. P. H. Van Santen, R. W. Sproat, J. P. Olive and J. Hirschberg (eds), *Progress in Speech Processing*, Springer-Verlag, Berlin, pp. 77–89.

Daelemans, W., Berck, P. and Gillis, S.(1997), Data mining as a method for linguistic analysis: Dutch diminutives, *Folia Linguistica*.

Daelemans, W., Gillis, S. and Durieux, G.(1994), The acquisition of stress: a data-oriented approach, *Computational Linguistics* **20**(3), 421–451.

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

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

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

Daelemans, W., Zavrel, J., Van der Sloot, K. and Van den Bosch, A.(1998b), TiMBL: Tilburg Memory Based Learner, version 1.0, reference manual, *Technical Report ILK-9803*, ILK, Tilburg University.

Kilgarriff, A. (ed.)(forthcoming), *Computers and the Humanities, special issue on Senseval*, Kluwer, Dordrecht, NL.

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

Ng, H. T. and Lee, H. B.(1996), Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach, *Proc. of 34th meeting of the Assiociation for Computational Linguistics*.

Van den Bosch, A., Daelemans, W. and Weijters, A.(1996), Morphological analysis as classification: an inductive-learning approach, *in* K. Oflazer and H. Somers (eds), *Proceedings of the Second International Conference on New Methods in Natural Language Processing, NeMLaP-2, Ankara, Turkey*, pp. 79–89.

Veenstra, J. B.(1998), Fast NP chunking using memory-based learning techniques, *Proceedings of BENELEARN'98*, Wageningen, NL, pp. 71–78.

Wilks, Y. and Stevenson, M.(1998), Word sense disambiguation using optimised combinations of knowledge sources, *COLING-ACL'98, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, August 10-14, 1998*, Université de Montréal, Montréal, Quebec, Canada, pp. 1398–1402.

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

Zavrel, J., Daelemans, W. and Veenstra, J.(1997), Resolving PP attachment ambiguities with memory-based learning, *in* M. Ellison (ed.), *Proc. of the Workshop on Computational Language Learning (CoNLL'97)*, ACL, Madrid.