Natural Language Engineering **8** (4): 311–325. © 2002 Cambridge University Press
DOI: 10.1017/S1351324902003005 Printed in the United Kingdom

Parameter optimization for machine-learning of word sense disambiguation

V. HOSTE[1], I. HENDRICKX[2], W. DAELEMANS[1,2]
and A. VAN DEN BOSCH[2,3]

[1]*CNTS Language Technology Group, University of Antwerp, Belgium*
e-mail: {hoste,daelem}@uia.ua.ac.be

[2]*ILK Computational Linguistics, Tilburg University, The Netherlands*
e-mail: {I.H.E.Hendrickx,Antal.vdnBosch}@kub.nl

[3]*WhizBang! Labs – Research, Pittsburgh, PA, USA*

(*Received 5 November 2001, revised 14 March 2002*)

Abstract

Various Machine Learning (ML) approaches have been demonstrated to produce relatively successful Word Sense Disambiguation (WSD) systems. There are still unexplained differences among the performance measurements of different algorithms, hence it is warranted to deepen the investigation into which algorithm has the right 'bias' for this task. In this paper, we show that this is not easy to accomplish, due to intricate interactions between information sources, parameter settings, and properties of the training data. We investigate the impact of parameter optimization on generalization accuracy in a memory-based learning approach to English and Dutch WSD. A 'word-expert' architecture was adopted, yielding a set of classifiers, each specialized in one single wordform. The experts consist of multiple memory-based learning classifiers, each taking different information sources as input, combined in a voting scheme. We optimized the architectural and parametric settings for each individual word-expert by performing cross-validation experiments on the learning material. The results of these experiments show that the variation of both the algorithmic parameters and the information sources available to the classifiers leads to large fluctuations in accuracy. We demonstrate that optimization per word-expert leads to an overall significant improvement in the generalization accuracies of the produced WSD systems.

1 Introduction

The task of Word Sense Disambiguation (WSD) is to assign a sense label to a word in context. As in most areas of computational linguistics, both knowledge-based and statistical methods have been applied to the problem. See Ide and Véronis (1998) for an introduction to the area. More recently, several machine learning methods have been succesfully applied to this problem. In a seminal paper on the comparison of the accuracy of different Machine Learning (ML) methods on the task of WSD, Mooney (1996) tested seven ML algorithms with different biases on their ability to disambiguate the word *line*. In ML, the concept of bias is used to describe the heuristics implicit in the way a ML algorithm represents its hypotheses about the concept to be learned, and the way in which it searches the space of possible hypotheses. For example, decision-tree learning methods have a bias towards constructing simple decision trees, and use the information-theoretic construct of information gain to find maximally short orderings of principal components of information to split the learning material. The more the bias of a learning algorithm fits the properties of the task, the better its induced model will generalize to new data of the same task. With WSD in mind, it is therefore relevant to investigate which algorithm has a bias best suited for the task of WSD. In Mooney (1996), the conclusion was put forward that within the class of symbolic machine learning methods, decision lists, as also used in Yarowsky (2000), are at an advantage for WSD because of their rule ordering bias. Analogous to decision trees and much like rule induction approaches, decision lists search for a minimal-size ordered set of high-accuracy rules, that disambiguate efficiently and effectively.

Although the methodological set-up of the comparison in Mooney (1996) is sound, and the results provide insight into the role of algorithm bias, we argue that these results are not reliable (cf. Daelemans and Hoste 2002). Mostly for practical reasons of experimental and computational complexity, Mooney's comparative study and many other studies in ML of NLP tasks are limited to default settings of algorithm parameters and a constant input representation. However, the effect of algorithm bias on generalization accuracy is easily overwhelmed by the effect of algorithm parameter variation, input feature representation and selection, and the interaction between both. A similar argument can be found in Banko and Brill (2001) for the effect of training set size: for the task of disambiguating *confusables* (words like *it's* and *its* which are easily confused in writing), increasing training data with a factor 10^3 has a significantly larger effect on generalization accuracy than the choice of algorithm on the 'smaller' (still 1 million cases) training set, and the effect of algorithm bias becomes considerably smaller at these large training set sizes.

In this paper, we demonstrate the drastic effects of architecture and algorithm parameter optimization and of selection of information sources on generalization performance in a *memory-based learning* approach to all-words WSD for Dutch and English. In the remainder of this paper, we first describe the memory-based learning algorithms used in our experiments. In section 3, we outline the system architecture used in the experiments for Dutch and English all-words WSD, and discuss our word-expert approach. Section 4 reports on the different information sources extracted from the data for training the classifiers. In section 5, the optimization procedure and its results are described in detail. In section 6, we report on the generalization accuracy achieved for the Senseval-2 test data for English and Dutch. We conclude with a discussion and interpretation of our results in section 7.

2 Memory-based learning and WSD

We interpret WSD as a classification task: given a possibly ambiguous word and its context as input features, a classifier assigns the contextually correct class (sense) to it. Information about the local context and information about keywords in the context is provided as the information sources, coded in a feature vector. In the memory-based supervised learning set-up adopted here, all contexts in which an ambiguous word occurs in the learning material are kept in memory to extrapolate from during the classification of new test data. The distinguishing feature of Memory-Based Learning (MBL) in contrast with minimal-description-length-driven or 'eager' ML algorithms (e.g. decision trees and decision lists) is that MBL keeps all training data in memory, and only abstracts at classification time by extrapolating a class from the most similar item(s) in memory to the new test item. This strategy is often referred to as 'lazy' learning. In recent work (Daelemans, van den Bosch and Zavrel 1999), we have shown that for typical natural language processing tasks, this lazy learning approach performs well because it allows extrapolation from low-frequency or exceptional cases, whereas eager methods tend to treat these as discardable noise. Also, 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 (Zavrel and Daelemans 1997). For our experiments, we used the MBL algorithms implemented in TIMBL[1]. We give a brief overview of the algorithms and metrics here, and refer to Daelemans, van den Bosch and Weijters (1997) and Daelemans, Zavrel, van der Sloot and van den Bosch (2001) for more information.

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 (Aha, Kibler and Albert (1991). Classification occurs via the *k-nearest-distances* rule: all memory items which are equally near at the nearest *k* distances surrounding the test item are taken into account in classification. The classification assigned to the test item is simply the majority class among the memory items at the *k* nearest distances.

Feature-weighted IB1: in most cases, not all features are equally relevant for solving the task; different types of weighting are available in TIMBL to assign differential cost to a feature value mismatch during comparison. Some of these are information-theoretic (based on measuring the reduction of uncertainty about the class to be predicted when knowing the value of a feature): information gain and gain ratio. Others are statistical (based on comparing expected and observed frequencies of value-class associations): chi-squared and shared variance.

IGTREE: an oblivious decision tree is created with features as tests, and ordered according to one of the feature weighting methods discussed earlier, as a heuristic approximation of the computationally more expensive pure k-nearest distance classifier.

3 WSD architecture

Our approach to memory-based all-words WSD for Dutch and English follows the memory-based approach of Ng and Lee (1996), who used PEBLS (Cost and Salzberg 1993) as memory-based learner; and the work by Veenstra, Van den Bosch,

[1] Available from http://ilk.kub.nl

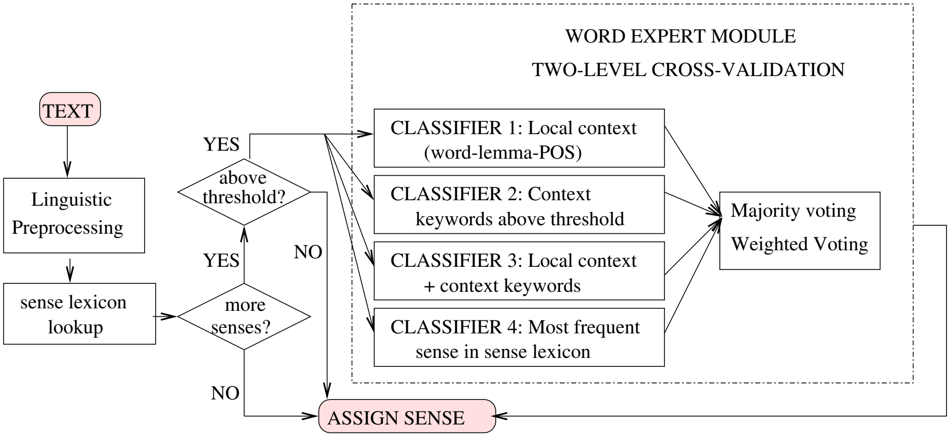
WORD EXPERT MODULE
TWO-LEVEL CROSS-VALIDATION
TEXT
Linguistic Preprocessing
sense lexicon lookup
more senses?
YES
NO
above threshold?
YES
NO
CLASSIFIER 1: Local context (word-lemma-POS)
CLASSIFIER 2: Context keywords above threshold
CLASSIFIER 3: Local context + context keywords
CLASSIFIER 4: Most frequent sense in sense lexicon
Majority voting
Weighted Voting
ASSIGN SENSE

Fig. 1. Disambiguation process.

Buchholz, Daelemans and Zavrel (2000) on a memory-based approach to the English lexical sample task of Senseval-1. We borrow the classification-based approach, and the word-expert concept (Berleant 1995) of the latter: for each wordform (or, for our English experiments, the combination of a wordform plus its part-of-speech), a word-expert classifier is trained using local context and keyword features from the context of occurrences of the word in the available training data, and this classifier is used to classify previously unseen occurrences of the word in the test data.

To make this approach suited for an all-words WSD task, and to get insight into the role of optimization in ML of WSD, a systematic matrix of experiments was designed. In this section, we show how our WSD system was built on the basis of the training material. We describe the general set-up and design of the architecture and development of both English and Dutch WSD systems. Figure 1 outlines the designed architecture. First, the training text is linguistically analyzed at the level of tokenization, Part-Of-Speech (POS) tagging, and lemmatization. If a wordform has more than one sense, and the number of available training items (occurrences in the training data) is above a certain threshold, a word-expert module is trained for it. Otherwise the default, i.e. the most frequent sense is used.

A word-expert consists of four classifiers: (i) a memory-based learner trained on the local context of the occurrence of an ambiguous word (with information about word, POS tag and lemma); (ii) a memory-based learner trained on keywords, selected according to a statistical criterion; (iii) a memory-based learner trained on both of the previous information sources; and (iv) a lexical default; a classifier always providing the most frequent sense in the sense lexicon. The algorithm parameters and feature weights for each component classifier are optimized using cross-validation on the training data. The output of the optimized classifiers is combined using different types of voting methods. Among these four single classifiers and four voters, it is decided per word-expert in the train set which is the optimal classification method for that particular word in the test set. All experiments were performed using ten-fold cross-validation (Weiss and C. Kulikowski 1991) as experimental method for error estimation. For each cross-validation, the data set was first partitioned into ten equal-sized sets and then each set was in turn used as test set while the classifier was trained on the other nine sets concatenated into one training set. All reported results on the training data are averaged over the ten folds.

In subsequent sections we discuss the architectural design in more detail.

3.1 Preprocessing the training text

Our experiments are based on an English and a Dutch word-sense labeled text corpus. We briefly describe them here.

English data. For English, the Semcor corpus included in WordNet1.6 (Fellbaum 1998) was used for training. In this corpus, every word is linked to its appropriate sense in the WordNet lexicon. The texts that were used to create the semantic concordances were extracted from the Brown Corpus and then linked to senses in WordNet. The training corpus consists of 409,990 wordforms, of which 190,481 are sense-tagged. For each wordform, a lemma and a part-of-speech is given. As test data, we used the corpus which was created for the English all-words task in the SENSEVAL-2 competition for automatic WSD. The test data consist of three journalistic articles on different topics, with a total of 2473 words to be sense-tagged. The words to be sense-tagged were exclusively content words: nouns, verbs, adjectives and adverbs. To build our experimentation data, we extracted only the wordforms and associated senses from the training and the test corpus. Tokenization, lemmatization and POS-tagging were done with our own software. For the part-of-speech tagging, the memory-based tagger MBT (Daelemans, Zavrel, Berck and Gillis 1996), trained on the *Wall Street Journal* corpus[2], was used. On the basis of word and POS information, lemmatization was performed (van den Bosch and Daelemans 1999).

Dutch data. The Dutch WSD corpus was originally built as a part of a sociolinguistic project (Schrooten and Vermeer 1994). The corpus consists of texts of 102 illustrated children books for the age range of 4–12. The data was annotated by six people who all processed a different part of the data. Each word in the dataset is assigned a non-hierarchical, symbolic sense tag, realized as a mnemonic description of the specific meaning the word has in the sentence. The dataset also contains senses that span over multiple words, covering idiomatic expressions. The dataset consists of 152,758 sense-tagged tokens (words and punctuation tokens). For SENSEVAL-2, the dataset was divided in two parts. The training set consisted of 114,988 words (76 books) and the test set had 37,770 words (26 books). For the part-of-speech tagging, the memory-based tagger MBT, trained on the Dutch Eindhoven corpus (Daelemans *et al.* 1996) was used. No lemmatization was performed.

[2] ACL Data Collection Initiative CD-Rom 1, September 1991.

3.2 Building word-experts for the training data

After the preprocessing stage, the sense lexicons for both languages were used to guide the sense disambiguation process (cf. Figure 1). For English, WordNet1.7 functioned as sense lexicon. For every combination of a wordform and a part of speech, the sense lexicon was consulted to determine whether this combination had one or more possible senses. For Dutch, the sense lexicon was derived directly from the learning material. In case of only one possible sense, the appropriate sense was assigned (English: 37,681 out of 190,481 words; Dutch: 59,646 out of 114,988 words).

In principle, word-experts should be constructed for all words with more than one sense (English: 152,800 out of 190,481 words; Dutch: 55,342 out of 114,988 words). However, both training corpora are fairly small, and many ambiguous words will occur only a few times. Since more than a few examples are generally needed to induce a sensible ML model, it is likely that there is an optimal threshold in the number of examples above which word-experts could be built that would be better than baseline. This was determined, for each corpus separately, through cross-validation experiments in which the threshold was varied between a minimum of 10 and 100 training items. The minimum threshold was chosen as 10, since optimization was performed with ten-fold cross-validation. In the training sets, 110,296 words (English) and 54,121 words (Dutch) exceeded the threshold of 10, respectively, and were considered possible candidates for building a word-expert. For all words of which the frequency is lower than the threshold, the most frequent sense is predicted.

Figure 2 shows the results of the best classifiers in contrast with the baseline classifiers (always predicting the most frequent sense) for both data sets, when varying the lower bound threshold for building word-experts. These lower bound thresholds represent the minimal number of training items for the construction of a word-expert. For example, setting this threshold to 40 means that word-experts are built for words for which 40 and more training instances are available. For all

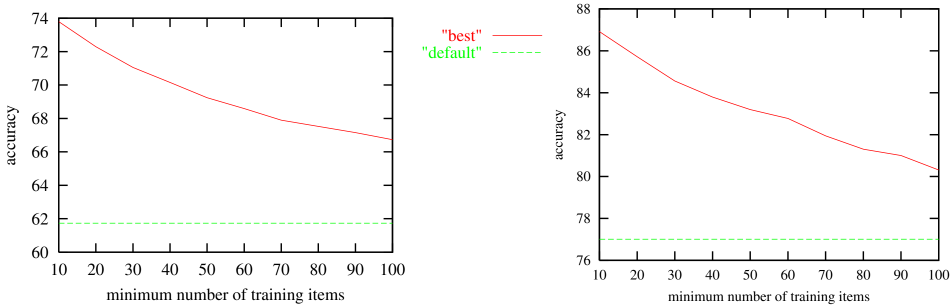

Fig. 2. Accuracies on the English (left) and Dutch (right) train set of the best performing classifier per word-expert in relation to a threshold varying between 10 and 100. Accuracies are calculated on words with more than one possible sense which qualify for the construction of a word-expert. The straight bottom graphs represent the baseline scores.

words below this threshold, the most frequent sense is predicted. The flat line in both plots represents the average accuracy of the baseline classifier over all words in the training set which qualify for the construction of a word-expert. The application of the WordNet baseline classifier yields a 61.7% classification accuracy for the English training set. For the Dutch training corpus, the training-set-based lexical default classifier produces 77.0% accuracy. The 'best' graph displays the accuracy when applying the optimal classifier for each single word-expert. The results illustrate that accuracy drops when the contribution of the baseline classifier increases. With a threshold of 10, the highest accuracy level is obtained for both English (73.8%) and Dutch (86.9%). When this threshold is raised to 100, generalization accuracy drops to 66.7% (English) and 80.3% (Dutch).

These results show that the MBL approach outperforms the strong statistical baseline already at small training set sizes. On the basis of this cross-validation step, the threshold for the construction of a word-expert was set to 10, leading to 2401 experts for the English training data, and 502 experts for Dutch.

4 Different information sources

The word-experts consist of different trained subcomponents (cf. Figure 1) which make use of different information sources (see also Stevenson and Wilks (2001) for recent research on combination of knowledge sources). The first classifier disambiguates senses on the basis of local information. To represent local information, a snapshot is made of the immediate local neighborhood surrounding the ambiguous word in question, the *focus* word. As a heuristic approximation of directly surrounding neighborhood, three wordforms to the left and three to the right are included in the local window. For the English focus words, the focus word itself, its lemma and part-of-speech are included in the representation as well, whereas for the Dutch focus word, only the parts-of-speech are included. For the local context wordforms in both data sets, the wordforms and their parts-of-speech are given. For example, the following instances are the training instances for the English sentence fragment "(...) no_matter whether he has short or long (...)" and the Dutch fragment "(...) de weg niet meer terug kunnen vinden (...) [not more able to find the road back]":

no_matter RB whether IN he PRP has have VBZ short JJ or CC long JJ have%2:42:00::
Art de N weg Adv niet Adv Adv terug V kunnen V vinden meer_adv

These training instances for the focus words 'has' (lemma: 'have' and part-of-speech: 'VBZ') and 'meer' (part-of-speech: 'Adv') shows at the first six positions the three words to the left of the focus together with their parts-of-speech. The local context words and parts-of-speech to the right of the focus word are represented in the last six features. At the end of the instance, the classification of the focus word is given. In the English data set, this is a so-called 'sense-key' (Fellbaum 1998), which uniquely identifies each word/sense pair in the WordNet lexicon. Each sense-key consists of a wordform, followed by a syntactic category (e.g. '2' stands for 'verb'), a semantic field and an identification number. In the Dutch instances, the classifications are mnemonic sense tags.

The second subcomponent of each word-expert is trained on information about possible disambiguating keywords in a context of three sentences: the sentence in which the ambiguous word occurs, the previous sentence, and the following sentence. The method used to extract these keywords for each sense is based on the work of Ng and Lee (1996). They determine the probability of a sense s of a focus word f given keyword k by dividing $N_{s,kloc}$ (the number of occurrences of a possible local context keyword k with a particular focus word-POS combination w with a particular sense s) by N_{kloc} (the number of occurrences of a possible local context keyword $kloc$ with a particular focus word-POS combination w regardless of its sense). In addition, we also took into account the frequency of a possible keyword in the complete training corpus N_{kcorp}:

$$p(s|k) = \frac{N_{s,kloc}}{N_{kloc}} \times \left(\frac{1}{N_{kcorp}}\right). \tag{1}$$

A word is a keyword for a given sense if (i) the word occurs more than M_1 times in that sense s, where M_1 is a predefined minimum number of times, and if (ii) $p(s|k) \geq M_2$ for that sense s, where M_2 is some predefined minimum probability. For our experiments, M_1 was set to 3 and M_2 to 0.001 after evaluating different values in a preliminary experiment. For English, we also used the extra information encapsulated in the WordNet information associated with ambiguous wordforms: all content words present in the example sentences that accompany the different sense definitions for a given focus word in the lexicon were encoded as keywords.

For each combination of a wordform (or wordform-POS combination in the English case) and sense, i.e. per word-expert, all keywords were selected and added to the input vector of the memory-based learner. Keywords are represented as binary features, with a value of 1 when the keyword is present in the example and 0 if not[3].

A third subcomponent of each word-expert is trained with both local-context and keyword information. An important aspect of memory-based learning is that it can take into account and integrate (by means of its automatic feature weighting and implicitly parallel similarity metric) diverse information sources.

Together with the fourth component, the straightforward lexical default (the most frequent sense in the sense lexicon), these classifiers are optimally combined as described in the next Section.

5 Optimization and voting

To improve the predictions of the different single learning algorithms, algorithm parameter optimization was performed where possible by cross-validating on the

[3] Since no length limitations were taken into account when building these vectors, they could grow very large. Therefore, a version of TIMBL was used, written by Jakub Zavrel, that indexes binary vectors only on the active bits.

training data. Furthermore, the possible gain in accuracy of different voting strategies was explored. The following algorithm parameters were optimized:

- *Optimization of different feature weighting metrics*: gain ratio weighting, information gain weighting, chi-squared weighting, shared variance weighting and log-likelihood weighting (see Daelemans *et al*. (2001)).
- *Optimization of the k value*, representing the number of nearest distances in which memory items are searched. In the experiments, k was varied between 1, 3, 5, 7, 9, 11, 15, 25, 35 and 45. Odd numbers are chosen to avoid ties. k was optimized both for the local-context learner and the keyword-based learner.
- *Optimization of distance metrics*: number of mismatches, number of matches and number of matches minus number of mismatches.

Figure 3 exemplifies the dramatic effect algorithm parameter optimization can have on accuracy for different word experts. The figure shows accuracy levels for different settings of weighting method and value of k for the English verb *be* and the English noun *basis*. For example, for the verb *be*, accuracy differs as much as 20.9% for the classifier which takes keyword information as input. Yet, optimal

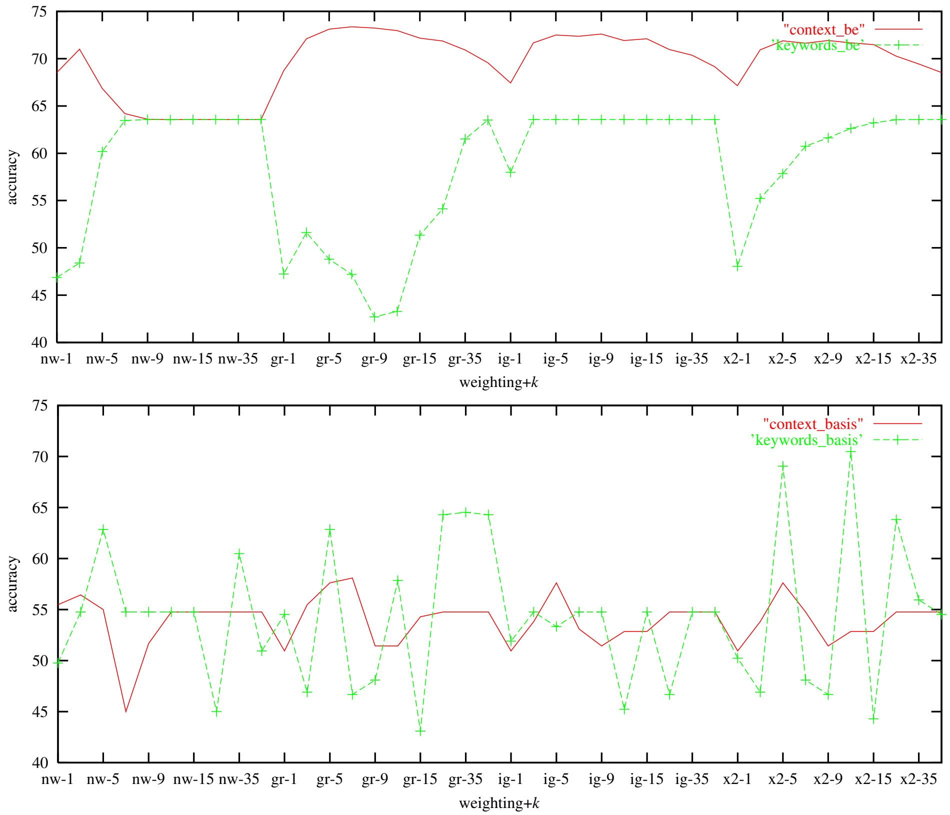

Fig. 3. Influence of the choice of information source on the generalization accuracy for different feature weighting methods (no weighting, gain ratio, information gain and chi-squared weighting) and for different k values (1, 3, 5, 7, 9, 11, 25, 35, 45). The two plots show the accuracy graphs for the English words ‘be’ and ‘basis’.

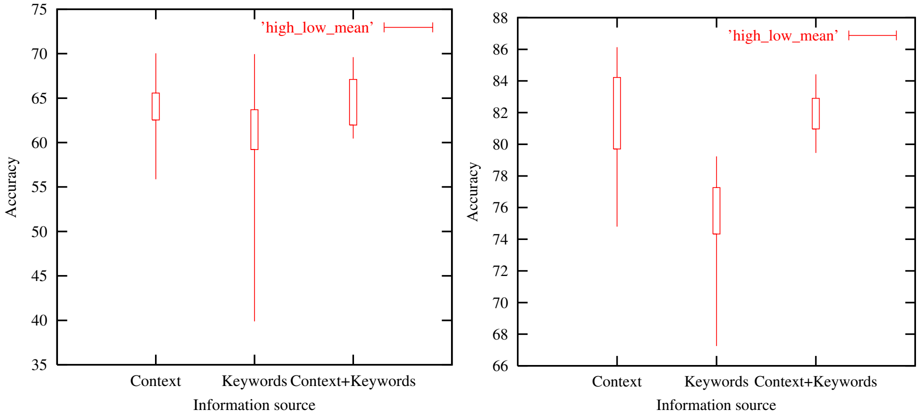

Fig. 4. Results of the three single classifiers over all parameter settings of all word-experts (weighted by frequency of the expert) for English (left) and Dutch (right). Graphs show the average difference between the accuracies obtained with the best and worst parameter setting per word-expert. The boxes in the graphs represent averages and standard deviations.

parameter settings for one word-expert cannot be generalized to other word experts. Furthermore, optimal parameter settings are dependent on the information source used as well.

Figure 4 displays the average between the best and worst accuracies on training material obtained with optimizing algorithmic parameters for all word-experts, varying among three information source settings: local context words only, keywords only, and the combination of the two. The classifiers trained only on keywords are especially sensitive to the algorithmic parameter setting: over all word-experts, accuracies vary between 39.9% and 70.0% for English and between 67.3% and 79.2% for Dutch. The results indicate that no general conclusion could be drawn concerning the importance of each individual information source when only the default settings of the algorithms would be used; considerable improvements may lie in parametric optimalization.

Subsequently, on the output of these three classifiers (optimized per word-expert) and the default most frequent sense according to the sense lexicon, both majority voting and weighted voting was performed. In case of majority voting, each sense-tagger is given one vote; the tag with most votes is selected. In weighted voting, more weight is given to the taggers with a higher overall accuracy. In the case of ties when voting over the output of four classifiers, the first decision (memory-based classifier based on local context information) was taken as output class. As an alternative, voting was also performed on the output of the three learning classifiers without taking into account the most frequent class.

On the basis of the results on the train set of all component classifiers and voting strategies, it was decided per word-expert which was the best performing algorithm. This classifier was then applied to the test data. Table 1 shows the number of times a specific classifier is selected as 'best' classifier. For both data sets, the 'local context' classifier is the preferred choice, followed by the 'keywords' classifier. In case of ties,

Table 1. *Number of word-experts in the test data for which a MBL classifier/voter is selected as best performing classifier*

	English	Dutch
Local context	186	134
Keywords	176	93
Local context + keywords	13	62
Majority voting	48	23
Majority voting (no default)	43	12
Weighted voting	66	44
Weighted voting (no default)	20	6
Default	29	13
Tie	14	96
Total	596	483

the WordNet baseline classifier was used for the classification of the English test items. When ties occured in the Dutch training set, the 'local context' classifier was applied on the test data.

6 Results on the hold-out test set

On the basis of the training material 2401 word-experts were built for English and 502 experts for Dutch. The optimal parameter settings of the different TIMBL classifiers within those word-experts were determined using ten-fold cross-validation. Once determined the optimal parameter settings for all word-experts, they were trained all over again with these settings, this time on the complete train set. They were then applied to the hold-out test corpus. For Dutch, the Senseval-2 test set was part of the corpus from which also the train set was extracted. For English, a test set was provided which was completely independent from the Semcor training set.

Table 2 shows the accuracy of our disambiguation system on the English and Dutch word forms in the Senseval-2 all-words test sets for which a word-expert exists. For English, 596 out of the 2401 train set word-experts can be applied to the test set. For the other train set word-experts, no testing material was provided. 1404 out of 2473 English test words are classified by a word-expert. In the Dutch test data, 483 of the 502 training set word-experts handle 17,456 of the 37,770 test items. The table shows that a word-expert approach to this WSD task task leads to a considerable increase in performance compared to the baseline classifier. Furthermore, the accuracy levels for the 'best' and the 'local context' classifiers are very close for both data sets. Table 1 already showed that the optimized 'local context' classifier was most frequently selected as best classifier. For English, the application of the 'best' classifier leads to a 54.6% top accuracy. The classifier which takes local context information as input performs best for the Dutch data: 84.2%.

A separate issue is the difference between the accuracies of the best English and Dutch systems on the words for which a word-expert is built. The senses discerned

Table 2. *Accuracy on the English and Dutch test words for which a word expert is built. The table includes the scores of all (optimized) component classifiers of the word-experts*

	English	Dutch
Best	**54.6**	84.0
Local context	53.9	**84.2**
Keywords	50.0	76.2
Local context + keywords	51.5	83.2
Majority voting	50.4	77.2
Majority voting (no default)	52.4	84.0
Weighted voting	51.9	84.0
Weighted voting (no default)	52.2	84.0
Default	48.9	76.8

in the English task are explicitly intended to be semantic and not syntactic, while the Dutch senses include syntactic differences – this coincides with the fact that English word-experts are based on word-POS tag combinations, while Dutch word-experts focus on wordforms. This makes the Dutch task partly a part-of-speech tagging task, which is generally held to be an easier task than word sense disambiguation. For example, the sense tags in the following instances for the focus word 'meer' are based on part-of-speech information.

N rand Prep van Art het Num V woont Adj woeste N Willem meer_N [border of the lake lives fierce Willem]
Art de N weg Adv niet Adv Adv terug V kunnen V vinden meer_adv [not more able to find the road back]

A similar approach of sense tag encoding – with equally high scores – to the one used in the Dutch task has also been reported in Stevenson and Wilks (2001). On a corpus of 5 articles in the *Wall Street Journal*, their system already correctly classifies 87.4% of the words when only using POS information (baseline: 78%).

We also calculated the accuracy of our disambiguation system on the complete English all words test set, which contains 2473 wordforms. For the English data, an accuracy of 63.6% and 64.5% were obtained according to the fine-grained and coarse-grained SENSEVAL-2 scoring, respectively. With these accuracies, our disambiguation system performed second best in the SENSEVAL-2 English all words competition. For further information, see Hoste, Kool and Daelemans (2001). In the Dutch test set, 91.9% of the words were correctly classified.

An important usability aspect of any WSD system applied in practice, apart from accuracy as analysed above, is its efficiency in terms of speed and memory. Arguably, efficiency in speed and memory during actual word-sense disambiguation is the most critical. We measured the total time (in seconds) it took in both systems to classify all word expert test material (i.e. all ambiguous words in the test sets). Both systems were installed separately on a stand-alone desktop computer[4], involving the deployment of all individual word experts (502 for the Dutch system, and 2401 for the English) as classification servlets. All word experts took the form of the optimized IB1 module trained on local context features only. The English system classified its 1404 test words in 24.5 s (elapsed wall clock time), or 57 words per second. The Dutch system classified its 17,469 test words in 47.0 s, or 372 words per second. Both speeds appear to be reasonable for real-time processing in possibly larger NLP systems. The relative slowness of the English system can be explained from the fact that the 1404 test words come from 596 different files, causing considerable file opening overhead in the total time.

7 Conclusions

We have shown that memory-based word-sense disambiguation systems for English and Dutch can benefit from optimizing architecture, information sources and algorithmic parameters on the basis of the learning material. For each different word-expert, different combinations of classifiers with different parameter settings are needed for optimal performance.

Our architecture for WSD allows more optimization of this type: in preliminary research on varying the values of the $M1$ and $M2$ parameters used for keyword selection in a cross-validation on the training data, accuracies vary between 63.8% and 67.7% for English, with selected keywords varying between 7.3 and 0.9. For Dutch, the variance is smaller: accuracies on the training material vary between 78.0% (1.8 keywords) and 79.6% (4.2 keywords selected). Again, this shows that parameter optimization can have a large effect on accuracy.

In general, we argue that changing any of the architectural variables (algorithm parameters, information sources, architectural parameters) can have great effects on the accuracy, making doubtful many conclusions in the literature based on default settings of algorithms only. It is our impression that these effects are at least intuitively known by most researchers working on ML of natural language, but little grounded evidence nor explanations are available in the literature. Moreover, there appears to be especially little understanding of the *interaction* between these variables. Many empirical findings, though illustrative, are observations on experiments in which one or two variables are alternated, but in which no overall optimization of parameters, architecture and feature representation is undertaken (Mooney 1996).

As long as no fundamental data-independent explanation is found for such phenomena, data-dependent cross-validation is likely to provide the right clues for improving word-sense disambiguation systems by considerable margins. We argue that cross-validating parametric settings of architecture and information sources should be included as a first step in constructing WSD systems – and NLP systems

[4] The test computer was running Linux on a i386 platform with 512 Mb of memory and one 900 Mhz processor.

in general. Memory-based learning, because of its efficiency, proved to be a good machine learning candidate to investigate this effect.

We believe that the dependence of system accuracy on the selection of information sources and algorithm parameters, which we demonstrated in this article for the case of memory-based learning, will be observable and testable with any machine learning method, including decision lists and decision trees and other methods often used for WSD. Although some explorative optimization is sometimes used in experiments with these methods, an exhaustive optimization of the type attempted here is not common. Only after such an optimization is achieved, different machine learning methods can be compared reliably.

References

Aha, D. W., Kibler, D. and Albert, M. (1991) Instance-based learning algorithms. *Machine Learning*, **6**: 37–66.

Banko, M. and Brill, E. (2001) Scaling to very very large corpora for natural language disambiguation. *Proceedings 39th Annual Meeting of the ACL*, pp. 26–33. ACL.

Berleant, D. (1995) Engineering word-experts for word disambiguation. *Natural Lang. Eng.* **1**(4): 339–362.

Cost, S. and Salzberg, S. (1993) A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* **10**(1): 57–78.

Daelemans, W. and Hoste, V. (2002) Evaluation of machine learning methods for natural language processing tasks. *Proceedings 3rd International Conference on Language Resources and Evaluation*, pp. 755–760.

Daelemans, W., van den Bosch, A. and Weijters, T. (1997) Igtree: Using trees for compression and classification in lazy learning algorithms. *Artif. Intell. Rev.* **11**: 407–423.

Daelemans, W., van den Bosch, A. and Zavrel, J. (1999) Forgetting exceptions is harmful in language learning. *Machine Learning*, **34**: 11–43.

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

Daelemans, W., Zavrel, J., van der Sloot, K. and van den Bosch, A. (2001) Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, ILK Technical Report 01-04.

Fellbaum, C. (1998) *WordNet: An Electronic Lexical Database*. MIT Press.

Hoste, V., Kool, A. and Daelemans, W. (2001) Classifier optimization and combination in the English all words task. *Proceedings 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 83–86.

Ide, I. and Véronis, J. (1998) Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, **24**(1): 1–40.

Mooney, R. (1996) Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. *Proceedings Conference on Empirical Methods in Natural Language Processing*, pp. 82–91.

Ng, H. T. and Lee, H. B. (1996) Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In: Joshi, A. and Palmer, M., editors, *Proceedings 34th Annual Meeting of the ACL*, pp. 40–47. Morgan Kaufmann.

Schrooten, W. and Vermeer, A. (1994) *Woorden in het basisonderwijs. 15.000 woorden aangeboden aan leerlingen.* TUP (Studies in meertaligheid 6).

Stevenson, M. and Wilks, Y. (2001) The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, **27**(3): 321–349.

van den Bosch, A. and Daelemans, W. (1999) Memory-based morphological analysis. *Proceedings 37th Annual Meeting of the ACL*, pp. 285–292.

Veenstra, J., van den Bosch, A., Buchholz, S., Daelemans, W. and Zavrel, J. (2000) Memory-based word sense disambiguation. *Comput. and the Humanities*, **34**(1/2): 171–177.

Weiss, S. and Kulikowski, C. (1991) *Computer Systems that Learn*. Springer Verlag.

Yarowsky, D. (1991) Hierarchical decision lists for word sense disambiguation. *Comput. and the Humanities*, **34**(1/2): 179–186.

Zavrel, J. and Daelemans, W. (1997) Memory-based learning: Using similarity for smoothing. *Proceedings 35th Annual Meeting of the ACL*, pp. 436–443.