Combined Optimization of Feature Selection and Algorithm Parameters in Machine Learning of Language

Walter Daelemans[1], Véronique Hoste[1], Fien De Meulder[1], and Bart Naudts[2]

[1] CNTS Language Technology Group
University of Antwerp
Universiteitsplein 1, B-2610 Antwerpen
{daelem,hoste,dmeulder}@uia.ua.ac.be

[2] Postdoctoral researcher of the Fund for Scientific Research, Flanders, Belgium
ISLAB, University of Antwerp
bart.naudts@ua.ac.be

Abstract. Comparative machine learning experiments have become an important methodology in empirical approaches to natural language processing (i) to investigate which machine learning algorithms have the 'right bias' to solve specific natural language processing tasks, and (ii) to investigate which sources of information add to accuracy in a learning approach. Using automatic word sense disambiguation as an example task, we show that with the methodology currently used in comparative machine learning experiments, the results may often not be reliable because of the role of and interaction between feature selection and algorithm parameter optimization. We propose genetic algorithms as a practical approach to achieve both higher accuracy within a single approach, and more reliable comparisons.

1 Introduction

Supervised machine learning methods are investigated intensively in empirical computational linguistics because they potentially have a number of advantages compared to standard statistical approaches. For example, Inductive Logic Programming (ILP) systems allow easy integration of linguistic background knowledge in the learning system, induced rule systems are often more interpretable, memory-based learning methods incorporate smoothing of sparse data by similarity-based learning, etc.

Frequently, research in machine learning (ML) of natural language takes the form of *comparative ML experiments*, either to investigate the role of different information sources in learning a task, or to investigate whether the bias of some learning algorithm fits the properties of natural language processing tasks better than alternative learning algorithms.

For the former goal, results of experiments with and without a certain information source are compared, to measure whether it is responsible for a statistically significant increase or decrease in accuracy. An example is text categorization: we may be interested in investigating whether part-of-speech tagging

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 84–95, 2003.

(adding the contextually correct morphosyntactic classes to the words in a document) improves the accuracy of a Bayesian text classification system or not. This can be achieved by comparing the accuracy of the classifier with and without the information source.

In the latter goal, investigating the applicability of an algorithm for a type of task, the *bias* of an algorithm refers to the representational constraints and specific search heuristics it uses. Some examples of bias are the fact that decision tree learners favor compact decision trees, and that ILP systems can represent hypotheses in terms of first order logic in contrast to most other learning methods which can only represent propositional hypotheses. In such experiments, two or more ML algorithms are compared for their accuracy on the same data. One example is a comparison between eager and lazy learning algorithms for language tasks: we may want to show that abstracting from infrequent examples, as done in eager learning, is harmful to generalization accuracy [5].

Apart from their inherent interest, the comparative machine learning approach has also gained importance because of the influence of competitive research evaluations such as SENSEVAL[1] and the CoNLL shared tasks[2], in which ML and other systems are compared on the same train and test data. SENSEVAL concerns research on *word sense disambiguation*, which we will use as a test case in this paper.

Word Sense Disambiguation (WSD) is a natural language processing task in which a word with more than one sense has to be disambiguated by using information from the context in which the word occurs. E.g. *knight* can (among others) refer to a chess piece or a medieval character. WSD is an essential subcomponent in applications such as machine translation (depending on the sense, *knight* will be translated into French as either *cavalier* or *chevalier*), language understanding, question answering, information retrieval, and so on. Over the last five years, two SENSEVAL competitions have been run to test the strengths and weaknesses of WSD systems with respect to different words, different aspects of language, and different languages in carefully controlled contexts [10,8]. Machine learning methods such as decision list learning and memory-based learning have been shown to outperform hand-crafting approaches in these comparisons, leading to a large body of comparative work of the two types discussed earlier.

A seminal paper by Mooney on the comparison of the accuracy of different machine learning methods [16] on the task of WSD is a good example of this classifier comparison approach. He tested seven ML algorithms on their ability to disambiguate the word *line*, and made several conclusions in terms of algorithm bias to explain the results. Many more examples can be found in the recent NLP literature of similar studies and interpretations [17,9,15], often with contradictory results and interpretations.

In the remainder of this paper, we will first describe standard methodology in Section 2 and show empirically that this methodology leads to conclusions that are not reliable for our WSD problem and for other machine learning tasks

[1] http://www.senseval.org.

[2] http://www.aclweb.org/signll.

inside and outside computational linguistics (Section 3). In Section 4 we show that the joint optimization of feature selection and algorithm parameters using a genetic algorithm is computationally feasible, leads in general to good results, and could therefore be used both to achieve higher accuracy and more reliable comparisons. Section 5 discusses our results in the light of related research.

2 Limitations of Standard Methodology

Crucial for objectively comparing algorithm bias and relevance of information sources is a methodology to reliably measure differences and compute their statistical significance. A detailed methodology has been developed for this [21] involving approaches like k-fold cross-validation [11,1,7] to estimate classifier quality (in terms of accuracy or derived measures like precision, recall, and F-score), as well as statistical techniques like McNemar [7] and paired cross-validation t-tests for determining the statistical significance of differences between algorithms or between presence or absence of information sources. Although this methodology is not without its problems [18], it is generally accepted and used both in machine learning and in most work in statistical NLP.

Many factors potentially play a role in the outcome of a (comparative) machine learning experiment: the data used (the sample selection and the sample size), the information sources used (the features selected) and their representation (e.g. as nominal or binary features), and the algorithm parameter settings (most ML algorithms have various parameters that can be tuned).

In a typical comparative machine learning experiment, two or more algorithms are compared for a fixed sample selection, feature selection, feature representation, and (default) algorithm parameter setting over a number of trials (cross-validation), and if the measured differences are statistically significant, conclusions are drawn about which algorithm is better suited to the problem being studied and why (mostly in terms of algorithm bias)[3]. Sometimes different sample sizes are used to provide a learning curve, and sometimes parameters of (some of the) algorithms are optimized on training data, or heuristic feature selection is attempted, but this is exceptional rather than common practice in comparative experiments. Interactions between different factors, like the effect of interleaved feature selection and algorithm parameter optimization, have to the best of our knowledge not yet been investigated systematically in comparative machine learning experiments for language processing problems.

In the remainder of this paper, we test the hypotheses that (i) feature selection, algorithm parameter optimization, and their joint optimization cause larger differences in accuracy within a single algorithm than differences observed between different algorithms using default parameter settings and feature input, and (ii) that the effect of adding and removing an information source when using default parameters can be reversed when re-optimizing the algorithm parameters. The implication of evidence for these hypotheses is that a large part of the comparative machine learning of language literature may not be reliable.

[3] A similar approach is taken for the comparison of information sources.

Another implication is that joint optimization can lead to significantly higher generalization accuracies, but this issue is not the focus of this paper.

3 Feature Selection, Parameter Optimization, and Their Interaction

In this Section, we analyze the impact of algorithm parameter optimization, feature selection, and the interaction between both on classifier accuracy in comparative experiments on WSD data and on the UCI benchmark datasets.

Feature (subset) selection is the process in which a subset of the available predictor features defining the input of the classification task are removed if they cannot be shown to be relevant in solving the learning task [11]. For computational reasons we used a *backward selection* algorithm. We start with all available features and look at the effect on accuracy of deleting one of the features, and continue deleting until no more accuracy increase is reported. Algorithm parameter optimization is the process in which parameters of a learning system (e.g. learning rate for neural networks, or the number of nearest neighbors in memory-based learning), are tuned for a particular problem. Although most machine learning systems provide sensible default settings, it is by no means certain that they will be *optimal* parameter settings for some particular task. In both cases (feature selection and parameter optimization), we are performing a *model selection* task which is well-known in machine learning. But as we mentioned earlier, whereas some published work in computational linguistics discusses either feature selection for some task, or algorithm parameter optimization for others, the effects of their interaction have, as far as we know, never been studied systematically.

The general set-up of our experiments is the following. Each experiment is done using a 10-fold cross-validation on the available data. This means that the data is split in 10 partitions, and each of these is used once as test set, with the other nine as corresponding train set. For each dataset, we provide information about the accuracy of two different machine learning systems under four conditions:

1. Using their respective default settings.
2. After optimizing the feature subset selection (backward selection) using default parameter settings, for each algorithm separately. (Optimization step 1).
3. After optimizing the algorithm parameters for each algorithm individually. Each “reasonable” parameter setting is tested with a 10-fold cross-validation experiment. (Optimization step 2).
4. After performing feature selection interleaved with optimization of the parameters for each algorithm in turn. (Optimization step 3).

We expect from our first hypothesis that each optimization step can increase the accuracy of the best result (as measured by the average result over the 10 experiments) considerably. In general, we expect the differences we record for the same algorithm over the four conditions to be much larger than the difference between the two learning algorithms when using default settings. As we are primarily interested in showing the variability of results due to the different optimizations, all results reported will be cross-validation results, and not results on an *additional* held-out dataset (this would imply an additional cross-validation loop within the cross-validation loop, which is computationally infeasible). For the WSD data we will report results on test datasets used by SENSEVAL which give an indication of the usefulness of the approach for improving accuracy.

3.1 Machine Learning Methods and Data

We chose two machine learning techniques for our experiments: the memory-based learning package TIMBL [6][4], and the rule induction package RIPPER [3]. These two approaches provide extremes of the *eagerness* dimension in ML (the degree in which a learning algorithm abstracts from the training data in forming a hypothesis). TIMBL is an example of lazy learning, RIPPER of eager learning.

For TIMBL the following algorithm parameters were optimized: similarity metrics, feature weighting metrics, class voting weighting, and number of nearest neighbors (varied between 1 and 45). For RIPPER the following parameter settings were varied: class ordering, negative tests in the rule conditions, hypothesis simplification magnitude, and example coverage. See [6,3] for explanation of these parameters.

The "line" data set has become a benchmark dataset for work on word sense disambiguation. It was first produced and described by Leecock, Towell, and Voorhees [14]. It consists of instances of the word *line*, taken from the 1987-89 Wall Street Journal and a 25 million word corpus from the American Printing House for the Blind. 4,149 examples of occurrences of *line* were each tagged with one of six WordNet senses: text, formation, division, phone, cord, product. Because the "product" sense is 5 times more common than any of the other senses, a sampled dataset was also used in which all senses are represented equally. For each sense, 349 instances were selected at random, producing a total of 2094 examples with each sense having an equal share. This kind of sampling has been done before, and has also been reported in the literature [16]. However, we made our own sample as the other samples were not readily available. This means that our sampled data results cannot be compared directly to those of other systems.

Here is an example of an instance representing one occurrence of the word *line* in the corpus:

```
line-n.w9_15:17036:,pen,NN,writing,VBG,those,DT,lines,line,NNS,was,
VBD,that,IN,of,IN,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,text.
```

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

Table 1. Results of TIMBL and RIPPER on different WSD data sets when using (i) default settings, (ii) backward selection, (iii) parameter optimization, and (iv) interleaved backward selection and parameter optimization.

"line" (complete)		TIMBL	RIPPER
WORDS	(default)	60.2	63.9
	(feat. sel.)	62.7	63.9
	(param. opt.)	63.4	70.2
	(interleaved opt.)	64.5	91.3
WORDS+POS	(default)	57.8	63.8
	(feat. sel.)	62.7	64.7
	(param. opt.)	64.3	71.6
	(interleaved opt.)	64.9	76.4

"line" (sampled)		TIMBL	RIPPER
WORDS	(default)	59.1	40.4
	(feat. sel.)	60.3	40.9
	(param. opt.)	66.4	61.2
	(interleaved opt.)	66.7	63.3
WORDS+POS	(default)	56.9	41.4
	(feat. sel.)	61.5	41.6
	(param. opt.)	67.3	60.5
	(interleaved opt.)	68.1	61.1

The first entry (*line-n.w9_15:17036:*) is an ID tag for the instance, and is ignored in the learning process. Next are the three context words occurring before the focus word, together with their parts of speech. Then follows the form of the word *line*, in this case the plural, together with its base form (*line*), and its part of speech (*NNS*). Then we can see the right context of the focus words, also with parts of speech. The next 200 features are binary features, each indicating the presence or absence of the 200 most salient context words of *line*.

3.2 Results on the WSD Datasets

In Table 1, if we focus on the variation for a single algorithm over the four conditions, we can see that parameter optimization, feature selection, and combined feature selection with parameter optimization lead to major accuracy improvements compared to the results obtained with default parameter settings. These 'vertical' accuracy differences are much larger than the 'horizontal' algorithm-comparing accuracy differences.

The fact that we could observe large standard deviations in the optimization experiments, also confirms the necessity of parameter optimization (only the best result is represented in Table 1 for each optimization step).

With respect to the selected parameter settings and feature combinations, we found that parameter settings which are optimal when using all features are not necessarily optimal when performing feature selection. Furthermore, we could observe that the feature selection considered to be optimal for TIMBL was often different from the one optimal for RIPPER.

Table 2. Results of TIMBL and RIPPER on different UCI data sets when using (i) default settings, (ii) backward selection, (iii) parameter optimization, and (iv) interleaved backward selection and parameter optimization.

Dataset		TIMBL	RIPPER
database for fitting contact lenses	(default)	75.0	79.2
	(feat. sel.)	**87.5**	**87.5**
	(param. opt.)	**87.5**	**87.5**
	(interleaved opt.)	**87.5**	**87.5**
contraceptive method choice	(default)	48.5	46.8
	(feat. sel.)	52.2	48.2
	(param. opt.)	54.2	49.8
	(interleaved opt.)	**54.8**	49.8
breast-cancer-wisconsin	(default)	95.7	93.7
	(feat. sel.)	96.3	95.3
	(param. opt.)	97.4	95.7
	(interleaved opt.)	**97.6**	95.7
car evaluation database	(default)	94.0	87.0
	(feat. sel.)	94.0	87.0
	(param. opt.)	96.9	**98.4**
	(interleaved opt.)	96.9	**98.4**
postoperative patient data	(default)	55.6	**71.1**
	(feat. sel.)	**71.1**	**71.1**
	(param. opt.)	**71.1**	**71.1**
	(interleaved opt.)	**71.1**	**71.1**

We conclude that we have found evidence for our hypothesis (i) that the accuracy differences between different machine learning algorithms using standard comparative methodology will in general be lower than the differences in accuracy resulting from interactions between algorithm parameter settings and information source selection, at least for this task (see [4] for similar results on other language datasets).

3.3 Results on the UCI Benchmarks

We investigated whether the effect is limited to natural language processing datasets by applying the same optimalization to 5 UCI benchmark datasets[5]: “database for fitting contact lenses” (24 instances), “contraceptive method choice” (1473 instances) , “breast-cancer-wisconsin” (699 instances), “car evaluation database” (1728 instances) and “postoperative patient data” (90 instances). Compared to our language processing datasets, these datasets are small. From the results in Table 2, we nevertheless see the same effects: the default settings for the algorithms are not optimal; the difference in accuracy for a single algorithm in the four conditions generally overwhelms accuracy differences found between the algorithms, and in cases like the “car evaluation database”, we see

[5] http://www.ics.uci.edu/ mlearn/MLRepository.html.

that the initial result (TIMBL outperforms RIPPER) is reversed after optimization. Similar effects explain why in the ML of natural language literature, so many results and interpretations about superiority of one algorithm over the other are contradictory.

4 Genetic Algorithms for Optimization

Our results of the previous Section show that a proper comparative experiment requires extensive optimization of a combinatorially explosive nature, and that the obtainable accuracy increase by going to this trouble are considerable. Optimization and model selection problems of the type described in this paper are of course not unique to machine learning of language. Solutions like *genetic algorithms* (GAs) have been used for a long time as domain-independent techniques suitable for exploring optimization in large search spaces such as those described in this paper. We applied this optimization technique to our datasets.

The evolutionary algorithm used to optimize the feature selection and parameter optimization employs an algorithmic scheme similar to that of *evolution strategies*: a population of μ individuals forms the genetic material from which λ new individuals are created using crossover and mutation. The μ best individuals of this bigger temporary population are selected to become the next generation of the algorithm.

An individual contains particular values for all algorithm parameters and for the selection of the features. E.g., for TIMBL, the large majority of these parameters control the use of a feature (ignore, weighted overlap, modified value difference), and are encoded in the chromosome as ternary alleles. At the end of the chromosome the 5-valued weighting parameter `w` and the 4-valued neighbor weighting parameter `d` are encoded, together with the `k` parameter which controls the number of neighbors. The latter is encoded as a real value which represents the logarithm of the number of neighbors. The quality or fitness of an individual is the classification result returned by TIMBL with these particular parameter settings. A similar approach is followed for encoding the RIPPER parameters into an individual.

The initial population is filled with individuals consisting of uniformly sampled values. The mutation operator replaces, independently for each position and with a small probability, the current value with an arbitrary other value. The mutation rates of the features are set independently of that of `w` and `d`. In the case of the `k` parameter, Gaussian noise is added to the current value. The crossover operators used are the traditional 1-point, 2-point and uniform crossovers. They operate on the whole chromosome. The selection strength can be controlled by tuning the proportion μ/λ; an alternative strategy chooses the μ best individuals from the combination of μ parents and λ children. The GA parameters were set using limited explorative experimentation. We are aware that the algorithm parameter optimization problem we try to solve with GAs also applies to the GAs themselves.

Table 3. Validation results for TIMBL on five word experts, for datasets with and without keyword information. For the smaller datasets, interleaved backward keyword selection and parameter optimization results are included and are shown to be worse than those of the GA. For the larger dataset, only the GA could be used to perform interleaved optimization because the other method had become too computationally expensive. SENSEVAL test set results are between brackets for default vs. GA with keywords.

WE	Words+POS		
	Def.	Opt.	GA
bar	48.1	55.3	*66.3*
channel	60.9	70.5	*73.9*
develop	19.3	**29.6**	**29.6**
natural	42.8	52.7	*58.9*
post	60.2	66.5	*75.6*
WE	Words + POS + Keywords		
	Def.		GA
bar	44.8 (47.0)		**66.9** (59.6)
channel	63.3 (50.7)		**75.4** (53.4)
develop	17.0 (37.7)		**29.6** (29.0)
natural	40.3 (31.1)		**61.3** (43.7)
post	57.4 (51.9)		**77.8** (58.2)

4.1 Results

The WSD data sets discussed in Table 3 were selected from the SENSEVAL-2 data, which provided training and test material for different ambiguous words. Each word was given a separate training and test set. We chose five of these words randomly, taking into account the following restrictions: at least 150 training items should be available, and the word should have at least 5 senses, each sense being represented by at least 10 training items. This process came up with the words *bar*, *channel*, *develop*, *natural*, *post*. Instances were made for each occurrence of each word in the same way as for the *line* data.

We see that the GA succeeds in finding solutions that are significantly better than the default solutions and the best solutions obtained by a heuristic combined feature selection and algorithm parameter optimization approach. The main advantage of the GA is that it allows us to explore much larger search spaces, for this problem e.g., also the use of context keywords, which would be computationally impossible with the heuristic methods in Section 3.

For the words with POS and keywords results we added between brackets for the default and GA results the results on the SENSEVAL test sets, showing that the optimization can indeed be used not only for showing the variability of the results, but also for obtaining higher predictive accuracy (although this should ideally be shown using two embedded cross-validation loops which turned out to be computationally infeasible for our data).

Table 4. Results of TIMBL with default settings and after interleaved feature selection and parameter optimization with a GA on the different WSD data sets for different information sources.

	Default			GA		
			words+POS			words+POS
dataset	words	words+POS	+keywords	words	words+POS	+keywords
“bar ”	**50.0**	48.1	44.8	56.4	66.3	**66.9**
“channel”	62.3	60.9	**63.3**	72.0	73.9	**75.4**
“develop”	16.3	**19.3**	17.0	**34.8**	29.6	29.6
“natural”	41.6	**42.8**	40.3	55.6	58.9	**61.3**
“post”	**62.5**	60.2	57.4	71.0	75.6	**77.8**
“line” (sampled)	**59.1**	56.9	57.0	**66.9**	**66.9**	**66.9**

4.2 Results on the Comparison of Information Sources

In Table 4, we find evidence for our second hypothesis (the effect of adding an information source can switch between positive and negative depending on the optimization). E.g. where results with the default settings would lead to a conclusion that keyword features don't help for most WSD problems except “channel”, the GA optimization shows that combinations of parameter settings and feature selection can be found for all WSD problems except “develop” which show exactly the opposite.

5 Related Research and Conclusion

Most comparative ML experiments, at least in computational linguistics, explore only one or a few points in the space of possible experiments for each algorithm to be compared. We have shown that regardless of the methodological accuracy with which the comparison is made, there is a high risk that other areas in the experimental space may lead to radically different results and conclusions. In general, the more effort is put in optimization (in this paper by exploring the interaction between feature selection and algorithm parameter optimization), the better the results will be, and the more reliable the comparison will be. Given the combinatorially explosive character of this type of optimization, we have chosen for GAs as a computationally feasible way to achieve this; no other heuristic optimization techniques allow the complex interactions we want to optimize. As a test case we used WSD datasets. In previous work [4] we showed that the same effects also occur in other tasks, like part of speech tagging and morphological synthesis.

The current paper builds on results obtained earlier on WSD [19,20] in which we found that independent optimization of algorithm parameters for each word to be disambiguated led to higher accuracy, which at one point we thought to be a limitation of the method used (memory-based learning). In this paper, we show that the problem is much more general than for a single algorithm (e.g. RIPPER behaves similarly). We also showed in this paper that feature selection and algorithm parameter optimization interact highly, and should be jointly optimized. We also build on earlier, less successful attempts to use GAs for optimization in memory-based learning [12,13]. GAs have been used for parameter optimization in ML a great deal, including for memory-based learning. A different discussion point concerns the lessons we have to draw from the relativity of comparative machine learning results. In an influential recent paper, Banko and Brill [2] conclude that “We have no reason to believe that any comparative conclusions drawn on one million words will hold when we finally scale up to larger training corpora”. They base this point of view on experiments comparing several machine learning algorithms on one typical NLP task (confusable word disambiguation in context) with data selection sizes varying from 1 million to 1 billion. We have shown in this paper that data sample size is only one aspect influencing comparative results, and that accuracy differences due to algorithm parameter optimization, feature selection, and especially the interaction between both easily overwhelm the accuracy differences reported between algorithms (or information sources) in comparative experiments. Like the Banko and Brill study, this suggests that published results of comparative machine learning experiments (and their interpretation) may often be unreliable.

The good news is that optimization of as many factors as possible (sample selection and size, feature selection and representation, algorithm parameters), when possible, will offer important accuracy increases and (more) reliable comparative results. We believe that, in the long term, a GA approach offers a computationally feasible approach to this huge optimization problem.

References

1. Ethem Alpaydin. Combined 5×2 cv F test for comparing supervised classification learning algorithms. *Neural Computation*, 11(8):1885–1892, 1999.
2. Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics, 2001.
3. William W. Cohen. Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.
4. Walter Daelemans and Véronique Hoste. Evaluation of machine learning methods for natural language processing tasks. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 755–760, 2002.
5. Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–41, 1999.
6. Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, ILK Technical Report 01-04, 2001.
7. Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
8. Phil Edmonds and Adam Kilgarriff, editors. *Journal of Natural Language Engineering special issue based on Senseval-2*, volume 9. Cambridge University Press, 2003.

9. Gerard Escudero, Lluis Marquez, and German Rigau. Boosting applied to word sense disambiguation. In *European Conference on Machine Learning*, pages 129–141, 2000.
10. Adam Kilgarriff and Martha Palmer, editors. *Computers and the Humanities special issue based on Senseval-1*, volume 34. 1999.
11. Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–323, 1997.
12. Anne Kool, Walter Daelemans, and Jakub Zavrel. Genetic algorithms for feature relevance assignment in memory-based language processing. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 103–106. Association for Computational Linguistics, Somerset, New Jersey, 2000.
13. Anne Kool, Jakub Zavrel, and Walter Daelemans. Simultaneous feature selection and parameter optimization for memory-based natural language processing. In Ad Feelders, editor, *Proceedings of the 10th BENELEARN meeting*, pages 93–100. Tilburg, The Netherlands, 2000.
14. Claudia Leacock, Geoffrey Towell, and Ellen Voorhees. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 260–265, March 1993.
15. Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 41–48, 2002.
16. Raymond J. Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–91. Association for Computational Linguistics, Somerset, New Jersey, 1996.
17. Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 40–47, San Francisco, 1996. Morgan Kaufmann Publishers.
18. Steven L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–327, 1997.
19. Jorn Veenstra, Antal Van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. Memory-based word sense disambiguation. *Computing and the Humanities*, 2000.
20. Walter Daelemans Véronique Hoste, Iris Hendrickx and Antal van den Bosch. Parameter optimization for machine-learning of word sense disambiguation. *Natural Language Engineering*, pages 311–325, 2002.
21. Sholom Weiss and Nitin Indurkhya. *Predictive Data Mining: A Practical Guide.* Morgan Kaufmann, San Francisco, 1998.