Exemplar-Based Word Sense Disambiguation: Some Recent Improvements

Hwee Tou Ng
DSO National Laboratories
20 Science Park Drive
Singapore 118230
nhweetou@dso.org.sg

Abstract

In this paper, we report recent improvements to the exemplar-based learning approach for word sense disambiguation that have achieved higher disambiguation accuracy. By using a larger value of k, the number of nearest neighbors to use for determining the class of a test example, and through 10-fold cross validation to automatically determine the best k, we have obtained improved disambiguation accuracy on a large sense-tagged corpus first used in (Ng and Lee, 1996). The accuracy achieved by our improved exemplar-based classifier is comparable to the accuracy on the same data set obtained by the Naive-Bayes algorithm, which was reported in (Mooney, 1996) to have the highest disambiguation accuracy among seven state-of-the-art machine learning algorithms.

1 Introduction

Much recent research on word sense disambiguation (WSD) has adopted a corpus-based, learning approach. Many different learning approaches have been used, including neural networks (Leacock et al., 1993), probabilistic algorithms (Bruce and Wiebe, 1994; Gale et al., 1992a; Gale et al., 1995; Leacock et al., 1993; Yarowsky, 1992), decision lists (Yarowsky, 1994), exemplar-based learning algorithms (Cardie, 1993; Ng and Lee, 1996), etc.

In particular, Mooney (1996) evaluated seven state-of-the-art machine learning algorithms on a common data set for disambiguating six senses of the word "line". The seven algorithms that he evaluated are: a Naive-Bayes classifier (Duda and Hart, 1973), a perceptron (Rosenblatt, 1958), a decision-tree learner (Quinlan, 1993), a k nearest-neighbor classifier (exemplar-based learner) (Cover and Hart, 1967), logic-based DNF and CNF learners (Mooney, 1995), and a decision-list learner (Rivest, 1987). His results indicate that the simple Naive-Bayes algorithm gives the highest accuracy on the "line" corpus tested. Past research in machine learning has also reported that the Naive-Bayes algorithm achieved good performance on other machine learning tasks (Clark and Niblett, 1989; Kohavi, 1996). This is in spite of the conditional independence assumption made by the Naive-Bayes algorithm, which may be unjustified in the domains tested. Gale, Church and Yarowsky (Gale et al., 1992a; Gale et al., 1995; Yarowsky, 1992) have also successfully used the Naive-Bayes algorithm (and several extensions and variations) for word sense disambiguation.

On the other hand, our past work on WSD (Ng and Lee, 1996) used an exemplar-based (or nearest neighbor) learning approach. Our WSD program, LEXAS, extracts a set of features, including part of speech and morphological form, surrounding words, local collocations, and verb-object syntactic relation from a sentence containing the word to be disambiguated. These features from a sentence form an example. LEXAS then uses the exemplar-based learning algorithm PEBLS (Cost and Salzberg, 1993) to find the sense (class) of the word to be disambiguated.

In this paper, we report recent improvements to the exemplar-based learning approach for WSD that have achieved higher disambiguation accuracy. The exemplar-based learning algorithm PEBLS contains a number of parameters that must be set before running the algorithm. These parameters include the number of nearest neighbors to use for determining the class of a test example (i.e., k in a k nearest-neighbor classifier), exemplar weights, feature weights, etc. We found that the number k of nearest neighbors used has a considerable impact on the accuracy of the induced exemplar-based classifier. By using 10-fold cross validation (Kohavi and

John, 1995) on the training set to automatically determine the best k to use, we have obtained improved disambiguation accuracy on a large sense-tagged corpus first used in (Ng and Lee, 1996). The accuracy achieved by our improved exemplar-based classifier is comparable to the accuracy on the same data set obtained by the Naive-Bayes algorithm, which was reported in (Mooney, 1996) to have the highest disambiguation accuracy among seven state-of-the-art machine learning algorithms.

The rest of this paper is organized as follows. Section 2 gives a brief description of the exemplar-based algorithm PEBLS and the Naive-Bayes algorithm. Section 3 describes the 10-fold cross validation training procedure to determine the best k number of nearest neighbors to use. Section 4 presents the disambiguation accuracy of PEBLS and Naive-Bayes on the large corpus of (Ng and Lee, 1996). Section 5 discusses the implications of the results. Section 6 gives the conclusion.

2 Learning Algorithms

2.1 PEBLS

The heart of exemplar-based learning is a measure of the similarity, or distance, between two examples. If the distance between two examples is small, then the two examples are similar. In PEBLS (Cost and Salzberg, 1993), the distance between two symbolic values v_1 and v_2 of a feature f is defined as:

$$d(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)|$$

where n is the total number of classes. $P(C_i|v_1)$ is estimated by $\frac{N_{1,i}}{N_1}$, where $N_{1,i}$ is the number of training examples with value v_1 for feature f that is classified as class i in the training corpus, and N_1 is the number of training examples with value v_1 for feature f in any class. $P(C_i|v_2)$ is estimated similarly. This distance metric of PEBLS is adapted from the value difference metric of the earlier work of (Stanfill and Waltz, 1986). The distance between two examples is the sum of the distances between the values of all the features of the two examples.

Let k be the number of nearest neighbors to use for determining the class of a test example, $k \geq 1$. During testing, a test example is compared against *all* the training examples. PEBLS then determines the k training examples with the shortest distance to the test example. Among these k closest matching training examples, the class which the majority of these k examples belong to will be assigned as the class of the test example, with tie among multiple majority classes broken randomly.

Note that the nearest neighbor algorithm tested in (Mooney, 1996) uses Hamming distance as the distance metric between two symbolic feature values. This is different from the above distance metric used in PEBLS.

2.2 Naive-Bayes

Our presentation of the Naive-Bayes algorithm (Duda and Hart, 1973) follows that of (Clark and Niblett, 1989). This algorithm is based on Bayes' theorem:

$$P(C_i| \wedge v_j) = \frac{P(\wedge v_j|C_i)P(C_i)}{P(\wedge v_j)} \qquad i = 1 \ldots n$$

where $P(C_i| \wedge v_j)$ is the probability that a test example is of class C_i given feature values v_j. ($\wedge v_j$ denotes the conjunction of all feature values in the test example.) The goal of a Naive-Bayes classifier is to determine the class C_i with the highest conditional probability $P(C_i| \wedge v_j)$. Since the denominator $P(\wedge v_j)$ of the above expression is constant for all classes C_i, the problem reduces to finding the class C_i with the maximum value for the numerator.

The Naive-Bayes classifier assumes independence of example features, so that

$$P(\wedge v_j|C_i) = \prod_j P(v_j|C_i)$$

During training, Naive-Bayes constructs the matrix $P(v_j|C_i)$, and $P(C_i)$ is estimated from the distribution of training examples among the classes. To avoid one zero count of $P(v_j|C_i)$ nullifying the effect of the other non-zero conditional probabilities in the multiplication, we replace zero counts of $P(v_j|C_i)$ by $P(C_i)/N$, where N is the total number of training examples. Other more complex smoothing procedures (such as those used in (Gale et al., 1992a)) are also possible, although we have not experimented with these other variations.

For the experimental results reported in this paper, we used the implementation of Naive-Bayes algorithm in the PEBLS program (Rachlin and Salzberg, 1993), which has an option for training and testing using the Naive-Bayes algorithm. We only changed the handling of zero probability counts to the method just described.

3 Improvements to Exemplar-Based WSD

PEBLS contains a number of parameters that must be set before running the algorithm. These parameters include k (the number of nearest neighbors to use for determining the class of a test example), exemplar weights, feature weights, etc. Each of these parameters has a default value in PEBLS, eg., $k = 1$, no exemplar weighting, no feature weighting, etc. We have used the default values for all parameter settings in our previous work on exemplar-based WSD reported in (Ng and Lee, 1996). However, our preliminary investigation indicates that, among the various learning parameters of PEBLS, the number k of nearest neighbors used has a considerable impact on the accuracy of the induced exemplar-based classifier.

Cross validation is a well-known technique that can be used for estimating the expected error rate of a classifier which has been trained on a particular data set. For instance, the C4.5 program (Quinlan, 1993) contains an option for running cross validation to estimate the expected error rate of an induced rule set. Cross validation has been proposed as a general technique to automatically determine the parameter settings of a given learning algorithm using a particular data set as training data (Kohavi and John, 1995).

In m-fold cross validation, a training data set is partitioned into m (approximately) equal-sized blocks, and the learning algorithm is run m times. In each run, one of the m blocks of training data is set aside as test data (the holdout set) and the remaining $m-1$ blocks are used as training data. The average error rate of the m runs is a good estimate of the error rate of the induced classifier.

For a particular parameter setting, we can run m-fold cross validation to determine the expected error rate of that particular parameter setting. We can then choose an optimal parameter setting that minimizes the expected error rate. Kohavi and John (1995) reported the effectiveness of such a technique in obtaining optimal sets of parameter settings over a large number of machine learning problems.

In our present study, we used 10-fold cross validation to automatically determine the best k (number of nearest neighbors) to use from the training data. To determine the best k for disambiguating a word on a particular training set, we run 10-fold cross validation using PEBLS 21 times, each time with $k = 1, 5, 10, 15, \ldots, 85, 90, 95, 100$. We compute the error rate for each k, and choose the value of k with the minimum error rate. Note that the automatic determination of the best k through 10-fold cross validation makes use of *only* the training set, without looking at the test set at all.

4 Experimental Results

Mooney (1996) has reported that the Naive-Bayes algorithm gives the best performance on disambiguating six senses of the word "line", among seven state-of-the-art learning algorithms tested. However, his comparative study is done on only one word using a data set of 2,094 examples. In our present study, we evaluated PEBLS and Naive-Bayes on a much larger corpus containing sense-tagged occurrences of 121 nouns and 70 verbs. This corpus was first reported in (Ng and Lee, 1996), and it contains about 192,800 sense-tagged word occurrences of 191 most frequently occurring and ambiguous words of English.[1] These 191 words have been tagged with senses from WORDNET (Miller, 1990), an on-line, electronic dictionary available publicly. For this set of 191 words, the average number of senses per noun is 7.8, while the average number of senses per verb is 12.0. The sentences in this corpus were drawn from the combined corpus of the 1 million word Brown corpus and the 2.5 million word Wall Street Journal (WSJ) corpus.

We tested both algorithms on two test sets from this corpus. The first test set, named BC50, consists of 7,119 occurrences of the 191 words appearing in 50 text files of the Brown corpus. The second test set, named WSJ6, consists of 14,139 occurrences of the 191 words appearing in 6 text files of the WSJ corpus. Both test sets are identical to the ones reported in (Ng and Lee, 1996).

Since the primary aim of our present study is the comparative evaluation of learning algorithms, not feature representation, we have chosen, for simplicity, to use local collocations as the only features in the example representation. Local collocations have been found to be the single most informative set of features for WSD (Ng and Lee, 1996). That local collocation knowledge provides important clues to WSD has also been pointed out previously by Yarowsky (1993).

Let w be the word to be disambiguated, and let $l_2\ l_1\ w\ r_1\ r_2$ be the sentence fragment containing w. In the present study, we used seven features in the representation of an example, which are the local collocations of the surrounding 4 words. These seven features are: l_2_l_1, l_1_r_1, r_1_r_2, l_1, r_1, l_2, and r_2. The first three features are concatenation of two words.[2]

The experimental results obtained are tabulated in Table 1. The first three rows of accuracy fig-

[1] This corpus is available from the Linguistic Data Consortium (LDC). Contact the LDC at ldc@unagi.cis.upenn.edu for details.

[2] The first five of these seven features were also used in (Ng and Lee, 1996).

Algorithm	BC50	WSJ6
Sense 1	40.5%	44.8%
Most Frequent	47.1%	63.7%
Ng & Lee (1996)	54.0%	68.6%
PEBLS ($k = 1$)	55.0%	70.2%
PEBLS ($k = 20$)	58.5%	74.5%
PEBLS (10-fold c.v.)	58.7%	75.2%
Naive-Bayes	58.2%	74.5%

Table 1: Experimental Results

ures are those of (Ng and Lee, 1996). The default strategy of picking the most frequent sense has been advocated as the baseline performance for evaluating WSD programs (Gale et al., 1992b; Miller et al., 1994). There are two instantiations of this strategy in our current evaluation. Since WORDNET orders its senses such that sense 1 is the most frequent sense, one possibility is to always pick sense 1 as the best sense assignment. This assignment method does not even need to look at the training examples. We call this method "Sense 1" in Table 1. Another assignment method is to determine the most frequently occurring sense in the training examples, and to assign this sense to all test examples. We call this method "Most Frequent" in Table 1.

The accuracy figures of LEXAS as reported in (Ng and Lee, 1996) are reproduced in the third row of Table 1. These figures were obtained using all features including part of speech and morphological form, surrounding words, local collocations, and verb-object syntactic relation. However, the feature value pruning method of (Ng and Lee, 1996) only selects surrounding words and local collocations as feature values if they are indicative of some sense class as measured by conditional probability (See (Ng and Lee, 1996) for details).

The next three rows show the accuracy figures of PEBLS using the parameter setting of $k = 1$, $k = 20$, and 10-fold cross validation for finding the best k, respectively. The last row shows the accuracy figures of the Naive-Bayes algorithm. Accuracy figures of the last four rows are all based on only seven collocation features as described earlier in this section. However, all possible feature values (collocated words) are used, without employing the feature value pruning method used in (Ng and Lee, 1996).

Note that the accuracy figures of PEBLS with $k = 1$ are 1.0% and 1.6% higher than the accuracy figures of (Ng and Lee, 1996) in the third row, also with $k = 1$. The feature value pruning method of (Ng and Lee, 1996) is intended to keep only feature values deemed important for classification. It seems that the pruning method has filtered out some useful collocation values that improve classification accuracy, such that this unfavorable effect outweighs the additional set of features (part of speech and morphological form, surrounding words, and verb-object syntactic relation) used.

Our results indicate that although Naive-Bayes performs better than PEBLS with $k = 1$, PEBLS with $k = 20$ achieves comparable performance. Furthermore, PEBLS with 10-fold cross validation to select the best k yields results slightly better than the Naive-Bayes algorithm.

5 Discussion

To understand why larger values of k are needed, we examined the performance of PEBLS when tested on the WSJ6 test set. During 10-fold cross validation runs on the training set, for each of the 191 words, we compared two error rates: the minimum expected error rate of PEBLS using the best k, and the expected error rate of the most frequent classifier. We found that for 13 words out of the 191 words, the minimum expected error rate of PEBLS using the best k is still higher than the expected error rate of the most frequent classifier. That is, for these 13 words, PEBLS will produce, on average, lower accuracy than the most frequent classifier.

Importantly, for 11 of these 13 words, the best k found by PEBLS are at least 85 and above. This indicates that for a training data set when PEBLS has trouble even outperforming the most frequent classifier, it will tend to use a large value for k. This is explainable since for a large value of k, PEBLS will tend towards the performance of the most frequent classifier, as it will find the k closest matching training examples and select the majority class among this large number of k examples. Note that in the extreme case when k equals the size of the training set, PEBLS will behave exactly like the most frequent classifier.

Our results indicate that although PEBLS with $k = 1$ gives lower accuracy compared with Naive-Bayes, PEBLS with $k = 20$ performs as well as Naive-Bayes. Furthermore, PEBLS with automatically selected k using 10-fold cross validation gives slightly higher performance compared with Naive-Bayes. We believe that this result is significant, in light of the fact that Naive-Bayes has been found to give the best performance for WSD among seven state-of-the-art machine learning algorithms (Mooney, 1996). It demonstrates that an exemplar-based learning approach is suitable for the WSD task, achieving high disambiguation accuracy.

One potential drawback of an exemplar-based learning approach is the testing time required, since each test example must be compared with every training example, and hence the required testing time grows linearly with the size of the training set. However, more sophisticated indexing methods such as that reported in (Friedman et al., 1977) can reduce this to logarithmic expected time, which will significantly reduce testing time.

In the present study, we have focused on the comparison of learning algorithms, but not on feature representation of examples. Our past work (Ng and Lee, 1996) suggests that multiple sources of knowledge are indeed useful for WSD. Future work will explore the addition of these other features to further improve disambiguation accuracy.

Besides the parameter k, PEBLS also contains other learning parameters such as exemplar weights and feature weights. Exemplar weighting has been found to improve classification performance (Cost and Salzberg, 1993). Also, given the relative importance of the various knowledge sources as reported in (Ng and Lee, 1996), it may be possible to improve disambiguation performance by introducing feature weighting. Future work can explore the effect of exemplar weighting and feature weighting on disambiguation accuracy.

6 Conclusion

In summary, we have presented improvements to the exemplar-based learning approach for WSD. By using a larger value of k, the number of nearest neighbors to use for determining the class of a test example, and through 10-fold cross validation to automatically determine the best k, we have obtained improved disambiguation accuracy on a large sense-tagged corpus. The accuracy achieved by our improved exemplar-based classifier is comparable to the accuracy on the same data set obtained by the Naive-Bayes algorithm, which was recently reported to have the highest disambiguation accuracy among seven state-of-the-art machine learning algorithms.

7 Acknowledgements

Thanks to Ray Mooney for helpful discussions, and the anonymous reviewers for their comments.

References

Rebecca Bruce and Janyce Wiebe. 1994. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

Claire Cardie. 1993. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 798–803, Washington, DC.

Peter Clark and Tim Niblett. 1989. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283.

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

T. M. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Richard Duda and Peter Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.

J. Friedman, J. Bentley, and R. Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992a. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26:415–439.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992b. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware.

William Gale, Kenneth Ward Church, and David Yarowsky. 1995. Discrimination Decisions for 100,000 Dimensional Spaces. *Annals of Operations Research*, 55:323–344.

Ron Kohavi and George H. John. 1995. Automatic parameter selection by minimizing estimated error. In *Machine Learning: Proceedings of the Twelfth International Conference*.

Ron Kohavi. 1996. Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*.

Claudia Leacock, Geoffrey Towell, and Ellen Voorhees. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Human Language Technology Workshop*.

George A. Miller, Ed. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop.*

Raymond J. Mooney. 1995. Encouraging experimental results on learning CNF. *Machine Learning*, 19(1):79–92.

Raymond J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 40–47.

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

John Rachlin and Steven Salzberg. 1993. PEBLS 3.0 User's Guide.

R. L. Rivest. 1987. Learning decision lists. *Machine Learning*, 2(3):229–246.

F. Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

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

David Yarowsky. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 454–460, Nantes, France.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop.*

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.