Proceedings of the Conference on Empirical Methods in Natural
Language Processing (EMNLP), Philadelphia, July 2002, pp. 41-48.
Association for Computational Linguistics.

An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation

Yoong Keok Lee and **Hwee Tou Ng**
Department of Computer Science
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{leeyoong, nght}@comp.nus.edu.sg

Abstract

In this paper, we evaluate a variety of knowledge sources and supervised learning algorithms for word sense disambiguation on SENSEVAL-2 and SENSEVAL-1 data. Our knowledge sources include the part-of-speech of neighboring words, single words in the surrounding context, local collocations, and syntactic relations. The learning algorithms evaluated include Support Vector Machines (SVM), Naive Bayes, AdaBoost, and decision tree algorithms. We present empirical results showing the relative contribution of the component knowledge sources and the different learning algorithms. In particular, using all of these knowledge sources and SVM (i.e., a single learning algorithm) achieves accuracy higher than the best official scores on both SENSEVAL-2 and SENSEVAL-1 test data.

1 Introduction

Natural language is inherently ambiguous. A word can have multiple meanings (or senses). Given an occurrence of a word w in a natural language text, the task of word sense disambiguation (WSD) is to determine the correct sense of w in that context. WSD is a fundamental problem of natural language processing. For example, effective WSD is crucial for high quality machine translation.

One could envisage building a WSD system using handcrafted rules or knowledge obtained from linguists. Such an approach would be highly labor-intensive, with questionable scalability. Another approach involves the use of dictionary or thesaurus to perform WSD.

In this paper, we focus on a corpus-based, supervised learning approach. In this approach, to disambiguate a word w, we first collect training texts in which instances of w occur. Each occurrence of w is manually tagged with the correct sense. We then train a WSD classifier based on these sample texts, such that the trained classifier is able to assign the sense of w in a new context.

Two WSD evaluation exercises, SENSEVAL-1 (Kilgarriff and Palmer, 2000) and SENSEVAL-2 (Edmonds and Cotton, 2001), were conducted in 1998 and 2001, respectively. The lexical sample task in these two SENSEVALs focuses on evaluating WSD systems in disambiguating a subset of nouns, verbs, and adjectives, for which manually sense-tagged training data have been collected.

In this paper, we conduct a systematic evaluation of the various knowledge sources and supervised learning algorithms on the English lexical sample data sets of both SENSEVALs.

2 Related Work

There is a large body of prior research on WSD. Due to space constraints, we will only highlight prior research efforts that have investigated (1) contribution of various knowledge sources, or (2) relative performance of different learning algorithms.

Early research efforts on comparing different learning algorithms (Mooney, 1996; Pedersen and Bruce, 1997) tend to base their comparison on only one word or at most a dozen words. Ng (1997) compared two learning algorithms, k-nearest neighbor and Naive Bayes, on the DSO corpus (191 words). Escudero et al. (2000) evaluated k-nearest neighbor, Naive Bayes, Winnow-based, and LazyBoosting algorithms on the DSO corpus. The recent work of Pedersen (2001a) and Zavrel et al. (2000) evaluated a variety of learning algorithms on the SENSEVAL-1 data set. However, all of these research efforts concentrate only on evaluating different learning algorithms, without systematically considering their interaction with knowledge sources.

Ng and Lee (1996) reported the relative contribution of different knowledge sources, but on only one word "interest". Stevenson and Wilks (2001) investigated the interaction of knowledge sources, such as part-of-speech, dictionary definition, subject codes, etc. on WSD. However, they do not evaluate their method on a common benchmark data set, and there is no exploration on the interaction of knowledge sources with different learning algorithms.

Participating systems at SENSEVAL-1 and SENSEVAL-2 tend to report accuracy using a particular set of knowledge sources and some particular learning algorithm, without investigating the effect of varying knowledge sources and learning algorithms. In SENSEVAL-2, the various Duluth systems (Pedersen, 2001b) attempted to investigate whether features or learning algorithms are more important. However, relative contribution of knowledge sources was not reported and only two main types of algorithms (Naive Bayes and decision tree) were tested.

In contrast, in this paper, we systematically vary *both* knowledge sources *and* learning algorithms, and investigate the *interaction* between them. We also base our evaluation on both SENSEVAL-2 and SENSEVAL-1 official test data sets, and compare with the official scores of participating systems.

3 Knowledge Sources

To disambiguate a word occurrence w, we consider four knowledge sources listed below. Each training (or test) context of w generates one training (or test) feature vector.

3.1 Part-of-Speech (POS) of Neighboring Words

We use 7 features to encode this knowledge source: $P_{-3}, P_{-2}, P_{-1}, P_0, P_1, P_2, P_3$, where P_{-i} (P_i) is the POS of the ith token to the left (right) of w, and P_0 is the POS of w. A token can be a word or a punctuation symbol, and each of these neighboring tokens must be in the same sentence as w. We use a sentence segmentation program (Reynar and Ratnaparkhi, 1997) and a POS tagger (Ratnaparkhi, 1996) to segment the tokens surrounding w into sentences and assign POS tags to these tokens.

For example, to disambiguate the word *bars* in the POS-tagged sentence *"Reid/NNP saw/VBD me/PRP looking/VBG at/IN the/DT iron/NN bars/NNS ./."*, the POS feature vector is $< IN, DT, NN, NNS, ., \epsilon, \epsilon >$ where ϵ denotes the POS tag of a null token.

3.2 Single Words in the Surrounding Context

For this knowledge source, we consider all single words (unigrams) in the surrounding context of w, and these words can be in a different sentence from w. For each training or test example, the SENSEVAL data sets provide up to a few sentences as the surrounding context. In the results reported in this paper, we consider all words in the provided context.

Specifically, all tokens in the surrounding context of w are converted to lower case and replaced by their morphological root forms. Tokens present in a list of stop words or tokens that do not contain at least an alphabet character (such as numbers and punctuation symbols) are removed. All remaining tokens from all training contexts provided for w are gathered. Each remaining token t contributes one feature. In a training (or test) example, the feature corresponding to t is set to 1 iff the context of w in that training (or test) example contains t.

We attempted a simple feature selection method to investigate if a learning algorithm performs better with or without feature selection. The feature selection method employed has one parameter: M_2. A feature t is selected if t occurs in some sense of w M_2 or more times in the training data. This parameter is also used by (Ng and Lee, 1996). We have tried $M_2 = 3$ and $M_2 = 0$ (i.e., no feature selection) in the results reported in this paper.

For example, if w is the word *bars* and the set of selected unigrams is $\{$*chocolate, iron, beer*$\}$, the feature vector for the sentence *"Reid saw me looking at the iron bars ."* is <0, 1, 0>.

3.3 Local Collocations

A local collocation $C_{i,j}$ refers to the ordered sequence of tokens in the local, narrow context of w. Offsets i and j denote the starting and ending position (relative to w) of the sequence, where a negative (positive) offset refers to a token to its left (right). For example, let w be the word *bars* in the sentence *"Reid saw me looking at the iron bars ."* Then $C_{-2,-1}$ is *the_iron* and $C_{-1,2}$ is *iron_._ϵ*, where ϵ denotes a null token. Like POS, a collocation does not cross sentence boundary. To represent this knowledge source of local collocations, we extracted 11 features corresponding to the following collocations: $C_{-1,-1}$, $C_{1,1}$, $C_{-2,-2}$, $C_{2,2}$, $C_{-2,-1}$, $C_{-1,1}$, $C_{1,2}$, $C_{-3,-1}$, $C_{-2,1}$, $C_{-1,2}$, and $C_{1,3}$. This set of 11 features is the union of the collocation features used in Ng and Lee (1996) and Ng (1997).

To extract the feature values of the collocation feature $C_{i,j}$, we first collect all possible collocation strings (converted into lower case) corresponding to $C_{i,j}$ in all training contexts of w. Unlike the case for surrounding words, we do not remove stop words, numbers, or punctuation symbols. Each collocation string is a possible feature value. Feature value selection using M_2, analogous to that used to select surrounding words, can be optionally applied. If a training (or test) context of w has collocation c, and c is a selected feature value, then the $C_{i,j}$ feature of w has value c. Otherwise, it has the value $\emptyset$, denoting the null string.

Note that each collocation $C_{i,j}$ is represented by one feature that can have many possible feature values (the local collocation strings), whereas each distinct surrounding word is represented by one feature that takes binary values (indicating presence or absence of that word). For example, if w is the word *bars* and suppose the set of selected collocations for $C_{-2,-1}$ is $\{$*a_chocolate, the_wine, the_iron*$\}$, then the feature value for collocation $C_{-2,-1}$ in the sentence *"Reid saw me looking at the iron bars ."* is *the_iron*.

1(a) *attention* (noun)
1(b) He turned his *attention* to the workbench .
1(c) <turned, VBD, active, left>
2(a) *turned* (verb)
2(b) He *turned* his attention to the workbench .
2(c) <he, attention, PRP, NN, VBD, active>
3(a) *green* (adj)
3(b) The modern tram is a *green* machine .
3(c) <machine, NN>

Table 1: Examples of syntactic relations (assuming no feature selection)

3.4 Syntactic Relations

We first parse the sentence containing w with a statistical parser (Charniak, 2000). The constituent tree structure generated by Charniak's parser is then converted into a dependency tree in which every word points to a parent headword. For example, in the sentence *"Reid saw me looking at the iron bars ."*, the word *Reid* points to the parent headword *saw*. Similarly, the word *me* also points to the parent headword *saw*.

We use different types of syntactic relations, depending on the POS of w. If w is a noun, we use four features: its parent headword h, the POS of h, the voice of h (*active*, *passive*, or $\emptyset$ if h is not a verb), and the relative position of h from w (whether h is to the *left* or *right* of w). If w is a verb, we use six features: the nearest word l to the left of w such that w is the parent headword of l, the nearest word r to the right of w such that w is the parent headword of r, the POS of l, the POS of r, the POS of w, and the voice of w. If w is an adjective, we use two features: its parent headword h and the POS of h. We also investigated the effect of feature selection on syntactic-relation features that are words (i.e., POS, voice, and relative position are excluded).

Some examples are shown in Table 1. Each POS noun, verb, or adjective is illustrated by one example. For each example, (a) shows w and its POS; (b) shows the sentence where w occurs; and (c) shows the feature vector corresponding to syntactic relations.

4 Learning Algorithms

We evaluated four supervised learning algorithms: Support Vector Machines (SVM), AdaBoost with decision stumps (AdB), Naive Bayes (NB), and decision trees (DT). All the experimental results reported in this paper are obtained using the implementation of these algorithms in WEKA (Witten and Frank, 2000). All learning parameters use the default values in WEKA unless otherwise stated.

4.1 Support Vector Machines

The SVM (Vapnik, 1995) performs optimization to find a hyperplane with the largest margin that separates training examples into two classes. A test example is classified depending on the side of the hyperplane it lies in. Input features can be mapped into high dimensional space before performing the optimization and classification. A kernel function (linear by default) can be used to reduce the computational cost of training and testing in high dimensional space. If the training examples are nonseparable, a regularization parameter C ($= 1$ by default) can be used to control the trade-off between achieving a large margin and a low training error. In WEKA's implementation of SVM, each nominal feature with n possible values is converted into n binary (0 or 1) features. If a nominal feature takes the ith feature value, then the ith binary feature is set to 1 and all the other binary features are set to 0. We tried higher order polynomial kernels, but they gave poorer results. Our reported results in this paper used the linear kernel.

4.2 AdaBoost

AdaBoost (Freund and Schapire, 1996) is a method of training an ensemble of weak learners such that the performance of the whole ensemble is higher than its constituents. The basic idea of boosting is to give more weights to misclassified training examples, forcing the new classifier to concentrate on these hard-to-classify examples. A test example is classified by a weighted vote of all trained classifiers. We use the decision stump (decision tree with only the root node) as the weak learner in AdaBoost. WEKA implements AdaBoost.M1. We used 100 iterations in AdaBoost as it gives higher accuracy than the default number of iterations in WEKA (10).

4.3 Naive Bayes

The Naive Bayes classifier (Duda and Hart, 1973) assumes the features are independent given the class. During classification, it chooses the class with the highest posterior probability. The default setting uses Laplace ("add one") smoothing.

4.4 Decision Trees

The decision tree algorithm (Quinlan, 1993) partitions the training examples using the feature with the highest information gain. It repeats this process recursively for each partition until all examples in each partition belong to one class. A test example is classified by traversing the learned decision tree. WEKA implements Quinlan's C4.5 decision tree algorithm, with pruning by default.

5 Evaluation Data Sets

In the SENSEVAL-2 English lexical sample task, participating systems are required to disambiguate 73 words that have their POS predetermined. There are 8,611 training instances and 4,328 test instances tagged with WORDNET senses. Our evaluation is based on all the official training and test data of SENSEVAL-2.

For SENSEVAL-1, we used the 36 trainable words for our evaluation. There are 13,845 training instances[1] for these trainable words, and 7,446 test instances. For SENSEVAL-1, 4 trainable words belong to the indeterminate category, i.e., the POS is not provided. For these words, we first used a POS tagger (Ratnaparkhi, 1996) to determine the correct POS.

For a word w that may occur in phrasal word form (eg, the verb "turn" and the phrasal form "turn down"), we train a separate classifier for each phrasal word form. During testing, if w appears in a phrasal word form, the classifier for that phrasal word form is used. Otherwise, the classifier for w is used.

6 Empirical Results

We ran the different learning algorithms using various knowledge sources. Table 2 (Table 3) shows

[1]We included 718 training instances from the HECTOR dictionary used in SENSEVAL-1, together with 13,127 training instances from the training corpus supplied.

Algorithm	POS	Surrounding Words		Collocations		Syntactic Relations		Combined	
	(i)	(ii) M_2=3	(iii) M_2=0	(iv) M_2=3	(v) M_2=0	(vi) M_2=3	(vii) M_2=0	(viii) = i+ii+iv+vi	(ix) = i+iii+v+vii
SVM									
- 1-per-class	54.7	51.6	57.7	52.8	60.5	49.1	54.5	61.5	**65.4**
AdB									
- normal	53.0	51.9	52.5	52.5	53.2	52.4	51.2	54.6	53.6
- 1-per-class	55.9	53.9	55.4	55.7	59.3	53.5	52.4	62.4	**62.8**
NB									
- normal	58.0	55.8	52.5	54.5	39.5	54.1	54.0	61.6	53.4
- 1-per-class	57.6	56.2	51.5	55.8	37.9	54.0	54.2	**62.7**	52.7
DT									
- normal	55.3	50.9	49.1	**57.2**	52.4	54.2	53.7	56.8	52.6
- 1-per-class	54.9	49.7	48.1	54.3	51.3	52.7	51.5	52.2	50.0

Table 2: Contribution of knowledge sources on SENSEVAL-2 data set (micro-averaged recall on all words)

Algorithm	POS	Surrounding Words		Collocations		Syntactic Relations		Combined	
	(i)	(ii) M_2=3	(iii) M_2=0	(iv) M_2=3	(v) M_2=0	(vi) M_2=3	(vii) M_2=0	(viii) = i+ii+iv+vi	(ix) = i+iii+v+vii
SVM									
- 1-per-class	70.3	65.5	70.3	69.5	74.0	65.1	69.8	76.3	**79.2**
AdB									
- normal	67.2	63.5	64.4	64.2	65.2	65.7	65.6	68.2	68.4
- 1-per-class	71.6	67.0	68.9	69.7	71.2	69.4	68.3	77.7	**78.0**
NB									
- normal	71.5	66.6	63.5	69.1	53.9	69.4	69.6	75.7	67.2
- 1-per-class	71.6	67.3	64.1	70.3	53.0	69.8	70.4	**76.3**	68.2
DT									
- normal	69.2	66.2	65.0	70.2	67.9	68.9	68.6	**73.4**	70.2
- 1-per-class	68.7	66.6	65.4	67.0	64.4	67.6	64.8	71.4	67.8

Table 3: Contribution of knowledge sources on SENSEVAL-1 data set (micro-averaged recall on all words)

POS	SVM	AdB	NB	DT	S1	S2	S3
noun	68.8	69.2	66.4	60.0	68.2	69.5	66.8
verb	61.1	56.1	56.6	51.8	56.6	56.3	57.6
adj	68.0	64.3	68.4	63.8	73.2	68.8	66.8
all	65.4	62.8	62.7	57.2	64.2	63.8	62.9

(a) SENSEVAL-2 data set

POS	SVM	AdB	NB	DT	s1	s2	s3
noun	85.2	84.9	82.3	81.3	84.9	80.6	80.8
verb	77.0	74.4	73.3	69.5	70.5	70.9	68.7
adj	75.8	74.6	74.5	70.9	76.1	74.3	73.5
indet	76.9	76.8	74.3	70.2	77.6	76.9	76.6
all	79.2	78.0	76.3	73.4	77.1	75.5	74.6

(b) SENSEVAL-1 data set

Table 4: Best micro-averaged recall accuracies for each algorithm evaluated and official scores of the top 3 participating systems of SENSEVAL-2 and SENSEVAL-1

the accuracy figures for the different combinations of knowledge sources and learning algorithms for the SENSEVAL-2 (SENSEVAL-1) data set. The nine columns correspond to: (i) using only POS of neighboring words (ii) using only single words in the surrounding context with feature selection ($M_2 = 3$) (iii) same as (ii) but without feature selection ($M_2 = 0$) (iv) using only local collocations with feature selection ($M_2 = 3$) (v) same as (iv) but without feature selection ($M_2 = 0$) (vi) using only syntactic relations with feature selection on words ($M_2 = 3$) (vii) same as (vi) but without feature selection ($M_2 = 0$) (viii) combining all four knowledge sources with feature selection (ix) combining all four knowledge sources without feature selection.

SVM is only capable of handling binary class problems. The usual practice to deal with multi-class problems is to build one binary classifier per output class (denoted “1-per-class”). The original AdaBoost, Naive Bayes, and decision tree algo-

POS	SVM			AdB			NB			DT		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
noun	$\sim$	$\sim$	$>$	$\sim$	$\sim$	$>$	$\sim$	$\ll$	$\sim$	$\ll$	$\ll$	$\ll$
verb	$\gg$	$\gg$	$\gg$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\ll$	$\ll$	$\ll$
adj	$\ll$	$\sim$	$\sim$	$\ll$	$<$	$\sim$	$\ll$	$\sim$	$\sim$	$\ll$	$\ll$	$<$
all	$>$	$>$	$\gg$	$<$	$\sim$	$\sim$	$<$	$\sim$	$\sim$	$\ll$	$\ll$	$\ll$

(a) SENSEVAL-2 data set (using micro-averaged recall)

POS	SVM			AdB			NB			DT		
	s1	s2	s3	s1	s2	s3	s1	s2	s3	s1	s2	s3
noun	$\sim$	$\gg$	$\gg$	$\sim$	$\sim$	$\gg$	$<$	$\sim$	$>$	$<$	$\sim$	$\sim$
verb	$\gg$	$\gg$	$\gg$	$>$	$\sim$	$\gg$	$>$	$\sim$	$\gg$	$\sim$	$\sim$	$\sim$
adj	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$<$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
indet	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
all	$>$	$\gg$	$\gg$	$\sim$	$>$	$\gg$	$\sim$	$\sim$	$\gg$	$\ll$	$\sim$	$\sim$

(b) SENSEVAL-1 data set (using macro-averaged recall)

Table 5: Paired t-test on SENSEVAL-2 and SENSEVAL-1 data sets: "$\sim$", ("$>$" and "$<$"), and ("$\gg$" and "$\ll$") correspond to the p-value > 0.05, $(0.01, 0.05]$, and ≤ 0.01 respectively. "$>$" or "$\gg$" means our algorithm is significantly better.

rithms can already handle multi-class problems, and we denote runs using the original AdB, NB, and DT algorithms as "normal" in Table 2 and Table 3.

Accuracy for each word task w can be measured by recall (r) or precision (p), defined by:

$$\text{r} = \frac{\text{no. of test instances correctly labeled}}{\text{no. of test instances in word task } w}$$

$$\text{p} = \frac{\text{no. of test instances correctly labeled}}{\text{no. of test instances output in word task } w}$$

Recall is very close (but not always identical) to precision for the top SENSEVAL participating systems. In this paper, our reported results are based on the official fine-grained scoring method.

To compute an average recall figure over a set of words, we can either adopt micro-averaging (mi) or macro-averaging (ma), defined by:

$$\text{mi} = \frac{\text{total no. of test instances correctly labeled}}{\text{total no. of test instances in all word tasks}}$$

$$\text{ma} = \frac{1}{N} \sum_{i \in \text{ word tasks}}^{N = |\text{word tasks}|} \text{recall for word task } i$$

That is, micro-averaging treats each test instance equally, so that a word task with many test instances will dominate the micro-averaged recall. On the other hand, macro-averaging treats each word task equally.

As shown in Table 2 and Table 3, the best micro-averaged recall for SENSEVAL-2 (SENSEVAL-1) is 65.4% (79.2%), obtained by combining all knowledge sources (without feature selection) and using SVM as the learning algorithm.

In Table 4, we tabulate the best micro-averaged recall for each learning algorithm, broken down according to nouns, verbs, adjectives, indeterminates (for SENSEVAL-1), and all words. We also tabulate analogous figures for the top three participating systems for both SENSEVALs. The top three systems for SENSEVAL-2 are: JHU (S1) (Yarowsky et al., 2001), SMUls (S2) (Mihalcea and Moldovan, 2001), and KUNLP (S3) (Seo et al., 2001). The top three systems for SENSEVAL-1 are: hopkins (s1) (Yarowsky, 2000), ets-pu (s2) (Chodorow et al., 2000), and tilburg (s3) (Veenstra et al., 2000). As shown in Table 4, SVM with all four knowledge sources achieves accuracy higher than the best official scores of both SENSEVALs.

We also conducted paired t test to see if one system is significantly better than another. The t statistic of the difference between each pair of recall figures (between each test instance pair for micro-averaging and between each word task pair for macro-averaging) is computed, giving rise to a p value. A large p value indicates that the two systems are not significantly different from each other. The comparison between our learning algorithms and the top three participating systems is given in Table 5. Note that we can only compare macro-averaged recall for SENSEVAL-1 systems, since the sense of each individual test instance output by the SENSEVAL-1 participating systems is not available. The comparison indicates that our SVM system is better than the best official SENSEVAL-2 and SENSEVAL-1 systems at the level of significance 0.05.

Note that we are able to obtain state-of-the-art results using a *single* learning algorithm (SVM), without resorting to combining multiple learning algorithms. Several top SENSEVAL-2 participating systems have attempted the combination of classifiers using different learning algorithms.

In SENSEVAL-2, JHU used a combination of various learning algorithms (decision lists, cosine-based vector models, and Bayesian models) with various knowledge sources such as surrounding words, local collocations, syntactic relations, and morphological information. SMUls used a k-nearest neighbor algorithm with features such as keywords, collocations, POS, and name entities. KUNLP used Classification Information Model, an entropy-based learning algorithm, with local, topical, and bigram contexts and their POS.

In SENSEVAL-1, hopkins used hierarchical decision lists with features similar to those used by JHU in SENSEVAL-2. ets-pu used a Naive Bayes classifier with topical and local words and their POS. tilburg used a k-nearest neighbor algorithm with features similar to those used by (Ng and Lee, 1996). tilburg also used dictionary examples as additional training data.

7 Discussions

Based on our experimental results, there appears to be no single, universally best knowledge source. Instead, knowledge sources and learning algorithms interact and influence each other. For example, local collocations contribute the most for SVM, while parts-of-speech (POS) contribute the most for NB. NB even outperforms SVM if only POS is used. In addition, different learning algorithms benefit differently from feature selection. SVM performs best without feature selection, whereas NB performs best with some feature selection ($M_2 = 3$). We will investigate the effect of more elaborate feature selection schemes on the performance of different learning algorithms for WSD in future work.

Also, using the combination of four knowledge sources gives better performance than using any single individual knowledge source for most algorithms. On the SENSEVAL-2 test set, SVM achieves 65.4% (all 4 knowledge sources), 64.8% (remove syntactic relations), 61.8% (further remove POS), and 60.5% (only collocations) as knowledge sources are removed one at a time.

Before concluding, we note that the SENSEVAL-2 participating system UMD-SST (Cabezas et al., 2001) also used SVM, with surrounding words and local collocations as features. However, they reported recall of only 56.8%. In contrast, our implementation of SVM using the two knowledge sources of surrounding words and local collocations achieves recall of 61.8%. Following the description in (Cabezas et al., 2001), our own re-implementation of UMD-SST gives a recall of 58.6%, close to their reported figure of 56.8%. The performance drop from 61.8% may be due to the different collocations used in the two systems.

References

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised sense tagging using support vector machines. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 59–62.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

Martin Chodorow, Claudia Leacock, and George A. Miller. 2000. A topical/local classifier for word sense identification. *Computers and the Humanities*, 34(1–2):115–120.

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

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 1–5.

Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 172–180.

Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156.

Adam Kilgarriff and Martha Palmer. 2000. Introduction to the special issue on SENSEVAL. *Computers and the Humanities*, 34(1–2):1–13.

Rada F. Mihalcea and Dan I. Moldovan. 2001. Pattern learning and active feature selection for word sense disambiguation. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 127–130.

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*, pages 82–91.

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*, pages 40–47.

Hwee Tou Ng. 1997. Exemplar-based word sense disambiguation: Some recent improvements. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 208–213.

Ted Pedersen and Rebecca Bruce. 1997. A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 604–609.

Ted Pedersen. 2001a. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86.

Ted Pedersen. 2001b. Machine learning with lexical features: The Duluth approach to Senseval-2. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 139–142.

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

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19.

Hee-Cheol Seo, Sang-Zoo Lee, Hae-Chang Rim, and Ho Lee. 2001. KUNLP system using classification information model at SENSEVAL-2. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 147–150.

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

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Jorn Veenstra, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34(1–2):171–177.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco.

David Yarowsky, Silviu Cucerzan, Radu Florian, Charles Schafer, and Richard Wicentowski. 2001. The Johns Hopkins SENSEVAL2 system descriptions. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 163–166.

David Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1–2):179–186.

Jakub Zavrel, Sven Degroeve, Anne Kool, Walter Daelemans, and Kristiina Jokinen. 2000. Diverse classifiers for NLP disambiguation tasks: Comparison, optimization, combination, and evolution. In *TWLT 18. Learning to Behave*, pages 201–221.