Training of Semantic Class Disambiguation Classifiers which are Applicable to All Words

Patanan Ariyakornwijit[1,a)] Kiyoaki Shirai[1,b)]

Abstract: This paper proposed a method to disambiguate semantic classes of a given word. Unlike previous approaches of supervised learning for Word Sense Disambiguation (WSD), our approach (1) uses a set of semantic classes (coarse grained word senses) that are common for all words as the sense inventory, (2) trains only a few classifiers which can be applicable to all words. Binary classifiers for semantic class disambiguation are trained by Support Vector classification with the conventional WSD features. Two kinds of the training data are considered and compared: one is monosemous words in a raw text, the other is polysemous words in a sense tagged corpus. Experimental results showed that the latter was appropriate for semantic class disambiguation. Our proposed method achieved 43.8% accuracy (the ratio of instances where chosen semantic classes are exactly agreed with the gold standard) and 43.6% average F-measures of binary classifiers, which are much improved than the baseline.

Keywords: Word Sense Disambiguation, Semantic Class, Supervised Learning, Data Sparseness Problem

1. Introduction

Word Sense Disambiguation (WSD) is the task to find the right meaning of a word in a given sentence. WSD is one of the important tasks in natural language processing such as machine translation, language understanding and information retrieval. In order to solve WSD problem, many algorithms are proposed. The supervised learning methods showed better performance than others. But it still suffers from a serious problem that it is rather difficult to prepare a large amount of training data. It is also known as 'knowledge acquisition bottleneck'.

In the previous work on supervised learning, WSD classifiers are trained for individual target words, since the sense inventories are different for target words. Therefore, it is necessary to train a bulk of classifiers in order to disambiguate senses of all words in a text. Obviously, it is difficult to prepare a sense tagged sentences for all kinds of words.

This paper proposes a method to train WSD classifiers which can be applicable to all words. In our approach, a set of semantic classes is used as the common sense inventory for all words. Here the semantic class means a coarse grained sense or rather abstract concept, such as 'artifact', 'event', 'group', 'person' and so on. Our motivation to use semantic classes as an universal set of senses is that trained classifiers could disambiguate senses of all words, especially low frequent words. It would be alleviate data sparseness problem or knowledge acquisition bottleneck.

Although semantic class disambiguation or the coarse grained WSD is not sufficient for some NLP applications, but it is still effective in several applications such as information retrieval (IR). For example, it would be useful if a search system could distinguish a meaning of 'apple' among three senses: a fruit, tree or company. As described in Section 3.1, the top concepts in WordNet are used as semantic classes in this research. Three senses of 'apple' correspond to a semantic class of 'fruit', 'plant' and 'group' in WordNet, respectively. In this way, not fine but coarse grained sense disambiguation would contribute to gain the performance of IR systems.

The rests of the paper are organized as follows. Section 2 discusses related work about coarse grained WSD. The proposed method is described in Section 3, which includes the definition of semantic classes, the system architecture, the features used for training classifiers and how to prepare the training data. We show results of several experiments to evaluate our method in Section 4. We finally conclude the paper and discuss future work in Section 5.

2. Related Work

Levin proposed classification of English verbs[1]. She classified over 3,000 English verbs with the assumption that a verb's meaning influences its syntactic behavior. She first describes that verbs can express their arguments in alternate ways. Then, she presents the classes of verbs that share a kernel of meaning and discover in detail of the behavior for each class. Finally, she draws classes and their alternations, which become the verb inventory. At that time, the verb inventory of Levin has one drawback; her classification of verbs are based on syntactic properties unlike those in WordNet[2].

A method for mapping WordNet entries into Levin classes is proposed by Korhonen[3]. Words in WordNet are arranged in hierarchical, and each node contains a set of synonym called synset. 1,616 synsets were automatically mapped to one of 32

[1] School of Information Science, Japan Advanced Institute of Science and Technology
[a)] patanan.a@jaist.ac.jp
[b)] kshirai@jaist.ac.jp

Levin classes, where the accuracy was 81%.

It is an open question how to define a set of common semantic classes for all words. It may depend on the applications requiring semantic class disambiguation. In this paper, WordNet is used for semantic class definition, however, any sets of semantic classes, including above verb classes, could be applicable for our method.

WSD with a coarse grained sense inventory has also been studied. Izquierdo et al. used Base Level Concepts (BLC) from WordNet in order to perform the class-based Word Sense Disambiguation[4]. He conducted the experiments under two different sets of BLC: all types of relations encoded in WordNet, and only the hyponymy relations. A naive most frequent classifier is able to perform a semantic tagging with accuracy figures over 75%.

Kohomban and Lee proposed a technique based on the similarity of word senses, which are coarser and more general concepts[5]. The general classes are mapped to fine grained senses with simple heuristics. Their proposed method trained a classifier for a word by using memory-based learner with 4 effective features: Local Context, Part-of-Speech, Collocation and Syntactic Relation. They reported that the accuracy was over 77%.

Semantic class disambiguation is not only well known in English but also another languages. Izquierdo et al. presented an approach of semantic disambiguation based on machine learning and semantic classes for Spanish[6]. They used semantic classes in order to collect a large number of examples for each class while the degree of polysemy is also reduced. Cast3LB, manually annotated corpus with Spanish WordNet senses, has been applied to Support Vector Machine with linear kernel in order to perform semantic disambiguation. The accuracy of disambiguation for nouns and verbs was 76.2%.

Resnik proposed an unsupervised WSD method based on selectional preferences [7]. Statistical model of selectinal restriction, which is an association score between a predicate and a conceptual class of a noun, is obtained from a corpus without sense tags and used for disambiguation of nouns. Although he evaluated his method for disambiguation of fine grained WordNet senses, his method could be used for coarse grained WSD using association scores for conceptual classes (i.e. coarse senses).

Past researches on coarse sense disambiguation tried to train classifiers for individual words. On the contrary, we aim to implement the universal model by training semantic class disambiguation classifiers that could be applicable to all words. We will further discuss the differences between previous work and our method in Subsection 3.2.

3. Proposed Method

3.1 Semantic Class

WordNet, broadly cited as a sense repository, offers hierarchical structure of senses (meaning). WordNet compiles 117,000 synsets, which are organized into forty-five lexicographer files based on syntactic category and logical groupings. Semantic classes in this research are defined as this coarsest level of the senses in WordNet. There are 45 semantic classes: 26 of a noun, 15 of a verb, 3 of an adjective, and 1 of an adverb. For our research, only 18 noun semantic classes and 14 verb semantic classes are used, since other semantic classes do not frequently appear in the test corpus used in the experiment in Section 4. TableA·2 in Appendix shows the list of semantic classes.

3.2 Architecture

As shown in Figure 1, in the most of previous work, WSD classifiers should be trained for individual target word w_i, since the sense inventories $\{\ldots, S_{ij}, \ldots\}$ are different. On the other hand, in our approach, we develop one system which can disambiguate all words in a text as shown in Figure 2. Note that our system choose semantic classes SC_i that are common for all words.

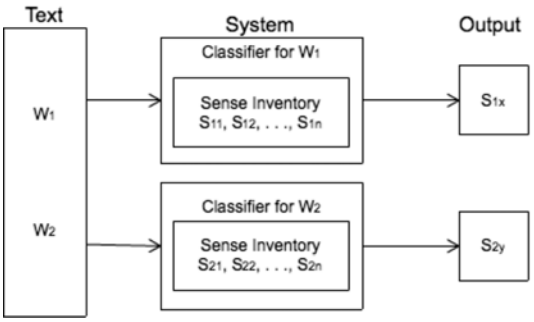

Fig. 1 Previous Approach

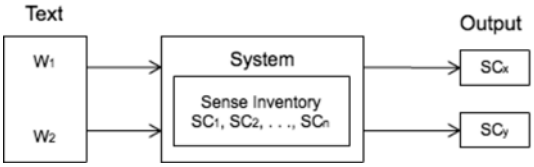

Fig. 2 Our Approach

Our system choose semantic classes for a given target word as follows. Fig. 3 also illustrates our procedure.

i. Part-of-speech (POS) of the target word is identified by POS tagger. Only nouns and verbs could be disambiguated.

ii. By looking up WordNet, all possible candidates of semantic classes $\{\cdots, SC_k, \cdots\}$, which is a subset of all noun or verb semantic classes, for the target word are retrieved.

iii. Each binary classifier CL_i judge if the target word has SC_i or not. The classifiers for individual semantic classes are trained in advance. For classification, features used for CL_i are extracted from a context of the target word.

iv. Finally the system outputs all SC_i where CL_i judges 'yes' as chosen semantic classes for the target word.

3.3 Classifiers

In general, a classifier is a model that has ability to identify which category an instance belongs to. For this research, the classifier (CL_i) has ability to judge whether the target word contains the semantic class SC_i or not. In this section, we present the

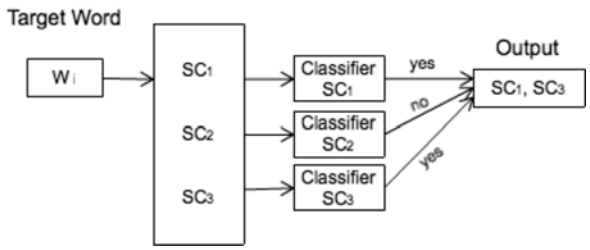

Fig. 3 Architecture of Our System

learning algorithm and the features that we use to implement the classifier.

3.3.1 Learning Algorithm

In this research, Support Vector Machines (SVM) is used as the classification algorithm. SVM is a kind of supervised learning, which can analyze data and recognize patterns. SVM is a binary classifier trained from a collection of positive and negative data. SVM works as follows. First, training data consisting of positive and negative samples is prepared. Then, the model is built by using SVM training algorithm. The model will separate two kind of samples with the clear gap. A new the test data will be consulted with the model and judged as positive or negative.

In this paper, we use Liblinear [8] as a supervised learning algorithm. We, first, tried to use Libsvm[9], but it is not a good option for evaluating a large number of instances and features. Thus we changed the learning algorithm from Libsvm to Liblinear. Without using kernels, Liblinear can quickly train a much larger set via a linear classifier. We use the L2-regularized L2-loss support vector classification with the default setting of Liblinear.

3.3.2 Features

The feature set was fairly simple; we borrow conventional features which have been successfully used in WSD. We used the features from Kohomban and Lee's method [5] with some modifications.

3.3.2.1 Local Context

Local context is a feature reptrsented by words around the target word. Local context features are extracted from a context with a window size n, that is, n words left and right from the target word. In the preliminary experiment, we changed the window size n as $\{3, 5, 10, 20\}$ and found that $n = 3$ was the best.

In our method, the punctuation marks and function words were removed. All words were converted into lower case. When the window did not exceed the boundaries of a document, i.e. there were not enough words to either side of the word within the window, those remaining positions are ignored.

3.3.2.2 Part-of-Speech

This feature consists of parts of speech of 2-gram, 3-gram and 4-gram including the target word itself. To obtain POS features, the sentences are analyzed by POS tagger [10]. When there were not enough words to either side of the target word, the value *"null"* was used to fill the vacancies.

3.3.2.3 Collocation

A collocation feature is the connection between the words under consideration (target word) and surrounding words, and it is used widely to solve WSD task. Collocation has ability to determine the sense of the ambiguous word it contains. Aditi explained the effectiveness of collocation by showing the example of "pound"; when it appears in "pound of [something]", its sense can be regarded as 'unit of measure' [11].

In this paper, we consider 3 types of collocation, 2-gram, 3-gram and 4-gram including the target word itself. In our approach, the classifiers are applied to all words, i.e. they accept many kinds of words as the target word. Therefore, the target word is replaced by wild card symbol '*'. Similar to Part of Speech feature, if there is not enough word on either side of a context, we replace vacancies with "null".

3.3.2.4 Syntactic Relation

Syntactic Relation feature represents more direct grammatical relationships, such as subject-verb or noun-adjective, between the target word and its surrounding word. We use the Stanford parser in order to extract the features[12]. Stanford parser provides two kinds of dependencies: typed dependencies and collapsed typed dependencies. The typed dependencies are a collection of direct dependencies between words in a sentence, where each word in the sentence (except the head of the sentence) is the dependent of one other word. While collapsed typed dependencies are obtained by collapsing a pair of typed dependencies into a single typed dependency, which is then labeled with a name based on the word between two dependencies.

We will use the collapsed typed dependencies of Stanford parser as the syntactic relation features. The word indices in the output of the parser are removed and the target word is replaced with '*'. In our model, not only dependencies associated with the target word but all dependences in the sentence are used as features.

3.4 Training Data

For our method, we use two kinds of training data: a collection of monosemous words without sense tagging and polysemous words with sense tagging.

Monosemous words

First, we use monosemous words, which have only one semantic class in WordNet, as the training data. For training the classifier of a semantic class SC_i, all words which has only one semantic class SC_i are used as positive samples, while words which have one semantic class other than SC_i are negative samples. A raw text can be used for the training data, since no manual annotation is required.

We propose another method to construct the training data considering balance of the number of positive and negative data. In this method, all monosemous words that has a SC_i are used as positive samples. On the other hand, for the negative samples, monosemous words that has a semantic class other than SC_i are randomly chosen so that the ratio of the number of positive and negative samples would be 1:1.

Polysemous words

The second data set that we use as a training data consists of

polysemous words (or ambiguous words). It is supposed that the correct semantic classes of polysemous words are annotated. Similar to monosemous words, positive and negative samples for training the classifier CL_i are prepared according to the annotated semantic classes of polysemous words.

There are both advantages and disadvantages of using either monosemous words or polysemous words as the training data. For monosemous words, it is easy to prepare a large number of training data because positive and negative samples are obtained from a raw text. However, it is still uncertain that unambiguous words are really useful for classification of ambiguous words. Furhtermore, words in the training and test data are totally different. Such gaps may cause negative impacts on semantic class disambiguation. On the other hand, polysemous words are real examples of ambiguous words. It would be expected that polysemous words are more appropriate as the training data than monosemous words. However, it is rather difficult to construct a large scale data since sense or semantic class annotation is required.

4. Evaluation

In this section, we evaluate our proposed method. First the corpora used for the experiments are introduced. Then we explain evaluation criteria. Finally, we report and discuss results of the experiments.

4.1 Data

Two experiments were conducted to evaluate our proposed method: one is 'monosemous words task' where the set of monosemous words are used as the training data, the other is 'polysemous words task' where the polysemous words are used. For monosemous words task, the training data of Senseval-3 English lexical sample task is used as a test data, and both Senseval-3 corpus and the Daily Yomiuri newspaper articles [14], which is a raw text, are used as the training data. Note that only polysemous and monosemous words in Senseval-3 corpus are used for test and training data, respectively. Thus the test and training data are mutually exclusive although Senseval-3 corpus is used for both data. For polysemous words task, Senseval-3 corpus is used as the test and training data by 5-fold cross validation.

For both tasks, the gold standard semantic classes are obtained by mapping gold sense tags in Senseval-3 corpus to semantic classes. In general, one target word may have one or more semantic classes as gold standard.

Table 1 shows number of target words (types), number of target instances (tokens) and average number of semantic classes per a target word in Senseval-3 data.

Table 1 Statistics of Senseval-3 Corpus

	Words	Instances	Semantic Classes
Nouns	20	3,593	3.90
Verbs	28	3,953	4.18

Table 2 reveals the average number of positive and negative samples per a semantic class in Senseval-3 and Yomiuri Shimbun corpus in monosemous words task. Note that the amount of the training data in monosemous words task is much greater than in polysemous.

Table 2 Training Data in Monosemous Words Task

	Senseval-3		Yomiuri	
	positive	negative	positive	negative
Nouns	163	4,080	2,370	59,100
Verbs	67	4,460	235	3,290

4.2 Criteria

The proposed methods are evaluated in terms of six kinds of criteria. They are separated into two groups: Instance Based Evaluation and Judgment Based Evaluation. Instance Based Evaluation is capable of evaluating the outputs for target instances or test sentences, while the Judgment Based Evaluation is able to evaluate the judgment of each clasifiers of a semantic class.

4.2.1 Instance Based Evaluation

Instance Based Evaluation, Accuracy (Exact Match) and Accuracy (Partial Match), is a measurement of the accuracy of semantic classes chosen for the target instances. Before describing definitions of these two criteria, we would like to briefly explain parameters that will be used for calculating the accuracies.

Exact Match (EM): the judgment is EM when the semantic classes chosen by the system is completely the same as the gold semantic class.

Partial Match (PM): the judgment is PM when the semantic class chosen by the system are not exactly same as gold, but at least one semantic class is correct.

Not Match (NM): the judgment is NM when the semantic classes chosen by the system DO NOT contain any gold semantic classes.

Table A·1 in Appendix shows examples of judgment of EM, PM and NM.

1. Accuracy (Exact Match):

It evaluates how the chosen semantic classes for an instance are completely correct. It defined as Eq. (1). $N(\cdot)$ stands for the number of instances judged as EM, PM or NM.

$$\textbf{Acc(Exact Match)} = \frac{N(EM)}{N(EM) + N(PM) + N(NM)} \quad (1)$$

2. Accuracy (Partial Match):

It loosely evaluates the correctness of chosen semantic classes, defiend as Eq. (2).

$$\textbf{Acc(Partial Match)} = \frac{N(EM) + N(PM)}{N(EM) + N(PM) + N(NM)} \quad (2)$$

4.2.2 Judgment Based Evaluation

We also evaluate the performance of individual classifiers CL_i. Judgment Based Evaluation contains 4 types of measurements: Agreement Ratio, Precision, Recall and F-measure. Agreement Ratio is the ratio of the cases where the system's judgment (yes or no) and the gold standard are agreed. In the next subsection, the averages of all classifiers for 32 semantic classes will be shown.

4.3 Results

The performance of the proposed method is compared with Baseline. Baseline is the system which always choose the most

frequent semantic class. Frequencies of semantic classes are obtained from either monosemous or polysemous words in training data.

4.3.1 Monosemous Words Task

As explained in Subsection 3.4, two kinds of training data are used. 'All:All' stands for the training data consisting of all positive and negative samples in the corpus, while 'Random 1:1' stands for the data with the equal number of positive and negative samples where negative samples are randomly chosen.

All:All

Table 3 and Table 4 show the results of Instance Based Evaluation and Judgment Based Evaluation on this experiment.

Table 3 Instance Based Evaluation of All : All

		Noun	Verb	All
System	Exact Match	3.1%	2.7%	2.9%
	Partial Match	3.9%	2.9%	3.4%
Baseline	Exact Match	24.2%	26.7%	25.4%
	Partial Match	30.0%	30.6%	30.3%

Table 4 Judgment Based Evaluation of All : All

		Noun	Verb	All
System	Agreement Ratio	74.8%	74.1%	74.4%
	Precision	32.2%	24.2%	29.3%
	Recall	2.2%	2.3%	2.2%
	F-measure	3.6%	3.7%	3.6%
Baseline	Agreement Ratio	66.6%	65.8%	66.2%
	Precision	8.7%	14.1%	11.1%
	Recall	19.2%	21.4%	20.2%
	F-measure	9.5%	13.6%	11.3%

As shown in Table 3, both measurements of Instance Based Evaluation are about 10 times lower than the Baseline. For the results of Judgment Based Evaluation in Table 4, only Agreement Ratio and Precision are higher than the Baseline.

According to our error analysis, almost all of the judgments by the classifiers are negative. We analyzed the number of positive and negative samples in the training data. The smallest ratio of number of positive to number of negative sample is 1:4, while the largest ratio is 1:1564. Such unbalance data might lead to the bias to negative judgment and misclassification of the system.

Random 1:1

Table 5 and Table 6 show the results of the Instance Based Evaluation and Judgment Based Evaluation on Random 1:1 experiment. Results of Judgement Based Evaluation of 32 classifiers for individual semantic classes are shown in Table A·3 in Appendix.

The closer quantity of negative samples and positive sample by using random selection method leads the scores higher than the Baseline in all evaluation criteria except for Agreement Ratio. The system achieved better performance for nouns than verbs in terms of all criteria.

Comparing to All:All, the performance of Random 1:1 shows great improvement. For instance, Accuracy (Exact Match) in Random 1:1 is roughly 10 times better, and Recall is significantly improved than All:All. On the other hand, Agreement Ratio is about 22% worse, and the precision is only 4.8% worse than All:All. In total, however, Random 1:1 is better than All:All since F-measure is about 9 times greater.

Table 5 Instance Based Evaluation of Random 1:1

		Noun	Verb	All
System	Exact Match	30.2%	25.3%	28.6%
	Partial Match	60.4%	42.5%	53.0%
Baseline	Exact Match	24.2%	26.7%	25.4%
	Partial Match	30.0%	30.6%	30.3%

Table 6 Judgment Based Evaluation of Random 1:1

		Noun	Verb	All
System	Agreement Ratio	60.1%	55.2%	58.0%
	Precision	29.6%	25.7%	27.9%
	Recall	48.9%	41.0%	45.4%
	F-measure	34.4%	27.1%	31.2%
Baseline	Agreement Ratio	66.6%	65.8%	66.2%
	Precision	8.7%	14.1%	11.1%
	Recall	19.2%	21.4%	20.2%
	F-measure	9.5%	13.6%	11.3%

These seem to be a good sign of improvement of the system. We can conclude that the unbalance of data could cause negative impacts in the judgment. Considering the balance of number of positive and negative samples seems important and effective when monosemous words are used as the training data.

4.3.2 Polysemous Words Task

The results of polysemous words task are shown in Table 7 and 8. In these tables, averages of 5-fold cross validation are shown. Results of each iteration are shown in Table 9 and 10. The five times of switching the partition of test data, Accuracy (Exact Match) is around 40 ~ 45% and the Accuracy (Partial Match) is 48 ~ 55%. These two measurements are higher than the Baseline roughly 1.8 times. Similarly, the proposed method outperformed the Baseline for all 4 criteria of Judgement Based Evaluation. The performance for nouns was better than verbs, however, differences were not so great as compared with monosemous words random 1:1.

Table 7 Instance Based Evaluation of Polysemous Words Task

		Noun	Verb	All
System	Exact Match	42.3%	45.3%	43.8%
	Partial Match	50.7%	49.5%	50.1%
Baseline	Exact Match	40.2%	36.6%	38.4%
	Partial Match	45.7%	39.4%	42.6%

Table 8 Judgment Based Evaluation of Polysemous Words Task

		Noun	Verb	All
System	Agreement Ratio	83.2%	82.3%	82.8%
	Precision	63.1%	61.6%	62.4%
	Recall	37.1%	36.4%	36.8%
	F-measure	43.1%	42.8%	43.0%
Baseline	Agreement Ratio	74.1%	72.7%	73.5%
	Precision	10.4%	24.1%	16.4%
	Recall	15.9%	19.2%	17.3%
	F-measure	12.1%	16.7%	14.1%

Using polysemous words as a training data shows a better performance than using the monosemous words. Although the number of positive training data using polysemous words is around a hundred or more, while the number of positive training using monosemous words is roughly a thousand or more. Regardless of tenth size of the training data, polysemous words are more effective. If more polysemous words are available for training the classifiers, the performance is expected to be improved more. In another view, there might be gaps between the monosemous and

Table 9 Instance Based Evaluation of 5 Trials of Cross Validation in Polysemous Words Task

		1st	2nd	3rd	4th	5th
System	Exact Match	42.1%	44.6%	44.2%	44.9%	40.3%
	Partial Match	48.8%	54.1%	50.8%	50.1%	49.2%
Baseline	Exact Match	37.6%	39.2%	40.2%	37.1%	38.0%
	Partial Match	42.3%	43.7%	41.1%	40.5%	42.5%

Table 10 Judgment Based Evaluation of 5 Trials of Cross Validation in Polysemous Words Task

		1st	2nd	3rd	4th	5th
System	Agreement Ratio	81.5%	83.7%	82.3%	83.0%	82.9%
	Precision	64.4%	56.4%	60.2%	66.1%	65.6%
	Recall	36.7%	36.4%	39.4%	37.4%	36.3%
	F-measure	42.5%	41.9%	44.3%	44.1%	43.8%
Baseline	Agreement Ratio	72.6%	74.5%	73.6%	73.0%	73.1%
	Precision	16.1%	15.6%	20.7%	15.6%	16.6%
	Recall	17.4%	17.4%	19.8%	16.9%	16.9%
	F-measure	14.1%	14.3%	17.1%	13.3%	13.9%

polysemous words in terms of the contexts where a certain semantic class appears.

The performance of our method is still low, although semantic class disambiguation or coarse grained WSD is relatively easy task. Further investigation is required to reveal which methodology, two approaches discussed in Subsection 3.2 or other unsupervised WSD methods, is appropriate to precisely disambiguate semantic classes.

5. Conclusion

This paper proposed the universal model for classifying semantic classes, which could be applicable to all words. We compare two kinds of classifiers, which are differentiated by the source of training data. One is the classifier trained from monosemous words in a raw text, while the other is the classifier using polysemous words in a sense tagged corpus. In our experiments, we found that (1) it is important to consider balance of number of positive and negative samples in monosemous words training data, (2) a relatively small amount of polysemous words is more appropriate than monosemous words. The best performance of our proposed method is that 43.8% accuracy (exact match) and 43.6% F-measure. They are significantly better than the Baseline, although there is much room to improve the performance for real NLP applications.

For future work, we are planning to add another corpus for monosemous words training data in order to enlarge the number of positive samples. The Daily Yomiuri newspaper articles in 2002 will be used. The motivation is quite simple: the more training data is, the higher performance is expected.

Furthermore, we are planning to use another learning algorithm such as K-nearest Neighbors to classify the semantic classes. Then, we could compare the performance of Support Vector Machine to K-nearest Neighbors.

References

[1] Beth Levin: English Verb Classes and Alternations, University of Chicago Press, Chicago, IL, 1993.

[2] Christiane Fellbaum, editor: WordNet An Electronic Lexical Database, *The MIT Press*, Cambridge, MA ; London, May 1998.

[3] Anna Korhonen: Assigning Verbs to Semantic Classes via WordNet, *In Proceedings of the COLING Workshop on Building and Using Semantic Networks*, 2002.

[4] Ruben Izquierdo, Armando Suarez and German Rigau: A Proposal of Automatic Selection of Coarse-Grained Semantic Classes for WSD, *In Procesamiento del lenguaje natural. N. 39 (sept. 2007)*, pages 189–196, 2007.

[5] Upali S. Kohomban and Wee Sun Lee: Learning Semantic Classes for Word Sense Disambiguation, *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 34-41, 2005.

[6] Rubén Izquierdo-Beviá, Lorenza Moreno-Monteagudo, Borja Navarro and Armando Suárez: Spanish all-words semantic class disambiguation using Cast3LB corpus, *In Proceedings of the 5th Mexican international conference on Artificial Intelligence*, MICAI'06, pages 879–888, 2006.

[7] Philip Resnik: Selectional Preference and Sense Disambiguation. *In Proceeding of ACL Siglex Workshop on Tagging Text with Lexical Semantics*, 1997.

[8] Rong-En Fan and Kai-Wei Chang and Cho-Jui Hsieh and Xiang-Rui Wang and Chih-Jen Lin: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, pages 1871–1874, 2008

[9] Chih-Chung Chang and Chih-Jen Lin: LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, pages 27:1–27:27, 2011.

[10] Yoshimasa Tsuruoka: An English Pos–Tagger, `http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/` [2008/07/28].

[11] Aditi Shrikumar: Two Approaches to All-Words Sense Disambiguation, *CS288 Final Project*, Berkley University, 2009.

[12] Marie catherine De Marneffe, Bill Maccartney, and Christopher D. Manning: Generating Typed Dependency Parses from Phrase Structure Parses, *In LREC 2006*.

[13] R. Mihalcea, T. Chklovski, and A. Kilgarriff: The Senseval-3 English lexical sample task, *In Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25-28, 2004.

[14] 'The Daily Yomiuri' newspaper articles in English, 2003.

Appendix

Table A·1 Examples of Instance Based Evaluation.

Target word	Sentence	Correct Semantic Classes	Semantic Classes chosen by System	Judgment
T_1	S_1	SC_1	SC_1	EM
T_1	S_2	SC_1, SC_2	SC_1, SC_2	EM
T_1	S_3	SC_1, SC_2	SC_1	PM
T_2	S_4	SC_1, SC_3	SC_1	PM
T_2	S_5	SC_3, SC_4	SC_3, SC_4, SC_5	PM
T_3	S_6	SC_5, SC_6	SC_3, SC_5	PM
T_4	S_7	SC_7	SC_8	NM
T_4	S_8	SC_7	-	NM

Table A·2 List of Semantic Classes in WordNet

ID	Name	Contents
03*	noun.Tops	unique beginner for nouns
04	noun.act	nouns denoting acts or actions
05*	noun.animal	nouns denoting animals
06	noun.artifact	nouns denoting man-made objects
07	noun.attribute	nouns denoting attributes of people and objects
08	noun.body	nouns denoting body parts
09	noun.cognition	nouns denoting cognitive processes and contents
10	noun.communication	nouns denoting communicative processes and contents
11	noun.event	nouns denoting natural events
12*	noun.feeling	nouns denoting feelings and emotions
13*	noun.food	nouns denoting foods and drinks
14	noun.group	nouns denoting groupings of people or objects
15	noun.location	nouns denoting spatial position
16*	noun.motive	nouns denoting goals
17	noun.object	nouns denoting natural objects (not man-made)
18	noun.person	nouns denoting people
19*	noun.phenomenon	nouns denoting natural phenomena
20*	noun.plant	nouns denoting plants
21	noun.possession	nouns denoting possession and transfer of possession
22	noun.process	nouns denoting natural processes
23	noun.quantity	nouns denoting quantities and units of measure
24	noun.relation	nouns denoting relations between people or things or ideas
25	noun.shape	nouns denoting two and three dimensional shapes
26	noun.state	nouns denoting stable states of affairs
27	noun.substance	nouns denoting substances
28*	noun.time	nouns denoting time and temporal relations
29	verb.body	verbs of grooming, dressing and bodily care
30	verb.change	verbs of size, temperature change, intensifying, etc.
31	verb.cognition	verbs of thinking, judging, analyzing, doubting
32	verb.communication	verbs of telling, asking, ordering, singing
33	verb.competition	verbs of fighting, athletic activities
34	verb.consumption	verbs of eating and drinking
35	verb.contact	verbs of touching, hitting, tying, digging
36	verb.creation	verbs of sewing, baking, painting, performing
37	verb.emotion	verbs of feeling
38	verb.motion	verbs of walking, flying, swimming
39	verb.perception	verbs of seeing, hearing, feeling
40	verb.possession	verbs of buying, selling, owning
41	verb.social	verbs of political and social activities and events
42	verb.stative	verbs of being, having, spatial relations
43*	verb.weather	verbs of raining, snowing, thawing, thundering

'*' denotes the semantic classes for which the classifier was not trained in our experiment.

Table A·3 Detail Results of All Classifiers for 32 Semantic Classes in Monosemous Words Task (Random 1:1)

	System				Baseline				# Positive Test Data	# Negative Test Data	# Test Data	# Positive Output	# Positive and Correct	# Positive Train Data	# Negative Train Data
	Agreement Ratio	Precision	Recall	F-measure	Agreement Ratio	Precision	Recall	F-Measure							
noun.act	0.604	0.263	0.472	0.338	0.516	0.301	0.955	0.458	178	652	830	319	84	7562	5842
noun.artifact	0.503	0.351	0.634	0.452	0.521	0.385	0.810	0.522	484	1015	1499	875	307	8130	6159
noun.attribut	0.538	0.433	0.686	0.531	0.615	0.478	0.103	0.170	522	846	1368	826	358	2149	1846
noun.body	0.716	0.869	0.750	0.805	0.221	0	0	0	204	57	261	176	153	409	389
noun.cognitio	0.467	0.325	0.624	0.427	0.682	0	0	0	455	974	1429	874	284	3307	2765
noun.commu	0.533	0.378	0.616	0.468	0.469	0.333	0.591	0.426	518	1033	1551	845	319	6690	5206
noun.event	0.720	0.224	0.241	0.232	0.825	0	0	0	79	371	450	85	19	1590	1385
noun.group	0.599	0.526	0.556	0.541	0.576	0	0	0	680	920	1600	718	378	5020	4027
noun.location	0.634	0.267	0.314	0.288	0.767	0	0	0	51	165	216	60	16	1803	1597
noun.object	0.549	0.140	0.508	0.219	0.877	0	0	0	59	416	475	215	30	691	634
noun.person	0.583	0.069	0.386	0.117	0.073	0.071	1.000	0.133	44	573	617	247	17	10752	7836
noun.posses	0.713	0.087	0.516	0.149	0.952	0	0	0	31	607	638	184	16	4910	3973
noun.process	0.387	0.080	0.553	0.140	0.912	0	0	0	38	381	419	261	21	135	133
noun.quantit	0.776	0.044	0.132	0.066	0.942	0	0	0	38	596	634	114	5	1266	1117
noun.relation	0.637	0.053	0.042	0.047	0.792	0	0	0	48	178	226	38	2	1862	1623
noun.shape	0.457	0.175	0.606	0.272	0.837	0	0	0	33	164	197	114	20	42	42
noun.state	0.577	0.340	0.447	0.386	0.703	0	0	0	253	595	848	332	113	2568	2217
noun.substar	0.824	0.701	0.712	0.707	0.705	0	0	0	66	155	221	67	47	1934	1677
verb.body	0.366	0.065	0.683	0.119	0.938	0	0	0	41	610	651	428	28	126	111
verb.change	0.580	0.331	0.263	0.293	0.474	0.309	0.477	0.375	665	1343	2008	528	175	559	465
verb.cognitio	0.388	0.255	0.739	0.380	0.744	0.475	0.122	0.194	238	702	940	689	176	426	371
verb.commur	0.397	0.310	0.845	0.453	0.296	0.296	1.000	0.457	470	1119	1589	1282	397	631	522
verb.compet	0.804	0.053	0.016	0.024	0.846	0	0	0	64	349	413	19	1	229	204
verb.consum	0.511	0.537	0.476	0.505	0.478	0	0	0	319	290	609	283	152	45	45
verb.contact	0.490	0.151	0.513	0.233	0.509	0	1	0	76	428	504	259	39	256	221
verb.creation	0.562	0.577	0.201	0.299	0.538	0	0	0	407	472	879	142	82	236	208
verb.emotion	0.731	0.064	0.333	0.108	0.953	0	0	0	15	293	308	78	5	185	167
verb.motion	0.530	0.392	0.247	0.303	0.568	0.462	0.303	0.366	198	281	479	125	49	311	272
verb.percept	0.404	0.378	0.645	0.477	0.424	0.113	0.054	0.073	389	535	924	664	251	202	176
verb.possess	0.670	0.185	0.253	0.214	0.823	0	0	0	150	696	846	205	38	174	158
verb.social	0.661	0.150	0.257	0.189	0.782	0.088	0.045	0.059	202	1109	1311	347	52	943	741
verb.stative	0.636	0.149	0.267	0.191	0.840	0	0	0	258	1345	1603	464	69	206	188
Average:	0.580	0.279	0.454	0.312	0.662	0.111	0.202	0.113							