| | |
|---|---|
| Title | Supervised Learning of Semantic Class Disambiguation Classifiers for All Words Task |
| Author(s) | Ariyakornwijit, Patanan |
| Citation | |
| Issue Date | 2012-09 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/10750 |
| Rights | |
| Description | Supervisor: Shirai Kiyoaki, 情報科学研究科, 修士 |

# Supervised Learning of Semantic Class Disambiguation Classifiers for All Words Task

Ariyakornwijit, Patanan (1010201)

School of Information Science,
Japan Advanced Institute of Science and Technology

August 9, 2012

**Keywords:** Word Sense Disambiguation, Semantic Class, Supervised Learning, Knowledge Acquisition Bottleneck.

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. The supervised learning methods showed better performance than others. But it still suffers from a serious problem 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 sense tagged sentences for all kinds of words. Unlike previous approaches of supervised learning for 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. Although semantic class disambiguation or the coarse grained WSD is not sufficient for some NLP applications, it is still effective in several applications such as information retrieval.

WordNet, broadly cited as a sense repository, offers hierarchical structure of senses (meaning). WordNet compiles 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. In the most of previous work, WSD classifiers should be trained for individual target words, since the sense inventories are different. On the other hand, in our approach, we develop one system which can disambiguate all words in a text. For each semantic class, a binary classifier, which can judge if any kinds of words in a sentence has the semantic class or not, is trained. For a given new word, all semantic classes judged as positive by classifiers are chosen as output.

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. We use a tool called Liblinear for training of classifiers. 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.

The feature set is fairly simple; we borrow conventional features which have been successfully used in WSD. We used the features from Kohomban and Lees method with some modifications. These features are local context, part-of-speech, collocation and syntactic relation. Local context is a feature represented by words around the target word. Local context features are extracted from a context with a window size 3. Part-of-speech consists of parts of speech of 2-gram, 3-gram and 4-gram including the target word itself. Collocation feature is the connection between the words under consideration (target word) and surrounding words. In this paper, we consider 2-gram, 3-gram and 4-gram including the target word itself. Syntactic Relation feature represents more direct grammatical relationships between the target word and its surrounding word. We use collapsed typed dependencies of Stanford parser in order to extract the features.

We use two kinds of training data: a collection of monosemous words without sense tagging and polysemous words with sense tagging. 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 have semantic class $SC_i$ are used as positive samples, while words which have semantic class other than $SC_i$ are negative samples. Furthermore, we propose three methods to prepare the training data using monosemous words, ‘All:All’, ‘Random 1:1’ and ‘At most P:N’. ‘All:All’ means the method using all positive and negative samples. Unfortunately, this system tends to always judge as negative. It may be caused by imbalance between number of positive and negative samples. ‘Random 1:1, means a method to construct the training data considering balance of the number of positive and negative data. In this method, all monosemous words that have a $SC_i$ are used as positive samples. On the other hand, for the negative samples, monosemous words that have a semantic class other than $SC_i$ are randomly chosen so that the ratio of positive and negative samples becomes 1:1. ‘At Most Method’ is proposed for considering variety of target words in the training data. Since a target word contains one or more contexts, At Most Method will limit the maximum contexts per each target word for each classifier in order to make the training data has difference target words as much as possible. ‘At Most P:N’ stands for the data where the ratio of positive(P) and negative(N) samples are adjusted to P:N by At Most Method. ‘At Most All:1’ means that the training data consists of all positive sample and the equal number of negative samples chosen by At Most Method.

The second data set 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.

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, Accuracy (Exact Match) and Accuracy (Partial Match), is a measurement of the accuracy of semantic classes chosen for the target instances. We also evaluate the performance of individual classifiers. Judgment Based Evaluation contains 4 types of measurements: Agreement Ratio, Precision, Recall and F-measure.

Several experiments are conducted to evaluate the proposed methods. First, Senseval-3 English lexical task corpus is used as test data. We compare monosemous and polysemous words training data. Two corpora are used for monosemous words training data: Senseval-3 corpus and Yomiuri Shimbun newspaper articles in 2003. All:All performs poorly, even worse than the baseline, which always choose the most frequent semantic class. Comparing to All:All, the performance of Random 1:1 shows great improvement of the measurements. For instance, Accuracy (Exact Match) in Random 1:1 is roughly 10 times better, and F-measure is about 9 times greater than All:All. For At Most P:N, we set up several experiment with different ratios: 1:1, 1:2, 1:3, 2:1 and All:1. At Most All:1 shows the best performance among them with 22% of Accuracy (Exact Match) and 26% of F-measure. However, the performance of At Most Method is worse than Random 1:1. To sum, Random 1:1 achieves the best among three methods using monosemous words training data.

Next, the classifiers trained from polysemous words are evaluated. Polysemous words are excerpted from Senseval-3 corpus. We conduct 5-fold cross validation on the polysemous words task. The performance is better than monosemous words task. Agreement Ratio is over 80%, and Precision is about 68.6%. Moreover, F-Measure is roughly 3 times better than the baseline, which means about 1.4 times greater than the monosemous method. In our experiments, we find 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.