Word Clustering for Collocation-Based Word Sense Disambiguation*

Peng Jin, Xu Sun, Yunfang Wu, and Shiwen Yu

Department of Computer Science and Technology
Institute of Computational Linguistics, Peking University, 100871, Beijing, China
{jandp, sunxu, wuyf, yusw}@pku.edu.cn

Abstract. The main disadvantage of collocation-based word sense disambiguation is that the recall is low, with relatively high precision. How to improve the recall without decrease the precision? In this paper, we investigate a word-class approach to extend the collocation list which is constructed from the manually sense-tagged corpus. But the word classes are obtained from a larger scale corpus which is not sense tagged. The experiment results have shown that the F-measure is improved to 71% compared to 54% of the baseline system where the word-class is not considered, although the precision decreases slightly. Further study discovers the relationship between the F-measure and the number of word-class trained from the various sizes of corpus.

1 Introduction

Word sense disambiguation (WSD) aims to identify the intended sense of a polysemous word given a context. A typical case is the Chinese word "讲" when occurring in "讲真话" ("tell the truth") and "讲实效" ("pay attention to the actual effect"). Correctly sense-tagging the word in context can prove to be beneficial for many NLP applications such as Information Retrieval [6], [14], and Machine Translation [3], [7].

Collocation is a combination of words that has certain tendency to be used together [5] and it is used widely to attack the WSD task. Many researchers used the collocation as an important feature in the supervised learning algorithms: Naïve Bayes [7], [13], Support Vector Machines [8], and Maximum Entropy [2]. And the other researches [15], [16] directly used the collocation to form decision list to deal with the WSD problem.

Word classes are often used to alleviate the data sparseness in NLP. Brown [1] performed automatic word clustering to improve the language model. Li [9] conducted syntactic disambiguation by using the acquired word-class. Och [12] provided an efficient method for determining bilingual word classes to improve statistical MT.

This paper integrates the contribution of word-class to collocation-based WSD. When the word-based collocation which is obtained from sense tagged corpus fails,

* Support by National Grant Fundamental Research 973 Program of China Under Grant No. 2004CB318102.

A. Gelbukh (Ed.): CICLing 2007, LNCS 4394, pp. 267 – 274, 2007.

class-based collocation is used to perform the WSD task. The results of experiment have shown that the average F-measure is improved to 70.81% compared to 54.02% of the baseline system where the word classes are not considered, although the precision decreases slightly. Additionally, the relationship between the F-measure and the number of word-class trained from the various sizes of corpus is also investigated.

The paper is structured as follows. Section 2 summarizes the related work. Section 3 describes how to extend the collocation list. Section 4 presents our experiments as well as the results. Section 5 analyzes the results of the experiments. Finally section 6 draws the conclusions and summarizes further work.

2 Related Work

The underlying idea is that one sense per collocation which has been verified by Yarowsky [15] on a coarse-grained WSD task. But the problem of data spars will be more serious on the fine-grained WSD task. We attempt to resolve the data sparseness with the help of word-class. Both of them are described as follows.

2.1 The Yarowsky Algorithm

Yarowsky [15] used the collocation to form a decision list to perform the WSD task. In his experiments, the content words (i.e., nouns, verbs, adjectives and adverbs) holding some relationships to the target word were treated as collocation words. The relationships include direct adjacency to left or right and first to the left or right in a sentence. He also considered certain syntactic relationships such as verb/object, subject/verb. Since similar corpus is not available in Chinese, we just apply the four co-occurrence words described above as collocation words. Different types of evidences are sorted by the equation 1 to form the final decision list.

$$Abs(Log(\frac{\Pr(Sense_1 \mid Collocation_i)}{\Pr(Sense_2 \mid Collocation_i)})) \quad (1)$$

To deal with the same collocation indicates more than two senses, we adapt to the equation 1. For example, "上 (shang4)" has fifteen different senses as an verb. If the same collocation corresponds to different senses of 上, we use the frequency counts of the most commonly-used sense as the nominator in equation 1, and the frequency counts of the rest senses as the denominator. The different types of evidence are sorted by the value of equation 1. When a new instance is encountered, one steps through the decision list until the evidence at that point in the list matches the current context under consideration. The sense with the greatest listed probability is returned.

The low recall is the main disadvantage of Yarowsky's algorithm to the fine-grained sense disambiguation. Because of the data sparseness, the collocation word in the novel context has little chance to match exactly with the items in the decision list. To resolve this problem, the word clustering is introduced.

2.2 Word Clustering

In this paper, we use an efficient method for word clustering which Och [12] introduced for machine translation. The task of a statistical language model is used to estimate the probability $P(w_1^N)$ of the word sequence $w_1^N = w_1 ... w_N$. A simple approximation of $P(w_1^N)$ is to model it as a product of bi-gram probabilities: $P(w_1^N) = \prod_{i=1}^{N} p(w_i \mid w_{i-1})$. Using the word class rather than the single word, we avoid the use of the most of the rarely seen bi-grams to estimate the probabilities. Rewriting the probability using word classes, we obtain the probability model as follow:

$$P(w_1^N \mid C) := \prod_{i=1}^{N} P(C(w_i) \mid C(w_{i-1})) \bullet P(w_i \mid C(w_i)) \tag{2}$$

Where the function C maps words to w their classes $C(w)$. In this model, we have two types of probabilities: the transition probability $P(C \mid C')$ for class C given its predecessor class C', and the membership probability $P(w \mid C)$ for word w given class C. To determine the optimal word classes $\hat{C}$ for a given number of classes, we perform a maximum-likelihood estimation:

$$\hat{C} = \arg\max_{C} P(w_1^N \mid C) \tag{3}$$

To the implementation, an efficient optimization algorithm is the exchange algorithm [13].It is necessary to set the number of word classes before the iteration.

Two word classes are selected for illustration. First is "花生 (peanut), 大豆 (bean), 棉花 (cotton), 水稻 (rice), 早稻 (early rice), 芒果 (mango), 红枣 (jujube), 柑桔 (orange), 银杏 (ginkgo)". To the target verb "吃" (which have five senses), these nouns can be its objects and indicate the same sense of "吃". Another word class is " 灌溉 (irrigate), 育秧 (raise rice seedlings), 施肥 (apply fertilizer), 播种 (sow), 移植 (transplant), 栽培 (cultivate), 备耕 (make preparations for plowing and sowing)". Most of them indicate the sense "plant" of the target noun "小麦 (wheat)" which has two senses categories: "plant" and "seed". For example, there is a collocation pair "灌溉 小麦" in the collocation list which is obtained from the sense tagged corpus, an unfamiliar collocation pair "备耕小麦" will be tagged with the intended sense of "小麦" because "灌溉" and "备耕" are clustered in the same word-class.

3 Extending the Collocation List

The algorithm of extending the collocation list which is constructed from the sense tagged corpus is quite straightforward. Given a new collocation pair exists in the novel context consists of the target word, the collocation word and the collocation type. If this specific collocation pair is found in the collocation list, we return the sense at the point in this decision list. While the match fails, we replace this collocation word with one of the words which are clustered in the same word-class to match again. The

process is finished when any match success or all words in the word-class are tried. If all words in this word-class fail to match, we let this target word untagged.

For example, "讲政治"(pay attention to the politics), "讲故事"(tell a story) are ordered in the collocation list. But to a new instance "讲笑话"(tell a joke), apparently we can not match the Chinese word "笑话" with any of the collocation word. Searching from the top of the collocation list, we check that "笑话" and "故事" are clustered in the same word-class. So the sense "tell" is returned and the process is ended.

4 Experiment

We have designed a set of experiments to compare the Yarowsky algorithm with and without the contribution of word classes. Yarowsky algorithm introduced in section 2.1 is used as our baseline. Both close test and open test are conducted.

4.1 Data Set

We have selected 52 polysemous verbs randomly with the four senses on average. Senses of words are defined with the Contemporary Chinese Dictionary, the Grammatical Knowledge-base of Contemporary Chinese and other hard-copy dictionaries. For each word sense, a lexical entry includes definition in Chinese, POS, Pinyin, semantic feature, subcategory framework, valence, semantic feature of subject, semantic feature of object, English equivalent and an example sentence.

A corpus containing People's Daily News (PDN) of the first three months of year 2000 (i.e., January, February and March) is used as our training/test set. The corpus is segmented (3,719,951 words) and POS tagged automatically before hand, and then is sense-tagged manually. To keep the consistency, a text is first tagged by one annotator and then checked by other two checkers. Five annotators are all native Chinese speakers. What's more, a software tool is developed to gather all the occurrences of a target word in the corpus into a checking file with the sense KWIC (Key Word in Context) format in sense tags order. Although the agreement rate between human annotators on verb sense annotation is only 81.3%, the checking process with the help of this tool improves significantly the consistency.

We also conduct an open test. The test corpus consists of the news of the first ten days of January 1998. The news corresponding to the first three months of 2000 are used as training set to construct the collocation list. The corpus which is used to word cluster amounts to seven months PDN.

4.2 Experimental Setup

Five-fold cross-validation method is used to evaluate these performances. We divide the sense-tagged three months corpus into five equal parts. In each process, the sense labels in one part are removed in order to be used as test corpus. And then, the collocation list is constructed from the other four parts of corpus. We first use this list to tag test corpus according to the Yarwosky algorithm and set its result as the baseline. After that the word-class is considered and the test corpus is tagged again according to the algorithm described in section 3.

To draw the learning curve, we vary the number of word-class and the sizes of corpus which used to cluster the words. In open test, the collocation list is constructed from the news corresponding to the first three months of year 2000.

4.3 Experiment Results

Table 1 shows the results of close test. It is achieved by 5-fold Cross-Validation with 200 word-clusters trained from the seven months corpus. "Tagged tokens" is referred to the occurrences of the polysemous words which are disambiguated automatically. "All tokens" means the occurrences of the all polysemous words in one test corpus.

We can see the performance of each process is stable. It demonstrates that the word class is very useful to alleviate the data sparse problem.

Table 1. Results with 200 Word Classes Trained from 7 Month Corpus

	Tagged Tokens	All Tokens	Precision	Recall	F-measure
T1	2,346	4237	0.9301	0.5537	0.6942
T2	2,969	4,676	0.9343	0.5766	0.7131
T3	2,362	4,133	0.9306	0.5715	0.7081
T4	2,773	4,721	0.9318	0.5874	0.7206
T5	2,871	4,992	0.9154	0.5751	0.7046
Ave.	2,664	4,552	0.9284	0.5729	0.7081

Table 2 shows the power of word-class. B1 and B2 denote individually the baseline in close and open test. S1 and S2 show the performance with the help of word-classes in these tests. Although the precision decreases slightly, the F-measures are improved significantly. Because in open test, the size of corpus used to training is bigger while the size of corpus used to test is less compared with the corpus in open test, the F-measure is even a bit higher than in close test.

Table 2. Results of Close and Open Test

	Tagged Tokens	All Tokens	Precision	Recall	F-measure
B1	1,691	4,552	0.9793	0.3708	0.5401
S1	2,664	4,552	0.9284	0.5729	0.7081
B2	874	2,325	0.9908	0.3559	0.5450
S2	1,380	2,325	0.9268	0.5935	0.7237

5 Discussion of Results

Fig 1 presents the relationship between the F-measure and the number of word-class trained from the various sizes of corpus. The reasons for errors are also explained.

5.1 Relationship Between F-Measure with Word-Class and Corpus

When we fix the size of the corpus which is used to cluster the word-class, we can see that the F-measure is verse proportional to the number of the word classes. However in our experiments, the precision is proportional to the number of the word classes (this can not be presented in this figure). The reason is straightforward that with the augment of the word classes, there are fewer words in every word-class. So the collocation which comes from test corpus has less chance of finding the word in the decision list belonging to the same word-class.

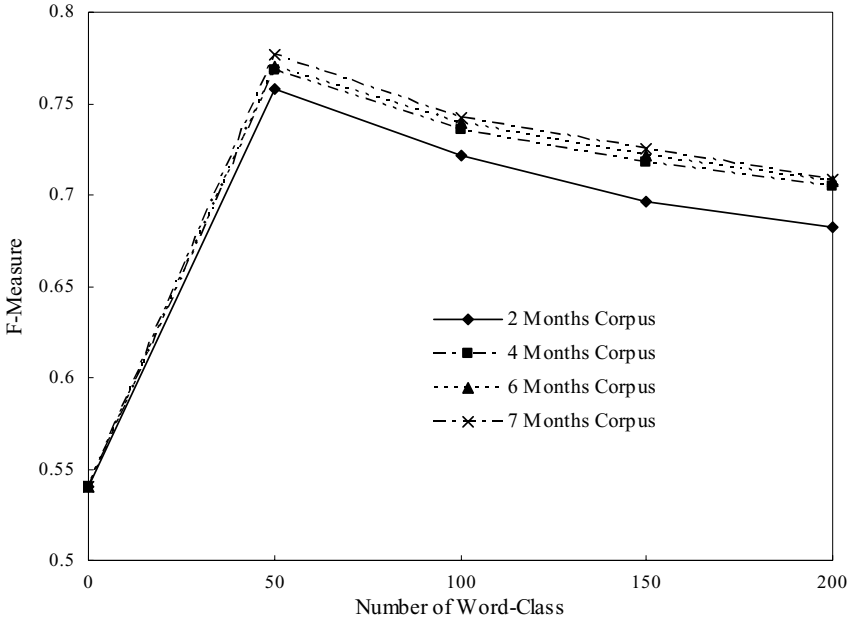

Fig. 1. F-measure at different number of word-class trained from the various sizes of corpus

When we fix the number of word classes, we can see that the F-measure increases with the size of the training corpus. This demonstrates that more data improve the system performance. But the increase rate is less and less. It shows there is a ceiling effect. That is to say, the effect on the performance will be less although more corpuses are trained for clustering the words.

5.2 Error Analysis

Unrelated words are clustered is the main cause of precision decreases. For example, there are two words "牛" (cattle) and "鞭炮" (cracker) are clustered in the same word-class. To the target word "放", "放牛" means "graze cattle" and "放鞭炮" means "fire crackers". To resolve this problem, we should pay much attention to improve the clustering results.

However, the reasonable word-classes also cause errors. Another example is "包饺子" (wrap dumpling) and "包午餐" (offer free lunch) . The word "饺子" (dumpling) and the word "午餐"(lunch) are clustered reasonable because both of them are nouns and related concepts. However, to the target polysemous word "包" , the sense is completely different: the former means "wrap" and the sense of the later is "offer free". It also explains why the WSD system benefits little from the ontology such as HowNet [4].

Although the collocation list obtained from the sense tagged corpus is extended by word classes, the F-measure is still not satisfied. There are still many unfamiliar collocations can not be matched because of the data sparseness.

6 Conclusion and the Future Work

We have demonstrated the word-class is very useful to improve the performance of the collocation-base method. The result shows that the F-measure is improved to 70.81% compared to 54.02% of the baseline system where the word clusters are not considered, although the precision decreases slightly. To open test, the performance is also improved from 54.50% to 72.37%.

This method will be used to help us to accelerate the construction sense tagged corpus. Another utility of word class is used as a feature in the supervised machine learning algorithms in our future research.

We can see that some words are highly sensitive to collocation while others are not. To the later, the performance is poor whether the word-class is used or not. We will further study which words and why they are sensitive to collocation from the perspectives of both linguistics and WSD.

References

1. Brown, P. F., Pietra, V. J., deSouza, P. V., Lai, J. C. and Mercer, R. L. Class-based N-gram Models of Natural Language. Computational Linguistics. 4 (1992) 467-479
2. Chao, G., Dyer, G.M. Maximum Entropy Models for Word Sense Disambiguation. Proceedings of the 19th International Conference on Computational Linguistics. Taipei, Taiwan (2002) 155–161
3. Dagan, D., Itai, A. Word Sense Disambiguation Using a Second Language Monolingual Corpus. Computational Linguistics. 4 (1994) 563–596
4. Dang, H. T., Chia, C., Palmer, M., Chiou, F. D., Rosenzweig J. Simple Features for Chinese Word Sense Disambiguation. Proceedings of the 19th International Conference on Computational Linguistics. Taipei, Taiwan (2002) 204–211
5. Gelbukh, A., G. Sidorov, S.-Y. Han, E. Hernández-Rubio. Automatic Enrichment of a Very Large Dictionary of Word Combinations on the Basis of Dependency Formalism. Proceedings of Mexican International Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence, N 2972, Springer-Verlag, (2004) 430-437
6. Kim, S.B., Seo, H.C., Rim, H.C. Information Retrieval Using Word Senses: Root Sense Tagging Approach, SIGIR'04, Sheffield, South Yorkshire, UK (2004) 258–265

7. Lee, H.A., Kim, G.C. Translation Selection through Source Word Sense Disambiguation and Target Word Selection. Proceedings of the 19th International. Conference on Computational Linguistics, Taipei, Taiwan (2002)
8. Lee, Y. K., Ng, H. T. and Chia, T. K. Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. Proceedings of SENSEVAL-3: Third International Workshop on the Evaluating Systems for the Semantic Analysis of Text, Barcelona, Spain. (2004)
9. Li, H. Word Clustering and Disambiguation Based on Co-occurrence Data. Natural Language Engineering. 8 (2002) 25-42
10. Li W.Y., Lu Q., Li W.J. Integrating Collocation Features in Chinese Word Sense Disambiguation. Proceeding of the Fourth SIGHAN Workshop on Chinese Language Processing (2005) 87–94
11. Martin, S., Liermann, J. and Ney, K. Algorithms for Bigram and Trigram Word Clustering. Speech Communication. 1 (1998) 19-37
12. Och, F. J. An Efficient Method for Determining Bilingual Word Classes. Proceeding of the Ninth Conference of the European Chapter of the Association for Computational Linguistics. (1999) 71-76
13. Pedersen, T. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. Proceeding of the first Annul Meeting of the North American Chapter for Computational Linguistics (2000) 63–69
14. Stokoe, C., Oakes, M.P., Tait, J. Word Sense Disambiguation in Information Retrieval Revisited. Proceeding of the 26th annul International ACM SIGIR conference On research and development in Information retrieval (2003)
15. Yarowsky, D. One Sense Per Collocation, Proceeding of ARPA Human Language Technology workshop. Princeton, New Jersey (1993)
16. Yarowsky, D. Hierarchical Decision Lists for Word Sense Disambiguation, Computers and the Humanities. 1 (2000) 179–186