Word Sense Disambiguation by Semi-supervised Learning

Zheng-Yu Niu[1], Donghong Ji[1], Chew-Lim Tan[2], and Lingpeng Yang[1]

[1] Institute for Infocomm Research,
21 Heng Mui Keng Terrace, 119613 Singapore
{zniu, dhji, lpyang}@i2r.a-star.edu.sg
[2] Department of Computer Science, National University of Singapore,
3 Science Drive 2, 117543 Singapore
tancl@comp.nus.edu.sg

Abstract. In this paper we propose to use a semi-supervised learning algorithm to deal with word sense disambiguation problem. We evaluated a semi-supervised learning algorithm, local and global consistency algorithm, on widely used benchmark corpus for word sense disambiguation. This algorithm yields encouraging experimental results. It achieves better performance than orthodox supervised learning algorithm, such as kNN, and its performance on monolingual benchmark corpus is comparable to a state of the art bootstrapping algorithm (bilingual bootstrapping) for word sense disambiguation.

1 Introduction

In this paper, we address the problem of word sense disambiguation (WSD), which is to assign an appropriate sense to an occurrence of a word in a given context. Many learning algorithms have been proposed or investigated to deal with this problem, including knowledge or dictionary based algorithms, and corpus based algorithms. Corpus based algorithms can be categorized as supervised learning algorithms, weakly supervised learning algorithms [1, 3, 5, 6, 7, 8], and unsupervised learning algorithms. In WSD task, we often face a shortage of labeled training data, but there is a large amount of unlabelled data which can be cheaply acquired. As a result, a great deal of work [1, 3, 5, 6, 7, 8] have been devoted to effective usage of unlabeled data for improving the performance of WSD systems.

Here we use a semi-supervised learning algorithm [9] to perform WSD. Compared with other weakly supervised learning based WSD algorithms, such as bootstrapping or co-training, semi-supervised learning algorithm explores the manifold structure to determine the labels of unlabeled points. Secondly, bootstrapping and co-training require that the class distribution should be fixed during the iteration procedure to avoid degenerate solutions.

This paper is organized as follows. In section 2 we will define feature vector and distance measure for WSD. In section 3 we will describe the semi-supervise

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 238–241, 2005.

learning algorithm used for WSD. Section 4 will give out the experimental results of a semi-supervised learning algorithm on widely used benchmark corpus. In section 5 we will conclude our work and suggest possible improvements.

2 Feature Set and Distance Measure

We use three types of features to capture contextual information: part-of-speech of neighboring words, unordered single words in topical context, and local collocation, following [2]. In later experiment, we conduct a simple feature selection by deleting features if they co-occurred less than three times with ambiguous word.

Let $V = \{v_i\}_{i=1}^{N}$, where v_i represents the feature vector of the i-th occurrence of ambiguous word w, and N is the total number of this ambiguous word's occurrences. Then the distance between symbol-valued vector v_i and v_j can be calculated using a modified Hamming distance:

$$\hat{d}_{ij} = \sum_{k} 1\{v_{ik} == v_{jk}, if\ v_{ik}\ \neq 0\ or\ v_{jk} \neq 0\}. \tag{1}$$

3 Semi-supervised Learning Algorithm

We will give a brief summary of the semi-supervised learning method, local and global consistency algorithm (LGC), introduced in [9].

Given a data set $X = \{x_1, ..., x_l, x_{l+1}, ..., x_n\}$, and a class label set $L = \{1, ..., c\}$, the first l points $x_i (1 \leq i \leq l)$ are labeled as y_i ($y_i \in L$) and remaining points $x_u (l+1 \leq u \leq n)$ are unlabeled. Define $Y \in N^{N \times c}$ with $Y_{ij} = 1$ if point x_i has label j and 0 otherwise. Let $\mathsf{F} \in R^{N \times c}$ denote all the matrices with nonnegative entries. A matrix $F \in \mathsf{F}$ is a matrix that labels all points x_i with a label $y_i = argmax_{j \leq c} F_{ij}$. Define the series $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ with $F(0) = Y$, $\alpha \in (0, 1)$. The entire algorithm is defined as follows:

1. Form the affinity matrix W by $W_{ij} = 1 - exp(-\frac{\hat{d}_{ij}}{2\sigma^2})$ if $i \neq j$ and $W_{ii} = 0$;
2. Compute $S = D^{-1/2} W D^{-1/2}$ with $D_{ii} = \sum_j W_{ij}$ and $D_{ij} = 0$ if $i \neq j$;
3. Compute the limit of series $lim_{t \to \infty} F(t) = F^* = (I - \alpha S)^{-1} Y$. Label each point x_i as $argmax_{j \leq c} F^*_{ij}$. I is $N \times N$ identity matrix.

The regularization framework for this method follows. The cost function associated with the matrix F with regularization parameter $\mu > 0 (\alpha = \frac{1}{1+\mu})$ is defined as:

$$Q(F) = \frac{1}{2}\left(\sum_{i,j=1}^{N} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \right\|^2 + \mu \sum_{i=1}^{N} \|F_i - Y_i\|^2\right). \tag{2}$$

Then the classifying function is

$$F^* = argmin_{F \in \mathsf{F}} Q(F). \tag{3}$$

In later experiments, we let Y be consistent with classification result of a supervised learning algorithm, such as kNN.

Table 1. Accuracy in [3] and accuracy of kNN and LGC with the size of labeled examples as $c \times b$. MB-D denotes monolingual bootstrapping with decision list as the classifier, MB-B monolingual bootstrapping with ensemble of Naive Bayes as the classifier, and BB bilingual bootstrapping with ensemble of Naive Bayes as the classifier

	Accuracies in [3]				Our Results		
Ambiguous Words	Major	MB-D	MB-B	BB	#labeled examples	kNN	LGC
interest	54.6%	54.7%	69.3%	75.5%	60	72.9%	76.6%
line	53.5%	55.6%	54.1%	62.7%	90	56.8%	61.9%

4 Experiments and Results

For comparison of semi-supervised learning algorithm with other weakly supervised learning method, such as bootstrapping algorithm, we evaluated it on widely used benchmark corpus, the corpora of four ambiguous words "hard", "interest", "line", and "serve".

We used kNN (k=1) as baseline, and ran kNN and LGC algorithm using all three types of features on four data sets. The α in LGC algorithm was simply fixed as 0.90. The width of the RBF kernel, σ, was set as 5. After calculation of affinity matrix, we use minimum spanning tree method to construct a connected and sparse graph for LGC.

In [3], they adopted "interest" and "line" corpora as test data. To the word "interest", they used its four major senses. For comparison to their results, we ran kNN and LGC on reduced "interest" corpus (constructed by retaining four major senses) and complete "line" corpus with the number of labeled examples as $c \times b$. c is the number of senses of ambiguous word, and b is the number of examples augmented in each iteration of bootstrapping procedure [3]. $c \times b$ can be deemed as the size of initial labeled examples in their bootstrapping algorithm. All the accuracies were averaged over 10 trials calculated on unlabeled data.

Figure 1 shows the accuracy curves of kNN and LGC versus different percentage of labeled examples. We see that LGC consistently outperformed the orthodox supervised learning algorithm kNN. It indicates that the incorporation of unlabeled data in learning procedure improves the classification results.

Table 1 shows that the performance of LGC algorithm is comparable to the bilingual bootstrapping algorithm (BB) and better than monolingual bootstrapping algorithms (MB-D and MB-B). It should be noted that LGC algorithm utilized only monolingual corpus. However BB achieved their performance with the requirement of two monolingual corpora (English text and Chinese text) and bilingual translation lexicon.

5 Conclusion and Future Work

In this paper we investigated the application of a semi-supervised learning algorithm for word sense disambiguation. In future work, we would like adopt feature

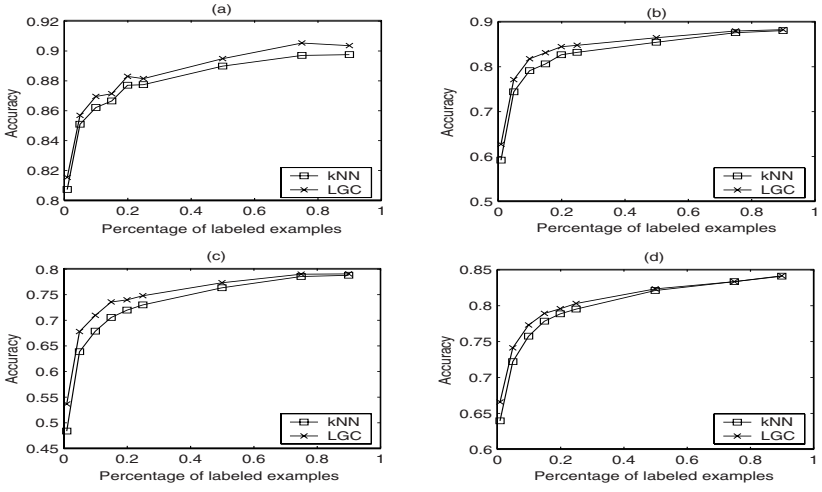

Fig. 1. Accuracy (axis Y) of kNN and LGC versus various percentage of labeled examples (axis X) on (a) hard, (b) interest, (c) line, and (d) serve corpus

clustering technique to deal with high dimensionality problem in feature vector representation of WSD.

References

1. Dagan, I. & Alon I.: Word Sense Disambiguation Using A Second Language Monolingual Corpus. Computational Linguistics, Vol. 20(4), pp. 563-596.(1994)
2. Lee, Y.K. & Ng, H.T.: An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, (pp. 41-48).(2002)
3. Li, H. & Li, C.: Word Translation Disambiguation Using Bilingual Bootstrapping. Computational Linguistics 30(1), 1-22.(2004)
4. Mihalcea R.: Bootstrapping Large Sense Tagged Corpora. Proceedings of the 3rd International Conference on Languages Resources and Evaluations.(2002)
5. Mihalcea R.: Co-training and Self-training for Word Sense Disambiguation. Proceedings of the Conference on Natural Language Learning.(2004)
6. Park, S.B., Zhang, B.T., & Kim, Y.T.: Word Sense Disambiguation by Learning from Unlabeled Data. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.(2000)
7. Su, W., Carpuat, M., & Wu, D.: Semi-Supervised Training of A Kernel PCA-Based Model for Word Sense Disambiguation. Proceedings of the 20th International Conference on Computational Linguistics.(2004)
8. Yarowsky, D.: Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189-196.(1995)
9. Zhou D., Bousquet, O., Lal, T.N., Weston, J., & Schölkopf, B.: Learning with Local and Global Consistency. Advances in Neural Information Processing Systems 16, pp. 321-328.(2003)