LSTC System for Chinese Word Sense Induction

Peng Jin, Yihao Zhang, Rui Sun
Laboratory of Intelligent Information Processing and Application
Leshan Teachers’ College
jandp@pku.edu.cn,yhaozhang@163.com,dram_218@163.com

Abstract

This paper presents the Chinese word sense Induction system of Leshan Teachers’ College. The system participates in the Chinese word sense Induction of task 4 in Back offs organized by the Chinese Information Processing Society of China (CIPS) and SIGHAN. The system extracts neighbor words and their POSs centered in the target words and selected the best one of four cluster algorithms: Simple KMeans, EM, Farthest First and Hierarchical Cluster based on training data. We obtained the F-Score of 60.5% on the training data otherwise the F-Score is 57.89% on the test data provided by organizers.

1. Introduction

Automatically obtain the intended sense of polysemous words according to its context has been shown to improve performance in information retrieval、information extraction and machine translation. There are two ways to resolve this problem in view of machine learning, one is supervised classification and the other is unsupervised classification i.e. clustering. The former is word sense disambiguation (WSD) which relies on large scale, high quality manually annotated sense corpus, but building a sense-annotated corpus is a time-consuming and expensive project. Even the corpus were constructed, the system trained from this corpus show the low performance on different domain test corpus. The later is word sense induction (WSI) which needs not any training data, and it has become one of the most important topics in current computational linguistics.

Chinese Information Processing Society of China (CIPS) and SIGHAN organized a task is intended to promote the research on Chinese WSI. We built a WSI system named LSTC-WSI system for this task. This system tried four cluster algorithms, i.e. Simple KMeans、EM、Farthest First and Hierarchical Cluster implemented by weak 3.7.1 [6], and found Simple KMeans compete the other three ones according to their performances on training data. Finally, the results returned by Simple KMeans were submitted.

2. Features Selection

Following the feature selection in word sense disambiguation, we extract neighbor words and their POSs centered in the target words. Word segmented and POS-tag tool adapted Chinese Lexical Analysis System developed by Institute of Computing Technology. No other resource is used in the system. The window size of the context is set to 5 around the ambiguous word. The neighbor words which occur only once were removed. Each sample is represented as a vector, and feature form is binary: if it occurs in is 1 otherwise is 0.

3. Clusters Algorithms

Four cluster algorithms were tried in our system. I will introduce them simply in the next respectively.

K-means clustering [1] is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The main idea is to define *k* centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other.

EM algorithm[2] is a method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. EM is an iterative method which alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables, and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step.

The Farthest First algorithm [3] is an implementation of the “Farthest First Traversal Algorithm” by Hochbaum and Shmoys (1985). It finds fast, approximate clusters and may be useful as an initialiser for k-means.

A hierarchical clustering [4] is the guarantee that for every k, the induced k clustering has cost at most eight times that of the optimal k-clustering. A hierarchical clustering of n data points is a recursive partitioning of the data into 2, 3, 4, . . . and finally n, clusters. Each intermediate clustering is made more fine-grained by dividing one of its clusters.

4. Development

4.1 Evaluation method

We consider the gold standard as a solution to the clustering problem. All examples tagged with a given sense in the gold standard form a class. For the system output, the clusters are formed by instances assigned to the same sense tag. We will compare clusters output by the system with the classes in the gold standard and compute F-score as usual [5]. F-score is computed with the formula below.

Suppose C_r is a class of the gold standard, and S_i is a cluster of the system generated, then

$$F-Score(C_r, S_i) = 2*P*R/(P+R) \quad (1)$$

$$p = \frac{\text{the number of correctly labeled examples for a cluster}}{\text{total cluster size}}$$

$$\mathrm{R} = \frac{\text{the number of correctly labeled examples for a cluster}}{\text{total cluster size}}$$

Then for a given class Cr,

$$F-score(C_r) = \max_{S_i}(F-score(C_r, S_i))$$

$$F-Score = \sum_{r=1}^{c} \frac{nr}{n} FScore(C_r) \quad (2)$$

where *c* is total number of classes, n_r is the size of class C_r, and n is the total size. Participants will be required to induce the senses of the target word using only the dataset provided by the organizers.

4.2 Data Set

The organizers provide 50 Chinese training data of SIGHAN2010-WSI-SampleData. The training data contain 50 Chinese words; each word has 50 example sentences, and gives each word the total number of sense. The total number of sense is ranging from 2 to 21, but more cases are 2. In order to facilitate the team participating in the contest to do experiment, the organizers also provide answer to each word.

In order to evaluating the system’s performance of all participating team, the organizers provide 100 test word and each word have 50 example sentences, the system of each participating team need to run out the results which the organizers need.

4.3 System Setup

We developed the LSTC-WSI system based on Weka. Firstly, we implemented the evaluation algorithm described in section 4.1. Then, the instances were represented as vectors according to the feature selection. Thirdly, four cluster algorithms from Weka were tried and set different thresholds for feature frequency. Because of paper length constraints, we could not list all the experience data we get. Table 1 listed system performance when frequency threshold set two and without POS information.

Table 1: The Performance on test data

Target word	Simple Kmeans	EM	Farthest First	Hierar chical
暗淡	0.618	0.680	0.538	0.649
把握	0.404	0.365	0.400	0.327
保安	0.711	0.557	0.672	0.636
保管	0.626	0.700	0.536	0.570
报销	0.571	0.555	0.572	0.573
背离	0.789	0.596	0.680	0.548
比重	0.704	0.617	0.704	0.682
便宜	0.568	0.495	0.461	0.583
标兵	0.5679	0.679	0.625	0.688
病毒	0.601	0.590	0.648	0.603
补贴	0.578	0.554	0.662	0.616
哺育	0.621	0.537	0.615	0.627
材料	0.560	0.429	0.466	0.527
采购	0.627	0.537	0.643	0.603
参加	0.610	0.538	0.643	0.638
草包	0.643	0.607	0.648	0.632
程序	0.615	0.545	0.662	0.603
澄清	0.621	0.616	0.615	0.658
吃饭	0.538	0.583	0.569	0.609
冲洗	0.603	0.540	0.632	0.569
冲撞	0.653	0.557	0.657	0.603
充电	0.627	0.622	0.652	0.690
出口	0.421	0.438	0.454	0.453
初二	0.609	0.528	0.583	0.627
春秋	0.634	0.667	0.486	0.652
戳穿	0.574	0.546	0.577	0.584
打	0.462	0.429	0.518	0.501
打断	0.661	0.584	0.584	0.602
打开	0.430	0.501	0.549	0.418
打破	0.596	0.644	0.647	0.654
打气	0.614	0.580	0.672	0.708
大军	0.666	0.600	0.615	0.595
大陆	0.638	0.590	0.540	0.678
大气	0.841	0.734	0.662	0.618
大人	0.613	0.562	0.670	0.568
单纯	0.635	0.617	0.646	0.649
导师	0.603	0.594	0.615	0.577
东北	0.644	0.635	0.661	0.560
东方	0.599	0.595	0.624	0.638
东西	0.588	0.575	0.587	0.508
动力	0.699	0.723	0.673	0.643
杜鹃	0.585	0.596	0.666	0.603
断交	0.643	0.639	0.666	0.656
断气	0.624	0.537	0.663	0.608
扼杀	0.632	0.525	0.629	0.617
发动	0.451	0.472	0.490	0.477
发展	0.613	0.625	0.6723	0.625
翻身	0.601	0.640	0.646	0.661
反射	0.591	0.585	0.663	0.639
调动	0.536	0.505	0.477	0.532

We tried two ways for feature selection: the frequency of features and neighbor words’ POS were taken into account or not. Table 2 shows the average performance on the test data via varying the parameter setting. Observing the results returned by Hierarchical cluster is very imbalance, we set the options “-L WARD” in order to balance the number.

Table 2: The Average Performance of 50 Training Data

Features	Simple Kmeans	EM	Farthest First	Hierarchical
Word, Windows 5	0.555	0.566	0.607	0.558
Word, Windows 5, Frequency 1	0.583	0.567	0.599	0.582
Word, Windows 5, Frequency 2	0.605	0.575	0.605	0.598
Word, Windows 5, Frequency 3	0.598	0.590	0.600	0.599
Word+POSs, Windows 5	0.562	0.582	0.618	0.569
Word+POSs, Windows 5, Frequency 1	0.589	0.580	0.610	0.594
Word+POSs, Windows 5, Frequency 2	0.589	0.580	0.610	0.594

Compared with the average performance of the 50 test data, we find the performance is best[1] when considering word only and setting the frequency is two at the same time simple KMeans was adapted. So, we use the same parameters setting and clustered the test data by simple KMeans. As table 2 shows, the F-Score is 60.5% on training data. But on test data, our system’s F-Score is 57.89% officially evaluated by task organizers.

5. Conclusion and Future Works

Four cluster algorithms are tried for Chinese word sense induction: Simple KMeans, EM, Farthest First and Hierarchical Cluster. We construct different feature spaces and select out the best combination of cluster and feature space. Finally, we apply the best system to the test data.

In the future, we will look for better cluster algorithms for word sense induction. Furthermore, we observe that it is different from word sense disambiguation, different part of speech will cause the polysemy. We will make use of this character to improve our system.

Acknowledgements

This work is supported by the Open Projects Program of Key Laboratory of Computational Linguistics(Peking University)，Ministry of Education, Grant No. KLCL-1002. This work is also supported by Leshan Teachers’ College, Grant No. Z1046.

References

[1] Dekang Lin, Xiaoyun Wu. Phrase Clustering for Discriminative Learning. Proceedings of ACL ,2009.

[2]Neal R, & Hinton G. A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in Graphical Models, 89, 355–368.

[3] Jon Gibson, Firat Tekiner, Peter Halfpenny. NCeSS Project: Data Mining for Social Scientists. Research Computing Services, University of Manchester, U.K.

[4] Sanjoy Dasgupta, Philip M. Long. Performance guarantees for hierarchical clustering. Journal of Computer and System Sciences, 555–569, 2005.

[5] Eneko Agirre, Aitor Soroa. Semeval-2007 Task 02:Evaluating Word Sense Induction and Discrimination Systems. Proceedings of SemEval-2007, pages 7–12, 2007.

[6] http://www.cs.waikato.ac.nz/ml/weka/

[1] Although “Farthest First” got the highest score, the results of “Farthest First” are too imbalance.