K-means and Graph-based Approaches for Chinese Word Sense Induction Task

Lisha Wang **Yanzhao Dou** **Xiaoling Sun** **Hongfei Lin**
Computer Science Department
Dalian University of Technology
{lisawang0110,yanzhaodou}@gmail.com
xlsun@mail.dlut.edu.cn hflin@dlut.edu.cn

Abstract

This paper details our experiments carried out at Word Sense Induction task. For the foreign language (especially English), there have been many studies of word sense induction (WSI), and the approaches and the techniques are more and more mature. However, the study of Chinese WSI is just getting started, and there has not been a better way to solve the problems encountered. WSI can be divided into two categories: supervised manner and unsupervised manner. But in the light of the high cost of supervised manner, we introduce novel solutions to automatic and unsupervised WSI. In this paper, we propose two different systems. The first one is called K-means-based Chinese word sense induction in an unsupervised manner while the second one is graph-based Chinese word sense induction. In the experiments, the first system has achieved a 0.7729 Fscore on average while the second one has achieved a 0.6067 Fscore.

1 Introduction

No matter in which kind of language, ambiguous terms always exist, Chinese is also not exceptional. According to statistics, although the percent of ambiguous terms in Chinese dictionary is only about 14.8%, the frequency of them is up to 42% in Chinese corpora. This phenomenon shows that the number of ambiguous terms is small in natural language, but their frequency is extremely high. Therefore, the key step in natural language processing (NLP) is to identify the specific meaning of a given target word according to its context. In this task, the input to a WSI algorithm is the sentences including the same ambiguous term, and our task is to cluster these sentences into different categories according to the meanings of this ambiguous term in every sentence. The study of WSI is earlier abroad and there has been a set of well-developed theories by now. However, the start of studying Chinese WSI is later and we need to find a better and appropriate way for Chinese WSI. In this paper, we develop two different systems. The first one is based on K-means algorithm which optimizes the initial centers and a Chinese thesaurus - TongYiCi CiLin is used to solve the problem of sparseness of a sentence's vector. The second one is a combination approach of graph-based clustering and K-means algorithm. We choose Chinese Whisper as the graph-based clustering approach.

2 K-means-based Chinese WSI in an Unsupervised Manner

Since the number of total meanings of an ambiguous term has been given in this task, our goal is to cluster those sentences which contain the same ambiguous term in an unsupervised manner. In this condition our primary problem is the selection of a suitable clustering method.

Clustering algorithms are generally divided into two categories, namely partitioning clustering algorithm and hierarchical clustering algorithm. Partitioning clustering algorithm is usually selected when the number of final clusters is known. Consequently, we need to input a parameter K as the number. Typical partitioning clustering algorithm contains K-means, K-medoids, CLARANS and so on. Among them, K-means clustering algorithm is

widely used and relatively simple. Hierarchical clustering algorithms are not required to input any parameters, which is their advantage compared to partitioning clustering algorithms. Typical hierarchical clustering algorithms contain BIRCH algorithm, DBSCAN algorithm, CURE algorithm and so on.

Considering the characters of WSI (e.g. the total number of a target word's sense has been given in advance), we should select partitioning clustering algorithm. In addition, considering the quality, the performance, and the degree of difficulty while being implemented among all kinds of partitioning clustering algorithms, we finally decide to use k-means algorithm, but we have improved it in order to obtain better clustering performance.

2.1 Traditional K-means Algorithm

The process of traditional K-means algorithm is as follows:

Input: the number of clusters (k) and n-data objects.

Output: k-clusters. The clusters should satisfy the following requirements: the objects in the same cluster have higher similarity, while the objects in different clusters have lower similarity.

The process steps:

(1) Choose k-objects randomly as initial cluster centers;
(2) Repeat;
(3) Compute each object's distance to each cluster's center, then object is assigned to the most similar cluster;
(4) Update the center of each cluster;
(5) Until the changes of all clusters' centers are smaller than a given threshold.

2.2 The Advantages and Disadvantages of Traditional K-means Algorithm

The greatest advantage of traditional K-means algorithm is comparatively simple. In addition, its implementation is quick, effective and does not need a high cost. However, from the idea and processes as illustrated, we can see that the traditional K-means algorithm has two disadvantages: (1) an over-reliance on the selection of initial points. If the selection is improper (e.g. just select some points in the same cluster as the initial points), the result will be poor. (2) the clustering results are sensitive to "noise" and isolated points. Small amounts of such data can greatly decrease the precision.

2.3 Maximum Distance-based Selection of the Initial Centers

Given the above considerations, this paper introduces a maximum distance-based selection of the initial centers.

The selection of initial centers has a great impact on the result in traditional K-means clustering algorithm. If the selection is more appropriate, then the result will be more reasonable, while the convergence rate will be faster. So we hope that the initial centers should be dispersed as far as possible, not be placed in a particular one or limited several clusters. The best selection should be that K-initial points belong to K-different clusters. In order to achieve this goal, we use the maximum distance. Specific method is processed as follows: Firstly, select an arbitrary point as the first cluster's center from the n-data objects, and then calculate its distance to the remaining (n-1) data objects, to find out the farthest point away from it as the second cluster's initial center. Secondly, calculate the distances of the remaining (n-2) data objects to both the clusters' center, compute the average of the two values, and then select the point with the maximum average value as the initial cluster center of the third. We repeat this process until find out K-initial points.

From Figure 1 we can see that the result of improved algorithm is much better than traditional K-means algorithm.

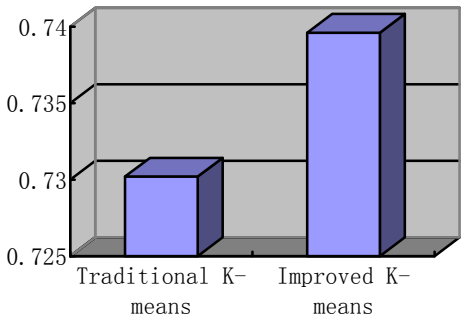

Figure 1: The results of traditional K-means algorithm and improved K-means algorithm.

2.4 The Context of the Target Words

During the process of WSI, we believe that the specific meaning of an ambiguous term is

determined by its context, that is to say, those target words with similar context should have similar meaning in theory. So the first step we have to do is to establish all sentences' context around a target word (we have carried out Chinese word segmentation and stop word filtering to these sentences). As the K-means algorithm can only handle numerical data, we change the context into numerical format and then represent it using VSM. But how to determine the window size of the context is necessary to be further discussed.

In this paper, we use the information gain proposed by Lu et al. to achieve the goal of determining the window size. We count out 3000 high frequency words from the given test set in this task, every word as a class, and then calculate the statistical uncertainty of the whole system (entropy), namely H (D) in equation (3); The next step is to calculate the uncertainty of the whole system on the premise of knowing relative position, namely the $\Sigma_{v \in Vp} P(v) \times H(D|v)$ in equation (3); Difference between the two values is just the amount of information on the entire system provided by this relative position. The amount of information (i.e. information gain) is the weight of this position in the whole system. In this way we can determine the windows size by the weight.

$$IG_p = H(D) - \Sigma_{v \in Vp} P(v) \times H(D \mid v) \qquad (1)$$

where

$$H(D) = -\Sigma_{d \in D} P(d) \times \log_2 P(d) \qquad (2)$$

$$P(d) = \frac{\left| fre(d) \right|}{\sum_i \left| fre(d_i) \right|} \qquad (3)$$

$\sum_i \left| fre(d_i) \right|$ is the sum of frequency of the 3,000 high frequency words appearing in the corpus; $\left| fre(d) \right|$ is the occurrence frequency of term d in the corpus.

We first separately select eight words before and after the target word in a sentence to constitute the context, expressed as the following form:

$<wd_{-8}, wd_{-7}, wd_{-6}, wd_{-5}, wd_{-4}, wd_{-3}, wd_{-2}, wd_{-1}$, ***focus-word***, $wd_{+1}, wd_{+2}, wd_{+3}, wd_{+4}, wd_{+5}, wd_{+6}, wd_{+7}, wd_{+8}>$

Table 1 Information gain of every position of context

Left context		Right context	
Position	Information gain	Position	Information gain
wd−1	3.979 875	wd+1	4.005 737
wd−2	2.800 943	wd+2	2.931 834
wd−3	2.183 287	wd+3	2.287 020
wd−4	1.709 504	wd+4	1.810 530
wd−5	1.361 637	wd+5	1.437 952
wd−6	1.074 606	wd+6	1.137 979
wd−7	0.304 546	wd+7	0.821 330
wd−8	0.298 992	wd+8	0.419 472

The amount of information provided by each position is presented in Table 1. According to the information gain in this table we can draw a conclusion: the closer a term to the target word, the more greatly it contributes to its meaning, and the ability to describe the target word's meaning decreases with the term's distance increasing to the focus-word. Because those words whose distance to the target word is more than 6 words contribute less to the meaning of the target word, we separately select at most 6 words before and after the target word as context.

2.5 Sparsity Problem

For those sentences containing the same target word we can respectively establish their context, and then merge the same words in those context to form a n-dimension space .Then we establish the vector model for each sentence. We have experimented with two different methods to represent weight in the vector: one is TF*IDF which is conventional and widely used in practice and the other one is Boolean. However, from Figure 2 we can see that the result of Boolean method is better. Analyzing the reasons, we can infer that the decisive role of a word to the target word is relevant whether the word appears or not, and has nothing to do with the times of appearance. Consequently, we select Boolean method to represent weight in the vector: if a word in the space appears in this sentence, the weight of this position in sentence's vector is 1, otherwise is 0.

Now we find a problem which should be solved: vector sparsity problem. In a few hundreds dimension vector space, a sentence contains only several limited words, thus the

vector is highly sparse. As we analyzed, there are two main causes: 1). The length of a sentence is too short, so the number of words contained by it is few. 2). When merging those words in the context of a target word, we don't take into account the semantic similarity between them. We know that if the vector is too sparse, the result will have large errors, even two sentences which should have belonged to the same class are divided into different clusters.

We can not solve the problem caused by the first factor, but we can improve the second one. In this paper we introduce TongYiCi CiLin from HIT to compress the vector's dimension.

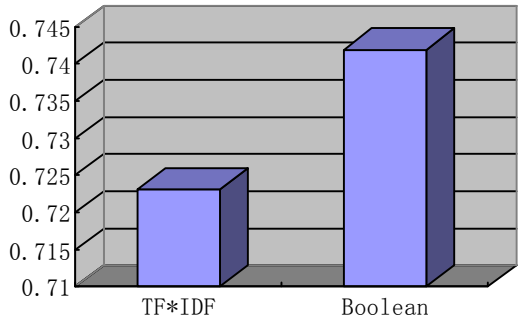

Figure 2: The results of two different methods to represent weight in the vector. Here we have selected improved K-means algorithm to optimize the initial centers.

2.6 Experiments

The whole process of experiment is as follows:

(1) Segment all sentences and filter stop-words for a given data set;
(2) Extract respectively six words before and after the focus-word from those sentences containing the same target words, and then use TongYiCi CiLin to merge these words into a lower n-dimension space;
(3) Establish the vector model for each sentence in this space;
(4) Cluster those sentences containing the same target words with maximum distance-based K-means algorithm proposed in this paper.

This experimental method is based on the following assumption: the similarity of target words' context determines the similarity of their meanings. In the framework of this assumption, we construct the context vector of each sentence, and then cluster those sentences containing the same target word.

In the experimental result, we have achieved 0.7729 Fscore on 100 ambiguous words.

3 Graph-based Chinese Word Sense Induction

In this system, we use a combination of graph-based clustering and K-means algorithm. At first we use Chinese Whisper to cluster the words in the corpus and the clustering result can be considered as an artificial synonyms dictionary. Secondly we construct corpus vectors using different methods, and now the vector dimension is decreased to the number of clusters. At last we cluster the vectors with the help of K-means algorithm.

3.1 Chinese Whisper Method

Many researches on WSI are based on word co-occurrence. The approach proposed by Chris Biemann has a wide range of applications, including language separation, acquisition of word class, word sense induction and so on. Chinese Whisper, which comes from a game called "Chinese Whisper", is a method used for graph clustering and its process is as follows:

(1) All nodes belong to different classes at the beginning;
(2) The nodes are processed for a small number of iterations and inherit the strongest class in the local neighborhoods. The sum of edge weights is maximal in this class.
(3) While updating a vertex i, each class, e.g. cl, receives a score equal to the weight of edge (i, j), here j has been assigned to cl. The maximum score determines the strongest class. If there are more than 2 strongest classes, only one is chosen randomly.
(4) While clustering, there are two important parameters to select: convergence constant and the iterations. From this we can see that this method has a great flexibility on parameter selection, and its clustering result is totally determined by the parameters.

In Chris Biemann's paper, using Chinese Whisper, his experiment about WSI based on British National Corpus (BNC) achieved 92.2% precision in adjective, 90% precision in noun, and 77.6% precision in verbs. Ioannis P.

Klapaftis and Suresh Manandhar use Chinese Whisper method for clustering and their experiment based on BNC achieved 81.1% FScore after trying 72 different parameters.

3.2 Graph Construction

When we construct the graph, every word is considered as a node in the graph and the weight of edge eij is measured by co-cocurence times of word i and word j. However, if we just use this method to construct the graph, the graph is very sparse. We use some methods proposed by IP Klapaftis to add new edges:

(1) Associate a vertex vector VC_i containing the vertices, which share an edge with vertex i in the graph.
(2) Calculate the similarity between each vertex vector VC_i and each vertex vector VC_j, here we use Jaccard similarity coefficient (JC) as a similarity measure:

$$JC(VC_i, VC_j) = \frac{|VC_i \bigcap VC_j|}{|VC_i \bigcup VC_j|} \quad (4)$$

Two nodes c_i and c_j are mutually similar if c_i is the most similar node to c_j and the other way round.
(3) Two mutually similar nodes c_i and c_j are clustered with the result that an occurrence of a node ck with one of c_i, c_j is also counted as an occurrence with the other node.

3.3 Experiments

K-means algorithm has a good performance for small corpus, but when the corpus size is too big, vector dimension will increase rapidly. So At first we use Chinese Whisper to cluster the words in the corpus after preprocessing, such as splitting the sentences, filtering stopwords and selecting context. Secondly we construct corpus vectors with VSM, and now the vector dimension is decreased to the number of clusters. At last we cluster the vectors using K-means algorithm analogous to the first system.

The choice of parameters is an important factor in Chinese Whisper and different parameters will result in different clusters. In this experiment we use batch process method in order to select the best parameters on training set. We select a group of parameters: convergence constant is from 0 to 1 and the step length is 0.1; iterations is from 1 to 30 and the step length is 1, which depends on the size of corpus. The process of experiment is as follows:

(1) Get a pair of parameters from the parameter group, cluster the corpus using Chinese Whisper, and then remove this pair of parameter from the parameter group.
(2) Construct vectors using the result of step (1).
(3) Cluster the vectors using K-means.
(4) The results are as the following two tables. From table 2 and table 3 we can see that if we use JC method to add new edges, the precision has a great improvement.

In the experimental result, we have achieved 0.6067 Fscore on 100 ambiguous words with the parameters: 0.8 and 12.

Table 2 Experimental results without using JC method

converge constance	iterations	precision (Boolean)
0.1	11	0.6119
0.1	15	0.6175
0.3	15	0.6210
0.5	15	0.6188

Table 3 Experimental results using JC method

converge constance	iterations	precision (Boolean)
0.6	17	0.6211
0.6	15	0.6251
0.7	11	0.6261
0.7	15	0.6287
0.8	12	0.6391
0.9	14	0.6192
1.0	16	0.6389
1.0	15	0.6300

4 Conclusion

In this paper, we propose two different systems for the task of Chinese WSI.

The result of the first system which is based on an improved K-means algorithm shows the proposed idea is feasible, and the precision is guaranteed. However, some problems still exist and need further to be resolved:

(1) The extended particle size of a word's synonym while using TongYiCi CiLin. If particle size is too large, the "noise" affects

the accuracy of the result; If particle size is too small, time complexity of the algorithm will increase drastically.

(2) The selection of initial centers in K-means algorithm remains to be further optimized. In addition to avoid the selected initial centers placing in one or several clusters, the problem of "noise" and isolated data need to be considered.

(3) The instability of this method. While we have got better results on most of ambiguous terms, but for those words with very many meanings, the induction effect is not so good. The reasons should be further analyzed and the solutions should be found out.

The result of the second system which is based on graph clustering shows that this method has a good performance in decreasing vector dimension. However, the number of clusters is too small, which made the performance of K-means algorithm poor.

Chinese Whisper has a good performance in WSI, but this is the first time to combine it with K-means together, thus there are lots of problems to be solved. As we have investigated, some methods can be used to improve the performance in the future work:

(1) Use a pair of words as a vertex of the graph instead of using a single word.

(2) Instead of using co-occurrence times as the weight of an edge, we can use conditional probability.

(3) Constrain words pair which can filter out some “noise”, i.e. only use those words whose co-occurrence times is greater than a given value threshold.

Acknowledgments

This work is supported by grant from the Natural Science Foundation of China (No.60673039 and 60973068), the National High Tech Research and Development Plan of China (No.2006AA01Z151), National Social Science Foundation of China (No.08BTQ025), the Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry and The Research Fund for the Doctoral Program of Higher Education (No.20090041110002).

References

Lu Song, Bai Shuo, and Huang Xiong. 2002. An Unsupervised Approach to Word Sense Disambiguation Based on Sense-Words in Vector Space Model. *Journal of Software*, 13(06): 1082-1089.

Stefan Bordag. 2004. Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. *In: Proceedings of HLT-NAACL, Workshop on Computational Lexical Semantics*, pages 137-144, Boston, Massachusetts.

Ioannis P.Klapaftis and Suresh Manandhar. 2008. Word Sense Induction Using Graphs of Collocations. *In: Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence, Frontiers in Artificial Intelligence and Applications*, pages 298-302, United Kingdom.

Chris Biemann. 2006. Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. *In: Proceedings of the HLT-NAACL 2006 Workshop on Textgraphs*, New York, USA.