Triplet-Based Chinese Word Sense Induction

Zhao Liu
School of Computer Science
Fudan University
Shanghai, China
ZLiu.fd@gmail.com

Xipeng Qiu
School of Computer Science
Fudan University
Shanghai, China
xpqiu@fudan.edu.cn

Xuanjing Huang
School of Computer Science
Fudan University
Shanghai, China
xjhuang@fudan.edu.cn

Abstract

This paper describes the implementation of our system at CLP 2010 bake-off of Chinese word sense induction. We first extract the triplets for the target word in each sentence, then use the intersection of all related words of these triplets from the Internet. We use the related word to construct feature vectors for the sentence. At last we discriminate the word senses by clustering the sentences. Our system achieved 77.88% F-score under the official evaluation.

1 Introduction

The goal of the CLP 2010 bake-off of Chinese word sense induction is to automatically discriminate the senses of Chinese target words by the use of only un-annotated data.

The use of word senses instead of word forms has been shown to improve performance in information retrieval, information extraction and machine translation. Word Sense Disambiguation generally requires the use of large-scale manually annotated lexical resources. Word Sense Induction can overcome this limitation, and it has become one of the most important topics in current computational linguistics research.

In this paper we introduce a method to solve the problem of Chinese word sense induction.For this task, Firstly we constructed triplets containing the target word in every instance, then searched the intersection of all the three words from the Internet with web searching engine and constructed feature vectors.Then we clustered the vectors with the sIB clustering algorithm and at last discriminated the word senses.

This paper is organized as following: firstly we introduce the related works. Then we talk about the methods in features selection and clustering. The method of evaluation and the result of our system is following. At last we discuss the improvement and the weakness of our system.

2 Related Works

Sense induction is typically treated as a clustering problem, by considering their co-occurring contexts, the instances of a target word are partitioned into classes. Previous methods have used the first or second order co-occurrence (Pedersen and Bruce, 1997; Schütze, 1998), parts of speech, and local collocations (Niu et al., 2007). The size of context window is also various, it can be as small as only two words before and after the target words. It may be the sentence where the target word is in. Or it will be 20 surrounding words on either side of the target words and even more words.

After every instance of the target word is represented as a feature vector, it will be the input of the clustering methods. Many clustering methods have been used in the task of word sense induction. For example, k-means and agglomerative clustering (Schütze, 1998). sIB (Sequential Information Bottleneck) a variation of Information Bottleneck is applied in (Niu et al., 2007). In (Dorow and

Widdows, 2003) Graph-based clustering algorithm is employed that in a graph a node represents a noun and two nodes have an edge between them if they co-occur in list more than a given number of times. A generative model based on LDA is proposed in (Brody and Lapata, 2009).

In our method, we use the triplets (Bordag, 2006) and their intersections from the Internet to construct the feature vectors then sIB is used as the clustering method.

3 Feature Selection

Our method select the features of the words similar to (Bordag, 2006) is also using the triplets. In Chinese there are no natural separators between the words as English, so the first step in Chinese language processing is often the Chinese word segmentation. In our system we use the FudanNLP toolkit[1] to split the words.

At the first stage, we split the instance of the target word and filter out the numbers, English words and stop words from it. So we get a sequence of the words. Then we select two words before the target and another two words after it. If there are no words before or after then leave it empty. After that we enumerate two words from the selected four words to construct a triplets together with the target words. So we can get several triplets for every instance of the target. Because the faulty of Chinese word segmentation and some special target word for example a single Chinese character as a word, there are some errors finding the position of the target words. If the word is a single Chinese character and the toolkit combine it with other Chinese characters to be a word, we will use that word as the target instead of the character to construct the triplets.

The second stage is obtaining corpus from the Internet. For every triplet we search the three words sequence in it with a pair of double quotation marks in Baidu web searching engine[2]. It gives the snippets of the webs which have all the three words in it. We select the first 50 snippets of each triplets. If the number of the snippets is less than 50, we will ignore that triplet. For some rare words the snippets searched from the Internet for all the triplets of the instance is less than 50. In that situation we will search the target word and another one co-occurring word in the searching engine to achieve enough snippets as features. After searching the triplets we select the first three triplets (or doublets) with largest amount of the webs searched by the searching engine. For every instance there are three or less triplets (or doublets) and we have obtained many snippets for them. After segmenting and filtering these snippets we use the bag of words from them as the feature for this instance.

The last stage of feature selection is to construct the feature vector for every instances containing the target word. In the previous stage we get a bag of words for each instance. For all the instances of one target word we make a statistic of the frequence of each word in the bags. In our system we select the words whose frequence is more than 50 as the dimensions for the feature vectors. From the tests we find that when this thread varies from 50 to 120 the result of our system is nearly the same, but outside that bound the result will become rather bad. So we use 50 as the thread. After constructing the dimension of that target word, we can get a feature vector for each instance that at each dimension the number is the frequence of that word occurs in that position.

We obtain the feature vectors for the target words by employing these three stage. The following work is clustering these vector to get the classes of the word senses.

4 The Clustering Algorithm

There are many classical clustering methods such as k-means, EM and so on. In (Niu et al., 2007) they applied the sIB (Slonim et al., 2002) clustering algorithm at SemEval-2007 for task 2 and it achieved a quite good result. And at first this algorithm is also introduced

[1] http://code.google.com/p/fudannlp/

[2] http://www.baidu.com

for the unsupervised document classification problem. So we use the sIB algorithm for clustering the feature vectors in our system.

Unlike the situation in (Niu et al., 2007), the number of the sense classes is provided in CLP2010 task 4. So we can apply the sIB algorithm directly without the sense number estimation procedure in that paper. sIB algorithm is a variant of the information bottleneck method.

Let d represent a document, and w represent a feature word, $d \in D$,$w \in F$. Given the joint distribution $p(d, w)$, the document clustering problem is formulated as looking for a compact representation T for D, which reserves as much information as possible about F. T is the document clustering solution. For solving this optimization problem, sIB algorithm was proposed in (Slonim et al., 2002), which found a local maximum of $I(T, F)$ by: given a initial partition T, iteratively drawing a $d \in D$ out of its cluster $t(d)$, $t \in T$, and merging it into t^{new} such that $t^{new} = argmax_{t \in T}\mathbf{d}(d, t)$. $\mathbf{d}(d, t)$ is the change of $I(T, F)$ due to merging d into cluster t^{new}, which is given by

$$\mathbf{d}(d, t) = (p(d)+p(t))JS(p(w|d), p(w|t)). \quad (1)$$

$JS(p, q)$ is the Jensen-Shannon divergence, which is defined as

$$JS(p, q) = \pi_p D_{KL}(p||\overline{p}) + \pi_q D_{KL}(q||\overline{p}), \quad (2)$$

$$D_{KL}(p||\overline{p}) = \sum_y p \log \frac{p}{\overline{p}}, \quad (3)$$

$$D_{KL}(q||\overline{p}) = \sum_y q \log \frac{q}{\overline{p}}, \quad (4)$$

$$\{p, q\} \equiv \{p(w|d), p(w|t)\}, \quad (5)$$

$$\{\pi_p, \pi_q\} \equiv \{\frac{p(d)}{p(d)+p(t)}, \frac{p(t)}{p(d)+p(t)}\}, \quad (6)$$

$$\overline{p} = \pi_p p(w|d) + \pi_q p(w|t). \quad (7)$$

In our system we use the sIB algorithm in the Weka 3.5.8 cluster package to cluster the feature vectors obtained in the previous section. The detailed description of the sIB algorithm in weka can refer to the website [3]. And the parameters for this Weka class is that: the number of clusters is the number of senses provided by the task, the random number seed is zero and the other parameters like maximum number of iteration and so on is set as default.

5 CLP 2010 Bake-Off of Chinese Word Sense Induction

5.1 Evaluation Measure

The evaluation measure is described as following:

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 (the sense tag with the highest weight for that instance). We will compare clusters output by the system with the classes in the gold standard and compute F-score as usual. 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

1. $F - score(C_r, S_i) = \frac{2*P*R}{P+R}$
2. P =the number of correctly labeled examples for a cluster/total cluster size
3. R =the number of correctly labeled examples for a cluster/total class size

Then for a given class C_r,

$$FScore(C_r) = \max_{S_i}(F - score(C_r, S_i)) \quad (8)$$

Then

$$FScore = \sum_{r=1}^{c} \frac{n_r}{n} FScore(C_r) \quad (9)$$

[3] http://pacific.mpi-cbg.de/javadoc/Weka/clusterers/sIB.html

Where c is total number of classes, n_r is the size of class C_r , and n is the total size.

5.2 DataSet

The data set includes 100 ambiguous Chinese words and for every word it provided 50 instances. Besides that they also provided a sample test set of 2500 examples of 50 target words with the answers to illustrate the data format.

Besides the sIB algorithm we also apply the k-means and EM algorithm to cluster the feature vectors. These algorithms are separately using the simpleKMeans class and the EM class in the Weka 3.5.8 cluster package. Except the number of clusters set as the given number of senses and number of seeds set as zero, all other parameters are set as default. For the given sample test set with answers the result is illustrated in the Table 1 below.

algorithm	F-score
k-means	0.7025
EM	0.7286
sIB	0.8132

Table 1: Results of three clustering algorithms

From Table 1 we can see the sIB clustering algorithm improves the result of the Chinese word sense induction evidently.

In the real test data test containing 100 ambiguous Chinese words, our system achieves a F-score 0.7788 ranking 6th among the 18 systems submitted. The best F-score of these 18 systems is 0.7933 and the average of them is 0.7128.

5.3 Discussion

In our system we only use the local collocations and the co-occurrences of the target words. But the words distance for the target word in the same sentence and the parts of speech of the neighboring word together with the target word is also important in this task.

In our experiment we used the parts of speech for the target word and each word before and after it achieved by the Chinese word segmentation system as part of the features vectors for clustering. With a proper weight on each POS dimension in the feature vectors, the F score for some word in the given sample test set with answers improved evidently. For example the Chinese word “便宜”, the F score of it was developed from 0.5983 to 0.7573. But because of the fault of the segmentation system and other reasons F score of other words fell and the speed of the system was rather slower than before, we gave up this improvement finally.

Without the words distance for the target word in the same sentence the feature vectors maybe lack some information useful. So if we can calculate the correlation between the target word and other words, we will use these word sufficiently. However because of quantity of the Internet corpus is unknown, we didn’t find the proper method to weigh the correlation.

From the previous section we find that the F score for the real test data test is lower than that for the sample test set. It is mainly because there are more single Chinese characters (as words) in the real test data set. Our system does not process these characters specially. For most of the Chinese characters we can’t judge their correct senses only from the context where they appear. Their meaning always depends on the collocations with the other Chinese characters with which they become a Chinese word. However our system discriminates the senses of them only referring to the context of them, it can’t judge the meaning of these Chinese characters properly. Maybe the best way is to search them in the dictionary.

However our system does not always have a very poor performance for any single Chinese character (as a word). The result is quite good for some Chinese characters. For example the Chinese character “谷” which has three meaning: valley, millet and a family name, the precision (P) of our system is 0.760. But for most of single Chinese characters such as “服” and “公”, it is so bad that the result in the sample test worked rather better than the real test.

In Chinese the former character “谷” tends to express a complete meaning and the other characters in the word which they combine often modify it such as the characters “山” and “稻” in the word “山谷” and “稻谷”. So this character can have a relatively high correlation with the words around and our system can deal with such characters like it. Unfortunately most characters need other characters to represent a complete meaning as the the latter “服” and “公” so they almost have no correlation with the words around but with those characters in the word in which they occur. But our system only uses the context features and even doesn't do any special process about these single Chinese characters. Therefore our system can not address those characters appropriately and we need to find a proper method to solve it, using a dictionary may be a choice.

This method works better for nouns and adjectives (in the sample test data set there are only 4 adjectives), but for verbs F score falls a little, illustrated in the Table 2 below.

POS	F-score
nouns	0.8473
adjectives	0.8543
verbs	0.7921

Table 2: Results of each POS in the sample test data set

Only using the local collocations in our system the F score is achieve above 80% (in the sample test), it demonstrates to some extent the information of collocations is so important that we should pay more attention to it.

6 Conclusion

The triplet-based Chinese word sense induction method is fitted to the task of Chinese word sense induction and obtain rather good result. But for some single characters word and some verbs, this method is not appropriate enough. In the future work, we will improve the method with more reasonable triplet selection strategies.

Acknowledgments

This work was (partially) funded by 863 Program (No. 2009AA01A346), 973 Program (No. 2010CB327906), and Shanghai Science and Technology Development Funds (No. 08511500302).

References

Bordag, S. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. *Proceedings of EACL-06. Trento.*

Brody, S. and M. Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111. Association for Computational Linguistics.

Dorow, B. and D. Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 79–82. Association for Computational Linguistics.

Niu, Z.Y., D.H. Ji, and C.L. Tan. 2007. I2r: Three systems for word sense discrimination, chinese word sense disambiguation, and english word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 177–182. Association for Computational Linguistics.

Pedersen, T. and R. Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 197–207.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Slonim, N., N. Friedman, and N. Tishby. 2002. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136. ACM.