ISCAS：A System for Chinese Word Sense Induction Based on K-means Algorithm

Zhenzhong Zhang* **Le Sun†** **Wenbo Li†**

*Institute of Software, Graduate University
Chinese Academy of Sciences
zhenzhong@nfs.iscas.ac.cn

†Institute of Software
Chinese Academy of Sciences
{sunle,wenbo02}@iscas.ac.cn

Abstract

This paper presents an unsupervised method for automatic Chinese word sense induction. The algorithm is based on clustering the similar words according to the contexts in which they occur. First, the target word which needs to be disambiguated is represented as the vector of its contexts. Then, reconstruct the matrix constituted by the vectors of target words through singular value decomposition (SVD) method, and use the vectors to cluster the similar words. Our system participants in CLP2010 back off task4-Chinese word sense induction.

1 Introduction

It has been shown that using word senses instead of surface word forms could improve performance on many nature language processing tasks such as information extraction (Joyce and Alan, 1999), information retrieval (Ozlem et al., 1999) and machine translation (David et al., 2005). Historically, word senses are represented as a fixed-list of definitions coming from a manually complied dictionary. However, there seem to be some disadvantages associated with such fixed-list of senses paradigm. Since dictionaries usually contain general definitions and lack explicit semantic, they can't reflect the exact content of the context where the target word appears. Another disadvantage is that the granularity of sense distinctions is fixed, so it may not be entirely suitable for different applications.

In order to overcome these limitations, some techniques like word sense induction (WSI) have been proposed for discovering words' senses automatically from the unannotated corpus. The word sense induction algorithms are usually base on the Distributional Hypothesis, proposed by (Zellig, 1954), which showed that words with similar meanings appear in similar contexts (Michael, 2009). And the hypothesis is also popularized with the phrase "a word characterized by the company it keeps" (John, 1957). This concept shows us a method to automatically discover senses of words by clustering the target words with similar contexts (Lin, 1998). The word sense induction can be regarded as an unsupervised clustering problem. First, select some features to be used when comparing similarity between words. Second, represent disambiguated words as vectors of selected features according to target words' contexts. Third, cluster the similar words using the vectors. But compared with European languages such as English, Chinese language has its own characteristics. For example, Chinese ideographs have senses while the English alphabets don't have. So the methods which work well in English may not be entirely suitable for Chinese.

This paper proposes a method for Chinese word sense induction, which contains two stage processes: features selecting and context clustering. Chinese ideographs and Chinese words which have two or more Chinese ideographs are used different strategies when selecting features. The vectors of target word's instances are put together to constitute a matrix, whose row is instances and column is features. Reconstruct the matrix through singular value decomposition to get a new vector for each instance. Then, K-means clustering algorithm is employed to cluster the vectors of disambiguated words' contexts. Each cluster to which some instances belong to identifies a sense of corresponding target word.

Our system participants in CLP2010 back off task4 - Chinese word sense induction.

The remainder of this paper is organized as follows. Section 2 presents the Chinese word senses induction algorithm. Section 3 presents the evaluation sheme and the results of our system. Section 4 gives some discussions and conclusions.

2 Chinese Word Senses Induction

This section will present the strategies of selecting features for disambiguated Chinese words and k-means algorithm for clustering vectors of the contexts.

2.1 Features Selection

Since the input instances of target words are unstructured, it's necessary to select features and transform them into structured format to fit the automatic clustering algorithm. Following the example in (Ted, 2007), words are chosen as features to represent the contexts where target words appear. A word *w* in the context of the target word can be represented as a vector whose *ith* component is the average of the calculated conditional probabilities of *w* and w_j.

The target words are usually removed from the corpus in the task of English word sense induction. But Chinese language is very different from European languages such as English. Chinese ideographs usually have meanings of their own while English alphabets don't have. In Chinese word senses induction tasks, the target word may be a Chinese word which could have one or more Chinese ideographs or a Chinese ideograph. And the meaning of Chinese ideographs is determined by the Chinese word where it appears. The following example shows us this case.

- 我国依靠推广超级稻累计增产稻*谷* 162亿公斤。
- 在木化石园附件的一处山*谷*，是大佛寺近期增加的五百罗汉堂。

In this example, the target word is Chinese ideograph "谷" displayed in italic in the contexts. In the first context, its meaning is paddy which is determined by the Chinese word "稻谷", and similarly in the second context its meaning is valley determined by "山谷". Since the meaning of the Chinese ideograph "谷" is determined by the word where it appears, it may not be appropriate to remove it from the contexts simply while the others of the word are left. Different strategies are employed to remove target words. If the target word contains two or more Chinese ideographs, it will be removed from the context. Otherwise it will be kept.

To solve the problem of data sparseness, we extracted extra 100 instances for each target word from Sogou Data and also used the thesauruses (TongYiCi CiLin of HIT) to reduce the dimensionality of the word space (feature space). Two filtering heuristics are applied when selecting features. The first one is the minimum frequency p_1 of words, and the second one is the maximum frequency p_2 of words.

Each selected word (feature) should be assigned a weight, which indicates the relative frequency of two co-occurring words. Using conditional probabilities for weighting for object/verb and subject/verb pairs is better than point-wise mutual information (Philipp et al., 2005). So we used conditional probabilities for weighting words pairs. Let $num_{i,j}$ denote the number of the instances where the word i and word j co-occur, and num_i denote the number of the instances in which the word i appears. Then the *jth* component of the vector of the word i can be calculated using the following equation.

$$w_{i,j} = \frac{p(j \mid i) + p(i \mid j)}{2}$$

Where

$$p(i \mid j) = \frac{num_{i,j}}{num_j}$$

The contexts of each target word are represented as the centroid of the vectors of the words occurring in the target contexts. Figure 1 shows an example of context vector, where the Chinese word "果实" co-occurs with Chinese words "水果"and "种子".

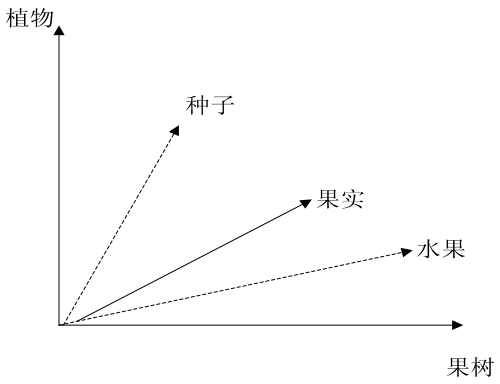

Figure 1: An example of a context vector for "果实", calculated as the centroid of vectors of "种子" and "水果".

2.2 Clustering Algorithm

K-means algorithm is applied to cluster the vectors of the target word. It assigns each element to one of K clusters according to which centroid the element is close to by the similarity function. The cosine function is used to measure the similarity between two vectors V and W:

$$sim(V,W) = \frac{V \bullet W}{|V| \times |W|} = \frac{\sum_{i=1}^{n} V_i W_i}{\sqrt{\sum_{i=1}^{n} V_i^2 \sum_{i=1}^{n} W_i^2}}$$

where n is the number of features in each vector. Before clustering the vectors of instances, we put together the vectors of instances in the corpus and obtain a co-occurrence matrix of instances and words. Singular value decomposition is applied to reduce the dimensionality of the resulting multidimensional space and finds the major axes of variation in the word space (Golub and Van Loan, 1989). After the reduction, the similarity between two instances can be measured using the cosine function mentioned as above between the corresponding vectors. The clustering algorithm stops when the centroid of each cluster does not change or the iteration of the algorithm exceed a user-defined threshold p_3. And the number of the clusters is determined by the corpus where the target word appears. Each cluster to which some instances belong represents one senses of the target word represented by the vector.

We also employed a graph-based clustering algorithm -Chinese Whispers (CW) (Chris, 2006) to deal with the task of Chinese WSI. CW does not require any input parameters and has a good performance in WSI (Chris, 2006). For more details about CW algorithm please refer to (Chris, 2006). We first constructed a graph, whose vertexes were instances of target word and edges' weight was the similarity of the corresponding two vertexes. Then we removed the edges with minimum weight until the percentage of the kept edges' sum respect the total was below a threshold p_4. CW algorithm was employed to cluster the graph and each clusters represented a sense of target word.

3 Evaluation

This section presents the evaluation scheme, set of parameters and the result of our system.

3.1 Evaluation Scheme

We use standard cluster evaluation methods to measure the performance of our WSI system. Following the former practice (Zhao and Karypis, 2005), we consider the FScore measure for assessing WSI methods. The FScore is used in a similar fashion to Information Retrieval exercises.

Let we assume that the size of a particular class s_r is n_r, the size of a particular cluster h_j is n_j and the size of their common instances set is $n_{r,j}$. The precision can be calculated as follow:

$$P(s_r, h_j) = \frac{n_{r,j}}{n_j}$$

The recall value can be defined as:

$$R(s_r, h_j) = \frac{n_{r,j}}{n_r}$$

Then FScore of this class and cluster is defined to be:

$$F(s_r, h_j) = \frac{2 \times P(s_r, h_j) \times R(s_r, h_j)}{P(s_r, h_j) + R(s_r, h_j)}$$

The FScore of class s_r, $F(s_r)$, is the maximum $F(s_r, h_j)$ value attained by any cluster, and it is defined as:

$$F(s_r) = \max_{h_j}(F(s_r, h_j))$$

Finally, the FScore of the entire clustering solution is defined as the weighted average FScore of each class:

$$FScore = \sum_{r=1}^{q} \frac{n_r \times F(s_r)}{n}$$

Where q is the number of classes and n is the total number of the instances where target word appears.

3.2 Tuning the Parameters

We tune the parameters of our system on the training data. But because of time restrictions, we do not optimize these parameters. The maximum frequency of a word (p_2) and the maximum number of the K-means' iteration (p_3) are tuned on the training data. The minimum frequency of a word (p_1) was set to two following our intuition. The last parameter K -the number of the clusters is determined by the test data in which the target word appears. When tuning parameters, we first fixed the parameter p_3 and found the best value of parameter p_2, which could lead to the best performance. The results have been shown in Table 1 and Table 2.

Parameters	FScore
P_3=300,p_2=35	0.7502
P_3=400,p_2=40	0.7523
P_3=500,p_2=40	**0.7582**

Table 1: The results of K-means with SVD

Parameters	FScore
P_3=300,p_2=40	0.7454
P_3=400,p_2=40	**0.7493**
P_3=500,p_2=45	0.7404

Table 2: The results of K-means

The performance of CW algorithm is shown in Table 3. The parameter p_4 is a threshold for pruning graph as describing in section 2.2.

Parameter	FScore
P_4=0.55	0.6325
P_4=0.6	0.6321
P_4=0.65	0.6278
P_4=0.7	**0.6393**
P_4=0.75	0.6289
P_4=0.8	0.6345
P_4=0.85	0.6326
P_4=0.9	0.6342
P_4=0.95	0.6355

Table 3: The results of CW.

The result shows that the K-means algorithm has a better performance than CW. That may because CW can't use the information of the number of clusters, but K-means could. Another problem for CW is that the size of corpus is small and the constructed graph can't reflect the inherent relation between the instances.

Based on the result of experiments, we employed K-means algorithm for our system and the parameters is shown in Table 4.

Parameters	Value
P_1: Minimum frequency of a word	2
P_2: Maximum frequency of a word	40
P_3: Maximum number of K-means iteration	500
K: the number of the cluster	-

Table 4: Parameters for the system. The last parameter K is provided by the test data.

3.3 Result

Our system participants in the CLP2010 backoff task4 and disambiguate 100 target words, total 5000 instances. The F-score of our system on the test data is 0.7209 against the F-score 0.7933 of the best system.

4 Conclusion

We have presented a model for Chinese word sense induction. Different strategies are applied to deal with Chinese ideographs and Chinese words that contain two or more Chinese ideographs. After selecting the features –words, singular value decomposition is used to find the major axes of variation in the feature space and reconstruct the vector of each context. Then we employ k-means cluster algorithm to cluster the vectors of contexts. Result shows that our system is able to induce correct senses. One drawback of our system is that it overlooks the infrequent senses because of lacking enough data. And our system only uses the information of word co-occurrences. So in the future we would like to integrate different kinds of information such as topical information, syntactic information and semantic information, and see if we could get a better result.

Acknowledgement

This work has been partially funded by National Natural Science Foundation of China under grant #60773027, #60736044 and #90920010

and by “863” Key Projects #2006AA010108, “863” Projects #2008AA01Z145. We would like to thank anonymous reviewers for their detailed comments.

References

Chris Biemann, 2006. *Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems*, In Proceedings of TextGraphs, pp. 73–80, New York, USA.

David Vickrey, Luke Biewald, Marc Teyssley, and Daphne Koller. 2005. *Word-sense disambiguation for machine translation*. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 771-778, Vancouver, British Columbia, Canada

Dekang Lin. 1998. *Automatic retrieval and clustering of similar words*. In Proceedings of the 17th international conference on Computational linguistics, volume 2, pages 768-774, Montreal, Quebec, Canada

Golub, G. H. and Van Loan, C. F. 1989. *Matrix Computations*. The John Hopkins University Press, Baltimore, MD

John, R., Firth. 1957. *A Synopsis of Linguistic Theory 1930-1955*, pages 1-32.

Joyce Yue Chai and Alan W. Biermann. 1999. *The use of word sense disambiguation in an information extraction system*. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, pages 850-855, Orlando, Florida, United States.

Michael Denkowski. 2009. *A Survey of Techniques for Unsupervised Word Sense Induction*.

Ozlem Uzuner, Boris Katz, and Deniz Yuret. 1999. *Word sense disambiguation for information retrieval*. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, page 985, Orlando, Florida, United States.

Philipp Cimiano, Andreas Hotho, and Steffen Staab, 2005. *Learning concept hierarchies from text corpora using formal concept analysis*, Journal of Artificial Intelligence Research (JAIR), 24, 305–339.

Ted Pedersen, 2007. *Umnd2: Senseclusters applied to the sense induction task of senseval-4*. In Proceedings of the Fourth International Workshop on Semantic Evaluations, pages 394–397, Prague, Czech Republic.

Zellig Harris. 1954. *Distributional Structure*, pages 146-162.

Ying Zhao and George Karypis. 2005. *Hierarchical clustering algorithms for document datasets.* Data Mining and Knowledge Discovery, 10(2):141.168.