Improving Word Sense Induction by Exploiting Semantic Relevance

Zhenzhong Zhang
Institute of Software, Graduate University,
Chinese Academy of Sciences, Beijing, China
Zhenzhong@nfs.iscas.ac.cn

Le Sun
Institute of Software, Chinese Academy of
Sciences, Beijing, China
sunle@iscas.ac.cn

Abstract

Word Sense Induction (WSI) is the task of automatically inducing the different senses of a target word from unannotated text. Traditional approaches based on the vector space model (VSM) represent each context of a target word as a vector of selected features (e.g. the words occurring in the context). These approaches assume that the words occurring in the context are independent and do not exploit semantic relevance between words. In this paper we propose a WSI method which can exploit semantic relevance between words by incorporating a word graph into the framework of clustering of context vectors. The method is evaluated on the testing data of the Chinese Word Sense Induction task of the first CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP2010). Experimental results show that our method significantly outperforms the baseline methods.

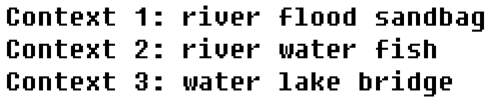
1 Introduction

It has been shown that using word senses instead of surface word forms can improve performance on many natural language processing tasks such as machine translation (Vickrey et al., 2005) and information retrieval (Uzuner et al., 1999; Véronis, 2004). Historically, using word senses usually involved the use of manually compiled resources in which word senses were represented as a fixed list of definitions. However, there seem to be some disadvantages associated with such fixed list of senses paradigm. Firstly, since dictionaries usually contain general definitions, they can not reflect the exact contents of the contexts where target words appear (Véronis, 2004). Secondly, because the "fixed list of senses" paradigm makes the fixed granularity assumption of the senses distinction, it may not be suitable in different applications (Kilgarriff, 1997; Brody and Lapata, 2009).

To overcome these limitations, some techniques like WSI have been proposed for discovering words senses automatically from unannoteted corpuses. WSI algorithms are usually based on the Distributional Hypothesis which shows that words with similar meanings appear in similar contexts (Harris, 1954). This concept can be leveraged to induce different senses of a target word by clustering the contexts where the target word appears.

Much work in WSI is based on the vector space model, in which each context of a target word is represented by a vector of selected features (e.g. the words occurring in the context). These context vectors are clustered and the resulting clusters are taken to represent the induced senses. However, when constructing context vectors, the approaches based on VSM assume that the words occurring in the contexts are independent and do not exploit semantic relevance between words. This will cause the problem that two contexts using semantically related but distinct words will show no similarity. Figure 1 shows a simple example of three context vectors taken from three contexts of the target word *bank*, which appears with one sense i.e. *sloping land*. If we assume that context words are independent, the similarity between context 1 and context 3 will be zero, which means that the senses of the target word *bank* in the two contexts are different. But, in practice, *bank* appears with one sense in the two contexts. Some methods have been proposed to use information beyond that which is found in the immediately surrounding context. For example, in (Schűtze, 1998), second order co-occurrence matrix was used to construct rich vectors of word contexts.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1387–1391,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP



```
Context 1: river flood sandbag
Context 2: river water fish
Context 3: water lake bridge
```

Figure 1: Three context vectors for the target word *bank*.

In this paper, we propose a WSI method which can exploit semantic relevance between words by incorporating a word graph into the framework of clustering of context vectors. Firstly, we build a graph, where each vertex corresponds to a selected word and edges between vertices are weighted based on the semantic relevance between their associated words. Then we adapt the Personalized PageRank method (Agirre and Soroa, 2009) for incorporating semantic relevance between words into context vectors. The resulting vectors are clustered and each cluster represents an induced sense of the target word. Our method bears some similarity with some graph-based methods of WSI since they all need a graph of words. But in our method the graph is used to incorporate semantic relevance between words into context vectors while in graph-based approaches of WSI it is clustered to induce different senses of a target word. We use two vector-based approaches and one graph-based approach as baselines. Our evaluation under the framework of CLP2010 Chinese Word Sense Induction task shows that our approach significantly outperforms the baseline systems.

The structure of this paper is as follows: in Section 2, we will present our approach. In Section 3, we will introduce the experimental setup and show the experimental results. We will end with a conclusion and future work in Section 4.

2 Our Approach

In this section, we introduce how to build a word graph and how to use Personalized PageRank method to incorporate semantic relevance between words into context vectors. Then we describe how to cluster the resulting context vectors to induce senses of target words.

2.1 Building A Word Graph

In this section, we aim to build a graph where each vertex corresponds to a word and edges between vertices are weighted based on the semantic relevance between their associated words. Initially, we construct a word-by-context matrix *P* with the entry $P_{i,j}$ giving the weight of word *i* in context *j*. In this paper, we set

$$P_{i,j} = \frac{n_{i,j}}{\sum_{i} n_{i,j}} \times \log \frac{N}{n(i)} \qquad (1)$$

where $n_{i,j}$ is the frequency of word *i* occurring in the context *j*, and *n(i)* is the number of the contexts containing the word *i* and *N* is the total number of contexts. Just like contexts can be seen as bags of words, words can be viewed as bags of contexts. So a row of the matrix *P* can be seen as the context-vector for a word. We assume that two words that have more correlated context-vectors will have a greater semantic relevance. In this case, the semantic relevance of two words is evaluated through the inner product of vectors corresponding to the two words. Support that $M=PP^T$, then $M_{i,j}$ gives the semantic relevance between word *i* and *j*. Singular Value Decomposition (SVD) is used to reduce the dimensionality of matrix *M*. *M* is evaluated by the equation (2)

$$M \approx \sum_{i=1}^{K} \lambda_i x_i x_i^T \qquad (2)$$

where x_i is the eigenvector of *M*, and λ_i is the corresponding eigenvalue and x_i^T denotes the transpose of x_i. *K* is the minimum *k* that satisfies $\sum_{i=1}^{k} \lambda_i \geq 0.85 \times \sum_{i=1}^{D} \lambda_i$, where *D* is the number of eigenvectors of *M*. Now we have built the graph, in which each vertex corresponds to a word and the weight of edge between vertex *i* and *j* is given by $M_{i,j}$ indicating the semantic relevance between the word *i* and *j*.

2.2 Incorporating semantic relevance between words into context vectors

Personalized PageRank algorithm is adapted from PageRank algorithm (Brin and Page, 1998). In the PageRank formulation ($\Pr = cM\Pr + (1-c)v$), the element values of the vector v are all$\frac{1}{N}$, where *N* is the total number of vertices in the graph. But in the Personalized PageRank, the vector v can be non-uniform and assign stronger probabilities to certain kinds of vertices.

We assume that the weight of a feature (word) in the context vector depends on not only its frequency in contexts but also the words that co-occur with it. This means, if two words *i* and *j* co-occur in a context and are semantic related, the weight of *i* (or *j*) should be strengthened by *j* (or *i*). We adapt Personalized PageRank algorithm for this process. The weight of the word i at t+1 step is defined as:

$$W(i)^{t+1} = (1-d)W(i)^{0} + d\sum_{j\in In(i)} \frac{M_{j,i}}{\sum_{k\in In(j)} M_{j,k}} W(j)^{t} \quad (3)$$

where $M_{j,i}$ is the weight of the edge between vertices *i* and *j*, which is defined in Section 2.1. $W(j)^t$ is the weight of the word *j* at t step and $W(i)^0$ is the initial weight of the word *i* in the context. *In(i)* stands for the set of vertices that connect to *i*. *d* is the damping factor and is usually set at 0.85.

The weight of each word is initialized based on its frequency and the Personalized PageRank algorithm iterates until convergence. After the running of the Personalized PageRank algorithm, each word gets a new weight. In this way, we incorporate semantic relevance between words into context vectors.

2.3 Inducing Word Senses

The k-means algorithm is used for clustering the resulting vectors produced in Section 2.2. The similarity between two objects is computed using cosine function. The number of clusters, k, is automatically determined using PK2 criterion function (Pedersen and Kulkarni, 2006). Each resulting cluster represents a kind of sense of the target word.

3 Experiments

3.1 Experimental Setup

Our experiments are based on the CLP2010 Chinese Word Sense Induction task testing dataset which contains 100 target words (22 target words are constituted by a single character and 78 target words are constituted by two or more characters) and total 5000 instances. The number of senses on average per word is 2.5. The original testing data contains the number of target word senses. But in practice, the number of target word senses is unknown and it need to be indentified automatically. So in the experiments we discover the number of target word senses automatically using the PK2 criterion.

Each instance of a target word is processed by segmenting Chinese word and removing stopwords and target words. The remaining words are used to build the word graphs and construct the vectors of contexts.

We use two vector-based approaches and one graph-based approach as baselines. The first one is a vector-based WSI approach, which represents the contexts of a target word using second order co-occurrence vectors (Schűtze, 1998). This approach constructs a word-by-word co-occurrence matrix by identifying bigrams whose number of occurrences is greater than a pre-specified threshold. A row in the matrix is the vector for a context word. Each context is represented by the centroid of all vectors of the words which make up the context. Then these context vectors are clustered to induce senses of target words. This approach has a good performance in public evaluation (e.g. Semeval-2007 task 02) (Agirre and Soroa, 2007).

The second one is also a vector-based WSI approach which represents the contexts of a target word using bag-of-words vectors and weights each feature (word) based on its TF and IDF. The two vector-based WSI approaches use k-means algorithm to cluster the context vectors of target words and the maximum number of k-means iterations is set to 100.

The third one is a graph-based WSI approach. We build a graph according to the approach described in (Agirre et al., 2006). Chinese Whispers algorithm (Biemann, 2006) is used to cluster the graph. The maximum number of Chinese Whispers iterations is set to 100. We also include the "one cluster per word" baseline (1c1w), where all instances of a target word are grouped into a single cluster. In SemEval-2010 task 14, none of the participating systems outperform this baseline in paired F-score (Artiles et al., 2009), which indicates that this baseline is quite strong.

According to (Pedersen, 2010), we employ paired F-score as evaluation measure. Let *C={C*$_j$*|j=1,2,…,n}* be a set of clusters generated by a WSI system and *S={G*$_i$*|i=1,2,…,m}* be the set of gold standard classes. For each cluster C_j, we generate $\binom{|C_j|}{2}$ instance pairs, in which $|C_j|$ is the total number of instances that belong to C_j. Similarly, we generate $\binom{|G_i|}{2}$ instance pairs for each gold standard class G_i. Let *F(C)* is the set of instance pairs generated from any clusters in *C* and *F(S)* is the set of instance pairs generated from any gold standard classes in *S*. Precision and recall are defined in Equation 4 and 5 respectively.

$$P = \frac{|F(C) \cap F(S)|}{|F(C)|} \quad (4)$$

$$R = \frac{|F(C) \cap F(S)|}{|F(S)|} \quad (5)$$

Then the paired F-score is defined in Equation 6.

$$Fs = \frac{2 \times P \times R}{P + R} \quad (6)$$

3.2 Experimental Results

Table 1 shows the results of experiments with 100 target words (total 5000 instances). Just like what have been shown in (Agirre and Soroa, 2007) and (Manandhar et al., 2010), the 1c1w baseline shows the best performance. However, it only discoveries one sense for each target word, which is the most frequent sense of the target word. In contrast, WSI_SR and WSI_SOV predict 2.1 and 2.4 senses on average per word respectively, which is more close to the actual number of senses (2.5).

System	Fs(%) (All)	Fs(%) (C)	Fs(%) (W)	#Cl
WSI_SR	58.5	42.78	62.82	2.1
WSI_SOV	56.37	42.46	60.28	2.4
WSI_BOW	51.24	40.68	54.25	3.7
WSI_CW	40.99	30.68	43.72	6.7
1c1w	60.6	44.5	65.14	1

Table 1: Evaluation of WSI systems. WSI_SR stands for our method which can exploit semantic relevance between words for WSI system, WSI_SOV for the WSI system based on second-order vectors, WSI_BOW for the WSI system based on bag-of-words vectors, WSI_CW for the graph-based WSI system which employs Chinese Whispers clustering algorithm. C stands for the target words which are constituted by one character while W stands for the target words which are constituted by two or more characters.

WSI_SR achieves 0.585 paired F-score, outperforming WSI_BOW with absolute improvements of 7.26%, which indicates that exploiting semantic relevance between words can improve the performance of WSI systems.

The performance of WSI_SR is well above that of WSI_SOV. This may be due to the fact that WSI_SOV only exploits semantic relevance between words occurring in the certain contexts while WSI_SR can exploit semantic relevance between words occurring in the all contexts. For example, in Figure 1, if we use the binary weighting scheme, WSI_SOV will set the weight for word *lake* and *bridge* to 0, which indicates that WSI_SOV cannot exploit the semantic relevance between the words occurring in context #1 and the two words. In contrast, WSI_SR sets the weight for word *lake* and *bridge* to 0.08, which shows that WSI_SR can exploit the semantic relevance between the words occurring in context #1 and them.

The system WSI_CW performs the worst. The possible reason is that the graph constructed from the testing dataset is made up of lots of unconnected subgraphs, which causes that the Chinese Whispers algorithm cannot cluster words correctly and induces too many senses. Compared to WSI_CW, WSI_SR incorporates the word graph into the framework of clustering of context vectors, which makes it avoid the drawback of WSI_CW.

A Chinese word can be constituted by a single character or multiple characters, which is different from English. Usually, the Chinese word containing only one character has more senses (e.g. Chinese word, “打” (beat), has twenty one senses in the testing dataset), which makes it more difficult to induce the senses of such words. In Table 1, we report the performance of systems on Chinese characters and Chinese words containing two or more characters. WSI_SR performs better than three baseline systems on Chinese words but has a similar performance with WSI_SOV on Chinese characters, which indicates that other information (e.g. syntactic information) should be exploited to improve the performance on Chinese characters.

4 Conclusions

In this paper, we present a WSI method, which can exploit semantic relevance between words by incorporating a word graph into the framework of clustering of context vectors. We build a word graph and use it to incorporate semantic relevance between words into context vectors. The resulting vectors are clustered to induce the senses of target words. Experimental results on the testing data of CLP2010 Chinese Word Sense Induction task demonstrate the effectiveness of our method.

Further work focuses on exploiting different kinds of information such as topic information and syntactic information to improve the performance of our method, especially for Chinese characters.

Acknowledgments

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. We would like to thank anonymous reviewers for their detailed comments.

References

Adam Kilgarriff, 1997. *I Don’t Believe in Word Senses,* Computers and the Humanities 31(2): 91-113.

Chris Biemann, 2006. *Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems*, In Proceedings of TextGraphs, pages 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

Eneko Agirre, David Martínez, Oier López de Lacalle and Aitor Soroa. 2006. *Two graph-based algorithms for state-of-the-art WSD.* In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 585–593, Sydney.

Eneko Agirre and Aitor Soroa. 2007. *Semeval-2007 task2: Evaluating word sense induction and discrimination systems*. In Proceedings of SemEval-2007. Association for Computational Llinguistics, pages 7-12, Prague.

Eneko Agirre and Aitor Soroa. 2009. *Personalizing PageRank for Word Sense Disambiguation*. In Proceedings of the 12th Conference of the European Chapter of the ACL, pages 33–41.

Hinrich Schűtze. 1998. *Automatic Word Sense Discrimination*. Computational Linguistic, Vol. 24, No. 1, pages 97-123.

Javier Artiles, Enrique Amig´o, and Julio Gonzalo. 2009. *The role of named entities in web people search*. In Proceedings of EMNLP, pages 534–542, pages 534–542, Singapore, August.

Jean. Véronis. 2004. *Hyperlex: lexical cartography for information retrieval*. Computer Speech & Language, 18(3):223.252.

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.

Samuel Brody and Mirella 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, Athens, Greece.

Sergey Brin and Lawrence Page. 1998. *The anatomy of a large-scale hypertextual web search engine.* Computer Networks and ISDN Systems, 30(1-7).

Suresh Manandhar, Ioannis P. Klapaftis, Dmitriy Dligach and Sameer S. Pradhan. 2010. *SemEval-2010 task 14: Word sense induction & disambiguation*. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 63–68, Uppsala, Sweden

Ted Pedersen and Anagha Kulkani, 2006. *Automatic cluster stopping with criterion functions and the gap statistic*. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume, pages 276–279, New York City, USA.

Ted Pedersen. 2010. *Duluth-WSI: SenseClusters applied to the sense induction task of SemEval-2*. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 363–366, Uppsala, Sweden.

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