Growing Related Words from Seed via User Behaviors: A Re-ranking Based Approach

Yabin Zheng **Zhiyuan Liu** **Lixing Xie**
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084,China
{yabin.zheng, lzy.thu, lavender087}@gmail.com

Abstract

Motivated by Google Sets, we study the problem of growing related words from a single seed word by leveraging user behaviors hiding in user records of Chinese input method. Our proposed method is motivated by the observation that the more frequently two words co-occur in user records, the more related they are. First, we utilize user behaviors to generate candidate words. Then, we utilize search engine to enrich candidate words with adequate semantic features. Finally, we reorder candidate words according to their semantic relatedness to the seed word. Experimental results on a Chinese input method dataset show that our method gains better performance.

1 Introduction

What is the relationship between "自然语言处理" (Natural Language Processing) and "人工智能" (Artificial Intelligence)? We may regard NLP as a research branch of AI. Problems arise when we want to find more words related to the input query/seed word. For example, if seed word "自然语言处理" (Natural Language Processing) is entered into Google Sets (Google, 2010), Google Sets returns an ordered list of related words such as "人工智能" (Artificial Intelligence) and "计算机" (Computer). Generally speaking, it performs a large-scale clustering algorithm that can gather related words.

In this paper, we want to investigate the advantage of user behaviors and re-ranking framework in related words retrieval task using Chinese input method user records. We construct a User-Word bipartite graph to represent the information hiding in user records. The bipartite graph keeps users on one side and words on the other side. The underlying idea is that the more frequently two words co-occur in user records, the more related they are. For example, "机器翻译" (Machine Translation) is quite related to "中文分词" (Chinese Word Segmentation) because the two words are usually used together by researchers in natural language processing community. As a result, user behaviors offer a new perspective for measuring relatedness between words. On the other hand, we can also recommend related words to users in order to enhance user experiences. Researchers are always willing to accept related terminologies in their research fields.

However, the method is purely statistics based if we only consider co-occurrence aspect. We want to add semantic features. Sahami and Helman (2006) utilize search engine to supply web queries with more semantic context and gains better results for query suggestion task. We borrow their idea in this paper. User behaviors provide statistic information to generate candidate words. Then, we can enrich candidate words with additional semantic features using search engine to retrieve more relevant candidates earlier. Statistical and semantic features can complement each other. Therefore, we can gain better performance if we consider them together.

The contributions of this paper are threefold. First, we introduce user behaviors in related word retrieval task and construct a User-Word bipartite graph from user behaviors. Words are used by users, and it is reasonable to measure relatedness between words by analyzing user behaviors. Second, we take the advantage of semantic features using search engine to reorder candidate words. We aim to return more relevant candidates earlier. Finally, our method is unsupervised and language independent, which means that we do not require any training set or manual labeling efforts.

The rest of the paper is organized as follows. Some related works are discussed in Section 2. Then we introduce our method for related words retrieval in Section 3. Experiment results and discussions are showed in Section 4. Finally, Section 5 concludes the whole paper and gives some future works.

Proceedings of the ACL 2010 Student Research Workshop, pages 49–54,
Uppsala, Sweden, 13 July 2010.

2 Related Work

For related words retrieval task, Google Sets (Google, 2010) provides a remarkably interesting tool for finding words related to an input word. As stated in (Zheng et al., 2009), Google Sets performs poor results for input words in Chinese language. Bayesian Sets (Ghahramani and Heller, 2006) offers an alternative method for related words retrieval under the framework of Bayesian inference. It computes a score for each candidate word by comparing the posterior probability of that word given the input, to the prior probability of that candidate word. Then, it returns a ranked list of candidate words according to their computed scores.

Recently, Zheng et al. (2009) introduce user behaviors in new word detection task via a collaborative filtering manner. They extend their method to related word retrieval task. Moreover, they prove that user behaviors provide a new point for new word detection and related word retrieval tasks. However, their method is purely statistical method without considering semantic features.

We can regard related word retrieval task as problem of measuring the semantic relatedness between pairs of very short texts. Sahami and Helman (2006) introduce a web kernel function for measuring semantic similarities using snippets of search results. This work is followed by Metzler et al., (2007), Yih and Meek, (2007). They combine the web kernel with other metrics of similarity between word vectors, such as Jaccard Coefficient and KL Divergence to enhance the result.

In this paper, we follow the similar idea of using search engine to enrich semantic features of a query word. We regard the returned snippets as the context of a query word. And then we reorder candidate words and expect more relevant candidate words can be retrieved earlier. More details are given in Section 3.

3 Related Words Retrieval

In this section, we will introduce how to find related words from a single seed word via user behaviors and re-ranking framework.

First, we introduce the dataset utilized in this paper. All the resource used in this paper comes from Sogou Chinese pinyin input method (Sogou, 2006). We use Sogou for abbreviation hereafter. Users can install Sogou on their computers and the word lists they have used are kept in their user records. Volunteers are encouraged to upload their anonymous user records to the server side. In order to preserve user privacy, usernames are hidden using MD5 hash algorithm.

Then we demonstrate how to build a User-Word bipartite graph based on the dataset. The construction can be accomplished while traversing the dataset with linear time cost. We will give more details in Section 3.1.

Second, we adopt conditional probability (Deshpande and Karypis, 2004) to measure the relatedness of two words. Intuitively, two words are supposed to be related if there are a lot of users who have used both of them. In other words, the two words always co-occur in user records. Starting from a single seed word, we can generate a set of candidate words. This is the candidate generation step.

Third, in order to take the advantage of semantic features, we carry out feature extraction techniques to represent generated candidate words with enriched semantic context. In this paper, we generally make use of search engine to conduct the feature extraction step. After this step, input seed word and candidate words are represented as feature vectors in the vector space.

Finally, we can reorder generated candidate words according to their semantic relatedness of the input seed word. We expect to retrieve more relevant candidate words earlier. We will make further explanations about the mentioned steps in the next subsections.

3.1 Bipartite Graph Construction

As stated before, we first construct a User-Word bipartite graph from the dataset. The bipartite graph has two layers, with users on one side and the words on the other side. We traverse the user records, and add a link between user *u* and word *w* if *w* appears in the user record of *u*. Thus this procedure can be accomplished in linear time.

In order to give better explanations of bipartite graph construction step, we show some user records in Figure 1 and the corresponding bipartite graph in Figure 2.

Fig. 1. User Records Sample

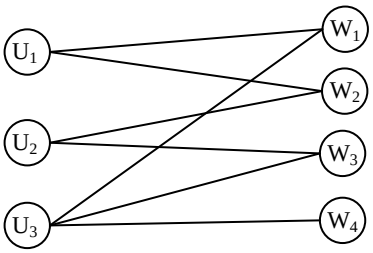

Fig. 2. Corresponding Bipartite Graph

From Figure 1, we can see that $Word_1$ and $Word_2$ appear in $User_1$'s record, which indicates that $User_1$ has used $Word_1$ and $Word_2$. As a result, in Figure 2, node $User_1$ is linked with node $Word_1$ and $Word_2$. The rest can be done in the same manner.

3.2 Candidates Generation

After the construction of bipartite graph, we can measure the relatedness of words from the bipartite graph. Intuitively, if two words always co-occur in user records, they are related to each other. Inspired by (Deshpande and Karypis, 2004), we adopt conditional probability to measure the relatedness of two words.

In particular, the conditional probability of word *j* occurs given that word *i* has already appeared is the number of users that used both word *i* and word *j* divided by the total number of users that used word *i*.

$$P(j \mid i) = \frac{Freq(ij)}{Freq(i)} \qquad (1)$$

In formula 1, *Freq(X)* is the number of users that have used words in the set *X*. We can clearly see that $P(j|i) \neq P(i|j)$, which means that conditional probability leads to asymmetric relations. The disadvantage is that each word *i* tends to have a close relationship with stop words that are used quite frequently in user records, such as "的" (of) and "一个" (a).

In order to alleviate this problem, we consider the conditional probabilities *P(j|i)* and *P(i|j)* together. Word *i* and word *j* is said to be quite related if conditional probabilities *P(j|i)* and *P(i|j)* are both relatively high. We borrow the idea proposed in (Li and Sun, 2007). In their paper, a *weighted harmonic averaging* is used to define the relatedness score between word *i* and word *j* because either *P(j|i)* or *P(i|j)* being too small is a severe detriment.

$$Score(i, j) = \left(\frac{\lambda}{P(i \mid j)} + \frac{1-\lambda}{P(j \mid i)} \right)^{-1} \qquad (2)$$

In formula 2, parameter $\lambda \in [0,1]$ is the weight for *P(i|j)*, which denotes how much *P(i|j)* should be emphasized. We carry out some comparative experiments when parameter λ varies from 0 to 1 stepped by 0.1. We also tried other co-occurrence based measures like mutual information, Euclidean and Jaccard distance, and found that weight harmonic averaging gives relatively better results. Due to space limitation, we are not able to report detailed results.

So far, we have introduced how to calculate the relatedness *Score(i, j)* between word *i* and word *j*. When a user enters an input seed word *w*, we can compute *Score(w,c)* between seed word *w* and each candidate word *c*, and then sort candidate words in a descending order. Top *N* candidate words are kept for re-ranking, we aim to reorder top *N* candidate words and return the more related candidate words earlier. Alternatively, we can also set a threshold for *Score(w,c)*, which keeps the candidate word *c* with *Score(w,c)* larger than the threshold. We argue that this threshold is difficult to set because different seed words have different score thresholds.

Note that this candidate generation step is completely statistical method as we only consider the co-occurrence of words. We argue that semantic features can be a complement of statistical method.

3.3 Semantic Feature Representation and Re-ranking

As stated before, we utilize search engine to enrich semantic features of the input seed word and top *N* candidate words. To be more specific, we issue a word to a search engine (Sogou, 2004) and get top 20 returned snippets. We regard snippets as the context and the semantic representation of this word.

For an input seed word *w*, we can generate top *N* candidate words using formula (2). We issue each word to search engine and get returned snippets. Then, each word is represented as a feature vector using bag-of-words model. Following the conventional approach, we calculate the relatedness between the input seed word *w* and a candidate word *c* as the cosine similarity between their feature vectors. Intuitively, if we introduce more candidate words, we are more likely to find related words in the candidate sets. However, noisy words are inevitably included. We will show how to tune parameter *N* in the experiment part.

As a result, candidate words with higher semantic similarities can be returned earlier with enriched semantic features. Re-ranking can be regarded as a complementary step after candidate generation. We can improve the performance of related word retrieval task if we consider user behaviors and re-ranking together.

4 Experiment

In this section, we demonstrate our experiment results. First, we introduce the dataset used in this paper and some statistics of the dataset. Then, we build our ground truth for related word retrieval task using Baidu encyclopedia. Third, we give some example of related word retrieval task. We show that more related words can be returned earlier if we consider semantic features. Finally, we make further analysis of the parameter tuning mentioned before.

4.1 Experiment Settings

We carry out our experiment on Sogou Chinese input method dataset. The dataset contains 10,000 users and 183,870 words, and the number of edges in the constructed bipartite graph is 42,250,718. As we can see, the dataset is quite sparse, because most of the users tend to use only a small number of words.

For related word retrieval task, we need to judge whether a candidate word is related to the input seed word. We can ask domain experts to answer this question. However, it needs a lot of manual efforts. To alleviate this problem, we adopt Baidu encyclopedia (Baidu, 2006) as our ground truth. In Baidu encyclopedia, volunteers give a set of words that are related to the particular seed word. As related words are provided by human, we are confident enough to use them as our ground truth.

We randomly select 2,000 seed words as our validation set. However, whether two words are related is quite subjective. In this paper, Baidu encyclopedia is only used as a relatively accurate standard for evaluation. We just want to investigate whether user behaviors and re-ranking framework is helpful in the related word retrieval task under various evaluation metrics.

We give a simple example of our method in Table 1. The input seed word is "机器学习" (Machine Learning). Generally speaking, all these returned candidate words are relevant to the seed word to certain degree, which indicates the effectiveness of our method.

特征向量(feature vector)	核函数(kernel function)
训练集(training set)	决策树(decision tree)
分类器(classifier)	测试集(test set)
降维(dimension reduction)	特征提取(feature extraction)

Table 1. Words Related to "Machine Learning"

4.2 Evaluation Metrics

In this paper, we use three evaluation metrics to validate the performance of our method:

1. Precision@N (**P@N**). P@N measures how much percent of the topmost results returned are correct. We consider P@5 and P@10.
2. Binary preference measure (**Bpref**) (Buckley and Voorhees, 2004). As we cannot list all the related words of an input seed word, we use Bpref to evaluate our method. For an input seed word with R judged candidate words where r is a related word and n is a nonrelated word. Bpref is defined as follow:

$$Bpref = \frac{1}{R}\sum_{r} 1 - \frac{|\, n \text{ ranked higher than } r \,|}{R} \quad (3)$$

3. Mean reciprocal rank of the first retrieved result (**MRR**). For a sample of input seed words W, rank_i is the rank of the first related candidate word for the input seed word w_i, MRR is the average of the reciprocal ranks of results, which is defined as follow:

$$MRR = \frac{1}{|W|}\sum_{i} \frac{1}{rank_i} \quad (4)$$

4.3 Candidate Re-ranking

In order to show the effectiveness of semantic features and re-ranking framework, we give an example in Table 2. The input seed word is "爱立信" (Ericsson), and if we only take user behaviors into consideration, top 5 words returned are shown on the left side. After using search engine and semantic representation, we reorder the candidate words as shown on the right side.

Input Seed Word: 爱立信 (Ericsson)	
Top 5 Candidates	After Re-ranking
北电 (Nortel)	索尼爱立信 (Sony Ericsson)
中兴 (ZTE Corporation)	索爱 (Sony Ericsson)
基站 (Base Station)	阿尔卡特 (Alcatel)
阿尔卡特 (Alcatel)	索尼 (Sony)
核心网 (Core Network)	华为 (Huawei)

Table 2. Candidate Re-ranking

As shown in Table 2, we can clearly see that we return the most related candidate words such as "索尼爱立信" (Sony Ericsson) and "索爱" (the abbreviation of Sony Ericsson in Chinese) in the first two places. Moreover, after re-ranking, top candidate words are some famous brands that are quite related to query word "爱立信" (Ericsson). Some words like "核心网" (Core Network) that are not quite related to the query word are removed from the top list. From this observation, we can see that semantic features and re-ranking framework can improve the performance.

4.4 Parameter Tuning

As discussed in Section 3, we have introduced two parameters in this paper. The first is the parameter λ in the candidate generation step, and the other is the parameter *N* in the re-ranking step. We show how these two parameters affect the performance. In addition, we should emphasize that the ground truth is not a complete answer, so all the results are only useful for comparisons. The absolute value is not very meaningful.

As we have shown in Section 3.2, parameter λ adjusts the weight of conditional probability between two word *i, j*. The parameter λ is varied from 0 to 1 stepped by 0.1. We record the corresponding values of P@5, P@10, Bpref and MRR. The results are shown in Figure 3.

We can clearly see that all the values increase when λ increases first. And then all the values decrease dramatically when λ is close to 1. This indicates that either *P(j|i)* or *P(i|j)* being too small is a severe detriment. The result reaches peak value when λ=0.5, i.e. we should treat *P(j|i)* and *P(i|j)*equally to get the best result. Therefore, we use λ=0.5 to generate candidate words, those candidates are used for re-ranking.

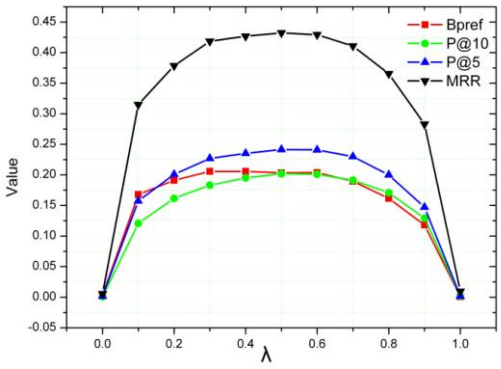

Fig. 3. Parameter λ for Candidate Generation

We also carry out the comparisons with Bayesian Sets, which is shown in Table 3. It is clear that our method gains better results than Bayesian Sets with different values of parameter λ. Results of Google Sets are omitted here because Zheng et al. (2009) have already showed that Google Sets performs worse than Bayesian Sets with query words in Chinese.

	Bpref	MRR	P@5	P@10
λ = 0.4	**0.2057**	0.4267	0.2352	0.195
λ = 0.5	0.2035	**0.4322**	**0.2414**	**0.2019**
λ = 0.6	0.2038	0.4292	0.2408	0.2009
Bayesian Sets	0.2033	0.3291	0.1842	0.1512

Table 3. Comparisons with Bayesian Sets

To investigate the effectiveness of re-ranking framework, we also conduct experiments on the parameter *N* that is used for re-ranking. The experimental results are shown in Figure 4.

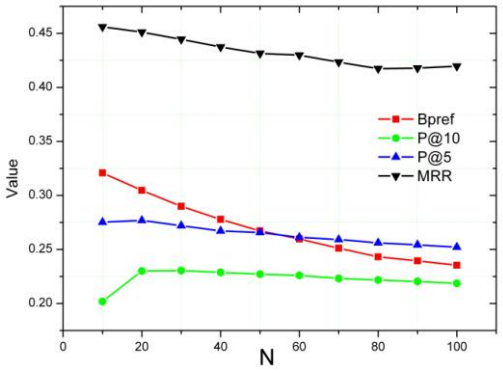

Fig. 4. Top *N* Candidates for Re-ranking

We can observe that more candidates tend to harm the performance as noisy words are introduced inevitably. For example, Bpref drops to less than 0.25 when N = 100. More comparative results are shown in Table 4. We can see that N = 20 gives relatively best results, which indicates that we should select Top 20 candidate words for re-ranking.

	Bpref	MRR	P@5	P@10
Non Re-ranking	0.2035	0.4322	0.2414	0.2019
N = 10	**0.3208**	**0.456**	0.2752	0.2019
N = 20	0.3047	0.4511	**0.2769**	0.2301
N = 30	0.2899	0.4444	0.272	**0.2305**

Table 4. Comparisons with Re-ranking Method

5 Conclusions and Future Work

In this paper, we have proposed a novel method for related word retrieval task. Different from other method, we consider user behaviors, semantic features and re-ranking framework together. We make a reasonable assumption that if two words always co-occur in user records, then

they tend to have a close relationship with each other. Based on this assumption, we first generate a set of candidate words that are related to an input seed word via user behaviors. Second, we utilize search engine to enrich candidates with semantic features. Finally, we can reorder the candidate words to return more related candidates earlier. Experiment results show that our method is effective and gains better results.

However, we also observed some noisy words in the returned results. As our dataset is generated from Chinese input method, users can type whatever they want, which will bring some noise in the dataset. We plan to remove noisy words in the future. Furthermore, we want to take the advantage of learning to rank literature (Liu, 2009) to further improve the performance of related word retrieval task. We may need to extract more features to represent the word pairs and build a labeled training set. Then various machine learning techniques can be used in this task.

Another important issue is how to build a complete and accurate ground truth for related word retrieval task. People may have different opinions about whether two words are related or not, which makes this problem complicate.

Thirdly, our method can only process a single seed word, so we aim to extend our method to process multiple seed words. In addition, we want to build a network of Chinese word association. We can discover how words are organized and connected within this network. And this word association network will be quite useful for foreigners to learn Chinese.

Fourthly, how to deal with ambiguous query word is also left as our future work. For example, query word "apple" can refer to a kind of fruit or an IT company. As a result, we are expected to return two groups of related words instead of mixing them together.

Finally, our dataset provides a new perspective for many interesting research tasks like new word detection, social network analysis, user behavior analysis, and so on. We are trying to release our dataset for research use in the future.

Acknowledgement

We thank Xiance Si and Wufeng Ke for providing the Baidu encyclopedia corpus for evaluation. We also thank the anonymous reviewers for their helpful comments and suggestions. This work is supported by a Tsinghua-Sogou joint research project.

References

Baidu. 2006. Baidu Encyclopedia. Available at http://baike.baidu.com

Chris Buckley and Ellen M. Voorhees. 2004. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp 25-32

Mukund Deshpande and George Karypis. 2004. Item-Based Top-N Recommendation Algorithms, *ACM Trans. Information Systems*, 22(1): 143-177

Zoubin Ghahramani and Katherine A. Heller. 2005. Bayesian Sets. In *Advances in Neural Information Processing Systems*

Google. Google Sets. Accessed on Feb. 9th, 2010, available at: http://labs.google.com/sets

Jingyang Li and Maosong Sun. 2007. Scalable term selection for text categorization, In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 774-782

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval, *Foundation and Trends on Information Retrieval*, Now Publishers

Donald Metzler, Susan T. Dumais, and Christopher Meek. 2007. Similarity measures for short segments of text. In *Proceeding of the 29th European Conference on Information Retrieval*, pp 16-27

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, pp 377-386

Sogou. 2006. Sogou Chinese Pinyin Input Method. Available at http://pinyin.sogou.com/

Sogou. 2004. Sogou Search Engine. Available at http://www.sogou.com

Wen-Tau Yih and Christopher Meek. 2007. Improving similarity measures for short segments of text. In *Proceedings of AAAI 2007*, pp 1489-1494

Yabin Zheng, Zhiyuan Liu, Maosong Sun, Liyun Ru, and Yang Zhang. 2009. Incorporating User Behaviors in New Word Detection. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pp 2101-2106