User Behaviors in Related Word Retrieval and New Word Detection: A Collaborative Perspective

ZHIYUAN LIU*, YABIN ZHENG*, LIXING XIE, MAOSONG SUN, LIYUN RU,
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
and Yang Zhang, Sohu Inc. R&D center

Nowadays, user behavior analysis and collaborative filtering have drawn a large body of research in the machine learning community. The goal is either to enhance the user experience or discover useful information hidden in the data. In this paper, we conduct extensive experiments on a Chinese input method data set, which keeps the word lists that users have used. Then, from the collaborative perspective, we aim to solve two tasks in natural language processing, i.e., related word retrieval and new word detection. Motivated by the observation that two words are usually highly related to each other if they co-occur frequently in users' records, we propose a novel semantic relatedness measure between words that takes both user behaviors and collaborative filtering into consideration. We utilize this measure to perform related word retrieval and new word detection tasks. Experimental results on both tasks indicate the applicability and effectiveness of our method.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: Probabilistic Algorithms; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms: Algorithms

Additional Key Words and Phrases: Related words retrieval, New word detection, Collaborative filtering, User behaviors, Natural language processing

ACM Reference Format:
Liu, Z., Zheng, Y., Xie, L., Sun, M., Ru, L., and Zhang, Y. 2011. User Behaviors in Related Word Retrieval and New Word Detection: A Collaborative Perspective. ACM Trans. Asian Language Inform. Process. 9, 4, Article 39 (March 2010), 24 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

What is the relationship between "Information Retrieval (IR)" and "Artificial Intelligence (AI)"? We might regard "IR" as a research branch of "AI". And which word is more related to "IR", "Data Mining" or "Genetic Algorithm"? We might think that the former is more related. A problem arises when we want to find more words that are related to the seed word IR. And the more related words should be returned earlier.

Zhiyuan Liu* and Yabin Zheng* contributed equally to this work. This work is supported by a Tsinghua-Sogou joint research project and National Natural Science Foundation of China under Grant No. 60873174.
Author's address: Zhiyuan Liu, Yabin Zheng, Lixing Xie, Maosong Sun and Liyun Ru, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, email: {lzy.thu, yabin.zheng, lavender087}@gmail.com, sms@tsinghua.edu.cn, riliyun@sohu-rd.com. Yang Zhang, Sohu Inc. R&D center, Beijing 100084, China, email: zhangyang@sohu-rd.com.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

We regard this problem as a **related word retrieval** task [Google 2010; Ghahramani and Heller 2006; Sarmento et al. 2007; Zheng et al. 2010]. For this task, Google Sets [2010] provides an interesting tool which takes one or several words as input, and returns some related words as output. In general, it performs a large-scale clustering algorithm to gather the related words. Due to its proprietary issue, we are not able to describe more details.

Recently, with the development of social network, such as facebook and twitter, new words are generated at a great speed. How to efficiently and correctly identify these new words becomes an important task. For **new word detection** task [Zheng et al. 2009; Zhang et al. 2002; Peng et al. 2004; Li et al. 2004; Chen and Bai 1998; Wang et al. 1995], we have some domain-specific lexicons at hand, and want to find some new words that are related to the specific domain but are not included in the original lexicons. These words are regarded as new words. Chinese new word detection is even more challenging because there are no natural separators between Chinese words. Previous works [Sproat and Emerson 2003] show that new words also have a crucial impact on the performance of Chinese word segmentation task.

In this paper, we want to investigate the advantage of exploiting user behaviors in Chinese Pinyin input method to measure the semantic relatedness between words, and then solve the related word retrieval and new word detection tasks. User records in Chinese Pinyin input method keep the words that users have used. As shown in Figure 1, $User_1$ has used two words $Word_1$ and $Word_2$. From this perspective, we can find connections between user records and typical collaborative filtering systems. Words in user records can be regarded as items in collaborative filtering systems and described using a bipartite graph. If a user buys an item or uses a word, we can add a link between them in the bipartite graph. Intuitively, if two words are highly related with each other, they tend to co-occur frequently in user records. Thus, we can quantify the the semantic relatedness between words through user behaviors and solve the related word retrieval task. Our method for related word retrieval is unsupervised without requiring any labeled data.

New words are created and used by users. As a result, we can detect new words from user records. We first utilize some domain-specific lexicons to find potential experts who use these domain-specific terminologies very frequently. Then, we detect domain-specific new words from these potential experts. The underlying idea is that experts tend to have the same tastes on certain domain-specific words, while common users seldom use them. In other words, these domain-specific words tend to appear more frequently among the potential experts than other users. We regard these words as the candidate new words. Our method for new word detection is weakly-supervised as we only require domain-specific lexicons. Experimental results indicate that it is reasonable and applicable to incorporate user behaviors and collaborative filtering in both tasks.

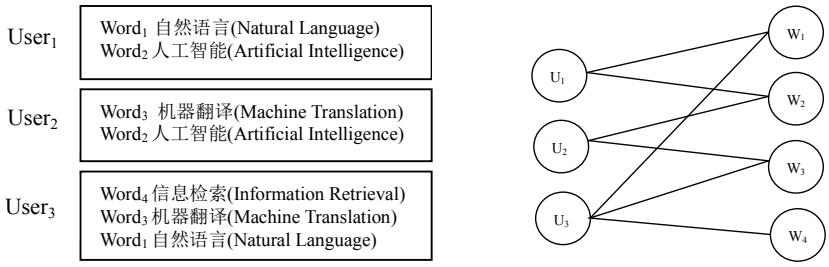

Fig. 1. User Records in Chinese Pinyin input method.

The contributions of this paper are threefold. First, we introduce user behaviors and collaborative filtering in the related word retrieval and new word detection tasks. Following the typical settings of collaborative filtering systems, we construct a User-Word bipartite graph using user records, which is different from other methods. Second, we define a semantic relatedness measure between words based on the constructed bipartite graph. We make a straightforward assumption that if two words are frequently used by the same user, they are very related with each other. And finally, our method is language-independent, which can be easily extended to other languages.

The rest of this paper is organized as follows. Related works about related word retrieval, new word detection, and collaborative filtering are discussed in section 2. We introduce the data set used in this paper as well as our method for related word retrieval in section 3. Then, we demonstrate our method for new word detection task in section 4. Experimental results for both tasks and discussions are shown in section 5. Finally, we present our conclusions and future work in section 6.

2. RELATED WORK

2.1. Related Word Retrieval

Given one or several words, how can you find other words that are related to the input query words in an ordered list? For this kind of related word retrieval task, Google Sets [2010] is a leading and interesting application. For input words in English, Google Sets can return reasonable results. As mentioned in [Zheng et al. 2009], the performance drops rapidly for input words in Chinese. Bayesian Sets [2006] solves the same problem under the framework of Bayesian inference. It computes a score for each candidate word by comparing the posterior probability of the word given the input words, to the prior probability of that candidate word. Bayesian Sets returns a ranked list of candidate words according to their semantic relatedness with the input words.

Some researchers pay their attentions to the entity set expansion task [Wang and Cohen 2007; 2008; 2009; Paşca et al. 2006; Van Durme and Paşca 2008; Etzioni et al. 2005; Paşca and Van Durme 2008; Paşca 2007; Pantel et al. 2009]. Entity set expansion is the task of finding related words that belong to the same entity class. For example, from the seed words "Toyota" and "Ford", we can expand the entity class to find other car brands. Most approaches harvest related words belonging to the same entity class from the Web. Some approaches [Wang and Cohen 2007; 2008; 2009] expand the target entity class from semi-structured documents written in markup languages (e.g. HTML, XML). Other approaches adopt distributional similarity [Pantel et al. 2009; Van Durme and Paşca 2008; Paşca et al. 2006] to find more entity words. The observation is that if two words belong to the same entity class, they tend to have similar distributional contexts. Finally, query logs from the search engine [Paşca and Van Durme 2008; Paşca 2007] provide a novel and effective perspective to entity set expansion.

The entity linking task [McNamee and Dang 2009; Ji et al. 2010] organized by NIST, aims to map named entities in texts to corresponding entries stored in an existing knowledge base. This task also needs to compute similarities between words. Recently, researchers propose to model the relationships between two texts that essentially have the same meaning, which is a paraphrase discovery task [Bannard and Callison-Burch 2005; Heilman and Smith 2010; Lin et al. 2010]. In paraphrase discovery, it is crucial to decide whether two pieces of superficially distinct texts indicate the same thing.

However, the entity set expansion and entity linking tasks only aim to harvest named entities like person, location and organization names. In this paper, we aim to harvest other kinds of words, not only named entities. Paraphrases refer to linguistic expressions that convey the same meaning. We focus on finding related words rather than words with the same meanings, which is a more general task.

We can formulate the related word retrieval task as the definition of semantic relatedness between words, and return the candidate words with higher semantic relatedness earlier. Recently, Zheng et al. [2009] introduce user behaviors in related word retrieval task via a collaborative filtering manner. The basic assumption is that the more frequently two words co-occur in user records, the more related they are.

However, our previous method [Zheng et al. 2009] is still preliminary. We only rank the candidate words based on their co-occurrence features on user records. In this paper, we aim to exploit context features to enhance the performance. First, we utilize co-occurrence features to produce candidate words. Then, we employ the idea of re-ranking to reorder these candidate words using context features extracted from external resources.

As discussed before, we can treat related word retrieval task as measuring the semantic relatedness between words. Word pairs can be regarded as pairs of very short texts like queries in search engine. Researchers try to propose accurate measures to define similarities between short texts, which is useful in query expansion [Riezler et al. 2008] and query suggestion tasks [Jones et al. 2006].

Sahami and Helman [2006] propose a web kernel function as a semantic relatedness measure between short texts based on snippets returned by search engines. The returned snippets of a search engine are regarded as enriched context features of the short text. This method is especially effective when the query words have no common terms. For example, the query word "SVM" and "support vector machine" indicate the same concept although they do not share any common word in the surface. But if we enter them into a search engine, the returned snippets are similar. Thus we can find that the two words are related by their context features. This work is followed by Metzler et al., [2007] Yih and Meek [2007]. They combine the web kernel with other similarity metrics, such as Dice Coefficient, Jaccard Coefficient and KL Divergence. Yih and Meek [2007] take some machine learning techniques into account to further improve the performance.

In this paper, we follow the idea of using search engine to enrich the context features of word pairs, and define a better semantic relatedness between words. We regard the returned snippets as the context of the query word and use vector space model [Salton et al. 1975] to measure the semantic relatedness between words.

In brief, we want to investigate the effectiveness and flexibility of user behaviors and re-ranking framework in the related word retrieval task. User behaviors are used to generate candidate words based on statistical features. Then we reorder the candidate words using context features and expect that more related candidate words can be retrieved earlier. Our related word retrieval algorithm can also process multiple seed words.

2.2. New Word Detection

Since Chinese does not have natural separators between words, segmentation is very important for Chinese text processing [Sproat and Emerson 2003]. Previous works [Zhang et al. 2002; Peng et al. 2004] indicate that the new word detection can be applied simultaneously with Chinese word segmentation to achieve better results on both tasks. In the past, many techniques have been proposed to solve the new word detection problem. We roughly classify these methods into two categories: rule-based methods and statistical machine learning methods.

As for rule-based methods, Wang et al. [Wang et al. 1995] use an n-grams based approach to identify and classify Chinese unknown words. Their approach adopts n-grams to obtain the collocating word sequences that are possible new words in Chinese. Some structural and semantic characteristics are introduced to make their approach more robust. In [Chen and Bai 1998], the authors present a corpus-based method that

derives sets of syntactic rules to identify new words. The advantage of their method is that it can perform automatic rule learning and evaluation of each generated rule. Experimental results show that their proposed method outperforms hand-crafted rules created by human experts.

Researchers in the statistical machine learning community have paid their attentions to the new word detection problem. Li et al. [2004] regard the new word detection task as a binary classification problem and use SVM classifier to identify the candidate words as new words or not. Meanwhile, various features, including the analogy between new words and lexicon words, in-word probability of a character, frequency in documents and anti-word list, have been extracted to solve empirically two types of new word where F-scores can achieve 64.4% and 54.7% respectively. In [Peng et al. 2004], they bring conditional random field into Chinese new word identification and integrate new word detection in the framework of Chinese word segmentation. They prove that new word detection and Chinese word segmentation can promote each other. In [Li et al. 2004], new words are identified according to their component tokens and context tokens. And then, they define a set of word roles to detect new words in real texts based on role tagging. Their method achieves acceptable results, especially for transliterations and person names.

Different from these previous methods, we incorporate user behaviors in the new word detection task. New words in this paper refer to the words that should be included in certain lexicons but are excluded, which is analogous to the definition of out-of-vocabulary words. Our method is weakly-supervised and follows a straightforward collaborative way, which is easy to understand and performs well.

2.3. Collaborative Filtering

Recommendation systems and collaborative filtering [Adomavicius and Tuzhilin 2005] have become a hot research topic since last decade. Recommendation systems are trying to predict the interests of users according to their historical tastes, and then recommend useful items that the users might be interested in. According to [Breese et al. 1998], collaborative filtering algorithms can be roughly divided into two categories: model-based and memory-based methods.

Model-based methods [Breese et al. 1998; Hofmann 2003; Deshpande and Karypis 2004] build a model using the collections of rating or purchasing records to discover the relations between different items and use the trained model to determine which items should be recommended. Item-based top-N recommendation algorithm [Deshpande and Karypis 2004] belongs to model-based methods. It analyzes the similarities between items and then recommends the items that are similar to the previous items that the user has purchased. We need to find a good similarity metric between items. This metric can be adopted in related word retrieval task if we treat words as items. Then, we utilize the idea of model-based methods in the related word retrieval task.

Memory-based methods [Nakamura and Abe 1998; Breese et al. 1998] make recommendations based on the previous rating records by users. The basic assumption is that each user can be classified in a large group of similar tastes and users have similar tastes tend to choose the same item in future. As a result, when the system tries to predict some items to a user, items frequently purchased or liked by the members of the same group can be used to generate candidates for recommendation. The system first searches similar users based on the historical rating records, which is an item-to-user step. The next step is to recommend items liked by most of those members, which is a user-to-item step.

We employ the idea of this method to perform new word detection task. Words in the user records are treated as items. If a user types in a word, we can consider that a user purchase an item. From this perspective, our settings are similar to typical

collaborative filtering systems. For the new word detection task, we have some domain-specific lexicons at hand. And we want to detect new words in that domain which are not included in the original lexicons. First, we find some potential experts who have typed in many terminologies in that domain. This can be recognized as an item-to-user step. Then we detect the candidate words that are used by most of those experts. This is a user-to-item step. It is clear that our method for new word detection follows the collaborative filtering methodology.

3. UNSUPERVISED RELATED WORD RETRIEVAL

In this section, we demonstrate how to discover related words from one or several seed words through user behaviors. The basic observation is that high association ratios or co-occurrence frequencies indicate a strong semantic relationship between words [Church and Hanks 1990]. In this paper, we adopt five co-occurrence measures to calculate semantic relatedness between words, namely Jaccard coefficient, Dice coefficient, Point-wise mutual information [Bollegala et al. 2007], Google Distance [Gracia et al. 2006] and conditional probability [Deshpande and Karypis 2004].

First, in order to give a better explanation, we introduce the data set used in this paper. All the resources used here are generated from Sogou Chinese Pinyin input method editor. We use Sogou-Pinyin for abbreviation hereafter. Users can use Sogou-Pinyin to type in Chinese words. The word lists that users have typed in are kept in their user records. Volunteers are encouraged to upload their anonymous user records to servers. The advantage is that they can download user records and personal settings on a different computer, which can greatly enhance user experience. In order to maintain user privacy, user information is hidden using MD5 hash algorithm. So, the data set is completely anonymous and user privacy is safeguarded.

We also introduce how to formulate related word retrieval in a typical collaborative filtering framework. We construct a User-Word bipartite graph with users on one side and words they have used on the other side. The construction can be accomplished while traversing the data set within linear time.

Second, we focus on how to perform the unsupervised related word retrieval task under this framework. Five co-occurrence based measures are proposed to generate candidate words. Intuitively, if users tend to use two words very frequently, i.e. two words always co-occur in user records, the two words are related with each other. Starting from a single seed word, we can identify candidate words using different measures.

Third, in order to further improve the performance, we can extract context features to supplement statistical features. We follow the methods introduced in [Bollegala et al. 2007; Sahami and Heilman 2006]. Search engine is used to extract context features for input seed words and candidate words. Each word is issued to a search engine, then the top 20 returned snippets are regarded as context features of the corresponding word. We can use vector space model and cosine distance to measure the similarities between two words.

Finally, we extend our method from a single seed word to multiple seed words. A candidate word is related to input seed words if it has close relationships with most of seed words. We make further explanations about the above steps in the following subsections.

3.1. Bipartite Graph Construction

In a typical collaborative filtering system, the recommendation problem is usually formalized in a bipartite graph, with users on one side and items on the other side. If an item is purchased or rated by a user, then we can add one edge between them in the bipartite graph.

Similarly, we can build a bipartite graph from user records in Figure 1. The bipartite graph has two layers, with users on one side and words they have used 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. This procedure can be accomplished in a linear time. As shown in Figure 1, we can see that $User_3$ has used $Word_1$, $Word_3$ and $Word_4$. As a result, node U_3 is connected with node W_1, W_3 and W_4 in the corresponding bipartite graph.

3.2. Candidate Words Generation

After building the bipartite graph, we can measure the semantic relatedness between two words from corresponding graph. Intuitively, two words are related if they always co-occur in user records. However, co-occurrence information alone does not accurately define semantic relatedness between words. For example, some stop words like "的" (of) and "我" (I) in Chinese tend to co-occur frequently with other words. As a result, we should consider not only the co-occurrence of two words w_1 and w_2, but also the individual occurrence of word w_1 and w_2 to assess the semantic relatedness between w_1 and w_2.

In this paper, we utilize five co-occurrence based measures: Jaccard coefficient, Dice coefficient, Point-wise mutual information, Google Distance and conditional probability to calculate semantic relatedness between two words. We use notation $Score(w_1, w_2)$ to denote the semantic relatedness score between words w_1 and w_2. $Freq(X)$ denotes the number of users who have used *all* the words in the set X.

The Jaccard coefficient is used to calculate the similarity between two sets. $Jacc(w_1, w_2)$ is defined as

$$Jacc(w_1, w_2) = \frac{Freq(w_1, w_2)}{Freq(w_1) + Freq(w_2) - Freq(w_1, w_2)}. \tag{1}$$

$Freq(w_1, w_2)$ denotes the number of users who have used both words, which is the estimation of co-occurrence of w_1 and w_2. Formula (1) defines the ratio of the probability of finding a user who have used words w_1 and w_2 together over the probability of finding a user who have used either w_1 or w_2. If w_1 and w_2 are the same word, $Jacc(w_1, w_2)$ is equal to 1. If w_1 and w_2 never co-occur, $Jacc(w_1, w_2)$ is equal to 0.

Dice coefficient $Dice(w_1, w_2)$ is similar to the Jaccard coefficient and is calculated in Formula (2). Again, if w_1 and w_2 never co-occur, $Dice(w_1, w_2)$ is equal to 0. And $Dice(w_1, w_2)$ is equal to 1 if w_1 and w_2 are identical.

$$Dice(w_1, w_2) = \frac{2Freq(w_1, w_2)}{Freq(w_1) + Freq(w_2)}. \tag{2}$$

We can regard the number of users who have used word w_1 and w_2 as random variables. Then, point-wise mutual information can be adopted to measure the mutual dependence between the two words using the following Formula:

$$PMI(w_1, w_2) = \log \frac{\frac{Freq(w_1, w_2)}{U}}{\frac{Freq(w_1)}{U}\frac{Freq(w_2)}{U}}, \tag{3}$$

where U is the number of users. Mutual information quantifies the mutual dependence of two random variables and can be regarded as the reduction of uncertainty about one random variable when another is known. Low mutual information indicates a small reduction, i.e. two random variables are less related. Zero mutual information means two random variables are independent. High mutual information indicates a large reduction in uncertainty, i.e. two random variables are highly related. Mutual information reaches its maximum when two random variables are identical.

Inspired by Kolmogorov complexity, a page-count-based similarity measure called Normalized Google Distance [Cilibrasi and Vitanyi 2007; Gracia et al. 2006] is proposed to compute the semantic relatedness between words. Google distance based semantic relatedness measure $Google(w_1, w_2)$ is defined as

$$Google(w_1, w_2) = e^{-2\frac{max\{\log Freq(w_1), \log Freq(w_2)\} - \log Freq(w_1, w_2)}{\log U - min\{\log Freq(w_1), \log Freq(w_2)\}}} \tag{4}$$

Finally, we adopt the conditional probability [Deshpande and Karypis 2004] to measure the semantic relatedness between words. The conditional probability $Prob(w_2|w_1)$ can be estimated as the number of users who have used both w_1 and w_2 divided by the number of users who have used w_1,

$$Prob(w_2|w_1) = \frac{Freq(w_1, w_2)}{Freq(w_1)}. \tag{5}$$

We can clearly see that usually $Prob(w_2|w_1) \neq Prob(w_1|w_2)$, which indicates that conditional probability is an asymmetric measure. Suppose w_2 is a stop word that is used by most users, then all the other words tend to have a close relationship with w_1 according to Formula (5). In order to alleviate this disadvantage, we use a *weighted harmonic averaging* measure [Li and Sun 2007; Forman 2003] to consider both $Prob(w_2|w_1)$ and $Prob(w_1|w_2)$ together. Words w_1 and w_2 are expected to be highly related to each other when $Prob(w_2|w_1)$ and $Prob(w_1|w_2)$ are both relatively high, either $Prob(w_2|w_1)$ or $Prob(w_1|w_2)$ being too small is a severe detriment. The conditional probability based semantic relatedness measure $Cond(w_1, w_2)$ is defined as

$$Cond(w_1, w_2) = \Big(\frac{\lambda}{Prob(w_1|w_2)} + \frac{1-\lambda}{Prob(w_2|w_1)}\Big)^{-1}. \tag{6}$$

In Formula (6), parameter $\lambda \in [0, 1]$ is the weight for $Prob(w_1|w_2)$, which controls how much $Prob(w_1|w_2)$ should be emphasized. We carry out comparative experiments when parameter λ changes from 0 to 1, stepped by 0.1. Experimental results show that we can get best performance when $\lambda = 0.5$, which indicates that we should consider $Prob(w_2|w_1)$ and $Prob(w_1|w_2)$ equally.

So far, we have introduced how to compute the semantic relatedness between two words using various measures. We also want to investigate whether we can combine these co-occurrence based measures to obtain better performance, i.e., these measures could benefit from each other. More details and discussions are given in the experiment section.

For a single input seed word w, we can calculate the semantic relatedness score $Score(w, c)$ between seed word w and candidate word c. We can easily extend our method to multiple input seed words. For multiple input seed words, we calculate the semantic relatedness score between candidate word c and each seed word w_i. Then the final score of candidate word c is assigned with the average score values, as shown in the following Formula:

$$Score(c) = \frac{1}{M}\sum_i Score(w_i, c), \tag{7}$$

where M is the number of input seed words. $Score(w_i, c)$ can be computed using co-occurrence based measures discussed before. Then we sort these candidate words in a descending order. Finally, Top N candidate words are returned. Alternatively, we can also set a threshold score for $Score(c)$ and only keep those candidate words whose $Score(c)$ are higher than a threshold. However, this threshold is difficult to decide because different input seed words may have different score thresholds.

We can see that this candidate generation step is completely based on statistical features because we only consider the co-occurrence of two words and their individual occurrences. Inspired by [Sahami and Heilman 2006; Bollegala et al. 2007], we investigate whether the context features extracted from search engine can be a complement of our statistical method.

3.3. Context Feature Extraction and Re-ranking

As stated before, for one or several input seed words, we can generate top N related candidate words using statistical features. To enhance the performance, we use search engines to enrich seed words and candidate words with more context features. Context is proved to be a good feature for semantic similarity computation [Iosif and Potamianos 2010; Pantel et al. 2009]. The basic assumption is that two words appearing in similar contexts tend to be close in meaning. Then, seed words and candidates can be enriched with their corresponding context features [Church and Hanks 1990]. These candidate words are reordered according to their context features.

Intuitively, if we introduce more candidate words (increase the value of N), we are more likely to find related words from the candidate sets. However, noisy words are introduced inevitably. We show how parameter N affects the performance in the experiment part.

Specifically, we enter a word to Chinese search engine Sogou, and get top 20 returned snippets. For different query words, Sogou returns snippets of different lengths. Generally, each snippet contains about 100 Chinese characters. We show some snippet examples in Figure 2. We get context features for query word "爱立信" (Ericsson) using the returned snippets.

We assume that top 20 snippets contain enough context features about query word, and more snippets will bring some noise. Following the method proposed by Sahami and Helman [2006], each query word can be represented as a feature vector using bag-of-words model. We calculate a new semantic relatedness score between input seed words and a candidate word using the cosine distance between their feature vectors. Finally, top N candidate words are reordered according to the semantic relatedness scores.

Fig. 2. Snippets returned by Sogou for query word "爱立信" (Ericsson).

Notice that we only utilize snippets as features in re-ranking. Our re-ranking method is unsupervised, as we do not require any training data set. More features can be explored to enhance the performance, and we can also build labeled training data to adopt more sophisticated re-ranking framework. This is our future work.

As a result, re-ranking can be regarded as a complementary step after candidate generation. Candidate words that are more related to seed words can be returned earlier with enriched context features. As shown in the experiment part, we can enhance the performance of related word retrieval task by performing re-ranking.

However, we also observe some noisy results in related word retrieval experiments. The main reason is that our method is unsupervised, we do not have any prior knowledge about input seed words. For instance, for ambiguous seed word "苹果" (apple), we do not know whether it is a kind of fruit or an IT company. We can add labeled domain-specific knowledge to solve this ambiguous problem. Starting from the domain-specific lexicons which contain all kinds of fruit names, "苹果" (apple) in this lexicon refers to fruit. When "苹果" (apple) is included in computer science related lexicon, it means an IT company name. From this perspective, we can extend our unsupervised related word retrieval method to a weakly-supervised new word detection task. Given labeled domain-specific lexicons, our goal is to find related new words that are excluded in original lexicons.

4. WEAKLY-SUPERVISED NEW WORD DETECTION

In this section, we present our method for the new word detection task which takes user behavior and collaborative filtering into consideration. New word in this paper refers to word that should be included in specific lexicons but are excluded. In other words, we need to retrieve missing words that are related to the words included in specific lexicons. From this perspective, new word detection is similar to the related word retrieval task. Both tasks need to calculate semantic relatedness between words. The main difference is that, for new word detection task, we need to construct lexicons that contain words belong to the same domain, such as "computer science".

When using Chinese input method, users may need some domain-specific lexicons to help them type in terminologies more efficiently and accurately. For example, computer science researchers will be happy if they are provided with a lexicon that contains the latest terminologies in computer science. Fortunately, Sogou-Pinyin provides various domain-specific lexicons in different areas. From their configuration files, we can obtain the information of what kind of domain-specific lexicons users have used. Like the mode of Wikipedia, these lexicons are maintained by volunteers, which can keep terminologies up-to-date. However, this can also bring noisy words inevitably because everyone can modify them.

As discussed before, our new word detection algorithm mainly performs in three steps. First, we select the most representative words from domain-specific lexicons and remove the noisy words. Second, starting from the cleaned domain-specific lexicons, we aim to discover potential experts in this particular domain through their user records. In our opinion, users who use representative words quite often tend to be potential experts in that domain. In other words, the more domain-specific terminologies they have used, the more likely they are potential experts. Finally, we detect new words in that domain from the behaviors of those potential experts. The idea is that words used much more frequently in the community of potential experts than other users are candidate new words.

In brief, our new word detection method follows the collaborative filtering strategy: first from representative words to potential experts, then from these experts to find new words.

4.1. Representative Words Selection

Domain-specific lexicons are maintained by volunteers, which brings noise inevitably. Some noisy and unrelated words are included in the lexicon. For example, lexicon in computer science field contains words like "问题" (problem) and "联系" (contact). These words should be removed from domain-specific lexicons. Some words that are very related to the specific domain are excluded. In our experiment, words "复杂度" (complexity) and "递归" (recursion) are absent from the original lexicon in computer science. In fact, we are trying to identify these missing new domain-specific words to enrich the domain-specific lexicons.

In this subsection, we introduce how to filter noisy words in the lexicon, and select most representative words. A word is supposed to be representative if it is very related to specific domain (we use discriminability to denote whether a word is related to a domain) and has a wide coverage among the experts in specific domain. For example, word "假通过率" (pseudo pass rate) is a terminology in computer science. But only few persons use this word in their records because it is over specialized. Word "重定向" (redirect) is a word widely used among experts in computer science as well. As a result, word "重定向" (redirect) is more representative than "假通过率" (pseudo pass rate). Inspired by [Li and Sun 2007], we generally consider two facotrs: discriminability and coverage to select representative words.

First, given a domain-specific lexicon, we can infer whether a user has used this lexicon from his configuration file. Users who have used certain lexicon are labeled as *positive* users. Other users who have not used certain lexicon are labeled as *negative* users. A user can be labeled as either positive or negative given different domain-specific lexicons. For instance, a computer science expert is a positive user given a computer science related lexicon, while he/she is a negative user given a biology related lexicon.

Then, we adopt discriminability to measure how unbalanced the distribution of a word in positive and negative users is. If a word always appears in positive users and seldom occurs in negative users, it has a powerful discriminability that can distinguish positive users from negative users. According to [Forman 2003], we use a straightforward metric, probability ration, to define the discriminability of a word as following: (for brief, we use PR instead)

$$PR(w,l) = \frac{P(w|l_+)}{P(w|l_-)} = \frac{Freq(w,l_+)/Freq(l_+)}{Freq(w,l_-)/Freq(l_-)}, \tag{8}$$

In Formula (8), w refers to a word, $PR(w,l)$ refers to the discriminability of word w in a domain-specific lexicon, $l+$ and l_- refers to positive and negative users, respectively. $Freq(w,l_+)$ indicates the number of positive users who have used word w. $Freq(l_+)$ indicates the number of positive users. $Freq(w,l_-)$ indicates the number of negative users who have used word w. $Freq(l_-)$ indicates the number of negative users.

Second, we simply use the number of users who have used word w, i.e. $Freq(w)$ to measure the coverage of w. In general, coverage and discriminability have a slightly negative correlation. Words with high coverage always have poor discriminability. For stop words like "的" (of) and "我" (I), almost every user tends to use them in their user records. It is not reasonable to label a user as positive or negative using these words. On the other hand, if a word mainly appears in positive users which shows strong discriminability, it always has weak coverage because it rarely appears in negative users.

In this paper, we consider the discriminability and coverage together to select representative words. A representative word should have strong discriminability that can distinguish positive users from negative users. Meanwhile, it should have wide enough

coverage. If we select words that rarely appear in user records, we cannot discover enough potential experts in the next step, which will hurt the performance of the new word detection task. Therefore, we need to consider both discriminability and coverage at the same time. Similar to Formula (6), a weighted parametric representative words selection criterion is defined as follows:

$$\zeta(w, \lambda^{'}) = \Big(\frac{\lambda^{'}}{\log PR(w,l)} + \frac{1-\lambda^{'}}{\log Freq(w)}\Big)^{-1}. \tag{9}$$

$\lambda^{'} \in [0,1]$ is adopted to balance discriminability and coverage. In our experiment, we set $\lambda^{'}$ to 0.5, which means that we treat discriminability and coverage equally. Besides, we can select representative words using only discriminability or coverage. We select top 1000 most representative words from a domain-specific lexicons using three selection criterions. Experimental results show that we can get best results if we consider two facotrs together.

4.2. Potential Expert Search

In this subsection, we discuss how to search potential experts using the most representative words. Users who have used corresponding representative words very frequently are potential experts in certain field. If a user have used 90 percent of these selected representative words in computer science, we are confident enough to believe that this user is a potential expert in computer science. We sort users according to the percentages of representative words they have used.

Alternatively, we can simply mark the positive users as potential experts, because they explicitly choose domain-specific lexicons in their configuration files. However, as shown in the experiment part, we observe that many users indicate they are positive users in computer science, but they seldom use terminologies in computer science in their user records. As a result, it is not accurate if we label them as potential experts. We cannot rely on the configuration files.

We do not take all the words in a domain-specific lexicons into consideration because some words are not related to the domain. For example, original domain-specific lexicon in computer science contains noisy words like "问题" (problem) and "联系" (contact). We cannot label a user as an expert in computer science if he uses these two words very frequently.

4.3. Detecting New Words

In this subsection, we introduce how to detect new words from discovered potential experts in the previous step. As mentioned before, original domain-specific lexicons are maintained by volunteers. Obviously, humans are not able to enumerate all the terminologies in specific domain. This disadvantage causes that some words that should be included are missing. We regard these missing words as new words and try to detect them. As a result, we can make the domain-specific lexicons complete.

Intuitively, if a word w is frequently used among the community of these potential experts with high coverage and seldom appears in other users' records and w is excluded in the original domain-specific lexicon, then we can fully believe that w is a missing new word that should be detected. In other words, w can distinguish potential experts from other users.

Similar to representative words selection step, we also take coverage into account when detecting new words. Assume a word only appear *once* within one of the potential experts' record. Then, the ratio of distributions between potential experts and other users is infinite because the denominator in Formula (8) is 0. However, this word may be a personal word used by this expert, which is not a valid terminology in this domain.

To solve this problem, we consider the coverage in potential experts together. We adopt Formula (9) to sort our detected new words and evaluate our method based on the ranked results. More details are given in experiment part.

Our weakly-supervised new word detection method relies on prior domain-specific knowledge, either domain-specific lexicons or configuration files. It will fail in detecting new words in emerging new domains. To alleviate this problem, we can ask domain experts to annotate seed words that belong to the new domains. Then our method can iteratively detect new words from these seed words. How to automatically detect new domains and collect domain-specific knowledge is left as our future work.

5. EXPERIMENTAL RESULTS

5.1. Experiment Setting

We carry out our experiments on the Sogou Chinese input method data set. The data set contains 849,134 users and 165,820,037 words. We should note that users are freely to type in whatever they want in their user records. As a result, there are many noisy words in the original data set. In order to alleviate this problem, we remove the words that are used less by 10 users.

For the related word retrieval task, we randomly select 10,000 user records and construct a bipartite graph using the method described in section 3.1. The corresponding bipartite graph contains 10,000 nodes on user side, 183,870 nodes on the word side, and the number of edges is 42,250,718. As we can see, the data set is very sparse, because users tend to use only few words. For the evaluation of the related word retrieval task, we need judge whether a candidate word is related to the input seed words. The first way is to ask domain experts to do manual labeling. The second way is to use a standard data set with accurate labeling. Baidu encyclopedia, is used as our golden standard in this paper. We consider both expert labeling and Baidu encyclopedia to evaluate our method.

For the new word detection task, we use three different domain-specific lexicons which are in computer science, idiom and saying, and world of warcraft, respectively. These are the most popular domain-specific lexicons with 33,513, 47,225 and 27,622 users, respectively. The three lexicons contain 9,464, 31,758 and 20,194 domain-specific words, respectively. We show how to select the most representative words from the three lexicons. Experimental results show that it is effective if we consider the discriminability and coverage together. We find that weakly-supervised new word detection task generally performs better than unsupervised related word retrieval task, which indicates that we can get better results when given more prior knowledge.

5.2. Evaluation Metrics

We adopt three evaluation metrics to validate two tasks:

(1) Precision@N (**P@N**). P@N is the precision evaluate at a given cut-off rank N, which considers how much percent of the topmost results returned are correct. We consider P@5, P@10, P@15, P@20, P@30, P@100 and P@1000 in this paper.
(2) Binary preference measure (**Bpref**). As we cannot list all the related words of one or several input seed words or give a complete list of domain-specific words, we adopt Bpref [Buckley and Voorhees 2004] to evaluate our method with incomplete information. For an input query with R judged relevant results where r is a member of R relevant result and n is a member of the first R judge irreverent results, Bpref is defined as follows:

$$Bpref = \frac{1}{R}\sum_{r} 1 - \frac{|n\ ranked\ higher\ than\ r|}{R}. \tag{10}$$

(3) Mean reciprocal rank of the *first* relevant result (MRR). For a number of input queries Q, $rank_i$ is the rank of the first relevant result returned by Q_i. MRR is the average of the reciprocal ranks of results for input queries, MRR is defined as follows ($|Q|$ denotes the number of queries):

$$MRR = \frac{1}{|Q|} \sum_i \frac{1}{rank_i}. \tag{11}$$

All these metrics give rewards if we can return correct results earlier and give penalties if we discover wrong results earlier. Detailed experimental results are given in next subsections.

5.3. Related Word Retrieval

In this section, we show the experimental results for the unsupervised related retrieval task. We randomly select 2,000 seed words from Baidu encyclopedia for parameter tuning and automatic evaluation. In addition, we select 100 discriminative seed words in computer science for manual labeling and evaluation. We want to investigate whether discriminative seed words gain better performance than randomly selected seed words. We first investigate the related word retrieval performance under different co-occurrence based measures discussed in section 3.2. Then, we utilize search engines to extract context features and further improve the performance. Finally, we show experimental results of multiple seed words.

5.3.1. Retrieval Without Re-ranking. Before we get into the details, we first give some results of the related word retrieval task in Table I. Our method can deal with single seed word (the first column in Table I) or multiple seed words (last two columns in Table I). For an ambiguous seed word like "苹果" (apple), we can add more seed words to perform disambiguation. Intuitively, if "苹果" (apple) appears with other IT companies like "微软" (Microsoft), it refers to the IT company Apple. If it appears with other fruit names like "草莓" (strawberry), then it refers to a kind of fruit. In general, our method can return words that are relevant to the input seed words.

As discussed in section 3.2, we have introduced five co-occurrence based measures to define semantic relatedness between two words. The first four measures are parameter-free. For conditional probability, we introduce a parameter λ in Formula (6) which controls the weight of $Prob(w_1|w_2)$. In our experiment, λ varies from 0 to 1 stepped by 0.1. For each seed word selected from Baidu encyclopedia, we keep the first 100 returned candidate words ranked by Formula (6). Then we record the corresponding values of Bpref, MRR, P@5, P@10, and so on. The results are shown in Figure 3.

Table I. Related words returned by our method based on the given seed words.

Query: 机器学习 (machine learning)	Query: 苹果(Apple), 雅虎(Yahoo), 谷歌(Google) 微软(Microsoft)	Query: 苹果(apple), 西红柿(tomato), 胡萝卜(carrot), 草莓(strawberry)
特征向量(feature vector)	腾讯(Tencent)	鸡蛋(egg)
降维(dimension reduction)	正版(authorized copy)	黄瓜(cucumber)
训练集(training set)	新浪(Sina)	牛奶(milk)
支持向量机(SVM)	盗版(pirated copy)	水果(fruit)
分类器(classifier)	浏览器(browser)	巧克力(chocolate)
测试集(test set)	评论(comment)	香蕉(banana)
核函数(kernel function)	搜狗(Sogou)	蛋糕(cake)
特征提取(feature extraction)	输入法(input method)	西瓜(watermelon)
召回率(recall)	拼音(Pinyin)	牛肉(beef)
最近邻(nearest neighbor)	原版(original copy)	脖子(neck)

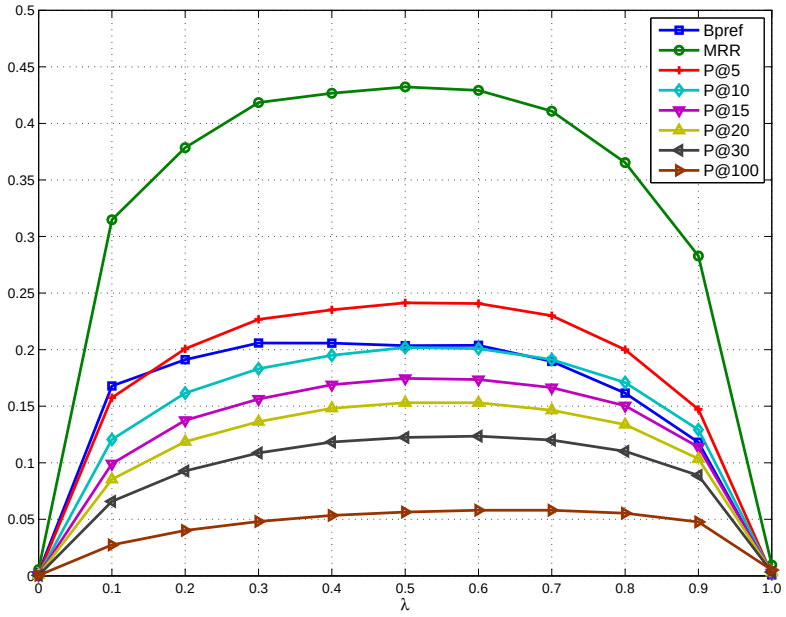

Fig. 3. Parameter λ in conditional probability for related word retrieval.

Table II. Performance with different co-occurrence based measures.

Measure	Bpref	MRR	P@5	P@10	P@15	P@20	P@30	P@100
Jaccard	0.2035	0.4322	0.2414	0.2019	0.1746	0.1530	0.1224	0.0564
Dice	0.2035	0.4322	0.2414	0.2019	0.1746	0.1530	0.1224	0.0564
PMI	0.0009	0.0096	0.0015	0.0020	0.0026	0.0031	0.0034	0.0051
$\lambda = 1.0$	0.0009	0.0096	0.0015	0.0020	0.0026	0.0031	0.0034	0.0051
Google	0.1806	0.3964	0.2174	0.1820	0.1565	0.1372	0.1096	0.0529
Bayesian Sets	0.1684	0.3348	0.1842	0.1512	0.1288	0.1110	0.0870	0.0382
Cond	**0.2035**	**0.4322**	**0.2414**	**0.2019**	**0.1746**	**0.153**	**0.1224**	**0.0564**

We can clearly see that all the evaluation values increase first when λ increases. Then, we get best performance when $\lambda = 0.5$, which indicates that we should consider $Prob(w_1|w_2)$ and $Prob(w_2|w_1)$ equally. After that, all the evaluation values decreases when λ keeps increasing. We note that performance decreases dramatically when λ is close to 0 or 1. This proves that either $Prob(w_1|w_2)$ or $Prob(w_2|w_1)$ being too small is a severe detriment. If there is no specific declaration, λ is set to be 0.5 hereafter.

We also carry out comparisons with other co-occurrence based measures and Bayesian Sets, which are shown in Table II. It is interesting that conditional probability gains similar performance as compared to Jaccard coefficient and Dice coefficient. When $\lambda = 0.5$, we can find that $Cond(w_1, w_2) = Dice(w_1, w_2)$:

$$\begin{aligned} Cond(w_1, w_2) &= \left(\frac{0.5}{Prob(w_1|w_2)} + \frac{0.5}{Prob(w_2|w_1)}\right)^{-1} \\ &= \left(\frac{0.5Freq(w_2)}{Freq(w_1, w_2)} + \frac{0.5Freq(w_1)}{Freq(w_1, w_2)}\right)^{-1} = Dice(w_1, w_2) \end{aligned} \tag{12}$$

As shown in Table II, Jaccard coefficient, Dice coefficient and conditional probability gain the best performance, followed by Google distance and Bayesian Sets. In general, the basic assumption of these co-occurrence based measures is that high association

or co-occurrence ratios indicate a semantic relatedness between words [Church and Hanks 1990]. Moreover, Iosif and Potamianos [2010] also verify the effectiveness of co-occurrence based measures for computing the semantic relatedness between words. They also study the properties and performances of various similarity measures. Unfortunately, based on their results, there are no complementarities among these measures. Similarly, in our experiment, we find that Jaccard coefficient, Dice coefficient and conditional probability return nearly the same related words for the same seed word, which indicate that we cannot combine them to improve the performance.

We observe that point-wise mutual information performs worst in our experiment. We further analyze the shortcoming of point-wise mutual information measure. For an input seed word w_1, we find that $PMI(w_1, w_2)$ tends to select words w_2 whose $Prob(w_1|w_2) = 1$. In other words, users who use w_2 will certainly use w_1 (w_2 co-occurs with w_1 with probability 1). Based on the observation that $Freq(w_1, w_2) \leq Freq(w_2)$, we get:

$$PMI(w_1, w_2) = \log \frac{\frac{Freq(w_1,w_2)}{U}}{\frac{Freq(w_1)}{U}\frac{Freq(w_2)}{U}} \leq \log \frac{\frac{Freq(w_2)}{U}}{\frac{Freq(w_1)}{U}\frac{Freq(w_2)}{U}} = \log \frac{U}{Freq(w_1)} \quad (13)$$

As shown in Formula (13), when $Prob(w_1|w_2) = 1$, or equivalently, $Freq(w_1, w_2) = Freq(w_2)$, $PMI(w_1, w_2)$ reaches its maximum. However, as shown in Figure 3, related word retrieval performance decreases rapidly if we only consider $Prob(w_1|w_2)$. For comparison, we also list results when $\lambda = 1.0$ in Table II. PMI gets exactly the same results with conditional probability when $\lambda = 1.0$, which confirms our inference.

From Table II, we observe that the performance is relatively poor using **randomly** selected seed words. Intuitively, we can get better performance if the seed words are chosen carefully. We select 100 discriminative seed words in computer science and carry out experiments on conditional probability, Bayesian Sets and Google Sets. For each seed word, we keep the first 10 returned candidate words. Domain experts are asked to label the results. We record the values of Bpref, MRR, P@5 and P@10 in Table III.

As shown in Table III, conditional probability gets comparable results with Bayesian Sets. Google Sets gives relatively worse results with seed words in Chinese. It does not return any results for seed words "框图" (block diagram), "分词" (word segmentation) and "加权" (weighting). However, Google Sets works well with seed words in English. We find that Google Sets generally harvests related words from structured or semi-structured web-pages. Chinese web-pages always contain much more noise as compared to English web-pages. Some Chinese web-pages are not written in standard HTML language. Chinese word segmentation makes it more difficult for finding related words in Chinese. As a result, Google Sets performs worse for seed words in Chinese. On the other hand, Bayesian Sets gains better performance than conditional probability with discriminative seed words, which indicates that Bayesian Sets benefits most if we select seed words carefully. For randomly selected seed words, conditional probability is better than Bayesian Sets. Finally, we can clearly see that the performance makes great progress under various evaluation metrics as compared to Table II. This confirms that we can gain better performance if we have prior knowledge of the seed words and choose them carefully.

Table III. Related word retrieval with discriminative seed words.

	Bpref	MRR	P@5	P@10
Google Sets	0.4308	0.6540	0.4383	0.3532
Bayesian Sets	**0.4701**	**0.7040**	**0.5060**	0.4390
Cond	0.4645	0.7018	0.4880	**0.4450**

Table IV. Candidate words Re-ranking for query word “爱立信”.

Query: 爱立信(Ericsson)	
Top 10 candidate words without Re-ranking	Top 10 candidate words 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)
运营商(operators)	西门子(Siemens)
网优(network optimization)	滑盖(slider)
西门子(Siemens)	网优(network optimization)
扩容(dilatation)	话务(telephone traffic)
话务(telephone traffic)	中兴(ZTE corporation)

5.3.2. Candidate Words Re-ranking. In this subsection, we investigate the effectiveness of our re-ranking framework. We first give an example in Table IV. The input seed word is “爱立信” (Ericsson), and the first column in Table IV shows the returned top 10 candidate words without re-ranking. After using search engines and context features, we reorder candidate words (as shown on the second column).

As shown in Table IV, we can return the most relevant candidate words such as “索尼爱立信” (Sony Ericsson) and “索爱” (the abbreviation of “索尼爱立信” in Chinese) in the first two places. Some famous brands like “阿尔卡特” (Alcatel) and “西门子” (Siemens) are also returned earlier after re-ranking. Words like “扩容” (dilatation) and “核心网” (Core Network) that are not very related are removed from the top 10 list. From these observations, we can see that the re-ranking framework and context features can improve the performance.

We also conduct experiments on the parameter N discussed in section 3.3. For a single seed word, top N candidate words are returned based on statistical features. Then, we utilize search engines to add context features for both the seed word and the candidate words. Finally, top N candidate words are reordered according to their context features. For each seed word, we keep the first 10 reordered candidate words. Detailed results are illustrated in Figure 4.

From Figure 4, We can see that too many candidate words tend to harm the performance because we may inevitably introduce noisy words. For example, Bpref drops to 0.25 when $N = 100$. In general, $N = 10$ and $N = 20$ give relatively best results. Our method usually returns useful candidate words in the first 20 places.

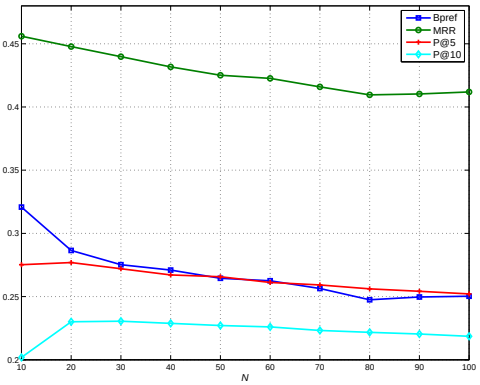

Fig. 4. Top N candidate words for re-ranking in related word retrieval.

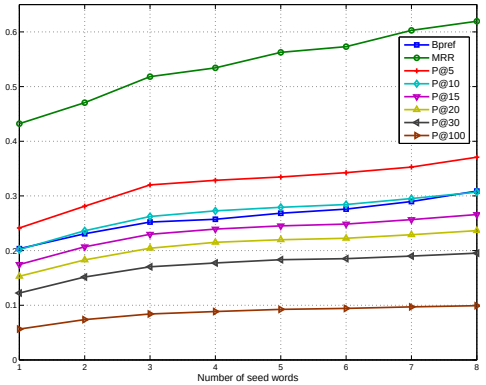

Fig. 5. Related word retrieval with multiple seed words.

Table V. Iterative related word retrieval from a single seed word.

Iteration	Query: 诸葛亮(Zhuge Liang)		
1	刘备(Liu Bei)	司马懿(Sima Yi)	蜀国(Shu)
2	黄忠(Huang Zhong)	魏国(Wei)	吴国(Wu)
3	张辽(Zhang Liao)	庖丁(Pao Ding)	典韦(Dian Wei)
4	张合(Zhang He)	曹仁(Cao Ren)	马岱(Ma Dai)
5	于禁(Yu Jin)	曹洪(Cao Hong)	夏侯(Xiahou)
Iteration	Query: 五道口(Wudaokou)		
1	城铁(City Rail)	知春路(Zhichunlu)	回龙观(Huilongguan)
2	上地(Shangdi)	西直门(Xizhimen)	霍营(Huoying)
3	西二旗(Xierqi)	西三旗(Xisanqi)	东直门(Dongzhimen)
4	天通苑(Tiantongyuan)	两居(Two bedrooms)	望京(Wangjing)
5	立水桥(Lishuiqiao)	四环(4th ring road)	一居(One bedroom)

5.3.3. Multiple Seed Words. In this subsection, we report experimental results for multiple seed words. Candidate words are ranked according to Formula (7). Number of seed words M changes from 2 to 8. We construct queries which contain M seed words, and keep top 100 returned candidate words for each query.

As shown in Figure 5, performance under various evaluation metrics improves steadily when the number of seed words increases. We can draw the conclusion that more seed words result in better results. We take the queries in Table I as an example. Word "苹果" (apple) is constrained to an IT company if it co-occurs with "微软" (Microsoft) and "谷歌" (Google) in the input, while it is constrained to a kind of fruit if it co-occurs with "胡萝卜" (carrot) and "草莓" (strawberry).

Furthermore, we can perform our related word retrieval task *iteratively* from a single seed word. In each iteration, we add the most related 3 candidate words to enlarge the set of seed words. Then, the enriched seed words are used for the next iteration. We show the iterative experiments for two seed words "诸葛亮(Zhuge Liang)" and "五道口" (Wudaokou) in Table V. The first seed word is a famous personage of the Three Kingdoms, and the second seed word is a place name in Beijing. As we can see, in each iteration, we can find proper related words from seed words. From seed word "诸葛亮(Zhuge Liang), we can find a list of personages of the Three Kingdoms. While from seed word "五道口" (Wudaokou), we can find a list of place names in Beijing. This experiment demonstrates the effectiveness of our method.

We have demonstrated that we can get better results if we have prior knowledge of the seed words and select discriminative seed words in previous subsections. What can we learn if we have **multiple discriminative** seed words? Fortunately, domain-specific lexicons contain some discriminative terminologies. Then from the domain-specific lexicons, we can detect related words in the specific domain accurately. Based on the above analysis, we extend our unsupervised related word retrieval method to solve the weakly-supervised new word detection task.

5.4. New Word Detection

As we can see, for the unsupervised related word retrieval task, we can gain better performance if we have prior knowledge about the seed words. Discriminative words, e.g. "机器学习" (machine learning), always generate better results than ambiguous words, e.g. "苹果" (apple). On the other hand, multiple seed words result in better performance than a single seed word. Ideally, from **multiple discriminative** seed words, we can detect related words with good performance. Domain-specific lexicons always contain some discriminative words. Then from domain-specific lexicons with label information, we aim to detect new words that are related to the specific domain but are not included. As a result, we extend our unsupervised related word retrieval method to solve the weakly-supervised new word detection task.

We use three domain-specific lexicons to conduct our representative words selection experiment. We remove noisy words and select discriminative words if we consider both coverage and discriminability together. Then, we select top 1000 words from each domain-specific lexicon as the most representative words. We can find potential experts based on the representative words and their user records. Finally, words used frequently by these potential experts are identified as new words.

5.4.1. Selecting Representative Words. As discussed before, we first try to filter noisy words in the original domain-specific lexicons. We focus on coverage and discriminability. Coverage reflects the popularity of words while discriminability emphasizes distinguishing positive users from negative users. We show words with good coverage or discriminability in three different lexicons respectively. In additional, we demonstrate we can get better results if we consider both coverage and discriminability.

Table VI shows words with good coverage in three lexicons. Take the computer science lexicon as an example, the most popular word in this lexicon is "问题" (problem), which is used by 78.33 percent of users. Some words such as "联系" (contact) and "学习" (study) are not related with computer science. The reason is that domain-specific lexicons are maintained by volunteers and everyone can modify them. In general, these widely-used words have poor discriminability. If a user uses these words very frequently, we are not confident enough to label him/her as a potential expert, because every user tends to use these words frequently.

Table VII shows the most discriminative words in three domain-specific lexicons. These words are quite related to their corresponding domains. Different from the words shown in Table VI, once these discriminative words appear in one user's record, he/she is very likely to be an expert because other users rarely use these words. However, we are encountering some problems if we select these words as representative words. The disadvantage is that these words have poor coverage. For example, word "并行数据库" (parallel database) only appears in about 10 user records. In other words, we are not able to discover enough potential experts if we use these words as representative words. This leads to a negative impact on the new word detection task.

Based on the above observations, we should consider both coverage and discriminability to select representative words. We treat coverage and discriminability equally to select representative words. We rank words according to Formula (9) and select top 1000 words as the representative words. Table VIII shows the corresponding results. In general, words listed in Table VIII have good discriminability as well as wide enough coverage. On one hand, they are used more frequently than words in Table VII. On the other hand, they maintain better discriminability than words in Table VI. As a result, we are confident to find enough potential experts using these words. We demonstrate how to find potential experts and detect new words in the next subsections.

Table VI. Representative words ranked by coverage.

Computer Science		Idiom and Saying		World of Warcraft	
Word	Coverage	Word	Coverage	Word	Coverage
问题(problem)	78.33%	不好意思(embarrassed)	48.48%	回家(go home)	67.50%
下载(download)	73.95%	乱七八糟(in a mess)	18.74%	其他(others)	62.45%
照片(picture)	67.26%	莫名其妙(be in a fog)	14.74%	速度(speed)	60.66%
空间(space)	66.74%	不知不觉(unwittingly)	9.43%	任务(task)	56.18%
联系(contact)	66.53%	一模一样(exactly alike)	9.02%	垃圾(garbage)	54.78%
学习(study)	63.51%	彼此彼此(same here)	8.79%	正常(normal)	54.31%
软件(software)	61.43%	一塌糊涂(in a mess)	8.47%	打开(open)	52.86%
情况(situation)	60.03%	胡思乱想(woolgather)	7.47%	专业(major)	52.73%
系统(system)	59.96%	不可思议(inconceivable)	7.15%	技术(skill)	51.83%
信息(message)	59.70%	原来如此(so that's it)	6.95%	全部(total)	49.12%

Table VII. Representative words ranked by discriminability.

Computer Science	Idiom and Saying	World of Warcraft
并行数据库 (parallel database)	进可替不 (recommend the wise men and supersede the incapable)	黑色作战迅猛龙 (black war raptor)
计时页(clocked page)	铺眉苫眼(put on an act)	考米克小屋(Kormek's hut)
线程控制块 (thread control block)	跌宕不羁 (unrestrained and reckless)	仪祭海袍 (flowing ritual robes)
系统目录表 (system directory list)	询根问底 (investigate thoroughly)	格兰戈瓦村 (Grangol'var village)
笛卡儿积(Cartesian product)	蓬头跣足(unkempt)	寒水魂精(winterfall E'ko)
环形缓冲 (circular buffer)	犯言直谏 (straight suggestion)	白鳄长靴 (albino Crocscale boots)
汉字特征(Hanzi features)	来者居上(catch up)	癞皮狼(mangy wolf)
服务接受者(service acceptor)	搬唇递舌(tell tales)	玛戈林(Magrin)
曼哈顿距离 (Manhatton distance)	报冰公事(hard job)	摩戈尔(Moggle)
操作系统病毒 (operating system virus)	卜昼卜夜 (day and night without cease)	抗寒图腾 (frost resistance totem)

Table VIII. Representative words ranked by coverage and discriminability.

Computer Science	Idiom and Saying	World of Warcraft
重定向 (redirect)	一个萝卜一个坑 (everyone has his own task, and there is nobody to spare)	瓦丝琪 (Vashj)
转义(escape)	两相情愿(consensual)	埃兰(Aran)
易用性(accessibility)	消遥自在(at large)	海度斯(Hydross)
可扩展性(extendibility)	指手划脚(make gestures)	禁魔监狱(The Arcatraz)
应用层(application layer)	翻复无常(vacillating in attitude)	摩摩尔(Murmur)
时间戳(timestamp)	急不可待(extremely anxious)	猫鼬(mongoose)
单线程(single thread)	张口结舌(see a wolf)	玛瑟里顿(Magtheridon)
多线程(multi thread)	流言蜚语(rumor)	纳迦(Naga)
可移植性(portability)	名符其实(be true to name)	绞喉(garrote)
通配符(wildcard character)	喜闻乐见(love to see and hear)	源生虚空(primal nether)

5.4.2. Finding Potential Experts. We have two ways to find potential experts in specific areas. The first way is to utilize user configuration files. We can label users who explicitly choose the domain-specific lexicons as experts. However, users may randomly choose some domain-specific lexicons and indicate that they are experts. For example, about 40.3 percent of users choose "idiom and saying" lexicon in their configuration files but use less than 10 words in that lexicon. It is not reliable to label these users as potential experts. Furthermore, we rank users according to the number of words they have used in the corresponding lexicons. We plot the data on a log-log graph in Figure 6, with the axes being $log(rank)$ and $log(number)$. This distribution roughly follows Zipf's law [2005], which indicates that many users use few domain-specific words although they choose domain-specific lexicons in their configuration files. As a result, it is not reliable to find potential experts using user configuration files.

Alternatively, we can find potential experts using representative words generated in the previous step. The more frequently these representative words appear in a user's record, the more likely he/she is a potential expert. We select 1000 representative words in each domain-specific lexicon. Similarly, we rank users according to the number of representative words they have used. We select top 8000 users as potential experts. As shown in Figure 7, these representative words are frequently used by a small percentage of users. These users are potential experts that we are trying to find. Then, we are trying to detect new words by analyzing user behaviors in the community of these potential experts.

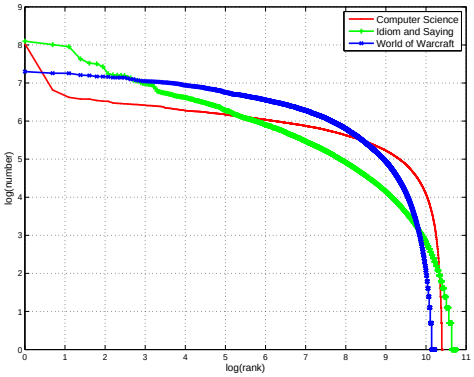

Fig. 6. Rank users according to the number of words they have used in domain-specific lexicons.

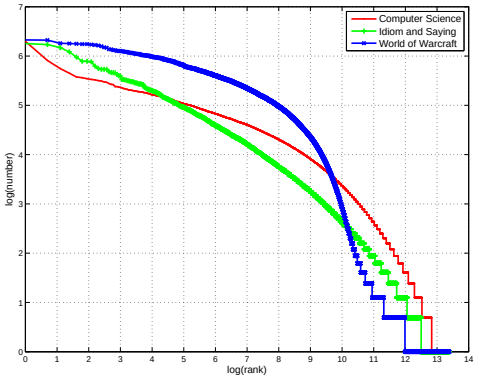

Fig. 7. Rank users according to the number of representative words they have used.

Table IX. Detected new words in computer science area.

Rank	Word	Rank	Word	Rank	Word
1	返回值(return value)	11	轮询(polling)	21	代码量(code size)
2	复杂度(complexity)	12	命令行(command line)	22	置为(set as)
3	序列化(serialization)	13	类库(class library)	23	类图(class diagram)
4	基类(serialization)	14	变量名(variable name)	24	轻量级(lightweight)
5	实例化(instantiation)	15	跨平台(cross platform)	25	头文件(header file)
6	分布式(distributed)	16	父类(super class)	26	架构师(architect)
7	注释掉(comment out)	17	设计模式(design pattern)	27	用例(use case)
8	正则(regular)	18	数据量(data volume)	28	健壮性(robustness)
9	操作符(operator)	19	业务逻辑(business logic)	29	扩展性(scalability)
10	类名(class name)	20	递归(recursion)	30	预定义(predefine)

Table X. New words detection performance in computer science area.

Bpref	Accuracy	P@5	P@10	P@30	P@100	P@1000
0.5563	0.8594	1.0000	1.0000	1.0000	0.9200	0.8750

5.4.3. Detecting New Words. As discussed before, we have discovered potential experts in particular domains. These experts tend to have similar tastes on certain terminologies, while other users seldom use them. Based on this observation, we detect new words according to how unbalanced the distributions of them in potential experts and other users are. If a word is widely used by potential experts and rarely appear in other users' records and it is absent from the original lexicon, then we believe that this word is a new word that should be included. Table IX shows top 30 new words in computer science detected by our method. Most of these words have strong relationships with computer science, which indicates that our method is effective.

Specifically, we adopt Formula (8) to measure the discriminability which can distinguish potential experts from other users. Then Formula (9) is used to rank the candidate new words. Candidates with higher scores are more likely to be new words. We select top 2000 words according to their scores. Then we filter words that are included in the original lexicon and get 1,629 new words. We ask five volunteers with strong background in computer science to judge whether a word is related to computer science. The final results are made based on their votes. As shown in Table X, our method gains up to 0.86 in accuracy, which means that 1,400 out of 1,629 detected new words are related to computer science. It is promising that our method gets 1.0 and 0.92 in P@30 and P@100 respectively. Based on these observations, we are confident to claim that it is helpful to incorporate user behaviors in the new word detection task.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel method for unsupervised related word retrieval and weakly-supervised new word detection tasks. Different from previous methods, we take user behaviors, collaborative filtering and re-ranking framework into consideration. We make a reasonable assumption that if two words frequently co-occur in user records, then they are likely to have a strong semantic connection with each other. Based on this assumption, we generate a list of candidate words that are related to the seed words. Then we utilize search engines to extract context features and further enhance the performance of related word retrieval task.

Furthermore, we observe that we can get better performance with multiple and discriminative seed words. Then we extend our unsupervised related word retrieval method to solve weakly-supervised new word detection task. From the domain-specific lexicons which contain some discriminative terminologies, we first find potential experts who use these terminologies frequently. Then, we investigate user behaviors of these potential experts. We believe that words used much more frequently among potential experts than other users are new words for the given domain. Experimental results on the both tasks show the effectiveness of our method. Our method for the both tasks is language independent and can be easily extend to other languages.

However, we observe some noisy results in both tasks. The main reason is that our data set is generated from Chinese input method users. Users can type in whatever they want when using an input method. How to filter the noisy words will be left as our future work. We also plan to utilize learning to rank literature [Liu 2009] to improve the performance of related word retrieval task. We can extract more features and build a labeled training set. Then, various machine learning techniques can be used for both tasks.

Another problem of our new word detection is that the set of representative words is relatively small. We require a lot of manual efforts to evaluate the performance of both tasks, which is time-consuming and expensive. We can try some extrinsic evaluation methods by applying our detected new words in other applications such as Chinese new word detection and automatic speech recognition. Moreover, how to automatically detect new words and collect domain-specific knowledge for emerging new domains is also an interesting problem.

Finally, it is important to build an accurate and complete ground truth for the related word retrieval task. This is difficult because we may need a lot of manual efforts. Moreover, people may have different opinions on whether two words are related or not which makes this task more complicated.

REFERENCES

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17,* 6, 734–749.

BANNARD, C. AND CALLISON-BURCH, C. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 597–604.

BOLLEGALA, D., MATSUO, Y., AND ISHIZUKA, M. 2007. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International Conference on World Wide Web*. ACM, 757–766.

BREESE, J., HECKERMAN, D., AND KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

BUCKLEY, C. AND VOORHEES, E. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 25–32.

CHEN, K. AND BAI, M. 1998. Unknown word detection for Chinese by a corpus-based learning method. *Computational Linguistics 3,* 1, 27–44.

CHURCH, K. AND HANKS, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics 16,* 1, 22–29.

CILIBRASI, R. AND VITANYI, P. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering 19,* 3, 370–383.

DESHPANDE, M. AND KARYPIS, G. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems 22,* 1, 143–177.

ETZIONI, O., CAFARELLA, M., DOWNEY, D., POPESCU, A., SHAKED, T., SODERLAND, S., WELD, D., AND YATES, A. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence 165,* 1, 91–134.

FORMAN, G. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research 3*, 1289–1305.

GHAHRAMANI, Z. AND HELLER, K. 2006. Bayesian sets. *Advances in Neural Information Processing Systems 18*, 435–442.

GOOGLE. 2010. Google Sets. http://labs.google.com/sets.

GRACIA, J., TRILLO, R., ESPINOZA, M., AND MENA, E. 2006. Querying the web: A multiontology disambiguation method. In *Proceedings of the 6th International Conference on Web Engineering*. ACM, 241–248.

HEILMAN, M. AND SMITH, N. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1011–1019.

HOFMANN, T. 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 259–266.

IOSIF, E. AND POTAMIANOS, A. 2010. Unsupervised Semantic Similarity Computation Between Terms Using Web Documents. *IEEE Transactions on Knowledge and Data Engineering 22,* 11, 1637–1647.

JI, H., GRISHMAN, R., DANG, H., GRIFFITT, K., AND ELLIS, J. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Text Analysis Conference*.

JONES, R., REY, B., MADANI, O., AND GREINER, W. 2006. Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web*. ACM, 387–396.

LI, H., HUANG, C., GAO, J., AND FAN, X. 2004. The use of SVM for Chinese new word identification. In *Proceedings of the First International Joint Conference on Natural Language Processing*. Springer, 723–732.

LI, J. AND SUN, M. 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*. Association for Computational Linguistics, 774–782.

LIN, D., CHURCH, K., JI, H., SEKINE, S., YAROWSKY, D., BERGSMA, S., PATIL, K., PITLER, E., LATHBURY, R., RAO, V., K., D., AND NARSALE, S. 2010. Unsupervised Acquisition of Lexical Knowledge from N-grams. Final Report of the 2009 JHU CLSP workshop. In *Proceedings of the 2009 JHU CLSP workshop*.

LIU, T. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval 3,* 3, 225–331.

MCNAMEE, P. AND DANG, H. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of the Text Analysis Conference*.

METZLER, D., DUMAIS, S., AND MEEK, C. 2007. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on Information Retrieval*. Springer-Verlag, 16–27.

NAKAMURA, A. AND ABE, N. 1998. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 395–403.

NEWMAN, M. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary physics 46,* 5, 323–351.

PANTEL, P., CRESTAN, E., BORKOVSKY, A., POPESCU, A., AND VYAS, V. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 938–947.

PAŞCA, M. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management*. ACM, 683–690.

PAŞCA, M., LIN, D., BIGHAM, J., LIFCHITS, A., AND JAIN, A. 2006. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 809–816.

PAŞCA, M. AND VAN DURME, B. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 19–27.

PENG, F., FENG, F., AND MCCALLUM, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, 562–568.

RIEZLER, S., LIU, Y., AND VASSERMAN, A. 2008. Translating queries into snippets for improved query expansion. In *Proceedings of the 22nd International Conference on Computational Linguistics*. Association for Computational Linguistics, 737–744.

SAHAMI, M. AND HEILMAN, T. 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*. ACM, 377–386.

SALTON, G., WONG, A., AND YANG, C. 1975. A vector space model for automatic indexing. *Communications of the ACM 18,* 11, 613–620.

SARMENTO, L., JIJKUON, V., DE RIJKE, M., AND OLIVEIRA, E. 2007. More like these: growing entity classes from seeds. In *Proceedings of the Sixteenth ACM conference on Conference on Information and Knowledge Management*. ACM, 959–962.

SPROAT, R. AND EMERSON, T. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*. Association for Computational Linguistics, 133–143.

VAN DURME, B. AND PAŞCA, M. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Twenty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 1243–1248.

WANG, M., HUANG, C., AND CHEN, K. 1995. The Identification and classification of Unknown Words in Chinese: A N-gram-Based Approach. In *Proceedings of the 1994 Kyoto Conference: A Festschrift for Professor Akira Ikeya*. The Logico-Linguistic Society of Japan, 113–123.

WANG, R. AND COHEN, W. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*. IEEE, 342–350.

WANG, R. AND COHEN, W. 2008. Iterative set expansion of named entities using the web. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. IEEE, 1091–1096.

WANG, R. AND COHEN, W. 2009. Automatic set instance extraction using the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, 441–449.

YIH, W. AND MEEK, C. 2007. Improving similarity measures for short segments of text. In *Proceedings of the 22nd National Conference on Artificial Intelligence*. AAAI Press, 1489–1494.

ZHANG, K., LIU, Q., ZHANG, H., AND CHENG, X. 2002. Automatic recognition of Chinese unknown words based on roles tagging. In *Proceedings of the First SIGHAN Workshop on Chinese Language Processing*. Association for Computational Linguistics, 1–7.

ZHENG, Y., LIU, Z., SUN, M., RU, L., AND ZHANG, Y. 2009. Incorporating user behaviors in new word detection. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2101–2106.

ZHENG, Y., LIU, Z., AND XIE, L. 2010. Growing related words from seed via user behaviors: a re-ranking based approach. In *Proceedings of the ACL 2010 Student Research Workshop*. Association for Computational Linguistics, 49–54.

Received November 2010; revised February 2011; accepted April 2011