Journal of Web Engineering, Vol. 15, No. 3&4 (2016) 181–202
© Rinton Press

ENHANCING KEYWORD SUGGESTION OF WEB SERACH BY LEVERAGING MICROBLOG DATA[a]

LIN LI
[1]*Hubei Collaborative & Innovative Center for Basic Educational Technology*
[2]*School of Computer Science & Technology, Wuhan University of Technology*
Wuhan, 430070, China
cathylilin@whut.edu.cn

LU QI
[1]*School of Computer Science & Technology, Wuhan University of Technology*
[2]*Hubei Key Laboratory of Transportation Internet of Things, Wuhan University of Technology*
Wuhan, 430070, China
chinaqilu18@whut.edu.cn

FANG DENG
College of Computer, Hubei University of Education
Wuhan, 430205, China
281588723@qq.com

SHENGWU XIONG JINGLING YUAN
School of Computer Science & Technology, Wuhan University of Technology
Wuhan, 430070, China
xiongsw@whut.edu.cn *yjl@whut.edu.cn*

Received December 12, 2014
Revised August 3, 2015

[a]This work was supported by projects of 15BGL048, 2015AA015403, 2015BAA072, and 61303029.

Query suggestion of Web search is an effective approach to help users quickly express their information need and accurately get the information they need. Most of popular web-search engines provide possible query suggestions based on their query log data, which is a kind of implicit relevance based approach. However, it is difficult to give suggestions to search queries that have no or few historical evidences in query logs. To solve this problem, traditional pseudo relevance based approaches directly extract additional keywords from the top-listed search results of a given search query as suggestions. However, for hot topic or event related search queries, users more like to browse the latest and newly appeared contents. In this paper, we follow the direction of pseudo relevance based suggestion approaches by mining microblog data that is inherent in fast information propagation and dissemination. Our graph based rank aggregation approach combines a frequency based ranking with considering words themself and a LDA (Latent Dirichlet Allocation) based ranking by mining hidden topics behinds words. A dataset is crawled from the posts of fourteen micro-topics of Sina microblog platform. The experimental results clearly demonstrate our proposed approach is more effective than traditional pseudo relevance based methods. Moreover, the suggested keywords extracted from the posts published by *authenticated users* are more effective than two traditional pseudo relevance based approaches, i.e., the posts submitted by *all users* and the top returned posts returned by Sina search engine. In addition, applying LDA on microbog posts alone is far from satisfactory, but the combination of the frequency based ranking and the LDA based ranking show much better performance.

Keywords: Search query, microblog posts, suggestion, pesudo relevance
Communicated by: G-J Houben & P. Fraternali

1 Introduction

Web search engines greatly change the way that people acquire information during the last ten years. As an end-user starts typing a query in a search box, most search engines assist users by providing a list of keywords, which is an effective suggestion service [32]. The user can quickly choose one of the suggested completions (in some cases, alternatives) and does not have to type the whole query herself. Feuer et al. [15] analyze approximately 1.5 million queries from the search logs of a commercial search engine and find that suggested queries are nearly 30% of the total queries and the engine with phrase suggestions performs better in terms of precision and recall than the same search engine without suggestions. Furthermore, Kelly et al. [22] observe that the use of offered query suggestions is more for difficult topics, i.e., topics on which users have little knowledge to formulate good queries. Yang et al. [33] present an optimal rare query suggestion framework by leveraging implicit feedbacks from users in the query logs. Sumit et al. [4] put forward a probabilistic mechanism for generating query suggestions from a corpus without using query logs and utilize the document corpus to extract a set of candidate words. In the literature, researchers and industries show great interests in query suggestion for enhancing web search quality and improving user search experiences.

Traditional implicit relevance based approaches think that the information need of a current user is searched before by some other users, so they rely on large amounts of past usage data to offer possible query suggestions. Although there are many works using query logs to suggest queries [1, 3, 6, 9, 11, 21, 27, 29, 33], When using general-purpose web search engines, end-users sometimes pose queries that are not or much less frequent in query logs. Especially with the rise of social network, there has been emerging a group of new network

vocabulary. When these newly appeared words formulate search queries, they always have few search history in query logs. It then becomes difficult to give useful suggestions in such cases.

On the other hand, pseudo relevance based approaches assume that the information need of a search query is closely related to its top search results where they usually mine representative keywords as suggestions [19, 26, 35]. No mater a query exists in query logs or not, it can produce suggestions. Indexed web pages in a common search engine are updated periodically. Although there exist new Web pages talking about the latest and newly appeared contents, especially for hot events and news, their rankings are relatively low due to ranking factors used by main search engines. If some keywords can be found to help users easily find new information, it will be good to improving search quality. Regarding this issue, one question is what is a potential suggestion resource with the latest and fresh information.

At present, as a widely used social medium platform, microblog's diverse features meet the people's new information requirements on interpersonal communicating and sharing. Compared with traditional media, microblogging as a new service has the following characteristics and advantages.

(1) Its information propagation is convenient and rapid.

(2) Its information dissemination is fast and posts are updated in time.

(3) It has great potential business value.

Among the above three features, the second one motivates our work. Nowadays the speed of information propagation through the microblog service is fast and a large amount of people are involved in it. The kind of intuitive, convenient, and efficient communication makes microblogging popular and micoblog posts updated quickly. Based on these observations, microblogging is selected as our suggestion resource. More specifically, it is the collection of the top posts returned by a microblog search engine.

Our other question is how to rank keywords as a suggestion list. Our pseudo relevance based approach utilizes a graph based rank aggregation which combines the frequency based ranking and the LDA [5] based ranking. Top ranked nouns and verb-nouns are shown to search users. Our main contributions are as follows.

(1) We find that adding microblog data can improve the quality of query suggestion and web search.

(2) Instead of using top returned posts like traditional pseudo relevance based approaches, using those posted by authentication users show comparable suggestion precision while saving run time.

(3) Typical frequency based ranking is more effective than the popular LDA based ranking and their combination shows the best suggestion results.

In addition, case study is presented and we find that the more related search results are obtained by using our suggestions.

The rest of the paper is organized as follows. In Section 2, we provide an overview of the prior work on query suggestion along with explaining how our approach differs from

them. In Section 3, we give a suggestion flowchart to describe our whole query suggestion process. In Section 4 to Section 6, we present our pseudo relevance based approaches in details. Experiments and results are presented in Section 7. Section 8 concludes the paper and outlines future research directions.

2 Related Work

There are a variety of researches on web query suggestion. Some of them focus on identifying past queries similar to a current user query. Others are interested in finding context (implicit or pseudo relevance feedback) to enhance web search such search results, search logs and so forth. Here we give a brief review of related work.

2.1 Query Clustering

Clustering queries submitted to search engines appears to be less explored than clustering Web pages or documents. The idea of exploiting the collaborative knowledge of users, embodied as a set of past search queries, was proposed early [1, 19, 31, 34]. Glance [19] introduced a software agent that collects queries from previous users, and determined the query similarity based on the Web pages returned by queries, and not the actual terms in the queries themselves. The goal of [31] was to accelerate the formation of optimal queries from past queries. Wen et al. [34] proposed to cluster similar queries to recommend URLs to frequently asked queries of a search engine. They combined similarities based on query contents and user clicks, and regarded user clicks as an implicit relevance feedback instead of using the top ranked Web pages. Baeza-Yates et al. [1] cluster queries presented in search logs. Given an initial query, similar queries from its cluster are identified based on vector similarity metrics and are then suggested to a user.

2.2 Using of Search Results

Fitzpatrick et al. [16] improved the effectiveness of a user-supplied query by identifying key terms from potentially relevant documents from past queries. In [25] Li et al. devised a novel enhanced web search approach by aggregating results of related Web queries, which aims at facilitating locating the information need of a user. Their search system took a couple of related queries as search inputs and output a final search result list which was the aggregation of the result lists of these input queries. The strength of the combined query collections can substantially enhance the utilization of query suggestion to improve Web search quality. In [26], the authors made use of URLs of search results and their search query to build bipartite graph for query recommendation.

2.3 Implicit Relevance Feedback

Barouni-Ebrahimi and Ghorbani utilized words frequently occurring in queries submitted by past users as suggestions [3]. By utilizing clickthrough data and session information, Cao et al. proposed a context aware query suggestion approach [9]. In order to deal with the data sparseness problem, they used concept based query suggestions where a concept was defined as a set of similar queries mined from the query-URL bipartite graph. On the other hand, in [2], the authors found related queries based on the content of clicked Web pages using click frequency as a weighting scheme. Their experiments showed that the content information was more accurate to measure query similarity than the URL information. As we discussed in

Section 1, rare queries are not frequently appeared in search logs, so their implicit relevance is not adequate. Our paper talks about pseudo relevance that is general and easy to get, which is orthogonal to researches on implicit relevance.

2.4 Pseudo Relevance Feedback

Gao et al. described a query suggestion mechanism for cross lingual information retrieval where for queries issued in one language, queries in other languages can also be suggested [17]. In [16] the authors improved the effectiveness of a user-supplied query by identifying key terms from potentially relevant documents from past queries. In [18], the authors introduced a software agent that collects queries from previous users, and determine the query similarity based on the Web pages returned by queries, and not the actual terms in the queries themselves. In [26], a tree distance based rank mechanism was devised for ordering the related queries using the merging distances of a hierarchical agglomerative clustering (HAC). Different from them, we study how to improve the quality of search query suggestion by leveraging microblogging data.

2.5 Rare Query Suggestion

Lately, Broder et. al studied an online expansion of rare queries in [8]. Their framework started by training an offline model that was able to suggest a ranked list of related queries to an incoming rare query. The rare query was then expanded by a weighted linear combination of the original query and the related queries according to their similarity. Yang et al. [33] also worked on rare query suggestion by using implicit feedbacks, while Sumit et al. [4] made use of a corpus instead of query logs. Jiafeng Gu et al. [20] introduced social annotation data into query recommendation as an additional resource and inferred what people might think when reading web pages.

2.6 Social Media Analysis

The rising popularity of online social networking services has spurred research into microblogs and their characteristics. There are a number of researchers exploring and studying English microblogging, i.e., twitter. Newman et al. [30] made the first quantitative study on the entire Twitter sphere and information diffusion on it. They studied the topological characteristics of Twitter and have found a non-powerlaw follower distribution, a short effective diameter, and low reciprocity, which all marks a deviation from known characteristics of human social networks. In 2010, the work in [24] further discussed the topological characteristics of Twitter and its power as a new medium of information sharing. Chen et al. [10] compared two kinds of approaches, traditional cosine-based approach and WordNet-based semantic approach, when computing similarities between microblog posts.

With the prevalence of Sina microblogging, some researchers began to study the new Chinese microblog media. Liu et al. [37] combined a translation-based method with a frequency-based method for keyword extraction. They extracted keywords for microblog users from the largest microblogging website in China, i.e., Sina Weibo. Different from them, we present how to extract and analyze microblog posts to produce effective suggestions which can better meet the information need of users, and experimentally discuss the selection of pseudo relevance resources in terms of precision and efficiency.

3 Keyword Suggestion Flowchart

As shown in Figure 1, the whole keyword suggestion process includes two modules. One is getting pseudo relevance via a microblog search engine. The other is selecting top N suggested keywords through the rank aggregation of the frequency based ranking and the LDA based ranking.

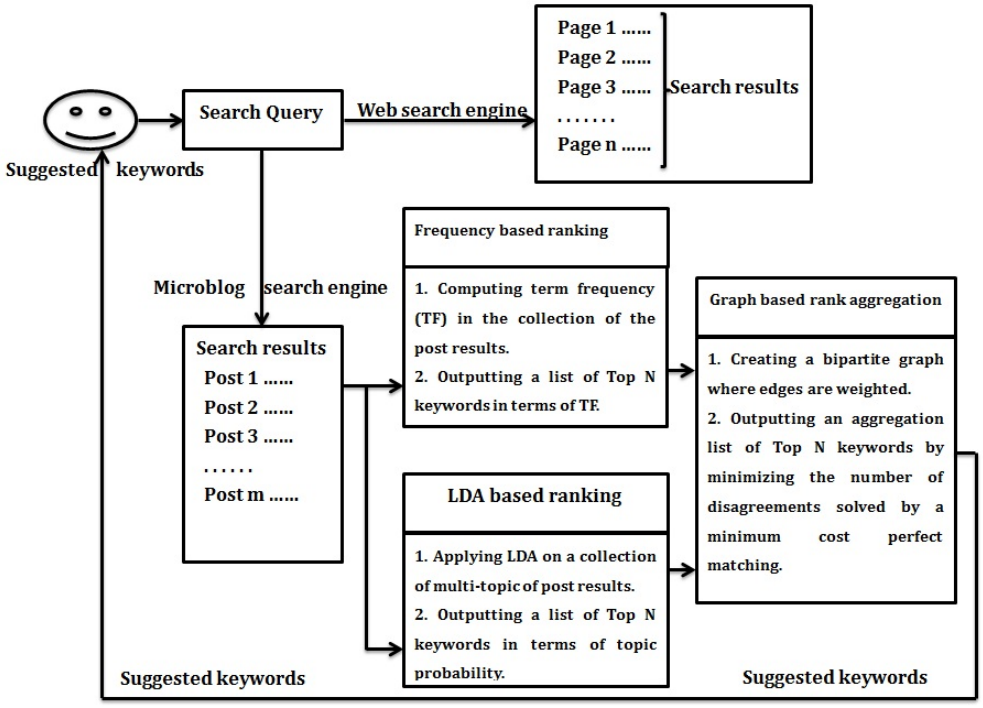

Fig. 1. The Flowchart Of Our Pseudo Relevance Based Approach

When a user submits a search query to a common Web search engine, our suggestion approach will put it to a microblog search engine. Microblog posts will be returned and they are used as our pseudo relevance. As we discussed in Section 1, one problem we will solve is how many or which part of returned posts should be involved in suggestion. The naive method is considering the top N returned posts. However, microblog posts are short and no longer than 140 characters. The quality of posts will affect the suggestion effectiveness. In our suggestion, we use the returned posts from authenticated users who are approved by a micoblogging service provider. In other words, we think authenticated users are more reliable to write high quality posts than others. Our experimental results in Section 7 show that the suggestion precision from the posts published by authenticated users is higher that that from the posts published by all users when using the TF based ranking. Moreover, it largely reduces run time that is vital for online suggestion service.

Since we treat microblog data as a kind of useful information for hot topic and event related suggestion, the next problem is how to select top N keywords as suggestions which can represent the meaning of a search query well. The intuitive method is the frequency

based ranking which orders keywords by their frequency statistics, such as term frequency in a document. In our suggestion context, we count the term frequency in a collection of returned posts, i.e., as discussed above, the collection posted by authenticated users. In addition, LDA is popular to discover a latent topic structure via estimating the probability distribution of the original co-occurrence activities. We can select top keywords ranked by their topic probabilities. Last, the two suggestion lists produced by the frequency based ranking and the LDA based ranking are aggregated together. We propose a graph based approach for rank aggregation and compare it with a modified single-winner election method, e.g., Board counts [23].

We summarize our suggestion idea here. When a user has an information need, she will transform the information need into a query and start typing the query in the query box of a search engine. The user has some information need but is not sure which words to use to formulate a query because traditional method that documents indexed by the search engine are not visible to the user. The terms selected by the user to formulate the queries often do not lead to a good retrieval performance due to the gap between query-term space and document-term space [12]. This problem is especially difficult for the search queries lacking context in query logs. To help the user formulate good search queries, keywords are extracted from the returned posts of authenticated users by a mircoblog search engine. Top N representative keywords are selected by the combination of the above two methods. The user can modify her query based on our suggested keywords and do a search again to meet her information need.

4 Frequency Based Ranking

In this section, we present the frequency based ranking for keyword suggestion using TF weighting. We compute term frequency not in an individual microblog post but in the collection of returned post results given a search query. A search query represents the information need of a user and the returned posts are regarded relevant to it. We want extract keywords from the whole returned collection that is potentially fresh according to the inherent characteristics of microblogging. Those keywords are as suggestion and help users find what they want. This ranking approach includes spams removing and frequency computing.

4.1 Removing Spammed Microblog Posts

Now there have been some spammed posts in the returned results of a search query. As spammed posts have a bad effect on the accuracy of understanding tweets, a long time ago the twitter has already started to address how to remove the spammed tweets. The twitter company suspends any user reported to be a spammer and use several well-known rules to clean tweets. For example, Twitter automatically deletes any tweets from accounts that are less than 24hrs old (or how long you specify).[b]

Based on our observations on Chinese microblogging, our approach is simple, but very effective for removing the spammed posts. We find that there exists some "Dirty data ", like this "sorry, this post has been deleted by the original author. For help, please contact customer service. http://t.cn/z0D6ZaQ ". This kind of posts has no any information for user name and its posted time. So before frequency computing, we make a data cleaning by

[b] https://addons.mozilla.org/en-US/firefox/addon/clean-tweets/

weeding out these kinds of data to ensure the quality of the data.

4.2 Frequency Computing

To complete text processing, two Chinese NLP tools is used [c]. Sometimes, some extremely common words that would appear to be of little value in helping select documents related to the information need of a user are excluded from the vocabulary entirely. These words are called stop words. Common stop words include articles, prepositions, conjunctions and the elimination of stop words decreased the number of terms. Therefore, we remove all stopwords. After it, we make Chinese participle preprocessing for the filtered posts by word segmentation and POS tagging. It can identify proper and newly appeared nouns and minimize the word granularity, such as the new word of weixin(a mobile phone chat software called WeChat in English) which is a new application launched by Tencent company in 2011. Otherwise, the word of weixin is divided into two words. Choosing suitable segmentation can improve the precision of participle processing results.

Last, we do word frequency statistic computing. It can not only make character frequency statistics but also make word frequency statistics. There is no doubt that we choose the word frequency here. Furthermore, we also observe that the processing time of the collection from authenticated users is 0.5 hours, much more less that that of the collection of all users, i.e., almost 4 hours.

5 LDA Based Ranking

LDA (Latent Dirichlet Allocation) is a probabilistic generative model for a text corpus. The basic idea is that documents are represented as random mixtures over latent topics and each topic is characterized by a distribution over words. Our basic idea is that the suggested keywords are ordered by their topic distributions. In other words, a keyword with the highest topic distribution score will be ranked in the first position of a suggestion list.

5.1 LDA Algorithm

Theoretically speaking, LDA is based on the assumption that there exists an unseen structure of "topics" or "themes" in the text corpus, which governs the co-occurrence observations. As such, the intuition behind LDA is to discover this latent topic structure via estimating the probability distribution of the original co-occurrence activities. The notations used in the algorithm are described in Table 1.

The generative procedure of LDA model is shown in Figure 2 and the pseudo codes of its implementation algorithm are shown in Table 2. The model formulation is also described as follows.

In LDA, a post retreated as a document $d_m = \{w_{mn}, n = 1, ..., N_m\}$ is generated by picking a distribution over the topics from a Dirichlet distribution ($Dir(\alpha)$). And given the topic distribution, we pick the topic assignment of each specific word. Then the topic assignment for each word $[m, n]$ is calculated by sampling a particular topic $z_{m,n}$ from the multinomial distribution of $Mult(\theta_m)$. And finally, a particular word of $w_{m,n}$ is generated for the placeholder $[m, n]$ by sampling its weight from the multinomial distribution of $Mult(\varphi_{z_{m,n}})$. Known from the above description, given Dirichlet parameters α and β, we can formulate a

[c] They are recommended by the website of http://www.china-language.gov.cn/index.htm

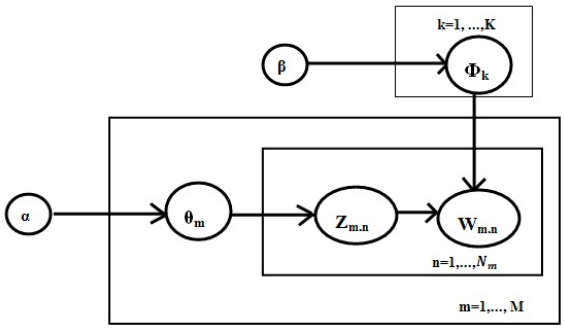

Fig. 2. The illustration of LDA

joint distribution of a post d_m, a topic mixture of d_m, i.e. θ_m , and a set of N_m topics, i.e. z_m as follows:

$$P_r(\theta_m, z_m, d_m, \Phi \,|\alpha, \beta) = P_r(\theta_m \,|\alpha)\, P_r(\Phi \,|\beta) \prod_{n=1}^{N_m} P_r\left(w_{m,n} \left|\varphi_{z_{m,n}}\right.\right) P_r(z_{m,n} \,|\theta_m)$$

And integrating over θ_m, $\phi_{z_{m,n}}$ and summing over z_m , we obtain the likelihood of the post d_m:

$$P_r(d_m \,|\alpha, \beta) = \int\int P_r(\theta_m \,|\alpha) P_r(\Phi \,|\beta) \prod_{n=1}^{N_m} P_r\left(w_{m,n} \left|\varphi_{z_{m,n}}\right.\right) P_r(z_{m,n} \,|\theta_m)\, d\theta_m d\Phi$$

Finally the likelihood of the post corpus $D = \{d_m, m = 1, ..., M\}$ is a product of the likelihood of all posts in the corpus, defined as

$$P_r(D \,|\alpha, \beta) = \prod_{m=1}^{M} P_r(d_m \,|\alpha, \beta)\,. \tag{1}$$

Under this generative model, posts can be modelled as probability distributions over multiple topics via an estimation procedure. In other words, the inputs of LDA model are terms of posts and its outputs are topic distributions.

5.2 Model Estimation

In order to use the topic model obtained by LDA, we need to solve the problem of computing the posterior distribution over the hidden topics given a post. In fact each post is equivalent

Table 1. Notations of LDA model

M:	the number of microblog posts
K:	the number of topics
V:	the size of vocabulary
α, β:	Dirichlet parameters
θ_m:	the topic assignment of the post m
Θ:	the topic estimations of the corpus, a $M \times K$ matrix
φ_k:	the word distribution of the topic k
Φ:	the word assignments of the topics, a $K \times V$ matrix

Table 2. Algorithm: Generation Process of LDA

for each of topics
sample the mixture of words $\phi_k \sim Dir(\beta)$
end
for each of posts $m = 1 : M$
sample the mixture of topics $\theta_m \sim Dir(\alpha)$
sample the lengths of posts $N_m \sim Poiss(\xi)$
for each of words $n = 1 : N_m$ in the post m
sample the topic index of $z_{m,n} \sim Mult(\theta_m)$
sample the weight of word $w_{m,n} \sim Mult(\phi_{z_{m,n}})$
end
end

to an especially short document, thus we employ model estimation on this post-term array. However, estimating the parameters of LDA by directly and exactly maximizing the likelihood of the whole data collection in Eq.(1) is intractable. The solution to this is to use alternative approximate estimation methods. Here we employ the variational EM algorithm [5] to find the variational parameters that maximize the total likelihood of the corpus with respect to the model parameters of α and β:

$$(\alpha_{est}, \beta_{est}) = \max l(\alpha, \beta) = \max \sum_{m=1}^{M} \log Pr(d_m \mid \alpha, \beta)\,. \tag{2}$$

The variational EM algorithm is briefly described as follows:

1. (E-step) For each post, find the optimizing values of variational parameters θ_m^* and φ_m^*.

2. (M-step) Maximize the resulting low bound on the likelihood with respect to model parameters α and β. This corresponds to finding maximum likelihood estimates with the approximate posterior which is computed in the E-step.

The E-step and M-step are executed iteratively until a maximum likelihood value reaches. Meanwhile, the calculated estimation parameters can be used to infer topic distribution of a new post by performing the variational inference. More details with respect to the variational EM algorithm are referred to [5].

5.3 Topic Inference

After doing the topic estimation using our training dataset (explained in Section 7.2.2), we eventually obtain the posterior parameters of the LDA model α and β as well as the topic assignments of posts and terms. Then the trained topic model could be used for topic inference for a target input post. In this case, the inference is done through the variational algorithm [5]. As a result, an inferred topic distribution of the target post is calculated, which reflects the likelihood of various topic assignments of the post. As the dimensionality of the topic space

is much smaller than that of the original term space, the LDA operation could be viewed as a dimensionality reduction method where the latent semantic topic of the document is optimally approximated. In other words, after the LDA model estimation, the expression of a post is transformed from a high-dimensional sparse term space to a low-dimensional latent topic space. For ranking, user's search query is LDA input. The learned latent topic with the highest topic probability is selected. The top 10 keywords are ranked by their topic probability in this selected topic.

6 Graph Based Rank Aggregation

We can get two different suggestion lists by using the frequency based ranking and the LDA based ranking. The former suggests keywords from the viewpoint of the term feature space of posts, while the later outputs keywords according to the latent topic feature space of posts. In our experiments, the two ranking approaches show quite different suggestion lists. We further study the problem of combining the two rank lists into a single rank list aiming at improving suggestion precision.

Voting provides us with a traditional class of algorithms to determine the aggregated rank list. The most common voting theory, named after its creator, is known as Borda's rule [7] which argues that the majority opinion is the truth, or at least the closest that we can come to determining it [36]. However, the problem with Borda's rule is that it does not optimize any criterion. We make use of Footrule distances [13] to weigh edges in a bipartite graph and then find a minimum cost matching. This method was proved in [14] to approximate the optimal ranking that approximately minimizes the number of disagreements with the given inputs.

6.1 Modified Borda's Rule

Borda's rule is a single winner election method. The winner of an election is determined by giving each candidate a certain number of points corresponding to the position in which each voter ranks her. Once all points have been counted, the candidate with the most points is the winner.

Our idea is that we treat a ranking approach as a voter. It means that a ranking approach orders the keywords in the same way as each voter selects a list of candidates. Let $A = a_1, a_2, \cdots, a_m$ be the set of positions in the rank list, and let the ranking approaches be named by elements of n (i.e., n voters in an election). We shall assume for the present that every element of n can be expressed by a linear order in the position set A. We denote a linear order by a sequence $A_i = a_{i_1}, a_{i_2}, \cdots, a_{i_m}$ where for $j < k$, a_{i_j} is preferred to a_{i_k}. For each voter, the ranked keywords should be given some points. The closer a keyword is to the top of the list, the more points it will be given. The voter awards the first-ranked candidate with one point (i.e., 1). The second-ranked candidate receives half of a point (i.e., 1/2), the third-ranked candidate receives on a third (i.e., 1/3), etc. This kind of point distribution gives more weights to the top keywords. When all elements of n have been counted, and each A_i can be thought of as a position vector, we sort the keywords by several formulas, defined as

$$L_1(a_k) = \sum_{i=1}^{n} 1/a_{i_k}, \quad L_2(a_k) = \sqrt{\sum_{i=1}^{n} (1/a_{i_k})^2}, \tag{3}$$

$$GM(a_k) \quad = \quad (\prod_{i=1}^{n} 1/a_{i_k})^{1/n} . \tag{4}$$

Equation 3 represents the L_1 norm and the L_2 norm of these position vectors, and the geometric mean of the n points is expressed in Equation 4. We take into consideration the median of the n points as well. Borda's rule is commonly classified as a positional voting system because from each voter, candidates receive a certain number of points. Computationally it is very easy, as it can be implemented in linear time.

6.2 Bipartite Graph

Borda's rule does not assure us that it can find the optimal rank list because it does not optimize any criterion. A graph theory based method is used here, to approximate the optimal ranking. We define a weighted balanced bipartite graph $G = (V_1 \cup V_2, W)$. $V_1 = r_1, r_2, \cdots, r_m$ is a set of keywords to be ranked. $V_2 = p_1, p_2, \cdots, p_m$ is the m available positions in the rank list. For any two vertices $r \in V_1$ and $p \in V_2$, rp is an edge in G; thus G is also a complete bipartite graph. The weight $W(r, p) =$ is the total distance of a ranking value that places r at position p. The task of rank aggregation is to minimize the number of disagreements with the respective lists. Therefore, if all the keywords are put in proper positions, the total distance (i.e., the number of disagreements) should be the smallest. Now we meet two difficulties in achieving this goal. One is how to compute the distance. The other one is what kind of approaches can minimize the distance.

To weigh the edges in G, according to Diaconis et al. [13], the two distance measures that we consider are:

$$Footrule\_D(\pi, \sigma) = \sum_{i=1}^{n} \mid \pi(i) - \sigma(i) \mid , \; Footrule\_S(\pi, \sigma) = \sum_{i=1}^{n} (\pi(i) - \sigma(i))^2 , \tag{5}$$

where π and σ are regarded as rank lists. Diaconis et al. [13] also suggest two other measures. One roughly seems similar to $Footrule\_D$, and the other is unsuitable for general use, having very small variance about a mean that is very close to its maximum value. Therefore, we choose $Footrule\_D$ and $Footrule\_S$ here. We then adjust the two measures to compute the total distance that is the weight in an edge, now defined as $\sum_i^n \mid A_i(r) - p \mid$ or $\sum_i^n (A_i(r) - p)^2$. Minimizing the total distance to n could be solved by the well-known Hungarian algorithm that finds a minimum cost perfect matching in the bipartite graph. A matching in a graph is a set of edges where no two of which share an endpoint. Dwork et al. [14] uses $Footrule\_D$ as the distance measure to effectively combat *spam*. Our experiments compared the two measures and observed that the largest improvement is reached by $Footrule\_S$.

7 Experiment

In this section, we first introduce our data sets and evaluation method. Then we present experimental results. Finally, a case study is given.

7.1 Data Statistics

We have collected microblog posts given 14 Sina microblog trending topics from March 8th to June 29th, 2012 by crawling. Under our preliminary statistics, there are 63,354 posts form all users. The number of tweets by the authenticated users is about 22,724, accounting for

Table 3. Statistics of crawled Posts from 14 Sina trending topics

topics	new ipad	iphone	ipad show	apple ceo salary	apple app store	ce2012	HTC
users	4893	1020	5569	4111	5289	3792	934
posts	6043	1242	6889	5600	6760	6126	1234
retweeted posts	823	169	1415	660	1401	1313	146
total posts	6874	1419	8313	6324	8175	7453	1385
clean posts	6866	1411	8304	6320	8161	7439	1383
spammed posts	8	8	9	4	14	14	2
users/tweets	0.8097	0.8213	0.8084	0.7263	0.7823	0.619	0.7551

Table 4. Statistics of crawled Posts from 14 Sina trending topics

topics	tablet	Kodak bankruptcy	Huawei	iphone4s	windows 8	iOS 5.0.1	facebook
users	326	5648	933	4921	4537	1063	3971
posts	1382	7137	1116	6714	6516	1645	5046
retweeted posts	255	1931	332	1324	607	282	590
total posts	1556	9084	1448	8085	7125	1932	5639
clean posts	1538	9068	1448	8038	7123	1927	5636
spammed posts	18	16	0	47	2	5	3
users/tweets	0.2541	0.7913	0.836	0.7329	0.6963	0.6462	0.7869

about 35.9% of total users' posts. Repeated and spammed posts are removed. The statistical results are shown in Table 3 and Table 4 which reflect an overall distribution of these topics.

From the tables we can see that the users of the topic "iphone4s" talking about iPhone 4s sales have risen up to 4921 and the number of its tweets is about 8038. But in contrast to another topic of "Tablet", the number of users is only 326 and the number of tweets is only 1538. From these figures we can say that microblog persons pay more attention to the topic of "iphone4s". This topic is very popular at that moment. The topic of "Kodak bankruptcy" has the largest number of tweets and users in this period of our crawled time. Thus we can say that this topic is the most popular among 14 topics.

7.2 Experimental Setup

Here, we present some experimental details using the aforementioned three approaches. Each topic extracted by us is treated a search query representing user's information need. The collection of published posts in each topic is the search results returned by Sina microblog search engine.

Here, in Table 5, we have made a specific explanation for all abbreviations appeared in the following. Most of traditional pseudo relevance approaches use the collection of top returned search results as relevance. Since microblog search engines usually order returned posts by time to show the timeline of a search topic, we divide the whole posts of a search topic into 7 collections in every 15 days. Each 15-day collection (called "TOP") represents the top returned post results at that time and is used as relevance. In addition, the suggestion performances of two collections of posts are mainly compared. One is collected from all users in a topic, called "ALL"; the other is collected only from authenticated users in a topic, called "AU".

7.2.1 Frequency Based Ranking

For the posts of each of topic we conduct the preprocessing of removing stopwords and Chinese participle preprocessing. Then word frequency statistics are done and top 10 representative nouns or verb-nouns are extracted. We think that nouns or verb-nouns can better represent the information need of a user. We compare the suggestion quality of using "AU" with that of

Table 5. Abbreviation explanation

	Meaning
AU: Authenticated posts	The posts published by authenticated users in a collection is used,
	which is our proposed idea.
ALL: Total posts	The whole posts returned are selected,
	which is the main idea of traditional pseudo relevance based approaches.
TOP: Top m posts	The top m posts ordered by publish time are returned by search engine,
	which is the main idea of traditional pseudo relevance based approaches.

"ALL" and "TOP". Notice that using the collection of top returned post results (i.e., "TOP') is what traditional pseudo relevance approaches do.

7.2.2 LDA Based Ranking

The first component of our LDA approach is the training data. The topics analyzed from this training data directly influence the learning and suggesting performance of our approach. Our training data is from the whole collection of microblog posts in 14 topics. We assume that the crawled posts working as a small post archive can cover the content topics that are relevant to the posts themselves for topic model learning and inference. Although we can crawl more microblog posts, our experimental results show that LDA is not good at dealing with short texts like posts. Other public universal Web sources such as ODP, TREC and etc., are relatively static, and thus they cannot catch the newly appeared things as microblogging can. Finding a suitable training data with rich text representation and kept updated is an interesting future work.

Last, we report the computing setting parameters of LDA model. At the initialization stage, α is set to be 0.1 and β is accordingly generated by the program shown in Table 2 and based on α and the training data. Then, the LDA model estimation uses the variational EM algorithm [5] to optimally choose the variational parameters that maximize the total likelihood of the corpus. The formula is given in eq.(2) where the EM algorithm is executed iteratively until a maximum likelihood value reaches (i.e., the iterations repeat until $(l_{k+1} - l_k)/l_k < 1e-5$).

7.2.3 Graph Based Rank Aggregation

In the context of web search, most search users just browse top 10 or 20 results. The top ranked items are more likely clicked. Therefore, we consider top 10 keywords in a final suggestion list. Since the above two ranking approaches may produce different keywords, the total candidate keywords are more than 10. Our final aggregated list still provides 10 positions. Methods presented in Section 6 are compared in the following experiments.

7.3 Evaluation Method

The precision (P) at the top N keywords of a recommendation list is defined as:

$$Precision@N = \frac{\#related\ keywords\ in\ a\ suggestion\ list}{N}. \tag{6}$$

The measure $Precision@N$ means how many valuable answers our algorithm gives at the top of recommendation lists. We set N=5 and 10 in the following evaluations. Query recommen-

Table 6. The average precision scores using the frequency based ranking

	Precision@5	Precision@10
TOP (3.08–3.23)	0.0929	0.1357
TOP (3.24–4.09)	0.25	0.3214
TOP (4.10–4.25)	0.3	0.3785
TOP (4.26–5.11)	0.2642	0.3571
TOP (5.12–5.27)	0.2357	0.3071
TOP (5.28–6.13)	0.2143	0.2929
TOP (6.14–6.29)	0.2071	0.2714
ALL	0.3214	0.4286
AU	**0.3286**	**0.4357**

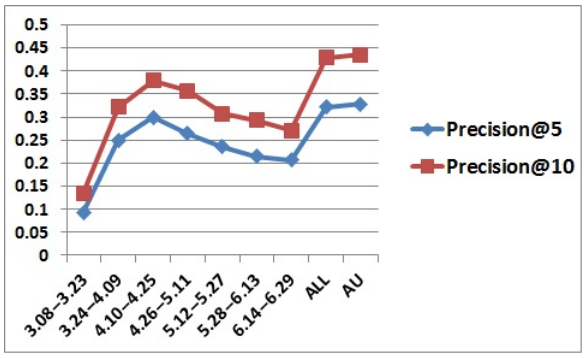

Fig. 3. The change trend of precison scores with time

dation is a ranking problem. We can evaluate our system by other evaluation measures like MAP and $NDCG$ [28] which are also used for ranking problems, especially in Web search. Because suggested keywords are much shorter than that of a Web page, Web users can read the top ten or twenty recommended keywords much more quickly than they browse and click the top Web pages one by one. In evaluation, we are more interested in the number of related keywords in a suggestion list, i.e., *Precision* than the order in which the queries are returned such as MAP, $NDCG$ and so on.

When we take the extracted top N representative nouns or verb-nouns as the suggested keywords, we manually judge whether these keywords can be considered to accurately reflect its search query. The precision values of 14 search queries are computed and their average score is reported in our experiments.

7.4 Experimental Results

Here, we discuss the experimental results of the three ranking approaches in terms of *Precision*@N as defined in Equation 6.

7.4.1 Frequency Based Ranking

The average results of all the 14 topics are listed in Table 6 and how they change with time is show in Figure 3.

"TOP (3.08–3.23)" means the precision score of using the top returned posts until 23rd March 2012. With the time goes on, a new topic will attract more and more users to publish

posts for representing their own opinions. It means more posts are returned as relevance to suggest keywords. However, the increasing trend in Figure 3 is interrupted at some time point and the precision scores are decreased. In our experiments, the turn point is 10th April to 25th April. This observation tells us that a topic in mircoblogging has strong characteristics of timeline. Users lost their interests in some topic and then publish less posts than before, so the suggestion relevance is not rich enough. Using top returned posts like traditional pseudo relevance based approaches cannot give good suggestion quality.

We consider how about the suggestion quality if we cover more returned posts in suggestion. To answer this question, we enlarge the relevance coverage by combining the returned posts from 23rd March to 29th March, i.e., "ALL" in Table 6 and Figure 3. "ALL" produces 0.3214 in terms of precision@5 and 0.4286 in terms of precision@10, which is better than that of any "TOP". Can we conclude that larger size of relevance is better?

We further think that the quality of relevance may be more important than its size. microblogging has its own characteristics in information sharing. Usually authenticated users are more reliable than others not only in information authenticity but also in text quality. Therefore, we extract the posts published by authenticated users from the whole posts by all users and conduct experiments again. As show in Table 6 and Figure 3, the precision scores of "AU" are slightly higher than those of "ALL", i.e., **0.3286 VS 0.3214 and 0.4357 VS 0.4286**. The finding is useful for online suggestion service. Text processing is time-consuming and largely affects the suggestion efficiency. The text size of "AU" is one third of that of "ALL" and the ratio is 0.3575. It takes us about 4 hours to do text processing for "ALL" and only 30 minutes for "AU" run by a PC of a 32-bit operating system, dual-core CPU and 3.00GB memory. "AU" can run faster 8 times than "ALL", but it still shows better suggestion quality.

7.4.2 LDA Based Ranking

This part will report the suggestion results by using the LDA based ranking. Search query is the input variables of LDA. Then we get their topic distributions and the topic with the highest topic probability is selected. Last, the top 10 keywords with highest probability in this selected topic are in our suggestion set. Since 14 search queries are used in our experiments with representing 14 search topics, the parameter K in LDA is set to be 10 and 15 which are close to 14. We also test other values of K in evaluation, but their suggestion precision scores are lower than that of using *K=10 or K=15*.

As shown in Table 7, the precision scores of "AU" are slightly lower than those of "ALL", i.e., **0.2243 VS 0.2429 and 0.2857 VS 0.3** when *K=10*. When comparing Table 6 with Table 7, we find that the frequency based ranking works much better than the LDA based ranking. For example, in terms of *precision@10*, using "ALL" data, their precision scores are 0.4286 and 0.3, respectively. LDA tries to discover latent topic structure via estimating the probability distribution of the original co-occurrence activities. Microblog posts are shorter than 140 characters, so their co-occurrences are not so richer than traditional long text web pages.

Although the LDA based ranking provides less related keywords, we check its suggestion lists and find that its suggestions are quite different from those given by the frequency based ranking. For more intuitive observations, we compare the specific keywords of a search query

Table 7. The average precision scores using LDA based ranking

K=10

	Precision@5	Precision@10
ALL	0.2429	0.3
AU	0.2243	0.2857

K=15

	Precision@5	Precision@10
ALL	0.1571	0.3
AU	0.1571	0.2632

Table 8. Two list of top 10 suggested keywords

	LDA based ranking	*Frequency based ranking*
	Keyword	Keyword
1	Kodak	Kodak
2	Company	Bankrupt
3	Media	Roll Film
4	Software	Digital Camera
5	Custom	Application
6	System	Protection
7	China	World
8	Sharing	Digital
9	America	Understanding
10	HongKong	Market
Precision	0.3	0.5

"Kodak bankrupt" suggested by frequency based ranking with LDA based ranking, as listed in Table 8. We can see that the LDA based ranking generates more abstract and topic related keywords while the suggestions of the frequency based ranking are more specific.

7.4.3 Graph Based Rank Aggregation

From the above results, we see that the suggested keywords are quite different between the frequency based ranking and the LDA based ranking. The former considers word itself, while the latter takes into account co-occurrence relationship between words. Thus, the LDA base ranking produces related words in terms of hidden topic feature space. Combining their suggestion would give users more choices. We report experimental results of the graph based ranking using Equation 3 to Equation 5 in Table 9. *Footrule_S*, a bipartite graph based ranking, gains the highest precision. We notice that the rank aggregation largely improves the performance of a single ranking. For example, in terms of precision@10, the graph based rank aggregation achieves 0.72534, while the frequency based ranking and the LDA based ranking are 0.4286 and 0.3, respectively. The result is consistent with our observations that the two single rankings works differently and generates different suggestions from their own viewpoints. Their combination gives us higher suggestion quality. In addition, the precision score of "AU" are highly comparable with "ALL", i.e., **0.5513 VS 0.5601 and 0.7211 VS 0.7234**. The post quality of "AU" is higher than that of "ALL" and thus shows better suggestions.

Table 9. The average precision scores using graph based ranking

		L_1	*L_2*	*GM*	*Footrule_D*	*Footrule_S*
Precision@5	ALL	0.5521	0.5508	0.5513	0.4796	**0.5601**
Precision@5	AU	0.5411	0.5375	0.5398	0.4358	**0.5513**
Precision@10	ALL	0.7248	0.7215	0.7231	0.6582	**0.7234**
Precision@10	AU	0.7210	0.7198	0.7203	0.6326	**0.7211**

Table 10. The precision scores of different methods

		Frequency	*LDA*	*Borda*	*Footrule_S*
Precision@5	ALL	0.3214	0.1571	0.5521	**0.5601**
Precision@5	AU	0.3286	0.1571	0.5411	**0.5513**
Precision@10	ALL	0.4286	0.3	0.7248	**0.7234**
Precision@10	AU	0.4357	0.2632	0.7210	**0.7211**

For a clear view, we put all the discussed methods in Table 10. "ALL" is a traditional way to use pseudo relevance. When it is applied in traditional frequency based and LDA based ranking, the performance is lower than our proposed graph based rank aggregation. "AU" is our proposed pseudo relevance selection since microblogging has such a distinct feature. Authenticated users easily are identified, which helps us automatically select relatively high quality posts as relevance. When "AU" is applied in tradtional and our proposed ranking methods, it shows comparative precision quality with "ALL" while running much faster.

7.4.4 Case Study

Suggested keywords are used to improve search quality. In this part, we present case study by investigating the search results of suggested keywords. Let us take the search topic *Kodak* as an example. The top 10 representative nouns or verb-nouns produced by total posts ad authenticated posts as shown in Table 11.

The top 10 words comprise a suggestion list, as show in Table 11. According to the context of this search topic, users may think that these words of *Kodak*, *Bankrupt*, *Applying*, *Protection* are related keywords. When we enter the key word of *Kodak* into Baidu search box, the search engine returns a number of results. The top 10 result sets as shown in Table 12.

We give out the URLs of top 10 result sets and their brief content descriptions. So we can clearly see that most of webpage are about the general information of Kodak, such as official website, Baidu Baike, customer service call and so on. Among them, only one piece of information is about Kodak bankruptcy. We found that at that time, Kodak was in the news of bankrupt protection. However, traditional web search engine has difficulty to bring newly appeared content to the top list of search results.

If we respectively use the suggested keywords of *bankruptcy* and *protection* added into the Baidu search box. The result sets as shown in Table 13 and Table 14.

It is obvious that after we added our recommended word, almost 80% webpages are about the topic of *Kodak filed for bankruptcy protection*. Experiment results show that our approach is effective, especially the words like *bankruptcy* and *protection* that represented the latest news of Kodak.

Table 11. Top 10 representative nouns or verb-nouns produced by total tweets and uthenticated tweets

	ALL		*AU*	
	Words	Frequency	Words	Frequency
1	Kodak	8954	Kodak	3618
2	Bankrupt	2561	Bankrupt	1896
3	Roll Film	1615	Roll Film	909
4	Company	1196	Digital Camera	755
5	Nokia	1163	Application	753
6	Digital	1104	Protection	732
7	Digital Camera	1033	World	448
8	Application	998	Digital	430
9	Photographic Film	983	Understanding	416
10	Protection	909	Market	415
Precision		0.4		0.4

Table 12. Top 10 search results of the query *Kodak*

	URL	Description
1	www.kodak.com.cn/	Kodak Chinese official website
2	baike.baidu.com/view/60113.htm	Kodak Baidu Baike
3	www.baidu.com/s?tn=baidurt&rtt=1&bsst=1&wd	Kodak's latest relevant information
4	s.leho.com/kefu?keyword=%BF%C2%B4%F	Kodak customer service call
5	zhidao.baidu.com/question/371803669.html	Kodak bankruptcy
6	gouwu.baidu.com/s?ie=gbk&wd=%BF%C2%B4%EF	Kodak products
7	www.kodak.com/ek/US/en/Home.htm	Kodak
8	detail.zol.com.cn/digital_camera_index/subcate15_139_list_1.html	Kodak product
9	www.mvgod.com/theater/ShangHai/KDS/	Kodak film news
10	dcdv.zol.com.cn/manu_139.shtml	Kodak products

8 Conclusions and Future Work

In this paper, we present how to suggest top n keywords by adding microblog data. Microblog, as a new product, has the characteristics of high efficiency of information dissemination. That is very important to our work. It means that some newly appeared contents usually exist in the microblog, which can give effective suggestion for hot and new-event related search queries. We find that the frequency-based ranking is simple but show much better suggestion quality than LDA based ranking. Combining the two ranking can largely improve the suggestion quality. It is worth mentioning that the average precision score produced by the posts that from authenticated users is actually comparable with that that of the posts from all users, and it runs 8 times faster. Moreover, our approach shows much better suggestions than the traditional pseudo relevance based method that uses the top returned results. In the future, an interesting topic is how to combine other social evidence to enhance query suggestion quality.

References

1. R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query recommendation using query logs in search engines. In *Proceedings of the 2004 International Conference on Current Trends in Database Technology*, EDBT'04, pages 588–596, Berlin, Heidelberg, 2004. Springer-Verlag.
2. R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Improving search engines by query clustering.

Table 13. Top 10 search results of the query *Kodak bankruptcy*

	URL	Description
1	zhidao.baidu.com/question/371803669.html	Kodak bankruptcy
2	tech.qq.com/zt2012/kodakpochan/	Kodak filed for bankruptcy protection
3	tech.sina.com.cn/z/kodakbanrupt/	Kodak filed for bankruptcy protection
4	it.sohu.com/s2012/kodakfilesforbankruptcy/	Kodak filed for bankruptcy protection
5	wenku.baidu.com/view/2a7d4efaf705cc17552709b4.html	The Kodak bankruptcy Analysis
6	info.china.alibaba.com/detail/1073730381.html	Kodak officially filed for bankruptcy protection
7	www.36kr.com/p/78165.html	Kodak bankruptcy
8	topic.eastmoney.com/keda/	Kodak filed for bankruptcy
9	topic.weibo.com/it/19160	Kodak filed for bankruptcy
10	finance.youku.com/kodak	Kodak officially bankrupt

Table 14. Top 10 search results of the query *Kodak protection*

	URL	Description
1	http://tech.qq.com/zt2012/kodakpochan/	Kodak filed for bankruptcy protection
2	http://tech.sina.com.cn/z/kodakbanrupt/	Kodak filed for bankruptcy protection
3	http://finance.qq.com/zt2011/kodak/	Kodak filed for bankruptcy protection
4	http://it.sohu.com/s2012/kodakfilesforbankruptcy/	Kodak filed for bankruptcy protection
5	http://topic.weibo.com/hot/19166	Kodak filed for bankruptcy protection
6	http://tech.ifeng.com/it/special/goodbyekodak/	Kodak filed for bankruptcy protection
7	http://finance.sina.com.cn/focus/kodak/	Kodak filed for bankruptcy protection
8	http://bbs.hebnews.cn/thread-1096205-1-1.html	The reasons of Kodak filed for bankruptcy
9	http://business.sohu.com/s2012/kodak/	Kodak filed for bankruptcy protection
10	www.baidu.com/s?tn=baidurt&rtt=1&bsst=1&wd	Kodak's latest relevant information

JASIST, 58(12):1793–1804, 2007.

3. M. Barouni-Ebrahimi and A. A. Ghorbani. A novel approach for frequent phrase mining in web search engine query streams. In *CNSR*, pages 125–132. IEEE Computer Society, 2007.
4. S. Bhatia, D. Majumdar, and P. Mitra. Query suggestions in the absence of query logs. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 795–804, New York, NY, USA, 2011. ACM.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
6. P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 Workshop on Web Search Click Data*, WSCD '09, pages 56–63, New York, NY, USA, 2009. ACM.
7. J. Borda. Mmoire sur les lections au scrutin. *Comptes rendus de lAcadmie des sciences*, 44, 1781.
8. A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, D. Metzler, L. Riedel, and J. Yuan. Online expansion of rare queries for sponsored search. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 511–520, New York, NY, USA, 2009. ACM.
9. H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 875–883, New York, NY, USA, 2008. ACM.
10. X. Chen, L. Li, G. Xu, Z. Yang, and M. Kitsuregawa. Recommending related microblogs: A comparison between topic and wordnet based approaches. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada, AAAI*, pages 2417–2418. AAAI Press, 2012.
11. S. Cucerzan and R. W. White. Query suggestion based on user landing pages. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 875–876, New York, NY, USA, 2007. ACM.
12. H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proceedings of the Eleventh International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA*, pages 325–332. ACM, 2002.
13. P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

14. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. of the 10th Int'l Conf. on World Wide Web (WWW'01)*, pages 613–622, Hong Kong, China, 2001.
15. A. Feuer, S. Savev, and J. A. Aslam. Evaluation of phrasal query suggestions. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 841–848, New York, NY, USA, 2007. ACM.
16. L. Fitzpatrick and M. Dent. Automatic feedback using past queries: Social searching? In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '97, pages 306–313, New York, NY, USA, 1997. ACM.
17. W. Gao, C. Niu, J.-Y. Nie, M. Zhou, J. Hu, K.-F. Wong, and H.-W. Hon. Cross-lingual query suggestion using query logs of different languages. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 463–470, New York, NY, USA, 2007. ACM.
18. N. S. Glance. Community search assistant. In *In Artificial Intelligence for Web Search*, pages 91–96. AAAI Press, 2000.
19. N. S. Glance. Community search assistant. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, pages 91–96, New York, NY, USA, 2001. ACM.
20. J. Guo, X. Cheng, G. Xu, and H. Shen. A structured approach to query recommendation with social annotation data. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM*, pages 619–628. ACM, 2010.
21. R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 387–396, New York, NY, USA, 2006. ACM.
22. D. Kelly, K. Gyllstrom, and E. W. Bailey. A comparison of query and term suggestion features for interactive searching. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 371–378, New York, NY, USA, 2009. ACM.
23. C. Klamler. The dodgson ranking and the borda count: a binary comparison. *Mathematical Social Sciences*, 48(1):103–108, 2004.
24. H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.
25. L. Li, G. Xu, Y. Zhang, and M. Kitsuregawa. Random walk based rank aggregation to improving web search. *Knowl.-Based Syst.*, 24(7):943–951, 2011.
26. L. Li, Z. Yang, L. Liu, and M. Kitsuregawa. Query-url bipartite based approach to personalized query recommendation. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 1189–1194. AAAI Press, 2008.
27. H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 709–718, New York, NY, USA, 2008. ACM.
28. C. D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval, 2008.
29. Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 469–478, New York, NY, USA, 2008. ACM.
30. M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Phys. Rev. E*, 68:036122, Sept. 2003.
31. V. V. Raghavan and H. Sever. On the reuse of past optimal queries. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 344–350, Seattle, Washington, USA, 1995.
32. F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2010.
33. Y. Song and L. wei He. Optimal rare query suggestion with implicit user feedback. In M. Rappa,

P. Jones, J. Freire, and S. Chakrabarti, editors, *WWW*, pages 901–910. ACM, 2010.
34. J.-R. Wen, J.-Y. Nie, and H. Zhang. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20(1):59–81, 2002.
35. J.-M. Yang, R. Cai, F. Jing, S. Wang, L. Zhang, and W.-Y. Ma. Search-based query suggestion. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1439–1440, New York, NY, USA, 2008. ACM.
36. H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.
37. L. Zhiyuan, C. Xinxiong, and S. Maosong. Mining the interests of chinese microbloggers via keyword extraction. *Foundations and Trends in Information Retrieval*, 6(1):76–87, 2012.