Intent-Aware Query Similarity

Jiafeng Guo[†], Xueqi Cheng[†], Gu Xu[‡], Xiaofei Zhu[†]

[†]Institute of Computing Technology, CAS
Beijing, P. R. China
guojiafeng@software.ict.ac.cn,cxq@ict.ac.cn,
zhuxiaofei@software.ict.ac.cn

[‡]Microsoft Research Asia
Beijing, P. R. China
guxu@microsoft.com

ABSTRACT

Query similarity calculation is an important problem and has a wide range of applications in IR, including query recommendation, query expansion, and even advertisement matching. Existing work on query similarity aims to provide a single similarity measure without considering the fact that queries are ambiguous and usually have multiple search intents. In this paper, we argue that query similarity should be defined upon search intents, so-called *intent-aware query similarity*. By introducing search intents into the calculation of query similarity, we can obtain more accurate and also informative similarity measures on queries and thus help a variety of applications, especially those related to diversification. Specifically, we first identify the potential search intents of queries, and then measure query similarity under different intents using intent-aware representations. A regularized topic model is employed to automatically learn the potential intents of queries by using both the words from search result snippets and the regularization from query co-clicks. Experimental results confirm the effectiveness of intent-aware query similarity on ambiguous queries which can provide significantly better similarity scores over the traditional approaches. We also experimentally verified the utility of intent-aware similarity in the application of query recommendation, which can suggest diverse queries in a structured way to search users.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

Query Similarity, Search Intent, Regularized Topic Model, Pair-wise Measure, Graph-based Measure

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.
Copyright 2011 ACM 978-1-4503-0717-8/11/10 ...$10.00.

1. INTRODUCTION

Calculating similarities between queries is a key element of various IR applications. For example, query recommendation [2, 34] manages to provide similar queries to users and help them to reformulate their queries regarding their information needs. In query expansion [14, 17], similar terms (words) are added to improve the recall of search, and the expanded query can also be considered as a similar query to the original query. Besides, query similarity can also be helpful to advertisement matching [29]. However, due to the high ambiguity of queries, how to properly define the similarity between queries is not a trivial problem. For example, given query "apple", it is similar to "apple tree" if the searcher is looking for apple fruits, while it is also similar to "apple store" if the search intent is to find products of the apple company. The similarity is actually not comparable across different search intents, which means that we cannot say "apple tree" is more similar to "apple" than "apple store" and vice versa. In this paper, we argue that the similarity between queries should be defined upon search intents, so-called *intent-aware query similarity*. Note that search intent here refers to the goal or need of a user during a search. In this manner, we can provide more precise and also informative similarity information without being biased by popular intents or wrongly making queries from different intents similar. Evidently intent-aware query similarity will be especially helpful for various diversity problems.

To the best of our knowledge, none of existing work formally addressed the problem of similarity calculation with the awareness of search intents and thus it is the unique position of this paper. Also it is important to notice that traditional similarity measures may encounter some problems when dealing with ambiguous queries, namely the queries with multiple search intents. Various data sources, including search results [30], user clicks [2, 26, 13] and search sessions [34, 9], have been employed to enrich the representation of search queries. While similarity measures defined on these representations can be divided into two major categories:

Pair-wise Measures. The similarity is independently measured on each query pair, using similarity functions like cosine similarity [2, 33], Jaccard coefficient [3], kernel functions [30], etc. For ambiguous queries, the information from multiple search intents are actually mixed together and thus pair-wise measures without the awareness of search intents will be easily biased by dominant search intents and ignore unpopular ones, for example query "apple" is only similar to "apple store" but not related to "apple tree".

Graph-based Measures. The similarity between queries is defined on a query graph, or query relation graph [13, 9]. It means that the similarity is not pair-wise independent and the similarity of adjacent queries is considered. In some sense, graph-based measures can be considered as propagating the similarity on top of the query graph. However, if the queries are ambiguous, the propagation should not cross the boundary of different search intents. Without the awareness of search intents, the queries with different search intents will be wrongly connected, for example "apple store" becomes similar to "apple tree".

In this paper, as the first attempt, we cast some light on the problem of "intent-aware query similarity". Specifically, we propose first identifying the potential search intents of queries, and then measuring query similarity under different search intents using intent-aware representations. Note that the identification of potential search intents of queries are largely different from classifying the queries into a set of predefined categories [23, 12], since the search intents of queries are usually fine-grained and substantial. In our work, a regularized topic model is employed to automatically learn the potential intents of queries by using both the words from search result snippets and the regularization from query co-clicks. Based on the learned intents of queries, we then extract the query representation under each intent, and thus different measures can be applied to measure query similarity with respect to different search intents. Both pair-wise measures (e.g., cosine similarity measure) and graph-based measures (e.g., spectral embedding [4]) can be easily applied.

While there are many applications requiring a similarity measure between queries, one direct application in the context of search is query recommendation. In this paper, we adopt query recommendation as an example to demonstrate the usefulness of introducing search intent into the calculation of query similarity. With our approach, we can identify similar queries to the user's initial query with respect to different search intents, and then a structured query recommendation [19] can be easily built by grouping diverse recommendations based on the search intents.

We conducted experiments based on a large collection of query logs and search results from a commercial search engine, and demonstrated the effectiveness of our intent-aware query similarity by comparing with other baseline methods. Experimental results show that by identifying search intents of queries, we can measure query similarity significantly better than traditional approaches. We also demonstrate that by grouping query recommendations based on search intents and presenting them in structured way, we can largely enhance users' click behaviors on recommendations.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 describes our solution to intent-aware query similarity. Section 4 discusses applying our application to query recommendations. Experimental results are presented in Section 4. Conclusions are made in the last section.

2. RELATED WORK

Measuring similarity between queries is an interesting and difficult problem. A reliable query similarity measure can be very useful for a variety of applications such as query recommendation [2, 34, 26], query reformulation [32, 1, 22], query expansion [14, 17], and advertising [29]. However, since queries are usually very short and ambiguous, it is difficult to calculate their similarity only based on query terms. Many existing approaches resolve the information inadequate problem in query by leveraging auxiliary information, including search results [30], clickthrough [2, 26, 13] or search sessions [34, 9], to enrich query representation for better similarity measurement.

Based on the augmented representation of queries, two major approaches are then applied to measure similarity between queries:

Pair-wise Measures. The similarity is independently measured on each query pair using pair-wise metrics. For example, Beeferman et al. [3] leveraged Jaccard similarity coefficient over the clickthrough vector of queries as the similarity measure. Baeza-Yates et al. [2] calculated query similarity using cosine similarity based on the aggregation of the term-weight vectors of the URLs clicked after the query. Wen et al. [33] applied various similarity metrics over both query term vectors and clickthrough vectors to measure query similarity, including term overlap, cosine similarity, edit distance, and Jaccard coefficient. Typical approaches using the hybrid similarity measurements over queries can also be found in [34, 22]. In [16], Deng et al. introduced two new schemes for representing queries based on clickthrough and applied both cosine similarity and Jaccard coefficient for measuring query similarity. In [30], Sahami et al. proposed a Kernel Function for measuring the query similarity based on the Tf-Idf weighted vectors of search result snippets. However, for ambiguous queries, the information from multiple search intents are actually mixed together and thus pair-wise measures without the awareness of search intents will be easily biased by dominant search intents and ignore unpopular ones.

Graph-based Measures. The similarity between queries is defined on a query graph, or query relation graph [13, 9]. Therefore, the similarity is not pair-wise independent and the similarity of adjacent queries is considered. In some sense, graph-based measures can be considered as propagating the similarity on top of the query graph. For example, Craswell et al. [13] applied two types of random walk process to propagate the query similarity along the query-URL bipartite graph and obtain better similarity scores between queries. Mei et al. [26] described a random walk on the one-mode query graph and employed the hitting time as the similarity measure. In [1], Antonellis et al. used the bipartite SimRank on the query-URL graph to measure query similarity. In [24], matrix factorization was employed to calculate query similarity based on the user-query-URL tripartite graph. Recently, Bordino et al. [9] introduced the query-flow graph based on the session data and employed graph projection to measure query similarity. Work on such a graph to compute query similarity can also be found in [7, 8]. However, if the queries are ambiguous, the graph-based measures without the awareness of search intents will wrongly connect the queries with different search intents.

Related work also includes topic modeling. Topic modeling has been popularly used for data analysis in various domains including topic discovery, document classification, citation analysis, and social network analysis. Topic models, such as Probabilistic Latent Semantic Indexing (PLSI) [21] and Latent Dirichlet Allocation (LDA) [6] have shown impressive empirical success in revealing the hidden structures of documents and in related applications like document classification and collaborative filtering. Based on the above models, a set of variants and extensions [18, 5, 31] have been further proposed to address document modeling problems in different scenarios. Our work exploits topic modeling in a new application (i.e., query similarity measure), and we employed a regularized topic model to fully leverage both search result snippets and co-clicks to help learn the search intents of queries. There have been several regularized topic models proposed to incorporate auxiliary knowledge as a constraint into topic model learning process and show the resulting benefits. For example, Cai et al. proposed two topic models, Laplacian pLSI (LapPLSI) [10] and Locally-consistent Topic Modeling (LTM) [11], which incorporate manifold structure information as a constraint into the PLSI model to smooth the probability density functions. Similarly, Mei et al. [25] regularized the statistical topic model PLSI with a harmonic regularizer based on a graph structure in the data. In [20], Guo et al. introduced a weakly supervised topic model, i.e. WS-LDA, by incorporating human labels as a soft constraints into the LDA model to supervise the topic alignment.

3. OUR APPROACH

In our work, we propose to measure query similarity with the awareness of search intents. The key idea is that we first identify the potential search intents of queries, and then measure the query similarity under different search intents using their intent-aware representations. As aforementioned, there are a large amount fined-grained potential search intents behind different queries (e.g. the fruit or product intent for query "apple"), and it is usually difficult to define an appropriate prior taxonomy for search intents. Therefore, we propose to leverage the large amount of hidden topics in topic model to help capture the potential search intents of queries.

Specifically, a regularized topic model is employed to automatically learn the potential intents of queries by using both the words from search result snippets and the regularization from query co-clicks. Based on the learned intents of queries, we then extract the query representation under each intent, and thus different measures can be applied to measure query similarity with respect to different search intents. Both pair-wise measures (e.g., cosine similarity measure) and graph-based measures (e.g., spectral embedding) can be easily applied.

In this section, we will first introduce the auxiliary data we leveraged for identifying the search intents of queries. We then describe the regularized topic model for the key learning problem. Finally, we show how to extract the query representation under each intent, and apply different similarity metrics to measure query similarity with respect to search intent.

3.1 Data for Identifying Search Intents of Queries

In order to identify the potential search intents of queries, we need to leverage some auxiliary data to provide a rich representation for short queries. Different types of auxiliary data, including search results [30], clickthrough [2, 26, 13] or search sessions [34, 9], have been leveraged to enrich query representation for similarity measurement in many existing approaches. For ambiguous queries, such auxiliary data is a mixture of information from multiple search intents. However, traditional approaches usually directly leverage such kind of mixed-intent representations of queries for similarity measure. In our work, we leverage both search result snippets and clickthrough of queries to help learn the potential search intents of queries. This is largely different from the traditional work.

Although both the two types of data can provide rich information for identifying potential search intents of queries, they have some natural different characteristics. The search result snippets provide a great context for the query. We can thus construct a virtual document by the words from the snippets of the top search results to well "describe" the given query. Thanks to the advance of modern search engines, such a description often show a high recall of potential search intents of the given query. Take the query "office" for example, a typical search results from the commercial search engines (e.g. Google, Bing or Yahoo!) may contain the content about the microsoft office software, the office tv series, and even some office furniture or supplies. However, search results may also have some irrelevant, spam or advertisement information, and words itself may have ambiguity. All these factors may hurt the quality or precision of the search result snippets on resolving search intents of queries.

Meanwhile, the large amount of clickthough data from search logs also provide us useful information for identifying search intents of queries. Unlike the search result snippets, clickthrough information is a resource with higher precision but lower recall. As we can see, although one query may convey multiple search intents, it is often determined when a specific URL is clicked. Therefore, if two queries share a set of same clicked URLs, they will convey similar search intent [2, 3]. The "wisdom of the crowds" property thus makes clickthrough, especially the query co-clicks, a more precise resource for identifying similar search intent of queries. However, query co-click information is relative sparse (i.e. lower recall) in describing search intents compared with the search result snippets, since usually there are limited clicks for each query.

Based on the above analysis of the different characteristics in search result snippets and clickthrough, we propose to simultaneously leverage the two types of auxiliary data in our work so that they can supplement each other in resolving search intents of queries.

3.2 Learning Potential Search Intents with a Regularized Topic Model

From previous section we can see, the search result snippets provide rich context information for a given query. We can thus construct a virtual document by the words from the search result snippets to describe the given query and capture its major search intents. Since queries may convey multiple search intents, it is natural to apply a mixture model (e.g. topic model) over the virtual document of the query to help learn the potential search intents. Meanwhile, the clickthrough information, especially the query co-clicks, provides us clear evidence about which queries convey the similar search intent. Such evidence can be used as a powerful constraint over the intent distribution of queries to help us reveal the search intents of queries more accurately. Therefore, we propose to employ a regularized topic model to fully leverage the two types of auxiliary data to help learn the potential search intents of queries.

3.2.1 *The Regularized Topic Model*

Suppose there is a collection of N queries $\mathcal{Q} = \{q_1, \ldots, q_N\}$ sharing the same set of K potential search intents $\mathcal{S}$ =

$\{s_1, \ldots, s_K\}$, and each query is represented by a set of words $w \in \mathcal{W} = \{w_1, \ldots, w_M\}$, which are collected from its top search result snippets. By viewing queries as virtual "documents", words from top search result snippets as "words", and potential search intents as "topics", we can apply the Probabilistic Latent Semantic Indexing (PLSI) [21] to model the generation of each query and its words from top search result snippets by the following scheme:

1. select a query q_i with probability $P(q_i)$,
2. pick a potential search intent s_k with probability $P(s_k|q_i)$,
3. generate a word w_j with probability $P(w_j|s_k)$.

By summing out the latent variable s, the joint probability of an observed pair (q_i, w_j) can be computed as

$$\begin{aligned} P(q_i, w_j) &= P(q_i)P(w_j|q_i) \\ &= P(q_i)\sum_{k=1}^{K} P(w_j|s_k)P(s_k|q_i) \end{aligned} \quad (1)$$

Based on this joint probability, we can calculate the log-likelihood as

$$\tilde{\mathcal{L}} = \sum_{i=1}^{N}\sum_{j=1}^{M} n(q_i, w_j) \log\Bigg(P(q_i)\sum_{k=1}^{K} P(w_j|s_k)P(s_k|q_i)\Bigg) \quad (2)$$

where $n(q_i, w_j)$ denotes the number of times word w_j occurred in the top search result snippets of query q_i. Following the maximum likelihood principle, one can determine the model parameters $\{P(w_j|s_k), P(s_k|q_i)\}$ by maximizing the relevant part of Equation (2).

Recall that the query co-clicks provide us clear evidence about similar search intent between queries. That is, if two queries share many same clicked URLs, they convey similar search intent. Therefore, we can use the co-click information as a constraint over the search intents of queries to help us reveal the intents more accurately. More formally, we need to minimize the proximity of the probability distribution $P(s|q)$ of co-clicked query pairs, expressed by

$$\mathcal{R} = \sum_{i,j=1}^{N}\sum_{k=1}^{K} C_{ij}(P(s_k|q_i) - P(s_k|q_j))^2 \quad (3)$$

where matrix C contains the co-click relationship between queries with C_{ij} set as unique co-clicked URL number of the query pair (q_i, q_j). Note that other definitions for the co-click matrix C and proximity measure can also be used.

Now we give out the regularized topic model for identifying potential search intents of queries. The model adopts the generative scheme of PLSI. It aims to maximize the *regularized* log-likelihood as follows:

$$\begin{aligned} \mathfrak{L} &= \mathcal{L} - \lambda\mathcal{R} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(q_i, w_j) \log\Bigg(P(q_i)\sum_{k=1}^{K} P(w_j|s_k)P(s_k|q_i)\Bigg) \\ &\quad -\lambda\sum_{i,j=1}^{N}\sum_{k=1}^{K} C_{ij}(P(s_k|q_i) - P(s_k|q_j))^2 \end{aligned} \quad (4)$$

where λ is the regularization parameter. Note that the regularized topic model essentially have the same form of the LapPLSI model proposed by Cai et al. [10]. However, in their case, they adopted the manifold structure information from document dataset to help modeling latent topics in the same document collections. While in our problem, we adopt the regularized topic model to effectively combine the two different types of data (i.e. search result snippets and query co-clicks) in the learning task for identifying the potential search intents of queries.

3.2.2 Model Fitting

The standard procedure for maximum likelihood estimation in latent variable model is the Expectation Maximization (EM) algorithm [15]. Here we use the Generalized EM algorithm [27] for parameter estimation in our regularized topic model as [10]. The major difference between Generalized EM and traditional EM is that in the M-step, Generalized EM only finds parameters that "improve" the expected value of the *complete* data log-likelihood function rather than "maximizing" it.

In our model, we have $NK + MK$ parameters $\{P(w_j|s_k), P(s_k|q_i)\}$ to be estimated, which is the same as PLSA. For simplicity, we define $\Phi = \{P(w_j|s_k)\}$ and $\Theta = \{P(s_k|q_i)\}$.

E-step: The E-step of the regularized topic model is exactly the same as that of PLSI. By applying Bayes' formula, we compute posterior probabilities.

$$P(s_k|q_i, w_j) = \frac{P(w_j|s_k)P(s_k|q_i)}{\sum_{k'=1}^{K} P(w_j|s_{k'})P(s_{k'}|q_i)} \quad (5)$$

M-step: In M-step, we maximize the expected complete data log-likelihood which is

$$\begin{aligned} Q(\Phi, \Theta) &= Q_1(\Phi, \Theta) - \lambda Q_2(\Theta) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(q_i, w_j)\sum_{k=1}^{K} P(s_k|q_i, w_j)\log[P(w_j|s_k)P(s_k|q_i)] \\ &\quad -\lambda\sum_{i,j=1}^{N}\sum_{k=1}^{K} C_{ij}(P(s_k|q_i) - P(s_k|q_j))^2 \end{aligned} \quad (6)$$

The M-step re-estimation equation for Φ is exactly the same as PLSI since the regularization term does not include $P(w_j|z_k)$.

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(q_i, w_j)P(s_k|q_i, w_j)}{\sum_{j'=1}^{M}\sum_{i=1}^{N} n(q_i, w_{j'})P(s_k|q_i, w_{j'})} \quad (7)$$

However, we do not have a closed form re-estimation equation for Θ. Based on the concepts of Generalized EM, we re-estimate Θ by increasing $Q(\Theta)$ rather than maximizing it. Specifically, let $\{\Phi_n, \Theta_n\}$ denote the parameter values of the previous iteration and $\{\Phi_{n+1}, \Theta_{n+1}\}$ denote the parameter values of the current iteration. We first find $\{\Phi_{n+1}, \Theta_{n+1}^{(1)}\}$ which maximizes $Q_1(\Phi, \Theta)$ instead of the whole $Q(\Phi, \Theta)$ using the re-estimation Eqn. (7) and (8) for Φ and Θ respectively.

$$P(z_k|q_i) = \frac{\sum_{j=1}^{M} n(q_i, w_j)P(s_k|q_i, w_j)}{\sum_{j=1}^{M} n(q_i, w_j)} \quad (8)$$

We then try to start from $\Theta_{n+1}^{(1)}$ and decrease $Q_2(\Theta)$, by using the Newton-Raphson method [28]. Therefore, we can obtain the closed form solution for updating Θ_{n+1}

$$P(s_k|q_i)_{n+1}^{(t+1)} = (1-\gamma)P(s_k|q_i)_{n+1}^{(t)} + \gamma\frac{\sum_{j=1}^{N} C_{ij}P(s_k|q_j)_{n+1}^{(t)}}{\sum_{j=1}^{N} C_{ij}} \quad (9)$$

Algorithm 1 Generalized EM for Regularized Topic Model

```
Input: $N$: # of queries, $M$: size of vocabulary, $K$: # of topics,
    $C$: co-click matrix, $\lambda$: regularization parameter, $\gamma$: Newton
    step parameter, $\delta$: Termination condition.
Output: $\Phi = \{P(w_j|s_k)\}$,$\Theta = \{P(s_k|q_i)\}$
Randomly initialize $\Phi_0$ and $\Theta_0$.
$n \leftarrow 0$
while (true) do
    E-step:
    Compute $P(s_k|q_i, w_j)$ using $\Phi_n$ and $\Theta_n$ as in Eqn. (5)
    M-step:
    Re-estimate $\Phi_{n+1}$ as in Eqn. (7)
    Re-estimate $\Theta_{n+1}$ as in Eqn. (8)
    $\Theta^{(1)}_{n+1} \leftarrow \Theta_{n+1}$
    Compute $\Theta^{(2)}_{n+1}$ from $\Theta^{(1)}_{n+1}$ as in Eqn. (9)
    while $(Q(\Phi_{n+1}, \Theta^{(2)}_{n+1}) \geq Q(\Phi_{n+1}, \Theta^{(1)}_{n+1}))$ do
        $\Theta^{(1)}_{n+1} \leftarrow \Theta^{(2)}_{n+1}$
        Compute $\Theta^{(2)}$ from $\Theta^{(1)}$ as in Eqn. (9)
    end while
    if $(Q(\Phi_{n+1}, \Theta^{(1)}_{n+1}) \geq Q(\Phi_n, \Theta_n))$ then
        $\Theta_{n+t} \leftarrow \Theta^{(1)}_{n+t}$
    else
        $\Phi_{n+1} \leftarrow \Phi_n$
        $\Theta_{n+1} \leftarrow \Theta_n$
    end if
    if $(Q(\Phi_{n+1}, \Theta_{n+1}) - Q(\Phi_n, \Theta_n)) \leq \delta)$ then
        break
    end if
    $n \leftarrow n + 1$
end while
return $\Phi_{n+1}, \Theta_{n+1}$
```

where $0 \leq \gamma \leq 1$ is the step parameter. It can be easily verified that $\sum_{k=1}^{K} P(s_k|q_i)^{(t+1)}_{n+1} = 1$ and $P(s_k|q_i)^{(t+1)}_{n+1} \geq 0$ hold in Eqn. (9) as long as $\sum_{k=1}^{K} P(s_k|q_i)^{(t)}_{n+1} = 1$ and $P(s_k|q_i)^{(t)}_{n+1} \geq 0$. Note here Φ_{n+1} will be fixed since $Q_2(\Theta)$ does not include Φ.

The iteration of Eqn. (9) is repeated until $Q(\Phi_{n+1}, \Theta^{(t+1)}_{n+1}) \leq Q(\Phi_{n+1}, \Theta^{(t)}_{n+1})$. We then test whether $Q(\Phi_{n+1}, \Theta^{(t+1)}_{n+1}) \geq Q(\Phi_n, \Theta_n)$. If it is not true, we reject the proposal of $\{\Phi_{n+1}, \Theta^{(t+1)}_{n+1}\}$ and return $\{\Phi_n, \Theta_n\}$ as the result of the M-step, and continue with the next E-step. The specific model fitting algorithm for the regularized topic model is summarized in Algorithm 1.

Finally, we obtain the probability $P(s_k|q_i)$ for each query q_i, which denotes the distribution of potential search intents of the query, and the probability $P(w_j|s_k)$ for each intent s_k, which denotes the distribution of words under each search intent. Here we further cut off the search intent with the proportion under a pre-defined threshold (e.g. 0.1 in our case) for each query to only preserve its major intents and avoid potential noises.

3.3 Intent-Aware Similarity Measure

Based on the learned intents of queries, here we show how to extract the query representation under each intent, and apply different metrics to measure query similarity with respect to different search intents. Both pair-wise measures (e.g. cosine similarity) and graph-based measures (e.g. spectral embedding) are adopted in our work as examples.

Pair-wise Measures: For pair-wise measures, the similarity of each query pair is independently measured by pair-wise metrics over the corresponding representations of queries. A typical representation for query q_i is the word vector from its search result snippets, where the l-th element of query q_i's vector is denoted as

$$\vec{q}_i[l] = n(q_i, w_l)$$

Such a word vector is a mixed-intent representation since words describing different intents of the query are merged together. In order to apply pair-wised measures to estimate query similarity with respect to search intents, we need to extract the word vector representation of the query under different search intents. This can be easily obtained with the topic model learned above.

As we can see, with our learned model, we can infer the probability $P(s_k|q_i, w_l)$ using the Eqn. (5), which denotes the expected search intent distribution for each word occurrence w_l given query q_i. In other words, we have the specific topic assignment for each word w_l from the top search result snippets of query q_i. Therefore, it is straightforward to represent query q_i under the k-th search intent using the word vector with the l-th element defined as

$$\vec{q}_{ik}[l] = n(q_i, w_l)P(s_k|q_i, w_l)$$

With the query representation under different search intents in hand, we can directly apply pair-wise measures to calculate the similarity between queries with respect to search intents. Here we take the traditional *cosine similarity* for example. Given query q_i and q_j, the similarity between the two queries under the k-th search intent can be calculated as follows

$$Sim_k(q_i, q_j) = \frac{\vec{q}_{ik} \cdot \vec{q}_{jk}}{\| \vec{q}_{ik} \| \| \vec{q}_{jk} \|} \quad (10)$$

Graph-based Measures: For Graph-based measures, the similarity between queries is defined on a query graph, or a query relation graph [13, 9]. A typical query graph is the query similarity graph, which is an undirected graph $G = (V, E, A)$, where V is the set of unique queries $\mathcal{Q} = \{q_1, \ldots, q_N\}$, $E \subseteq V \times V$ is the set of edges, and $A = [W_{ij}]_{i,j=1,\ldots,N}$ is the adjacency matrix with W_{ij} defined as the similarity between the i-th and j-th queries. Here we simply define W_{ij} as the Jaccard coefficient on clicked URLs between the i-th and j-th queries for demonstration. Obviously, such a query graph is a mixed-intent representation of queries, since queries from different search intents are connected together. To leverage graph-based measures to estimate query similarity with respect to search intent, the key problem is to extract the query graph representation under different search intents.

In our topic model, we obtain the probability $P(s_k|q_i)$ for each query q_i, which denotes the probability that query q_i conveys the search intent s_k. Here, we define the probability that an edge will generated between query q_i with search intent s_k and query q_j with search intent $s_{k'}$ is $P(s_k|q_i)P(s_{k'}|q_j)$. This probability obviously satisfies the normalization condition $\sum_{k,k'} P(s_k|q_i)P(s_{k'}|q_j) = 1$. In this way, we can obtain the query graph representation under the k-th search intent by re-define the edge weight between query q_i and q_j as

$$W^k_{ij} = W_{ij}P(s_k|q_i)P(s_k|q_j)$$

With the query similarity graph representation under different search intents in hand, we can then apply graph-based measures to calculate the similarity between queries with respect to search intents. Here we take the graph projection method, referred as spectral embedding [4], as an example.

The basic idea of spectral embedding is to project the original graph into a low-dimensional Euclidean space and then measure distances between graph nodes by considering the distances of the corresponding projected points. The spectral embedding has the property of preserving the distances in the projected space. The process of applying spectral embedding on the query graph representation under k-th search intent is described briefly below:

1. Given the adjacency matrix $A_k = [W^k_{ij}]_{i,j=1,\ldots,N}$ of the graph G under the k-th search intent, we compute the eigenvalues and eigenvectors for the generalized eigenvector problem:
$$L_k\mathbf{y} = \lambda D_k\mathbf{y} \tag{11}$$
where D_k is diagonal matrix whose entries are column sums of A_k. $L_k = D_k - A_k$ is the Laplacian matrix.

2. Let $\mathbf{y}_0, \ldots, \mathbf{y}_{l-1}$ be the solution of Eqn. (11), ordered according to their eigenvalues with $\mathbf{y}_0$ having the smallest eigenvalue. We then obtain, for the k-th search intent, the query q_i under the embedding into the lower dimensional space $\mathbb{R}^m$ is given by $\vec{q}_{ik}=(\mathbf{y}_1(i), \ldots, \mathbf{y}_m(i))$

On the projected space under each search intent, we can take cosine similarity as the metric to measure the similarity between queries. The similarity between query q_i and q_j under the k-th search intent is obtained by

$$Sim_k(q_i, q_j) = \frac{1 + cos(\vec{q}_{ik}, \vec{q}_{jk})}{2} \tag{12}$$

Note here we follow the way in [9] to rescale the cosine to ensure a measure in $[0, 1]$.

4. APPLICATION TO QUERY RECOMMENDATION

After describing our approach for measuring query similarity with respect to search intents, we turn our attention to the task of developing a simple application based on our approach. While there are many applications requiring a similarity measure between queries, one particularly useful application in the context of search is query recommendation. Query recommendation is to suggest a set of potentially related queries to search users to help them refine their original query. Here we use query recommendation as one example to show the potential utility of our approach and also demonstrate its effectiveness.

Traditionally, query recommendation provide search users a list of related queries. Since queries are often ambiguous in search intent, the recommendation list is thus a mixture of related queries from different search intents, or even worse, dominated by related queries from one popular search intent. As proposed in [19], a structured approach is effective in providing users with diverse query recommendations and thus enhance users' click behavior on recommendations. With our approach to intent-aware similarity, we can identify similar queries to the user's initial query with respect to different search intents, and thus a structured query recommendation can be easily built by grouping diverse recommendations based on the search intents.

More specifically, the structured query recommendation based on our approach can be divided into two stages, i.e. the offline learning stage and the online testing stage. In the offline learning stage, we applied the regularized topic model described in previous section to learn the potential search intents of queries and extract the query representations under each search intent. We can then index all the result representations for fast retrieval later. In the online testing stage, for a given user query q^*, we find its major search intents and corresponding representations under those intents. For each of its major search intent, we retrieve all the existing queries in the repository that potentially match q^* (i.e., a query q's vector representation matches q^*'s in at least one dimension), calculate the similarity between them using certain metrics, and sort the queries under that search intent. Finally, we provide users the structured recommendation results, where similar queries under the same search intent are grouped together and the groups are further ordered according to the proportion of the search intents of the initial query.

5. EXPERIMENTS

We have conducted experiments to verify the effectiveness our similarity measure approach. In this section, we first compare the performance of query similarity calculation of our approach with two baseline methods. We then compare the performance on identifying search intents of queries between our approach and traditional PLSI model. Finally, we apply our approach to query recommendation and evaluate its effectiveness by users' click behavior.

5.1 Experiment Setting

In our experiments, we obtained a query clickthrough data set by randomly sampling from a commercial search engine's search logs during a time period of one month. This sampled clickthrough data set contains about 15 million records. The queries were processed via the following normalization steps (i) trimming of each query, (ii) converting letters into lower case, and (iii) space sequence reduction to one space character. Queries and corresponding clickthrough data containing adult content were filtered out. For each query, we then collected its top N (i.e. $N = 10$ in our case) search results from the same search engine. A virtual document was constructed for each query by aggregating all the words from its top search result snippets with the stop words removed. Finally, we obtained $11,524$ unique queries, $87,415$ unique URLs, and $45,882$ unique words.

Two types of similarity measures (i.e., pair-wise measures and graph-based measures) were adopted as baselines in our evaluation. For pair-wise measures, we used cosine similarity based on Tf-Idf weighted word vector from top search results as a baseline measure, referred as **Cos-Word**. For graph-based measures, we used the spectral embedding over the query similarity graph based on clickthrough (described in section 3.3) as another baseline measure, referred as **Embed-Click**. Note that a similar approach has been used in [9] for measuring query similarity. In our experiments, we empirically set the embedding dimension to 10 since we found that increasing the number of dimensions does not determine a considerable gain in terms of quality.

For our approach, we identified potential search intents of queries with our regularized topic model, and adopted pair-wise measure (i.e. cosine similarity) and graph-based measure (i.e. spectral embedding) based on the intent-aware representations of queries for similarity measure. We denote our two approaches as **Cos-Intent** and **Embed-Intent**, respectively. In our experiments, we empirically set the value

Table 1: Example Queries Pairs with Similarity Scores Calculated by Different Methods

Method	Intent†	apple					
		apple store	apple company	apple ipod	apple fruit	apple tree	apple juice
Cos-Word	N/A	0.86	0.78	0.65	0.17	0.15	0.11
Cos-Intent	fruit	0	0	0	0.44	0.41	0.39
	company	0.92	0.83	0.77	0	0	0
Embed-Click	N/A	0.89	0.81	0.87	0.46	0.37	0.41
Embed-Intent	fruit	0	0	0	0.83	0.77	0.79
	company	1	0.96	0.99	0	0	0

Method	Intent†	taylor					
		taylor swift	taylor swift new songs	taylor ice cream	taylor soft serve machine	taylor acoustic	taylor guitars
Cos-Word	N/A	0.55	0.51	0.49	0.58	0.62	0.59
Cos-Intent	singer	0.76	0.68	0	0	0	0
	instrument	0	0	0	0	0.87	0.85
	company	0	0	0.52	0.61	0	0
Embed-Click	N/A	0.48	0.47	0.47	0.46	0.44	0.51
Embed-Intent	singer	1	1	0	0	0	0
	instrument	0	0	0	0	0.60	0.63
	company	0	0	0.87	0.72	0	0

†the search intents are manually labeled for illustration

Table 2: Examples of Similar and Dissimilar Query Pairs

Type	Query Pair	Traditional Method		Intent-Aware Method†	
		Cos-Word	**Embed-Click**	**Cos-Intent**	**Embed-Intent**
Similar Pairs	(apple, apple store)	0.86	0.89	0|0.92	0|1
	(apple, apple fruit)	0.17	0.46	0.44|0	0.83|0
Dissimilar Pairs	(apple store, apple fruit)	0.09	0.37	0|0	0|0
	(apple ipod, apple tree)	0.08	0.34	0|0	0|0
Similar Pairs	(taylor, taylor swift)	0.55	0.48	0.76|0|0	1|0|0
	(taylor, taylor soft serve machine)	0.58	0.46	0|0|0.61	0|0|0.72
Dissimilar Pairs	(taylor swift, taylor soft serve machine)	0.28	0.36	0|0|0	0|0|0
	(taylor ice cream, taylor acoustic)	0.24	0.38	0|0|0	0|0|0

†similarity scores under different intents are separated by vertical bars for clarity

of Newton step parameter γ to 0.1, the value of the regularization parameter λ to 10, and the number of potential search intent to 500. We found that when the number of potential search intents is reasonable large (i.e. ≥ 500), the performance would be relatively stable.

5.2 Evaluation of Similarity Measurement

5.2.1 Qualitative Evaluation

To get a cursory evaluation for how well our approach performs, we show some example query pairs with the similarity scores calculated by different methods in Table 1. In this table, we take the two multi-intent queries "apple" and "taylor" as the central queries and show the similarity scores between the central query and their similar queries from different search intents.

From Table 1 we can see, both the Cos-Word and Embed-Click measures, which do not take search intent into account, assign a single similarity score for each pair of queries. Take the Cos-Word measure for example, the similarity score between "taylor" and "taylor swift" is 0.55, while the similarity score between "taylor" and "taylor soft serve machine" is 0.58. Since they are calculated under the same measure, it seems that we can derive a strange conclusion that "taylor soft serve machine" is more similar to "taylor" than "taylor swift". In fact, "taylor soft serve machine" and "taylor swift" are from different search intents of the query "taylor", one about a company and the other about a famous singer. The similarity is actually not comparable across different search intents in such case. The similar problem can also be found in the query "apple" which is also shown in Table 1 and many other multi-intent queries.

On the other hand, by identifying potential search intents of queries, both the Cos-Intent and Embed-Intent measures show a more proper similarity measure for the same set of queries. For example, we can see that queries like "taylor swift" and "taylor swift new songs" both obtain high similarity scores under the search intent about the singer, while low similarity scores (i.e. zero) under other search intents. Note here the names of the search intents are manually labeled for illustration. Clearly, under the search intent about the singer, "taylor swift" is much more similar to "taylor" than "taylor soft serve machine", while under the search intent about the company, it is the opposite. Similar results can also be found for the query "apple".

To have a deeper understanding of the problems in similarity measures without the awareness of search intents, we further extract some pairs of similar and dissimilar queries for comparison as shown in Table 2. From the results we can see, the pair-wise measure Cos-Word gives the query pair ("apple", "apple store") a much higher similarity score (i.e. 0.86) than the query pair ("apple", "apple tree"). The major reason is that, for the ambiguous query "apple", the representation (i.e. words from top search result snippets) used by Cos-Word for similarity measure is actually a mixture of its multiple search intents but dominated by the intent about the apple company. Therefore, without the awareness of

Table 3: Examples of Manually Built Test Set

Query	Major Intents
24 hours	1. tv show 24, 24 on fox, 24 the series
	2. 24 fitness, 24hr fitness, 24 hour gym
sigma	1. sigma aldrich, sigma chemicals, sigma biology
	2. greek alphabet sigma, sigma symbol, sigma maths
	3. sigma camera, sigma photo, sigma lenses
svm	1. svm cards, svm gift card, svm gas cards
	2. svm kernel, svm tutorial, support vector machine

search intents, the similarity measure Cos-Word is easily biased by dominant search intents and ignore unpopular ones. On the contrary, the intent-aware pair-wise measure Cos-Intent calculates the similarity for each pair under different search intents using intent-aware representations. Such intent-aware representations convey precise information of each query under each intent. Therefore, our approach can produce more reasonable similarity scores without the bias on only popular intents. For example, "apple tree" receives a much higher similarity score (i.e. 0.44) to "apple" than "apple store" (i.e. 0) under the fruit intent.

Moreover, we observe that the dissimilar query pairs also obtain reasonably large similarity scores under the traditional graph-based measure Embed-Click. For example, the query pair ("apple store", "apple fruit") receives a similarity score of 0.37 and the query pair ("taylor swift", "taylor soft serve machine") receives a similarity score of 0.36 under the Embed-Click measure. As we know, the Embed-Click measure leverages the spectral embedding technique, which in some sense propagates the similarity on top of the query graph during embedding. Therefore, since query "apple store" and "apple fruit" are both close to "apple" in the original graph, without the awareness of search intents, the similarity will be propagated cross the boundary of different search intents and thus query "apple store" and "apple fruit" will become similar after embedding. On the contrary, the intent-aware graph-based measure Embed-Intent extracts the intent-aware graph representation under each search intent, where queries not conveying that intent will not be connected (or connected with a very large distance). In this way, we can see that the similarity is properly propagated among queries, and dissimilar queries like "apple store" and "apple fruit" will not be wrongly connected any more.

5.2.2 Quantitative Evaluation

We further conducted quantitative comparison between our approaches and baseline approaches. For evaluation, we first constructed a test set based on our query logs. We collected a set of single-term queries that are likely to have more than one search intent. From these candidate queries, we asked three human judges to figure out the queries which have multiple search intents. Specifically, for each candidate query, queries that share co-clicks with it were collected as its potential similar queries. The human judges then exam each candidate query whether there are at least two distinct clusters in its potential similar queries. Finally, we collected 200 seed queries that have multiple search intents which are agreed by at least two human judges. For each seed query, we then created a test set of similar queries by extracting 3 representative similar queries under each major intent. In this way, we obtained a test set with total $1,581$ queries labeled. Such a test set represents the ground truth for our evaluation. Table 3 show examples of the test set.

Table 4: $\mathcal{H}_{\hat{S}}(Sim)$ for Different Similarity Measures

Method	$\mathcal{H}_{\hat{S}}(Sim)$	Significant differences[†]
Cos-Word	0.47±0.06	>Embed-Click***
Cos-Intent	0.08±0.03	>Cos-Word*** >Embed-Click***
Embed-Click	0.54±0.02	
Embed-Intent	0.09±0.03	>Cos-Word*** >Embed-Click***

[†]the significant levels are denoted as 0.1* 0.05 ** 0.01 ***

We then apply different similarity measures over the queries in the test set, and evaluate their agreement with human labeled search intents. Specifically, given a seed query, let $\mathcal{Q}$ be the set of its similar queries which can be categorized into K search intents $S = \{S_1, \ldots, S_K\}$. Obviously, we have $\mathcal{Q} = \cup_i S_i$. For a similarity measure $Sim(q, q')$, we introduce two scores similarly as [9]:

Expected Intra-intent Similarity:

$$IntraSim(S) = \frac{1}{K}\sum_{k=1}^{K}\left[\sum_{q_i,q_j \in S_k, i\neq j}\frac{2Sim(q_i,q_j)}{|S_k||S_k-1|}\right]$$

Expected Inter-intent Similarity:

$$InterSim(S)=\frac{1}{K(K-1)}\sum_{S_k,S_{k'}\in S, k\neq k'}\left[\sum_{q_i\in S_k}\sum_{q_j\in S_{k'}}\frac{Sim(q_i,q_j)}{|S_k||S_{k'}|}\right]$$

The intuition is that a similarity measure agrees with human labeled search intent if the expected inter-intent similarity score is small compared to the expected intra-intent similarity score. Therefore, we calculate the following expected ratio to evaluate the quality of a similarity measure

$$\mathcal{H}_{\hat{S}}(Sim) = E\left[\frac{InterSim(S)}{IntraSim(S)}\right]_{S\in\hat{S}} \tag{13}$$

Given two different similarity measures, the best one is the one that minimize the measure $\mathcal{H}_{\hat{S}}(Sim)$.

The results are reported in Table 4. From the results we can see, for the traditional measures which do not take search intent into account, the pair-wise measure Cos-Word performs better than the graph-based measure Embed-Click. The major reason is that the Embed-Click measure would generate a higher inter-intent similarity, since the similarity propagation often wrongly connects the dissimilar queries and assign them a large similarity score. By considering search intent of queries, both Cos-Intent and Embed-Intent can significantly outperform the baseline measures (p-value$<$ 0.01). With some further analysis, we find that our approach generates a much smaller inter-intent similarity as well as a larger intra-intent similarity as compared with the traditional measures. It shows that, by measuring query similarity with respect to search intent, we can generate more precise similarity scores for both similar and dissimilar queries.

5.3 Evaluation of Topic Models

In our approach, we employ the regularized topic model to learn the potential search intents of queries by using both words from search results and regularization from query co-clicks. A natural question is how about learning the search intents only based on words from search results, e.g., to just apply the traditional PLSI model over the virtual document of queries. In other words, does the regularization from query co-clicks really helps for the learning problem?

To answer this question, we compared our regularized topic model with traditional PLSI model on the performance of identifying potential search intents. We applied PLSI over

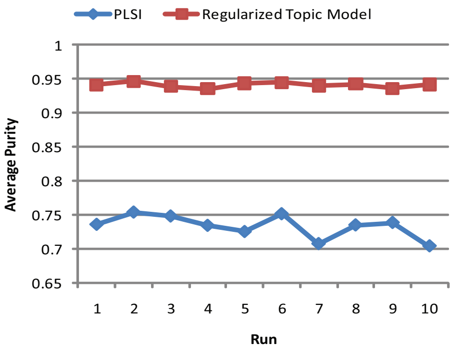

Figure 1: Comparison between PLSI and Regularized Topic Model on Average Purity Score

the search result snippets of queries to learn the search intents. To make the comparison fair, we use the same intent number for PLSI and our regularized topic model. For evaluation, we use the same test set described above. With the topic model learned in hand, we assign each query in the test set to an intent group according to its largest intent. The intuition is that the topic model learns better if the predicted intent groups are more like the human labeled results. For each seed query, let $S = \{s_1, \ldots, s_J\}$ denote the intent groups predicted by the topic model, and $\hat{S} = \{\hat{s}_1, \ldots, \hat{s}_K\}$ denote its manually labeled intent groups. We then borrow the *purity* metric [35] from traditional clustering problem to evaluate the quality of predicted intent groups, which is defined as

$$Purity(S, \hat{S}) = \frac{1}{N} \sum_j \max_K |s_j \cap \hat{s}_k| \tag{14}$$

where N is the total number of similar queries in the test set for a seed query. A higher purity score means a better prediction on intent groups.

The average purity scores for both PLSI and our regularized topic model over different runs are depicted in Fig. 1. The results show that our regularized topic model can consistently outperform PLSI. The average purity score obtained by our regularized topic model is about 0.94, while the average purity score obtained by PLSI is about 0.73. The results indicate that the query co-clicks can largely help resolve the ambiguity in search intents of queries. By leveraging the regularization from query co-clicks, we can learn the potential search intents of queries significantly better than using the search results alone. Moreover, we can also observe that the performance of our regularized topic model is more stable than PLSI. It shows that the regularization from query co-clicks can help our model produce high accuracy in prediction constantly.

5.4 Evaluation on Query Recommendation

In this section, we applied our approach to query recommendation. As described in Section 4, we can easily generate a structured query recommendation for a given query to suggesting diverse related queries from different search intents. We also implemented a traditional *list-based* recommendation system for comparison, where similar queries are ranked according to their similarity scores under a traditional measure and the top ones are returned in a list. For demonstration, here we employ the Cos-Word and Cos-Intent measures for the list approach and structured approach, respectively. We used the previous selected 200 multi-intent queries plus 200 randomly sampled queries for evaluation. For each query, top 10 recommendations are used for performance comparison.

Table 5: Comparisons between List Approach and Our Approach on Click Performance

	List Approach	Our Approach
Ave. CRN	4.10	4.63 (+12.9%)
Ave. CRS	0.43	0.47 (+9.3%)
Ave. TRS	0.15	0.17 (+13.3%)

We follow the way proposed in [19] to compare the performances of different recommendation methods by users' click behavior. The manual labeling process is almost the same, where human judges are required to label for each recommendation how likely he would like to click it with a 6-point scale (0, 0.2, 0.4, 0.6, 0.8, and 1) as the willingness measure. We asked 5 judges with or without computer science background to label the recommendations.

We also adopted the *Clicked Recommendation Number (CRN)*, *Clicked Recommendation Score (CRS)*, and *Total Recommendation Score (TRS)* as evaluation measures [19]. Given a query q, let $R = \{r_1, \ldots, r_k\}$ denote the k recommendations generated by a certain approach, and $L = \{l_1, \ldots, l_k\}$ denote the corresponding label scores on these recommendations. The three measures for a query q are then defined as follows

$$\begin{aligned} CRN_q &= |\{r_i | l_i > 0, i \in [1, k]\}|, \\ CRS_q &= \frac{\sum_{i=1}^{k} l_i}{CRN_q}, \\ TRS_q &= \frac{\sum_{i=1}^{k} l_i}{k}, \end{aligned}$$

where $| * |$ denotes the size of a set.

Table 5 shows the evaluation results of the two approaches. The numbers in the parentheses are the relative improvements of our approach compared with list approach. The results show that by recommending structured queries based on our intent-aware approach, we can largely improve both the click number and click willingness on recommendations. When compared with list approach, the relative improvements obtained by our approach are about 12.9%, 9.3% and 13.3% in terms of average CRN, average CRS and average TRS, respectively. This simple experiment demonstrates the utility and effectiveness of our intent-aware approach in real applications.

6. CONCLUSIONS

In this paper, we propose to measure query similarity with the awareness of potential search intents. A regularized topic model is employed to effectively identify search intents of queries using words from search result snippets and regularization from query co-clicks. We then extract the query representation under different intents, and apply two types of similarity measures to estimate query similarity with respect to search intent. Experimental results verify the effectiveness of our approach to intent-aware query similarity. We also demonstrate the utility of intent-aware similarity in the application of query recommendation, which can suggest diverse queries in a structured way to search users.

For the future work, we will consider using more context information of queries, e.g. search sessions, for identifying search intents better. It is also interesting to apply our intent-aware query similarity in other real applications in the search context.

7. ACKNOWLEDGMENTS

This research work was funded by the National High-tech R&D Program of China under grant No. 2010AA012500, and the National Natural Science Foundation of China under Grant No. 61003166 and Grant No. 60933005.

8. REFERENCES

[1] I. Antonellis, H. G. Molina, and C. C. Chang. Simrank++: query rewriting through link analysis of the click graph. *Proc. VLDB Endow.*, 1:408–421, August 2008.

[2] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Query recommendation using query logs in search engines. In W. Lindner, M. Mesiti, C. TÍźrker, Y. Tzitzikas, and A. Vakali, editors, *EDBT Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pages 588–596. Springer, 2004.

[3] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 407–416, New York, NY, USA, 2000. ACM.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14:585–591, 2001.

[5] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM.

[8] 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.

[9] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 515–522, New York, NY, USA, 2010. ACM.

[10] D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 911–920, New York, NY, USA, 2008. ACM.

[11] D. Cai, X. Wang, and X. He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 105–112, New York, NY, USA, 2009. ACM.

[12] B. Cao, J.-T. Sun, E. W. Xiang, D. H. Hu, Q. Yang, and Z. Chen. Pqc: personalized query classification. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1217–1226, New York, NY, USA, 2009. ACM.

[13] N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM.

[14] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Query expansion by mining user logs. *IEEE Trans. on Knowl. and Data Eng.*, 15:829–839, July 2003.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B(39):1–38, 1977.

[16] H. Deng, I. King, and M. R. Lyu. Entropy-biased models for query representation on the click graph. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 339–346, New York, NY, USA, 2009. ACM.

[17] 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.

[18] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, 2005.

[19] 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 international conference on Information and knowledge management*, CIKM '10, pages 619–628, New York, NY, USA, 2010. ACM.

[20] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM.

[21] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[22] 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.

[23] Y. Li, Z. Zheng, and H. K. Dai. Kdd cup-2005 report: facing a great challenge. *SIGKDD Explor. Newsl.*, 7:91–99, December 2005.

[24] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 709–718, New York, NY, USA, 2008. ACM.

[25] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 101–110, New York, NY, USA, 2008. ACM.

[26] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 469–478, New York, NY, USA, 2008. ACM.

[27] R. M. Neal and G. E. Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants*, pages 355–368. MIT Press, Cambridge, MA, USA, 1999.

[28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

[29] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 403–410, New York, NY, USA, 2008. ACM.

[30] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 377–386, New York, NY, USA, 2006. ACM.

[31] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[32] E. Terra and C. L. Clarke. Scoring missing terms in information retrieval tasks. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 50–58, New York, NY, USA, 2004. ACM.

[33] J.-R. Wen, J.-Y. Nie, and H. Zhang. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20(1):59–81, 2002.

[34] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 1039–1040, New York, NY, USA, 2006. ACM.

[35] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, 2002.