

Query intent mining with multiple dimensions of web search data

Di Jiang · Kenneth Wai-Ting Leung · Wilfred Ng

Received: 29 September 2014 / Revised: 15 December 2014 /
Accepted: 23 February 2015 / Published online: 19 March 2015
© Springer Science+Business Media New York 2015

Abstract Understanding the users' latent intents behind the search queries is critical for search engines. Hence, there has been an increasing attention on studying how to effectively mine the intents of search queries by analyzing search engine query log. However, we observe that the information richness of query log is not fully utilized so far and the information underuse heavily limits the performance of the existing methods. In this paper, we tackle the problem of query intent mining by taking full advantage of the information richness of query log from a multi-dimensional perspective. Specifically, we capture the latent relations between search queries via three different dimensions: the URL dimension, the session dimension and the term dimension. We first propose the *Result-Oriented Framework* (ROF), which is easy to implement and significantly improves both the precision and the recall of query intent mining. We further propose the *Topic-Oriented Framework* (TOF), in order to significantly reduce the online time and memory consumptions for query intent mining. TOF employs the *Query Log Topic Model* (QLTM) that derives the latent topics from query log to integrate the information of the three dimensions in a principled way. The latent topics that are considered as low-dimensional descriptions of the query relations and serve as the basis of efficient online query intent mining. We conduct extensive experiments on a major commercial search engine query log. Experimental results show that the two frameworks significantly outperform the state-of-the-art methods with respect to a variety of metrics.

Keywords Search engine · Query log · Topic model

D. Jiang (✉)
Baidu, Inc., Beijing, People's Republic of China
e-mail: jiangdi@baidu.com

K. Wai-Ting Leung · W. Ng
Hong Kong University of Science and Technology, Hong Kong, People's Republic of China

K. Wai-Ting Leung
e-mail: kwtleung@cse.ust.hk

W. Ng
e-mail: wilfred@cse.ust.hk

1 Introduction

In order to develop effective web search engines, it is fundamental to understand the users' latent intents behind the short search queries. Therefore, search query intent mining, which essentially maps the submitted queries into different intent categories, is gaining momentum in data mining research [1, 11, 22, 31, 32]. Based on the search engine query log, many supervised learning methods have been proposed for query intent mining [1, 31, 32]. However, as query log records very limited information for each query, the supervised methods usually need to augment the training features through external knowledge bases, which are typically costly to obtain. To avoid the undesirable feature augmentation, the recent trend of query intent mining is to expand the training data by semi-supervised learning paradigm [11, 22]. It is reported that, based on the feature of query terms alone, the semi-supervised learning approach works better than those with augmented features [22]. The key idea of the semi-supervised methods for query intent mining is to build a click graph, which is essentially a bipartite built upon queries and clicked URLs, and then expand the training data by mining the structure of the click graph. However, as a large proportion of search queries do not raise any clickthrough and two random queries rarely share the same clicked URLs [3], the click graph can only cover a small portion of the rich information in query log. Furthermore, the clicked URLs are inherently noisy and may also be biased from some users with malicious intents [12]. Thus, the robustness of methods based on click graph is rather unsatisfactory. Finally, the click graph is usually huge and practical issues such as reducing the time and memory consumption for semi-supervised intent mining are not satisfactorily addressed so far. The aforementioned problems which plague query intent mining give rise to the following research questions:

- Besides the click graph, how to identify extra information sources in the query log to further enrich the information coverage and enhance the robustness of the downstream query intent mining?
- Given multiple information sources that are derived from query log, how to effectively utilize them to improve the performance of query intent mining?
- As query intent mining is essentially conducted online in practice, how to significantly reduce its online time and memory consumptions?

To address the above three questions, we propose to model the relations between search queries in a multi-dimensional fashion, which provides comprehensive information coverage and robust information representation for the downstream query intent mining. Specifically, we capture the relations between queries from the following three dimensions: the URL dimension (whether they share the same URL clickthrough), the session dimension (whether they appear in the same search session) and the term dimension (whether they share the same query terms). By collectively utilizing the three dimensions, the multi-dimensional perspective captures much more information than the click graph. Based on the query relations captured by the three dimensions, we propose the *Result-Oriented Framework* (ROF), which is easy to be implemented in the multi-dimensional scenario and significantly improves both the precision and recall of query intent mining. In ROF, we first conduct query intent mining on each individual dimension and then the provisional results of different dimensions are combined according to strategies such as *Maximum Confidence* or *Majority Vote* to generate the final result. In order to significantly reduce the online computational cost of query intent mining, we further propose the *Topic-Oriented Frame-*

work (TOF), which utilizes the latent topics derived by *Query Log Topic Model* (QLTM) to represent the relations between search queries. TOF improves the precision, recall and scalability of query intent mining in term of both time and memory consumptions. Based on a large-scale query log from a commercial search engine, both ROF and TOF outperform several strong baselines with respect to a variety of metrics in the experiments.

The contributions of this paper are summarized as follows:

- We propose a multi-dimensional perspective to capture the latent query relations in search engine query log. The URL, session and term dimensions are proposed to collectively quantify the information richness in query log.
- Two new frameworks, ROF and TOF, are proposed for query intent mining. Both of the two frameworks significantly improve the quality of query intent mining results. The clear advantage of ROF lies in its ease of implementation while TOF is more superior in reducing the online computational cost of query intent mining.
- Extensive experiments are conducted to analyze the sensitivity of parameter settings and compare the proposed frameworks with the state-of-the-art methods. The results show that our frameworks significantly outperform the state-of-the-art methods with respect to various metrics.

The rest of the paper is organized as follows. We review the related work in Section 2. In Section 3, we detail the three dimensions of search engine query log. In Section 4, we formally define the problem of multi-dimensional query intent mining. In Sections 5 and 6, we detail the *Result-Oriented Framework* (ROF) and the *Topic-Oriented Framework* (TOF). We provide complexity analysis in Section 7. We present the experimental evaluations in Section 8 and conclude the paper in Section 9.

2 Related work

In recent years, search query intent mining has become a hot research issue and many methods have been proposed for mapping search queries into different intent categories. As a pioneering work, Broder [6] proposed the search intent taxonomy that is composed of three intents, namely, informational, navigational and transactional. As a further enrichment, [8] studied a wide range of facets that may be useful for user's intents identification. [32] used search engine results as features, including pages, snippets and titles, and built classifiers based on a document taxonomy. [7] transformed the problem of query classification to document classification which was solved directly in the target taxonomy. Another way to enhance feature representation is the use of word cluster features [2, 25]. In such approach, semantically similar words can be grouped into clusters, either by domain knowledge or by statistical methods, and be used as features to improve the generalization performance of a classifier. Similarly, the query classification methods in [1, 31] are also based on supervised learning and external knowledge bases are utilized to augment the training features. Radlinski [28] presented an approach for identifying the popular meanings of queries using user click behavior. Hu et al. [17] proposed a clustering algorithm to automatically mine the subtopics of queries. Dang et al. [13] clustered reformulated queries generated from publicly available resources. Sadikov et al. [30] addressed the problem of clustering the refinements of a user search query. As an orthogonal approach to tackle query intent mining, Li et al. [22] made the first attempt to increase the amount of training data for query intent min-

ing. A semi-supervised learning framework is proposed to evaluate query relevancy by the query-URL bipartite. Using the expanded training data and the unbiased training features such as words and phrases in queries themselves, their method yielded remarkable improvement in terms of both precision and recall. More recently, Celikyilmaz et al. [11] proposed a graph summarization algorithm for categorizing a given speech utterance into one of many semantic intent classes. Wang et al. [34] mines sequences of actions called search scripts from search query logs. Lee et al. [21] study whether and how we can automate the goal-identification process of web search. Yan et al. [39] proposed a keyword-based semantic entity search mechanism in dataspace. Qian et al. [27] proposed a method for mining new bursty intents from search query logs.

The present work is also related to query log analysis and probabilistic topic modeling. Boldi et al. [5] introduced the notion of query-flow graphs to represent the knowledge about latent querying behavior. Poblete et al. [26] introduced a graph representation of the Web, which includes both structural and usage information. Deng and Lyu [14] proposed the Co-HITS algorithm to incorporate the query-URL bipartite graph with content information and establish its relation between the HITS and personalized PageRank algorithms. Sadikov et al. [30] approximated user search behavior by a Markov graph and partitioned the graph into a predefined number of query clusters. Craswell et al. [12] applied Markov random walk model to a large click log, producing a probabilistic ranking of documents for a given query. Recently, topic modeling is gaining momentum in textual data analysis. Blei et al. [4] proposed Latent Dirichlet Allocation (LDA) which derives the latent topics to capture the semantic relation between words in articles. Some extensions of LDA achieved good performances on named entity recognition [24], article analysis [20] and query log analysis [10, 19].

We notice that there are some related work about query clustering [38] and query suggestion [9]. Although they are conducted based on query log as well, they solve problems that are different from ours. The reason that our work is significantly different from query clustering is as follows. Our methods are primarily semi-supervised, meaning that some labeled data and predefined categories of interest are required. In contrast, query clustering is essentially unsupervised and the resultant clusters can be significantly deviated from the categories of interest. Hence, our methods and query clustering are applied in very different applications scenarios. The differences between our methods and query suggestion are essentially twofold: (1) Query suggestion aims to find some “relevant” queries which are similar to the input query. Hence, the primary goal of query suggestion is not classifying queries, which, however, is the focus of our work. (2) In commercial search engines such as Baidu and Yahoo, query classification is typically an upstream application of query suggestion, i.e., query classification are typically conducted preceding query suggestion. Therefore, our methods can be applied to enhance the existing query suggestion methods.

The differences between our work and the previous ones are essentially twofold. First, rather than relying on external knowledge bases, we focus on taking full advantage of the information available in the query log itself. We propose to collectively utilize three dimensions to comprehensively capture the information in query log, which is under-used in previous work. This multi-dimensional perspective fits well in the task of query intent mining. Second, our work investigates the effective ways of integrating the multi-dimensional information for query intent mining. To the best of our knowledge, no previous work has been done to conduct query intent mining by using such comprehensive information sources collectively and systematically.

3 Query log dimensions

In this section, we present the details of the three dimensions that we utilize for query intent mining. Some examples of query log entries are presented in Table 1. Each entry contains the query identifier, the user identifier, the search query, the timestamp at which the query is submitted, the rank of clicked snippet (if any) and the clicked URL (if any). Based on this information, we model the relations between search queries by the following three dimensions.

3.1 URL dimension

The URL dimension aims at associating search queries via *URL co-clicking*. Search queries that result in the clickthrough on the same URL are associated with this dimension. For example, in Table 1, q_1 and q_5 can be related via the URL dimension since the two queries both result in clicking on the URL <http://www.innsocal.com/>. Similar to previous work [22], clickthrough on the same URL are considered as a piece of explicit evidence for the commonality of query intent. As the clicked URLs are recorded for each query, this relation can be straightforwardly obtained from the query log.

3.2 Session dimension

A session is defined as a series of queries that are sequentially submitted by a user to satisfy the same information need. For example, in Table 1, q_2 and q_3 are submitted by the same user to search for information about a movie named “chicken run”, thus, the two queries can be considered to be from the same session. The intuition of this dimension is that the semantically coherent session consisting of consecutive queries serves as a bridge for query relevance. This dimension is introduced to capture the *session co-existing* relation between queries. The query reformulation taxonomy proposed in [18] consists of a series of rules that evaluate the lexical similarity between queries and demonstrates high precision in detecting semantically relevant ones. Thus, we utilize this taxonomy to discover the sessions from the query log.

3.3 Term dimension

After tokenization, each search query can be segmented into one or more terms. The term dimension is proposed to capture the *term sharing* relation between queries. Search queries that share at least one term are related by this relation. For instance, in Table 1, q_1 and q_4

Table 1 Examples of search engine query log entries

QueryID	UserID	Query	Time	Rank	URL
q_1	25014	inn california	2012-03-13 11:37:28	1	http://www.innsocal.com/
q_2	45419	chicken run	2012-05-31 21:30:33		
q_3	45419	chicken run movie	2012-05-31 21:31:25	2	http://www.imdb.com/...
q_4	51162	residence inn san diego	2012-03-08 23:10:31	3	http://www.residenceinnsd.com/...
q_5	64732	hotel santa barbara	2012-06-11 22:11:35	3	http://www.innsocal.com/

share the same term “inn” and they can be related via the term dimension. In order to reduce the spurious relations, we use a stopwords table [23] to filter out the noninformative terms.

4 Problem definition

Based on the query relations captured by the three dimensions discussed in Section 3, we aim to infer the intents of a set of unlabeled queries by using a small set of seed queries, i.e., a set of queries with manually labeled intents. We denote the query-intent matrix as F , in which the entry f_{ij} is a real number indicating the probability that q_i belongs to the intent c_j . Given F , the posterior probability that q_i belongs to c_j is computed by the following formula:

$$P(c_j|q_i) = \frac{f_{ij}}{\sum_k f_{ik}}. \quad (1)$$

The instantiation of F is denoted as F^0 . If query q_i is manually labeled as c_j , then the entry f_{ij} in F^0 is 1, other entries in the i th row of F^0 are zeros. The entries which refer to the unlabeled queries are all zeros in F^0 . Based on the query relations captured by the three dimensions and F^0 , the query intent mining method estimates F , which determines the final intents of the queries. The notation used in this paper is presented in Table 2.

5 Result-oriented framework

In the face of the multi-dimensional information, *Result-Oriented Framework* (ROF) adopts a prediction-combination paradigm: the intermediate results are first obtained from each individual dimension and then they are combined to generate the final result. We first establish the three matrices B^U , B^S and B^T for the bipartite obtained via the URL, the session and the term dimension, respectively. The value of each entry in the three matrices is the frequency of observations associating the nodes from the rows and the columns:

- B^U : each row represents a unique query and each column represents a unique URL.
- B^S : each row represents a unique query and each column represents a session.
- B^T : each row represents a unique query and each column represents a unique term.

We then obtain the query affinity matrices W^U , W^S and W^T by multiplying B^U , B^S and B^T and their corresponding transposes. For each query affinity matrix, we apply Algorithm 1 to find the optimal intent estimation represented by the matrix F^* .

Algorithm 1 Intermediate Search Query Intent Mining

Input: initial query-class matrix F^0 ; query affinity matrix W

Output: query-class matrix F^*

- 1: Compute diagonal degree matrix D by $D_{ii} \leftarrow \sum_j W_{ij}$;
 - 2: Compute normalized graph Laplacian $\mathcal{L} \leftarrow D^{-1/2} W D^{-1/2}$;
 - 3: **repeat**
 - 4: Compute solution $F^i \leftarrow \frac{\mu}{1+\mu} \mathcal{L} F^{i-1} + \frac{1}{1+\mu} F^0$;
 - 5: **until** F^i converges to F^*
-

Table 2 Notation

Notation	Description
U	the URL dimension
S	the session dimension
T	the term dimension
B^U	the query-URL matrix
B^S	the query-session matrix
B^T	the query-term matrix
W^U	the query affinity matrix of URL dimension
W^S	the query affinity matrix of session dimension
W^T	the query affinity matrix of term dimension
W^{topic}	the query affinity matrix based on latent topics
F	the query-intent matrix
F^0	the instantiation of F
F_l	the manually labeled part of F^0
S^{topic}	the session-topic matrix
Z	the query-topic matrix
A	the topic-intent matrix
T_h	the threshold for query intent mining
C	the number of query intents
N	the number of unique queries
$U^\#$	the number of unique URLs
$\hat{U}^\#$	the number of URLs
$S^\#$	the number of sessions
$T^\#$	the number of unique terms
$\hat{T}^\#$	the number of terms

In Algorithm 1, during each iteration of line 4, each query node receives the information from its neighbors and also retains its initial information. The parameter μ quantifies the effect of the original value F^0 . Finally, F^* quantify the amount of information each query node has received from the classes during the iteration process. The sequence of F^i asymptotically converges to the following formula:

$$F^* = ((1 + \mu)I - \mu\mathcal{L})^{-1}F^0. \quad (2)$$

For the three query affinity matrices W^U , W^S and W^T , we apply Algorithm 1 to each of them and obtain three intermediate query-intent matrices, F^U , F^S and F^T . If we utilize any one of them as the final result, this framework reduces to using the information of one dimension to determine the query intents. As our goal is to investigate whether using results from different dimensions together can improve the performance of query intent mining, we propose two different strategies to effectively combine the information stored in F^U , F^S and F^T .

The first strategy is *Maximum Confidence*. This strategy regards f_{ij} as the confidence that the i th query embodies the j th intent. In the result combining procedure, the i th query is assigned with the intent that has the highest probability across the chosen dimensional combination. As for this strategy, there exist four different combinations, namely, $DC_1 =$

$\{U, S\}$, $DC_2 = \{U, T\}$, $DC_3 = \{S, T\}$, $DC_4 = \{U, S, T\}$. Without loss of generality, for any dimension combination DC , the query q_i is labeled as c_j if

$$\max_{k \in DC} f_{ij}^k = \max_{c=1}^C \max_{k \in DC} f_{ic}^k \tag{3}$$

and

$$\max_{k \in DC} f_{ij}^k > T_h. \tag{4}$$

Note that (3) labels a query as the intent that shows the highest probability across all the dimensions in DC . We adopt a conservative approach by using (4) and further constrain that the highest probability should also be higher than a probability threshold T_h , otherwise, the query remains unlabeled.

The second strategy is *Majority Vote*, which only has one dimensional combination $DC = \{U, S, T\}$. The query q_i is labeled as c_j if

$$\sum_{k \in DC} \Delta_{ij}^k = \max_{c=1}^C \sum_{k \in DC} \Delta_{ic}^k, \tag{5}$$

where

$$\Delta_{ij}^k = \begin{cases} 1, & \text{if } f_{ij}^k = \max_{c=1}^C f_{ic}^k \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$\tag{7}$$

In this strategy, we first associate q_i with an intermediate class label that shows the highest probability based on each dimension. In total, q_i receives three intermediate labels from the URL dimension, the session dimension and the term dimension, respectively. We then regard each intermediate result as a vote for a certain intent and then assign q_i the intent that receives the most votes. As there are three votes in total, we label a query as the intent that receives at least two votes. If no intent receives at least two votes, the query remains unlabeled.

Although the aforementioned two strategies seems simple and intuitive, to the best of our knowledge, they are the first to be proposed for the task of query intent mining. As we will show later in Section 8, a little surprisingly, these two strategies can significantly improve the accuracy of query intent mining.

6 Topic-oriented framework

ROF utilizes the “raw” information from the three dimensions and represent query relations in high-dimensional matrices such as W^U , W^S and W^T . In order to reduce the computational cost that is induced by the three high-dimensional matrices, we propose the *Topic-Oriented Framework* (TOF) in this section. In Section 6.1, we discuss the *Query Log Topic Model* (QLTM), which derives latent topics by integrating the information of the three dimensions in query log. In Sections 6.2 and 6.3, we discuss how to derive query-topic relations and the method of topic-based query intent mining.

6.1 Query log topic model

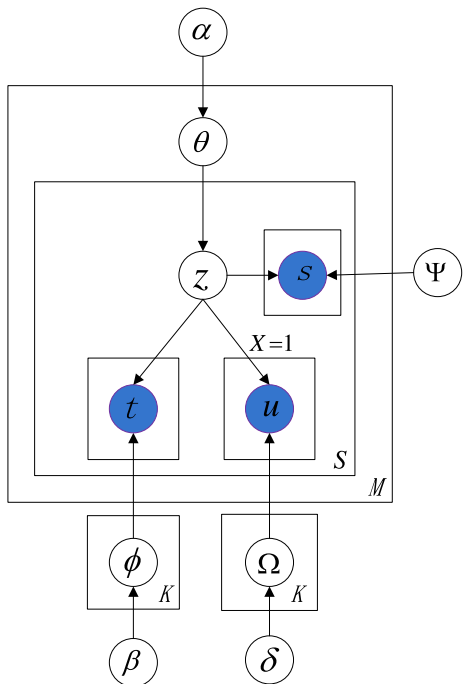
Query Log data has its unique characteristic and the general-purpose models can only work suboptimally for modeling query log data (as we will show in Section 8.3.1). Thus, we propose the *Query Log Topic Model* (QLTM), which is calibrated for query log data, in order

to integrate the information of the three dimensions and derive the semantically coherent topics to capture the latent relations between search queries in lower dimensions. In order to apply topic modeling on query log, we first group each user’s log entries as a document and then organize each document by sessions. Intuitively, query terms and the clicked URLs of the same session are utilized to satisfy the same information need and thus we constrain that the terms and URLs in the same session share the same topic. An important feature of web search is that it is essentially dynamic and thus we associate each topic with a Beta distribution to model its temporal prominence. The generative process of QLTM is described as follows:

- for each topic $k \in 1, \dots, K$
 - draw a term distribution $\phi_k \sim \text{Dirichlet}(\beta)$;
 - draw a URL distribution $\Omega_k \sim \text{Dirichlet}(\delta)$;
- for each document $m \in 1, \dots, M$
 - draw topic distribution $\theta_m \sim \text{Dirichlet}(\alpha)$;
 - for each session in m
 - choose a topic $z \sim \text{Multinomial}(\theta_m)$;
 - generate terms $t \sim \text{Multinomial}(\phi_z)$;
 - if the URL existence indicator $X_s = 1$, generate URLs $u \sim \text{Multinomial}(\Omega_z)$;
 - draw timestamps $s \sim \text{Beta}(\Psi_z)$;

The graphical model of QLTM is presented in Figure 1, which helps the reader to quickly capture the essence of QLTM. In Figure 1, α, β and δ are the hyperparameters. t, u and s are

Figure 1 Graphical model of QLTM



the observed variable. θ , ϕ and Ω are the latent parameters of interest. The graphical model illustrate a process as follows. We assume that each user has a topic distribution. When submitting the search queries in a session to the search engine, the user first decides the topic and then selects some query terms according to the chosen topic. For each session, the user needs to decide whether to click some URLs or not and any clicked URL is generated according to the chosen topic as well. Finally, the timestamps are generated according to the chosen topic’s temporal prominence. Hence, the topic assignment of a session is sensitive to the query terms, URLs and timestamps. Therefore, our model seamlessly integrate the information in query log and the latent topic space can be considered as an effective low-dimensional representation of the information in query log.

Now we discuss the parameter inference based on Gibbs sampling. The joint probability of the query terms, the URLs and the timestamps is given as follows:

$$P(\mathbf{t}, \mathbf{u}, \mathbf{s}, \mathbf{z}|\alpha, \beta, \delta, \mathbf{X}) = P(\mathbf{t}|\mathbf{z}, \beta)P(\mathbf{u}|\mathbf{z}, \delta, \mathbf{X})P(\mathbf{s}|\mathbf{z}, \Psi)P(\mathbf{z}|\alpha). \tag{8}$$

After combining the factorized formula terms, applying Bayes rule and folding the terms into the proportionality constant, the conditional probability of assigning the k th topic for the i th session is defined as follows:

$$\begin{aligned}
 P(z_i = k|\mathbf{z}_{-i}, \mathbf{t}, \mathbf{u}, \mathbf{s}, \Psi) &\propto \\
 &\frac{C_{mk}^{MK} + \alpha_k}{\sum_{k'}^K (C_{mk'}^{MK} + \alpha_{k'})} \prod_{j=1}^T \frac{(1 - s_j)^{\psi_{k1}-1} s_j^{\psi_{k2}-1}}{B(\psi_{k1}, \psi_{k2})} \\
 &\frac{\Gamma\left(\sum_{t=1}^T (C_{kt}^{KT} + \beta_t)\right)}{\Gamma\left(\sum_{t=1}^T (C_{kt}^{KT} + \beta_t + N_{it})\right)} \prod_{t=1}^T \frac{\Gamma(C_{kt}^{KT} + \beta_t + N_{it})}{\Gamma(C_{kt}^{KT} + \beta_t)} \\
 &\left\{ \frac{\Gamma\left(\sum_{u=1}^U (C_{ku}^{KU} + \delta_u)\right)}{\Gamma\left(\sum_{u=1}^U (C_{ku}^{KU} + \delta_u + N_{iu})\right)} \prod_{u=1}^U \frac{\Gamma(C_{ku}^{KU} + \delta_u + N_{iu})}{\Gamma(C_{ku}^{KU} + \delta_u)} \right\}^{I(X_i=1)}, \tag{9}
 \end{aligned}$$

where C_{mk}^{MK} is the number of sessions that are assigned topic k in document m , C_{kt}^{KT} is the number of times that the term t is assigned topic k , C_{ku}^{KU} is the number of times that the URL u is assigned topic k , N_{it} is the number of t in session i and N_{iu} is the number of u in session i . After each iteration of the sampling procedure, we update the parameters ψ_{k1} and ψ_{k2} for the k th topic as follows:

$$\psi_{k1} = \bar{s}_k \left(\frac{\bar{s}_k(1 - \bar{s}_k)}{v_k^2} - 1 \right), \tag{10}$$

$$\psi_{k2} = (1 - \bar{s}_k) \left(\frac{\bar{s}_k(1 - \bar{s}_k)}{v_k^2} - 1 \right), \tag{11}$$

where \bar{s}_k and v_k^2 denote the sample mean and biased sample variance of topic k ’s timestamps.

So far, we consider the session as the basic unit for topic assignment. However, the information about each search query has not been fully utilized. We now discuss an approach to capture the query information. It is well recognized that the query terms which frequently appear together in a query bear very coherent semantic meanings [35]. However, capturing the relations of query terms usually requires to breaks the *bag-of-words* assumption, which

assumes that the words in a document are exchangeable. Although there exists work such as the bigram topic model [36] that breaks the *bag-of-words* assumption, it is not suitable for query log data for two reasons: (1) search queries are short and are usually not grammatical. Thus, query terms do not demonstrate strong bigram relation as those in the conventional articles; (2) The computational cost of the bigram topic model is too high for the query log, which is usually in massive size.

To solve this problem, we now discuss an approach to embed the query information into QLTM by mining frequent patterns from query log. Due to the semantic coherency of the search query, we assume that the query terms that frequently appear together in a query are likely to be from the same topic. By considering each query as a *transaction* and mining the frequent closed patterns [16] of query terms, we can build a term-term relevance matrix R^T , the entries in which is calculated as follows:

$$R_{ij}^T = \text{sup}(t_i, t_j),$$

where t_i and t_j are the terms and sup is the support of the pattern. Similarly, the URLs that are frequently clicked for the same query are also semantically coherent and we also build an URL-URL relevance matrix R^U based on counting the frequency of URL co-occurrence. After obtaining the constraint matrix R^T and R^U , we utilize them as the priors of ϕ_k and Ω_k as follows:

$$p(\phi_k | R^T) \propto (\phi_k^\top R^T \phi_k). \tag{12}$$

$$p(\Omega_k | R^U) \propto (\Omega_k^\top R^U \Omega_k). \tag{13}$$

The log posteriors for MAP estimation are given by:

$$L_{MAP}^T = \sum_{t=1}^T C_{kt}^{KT} \log \phi_{t|k} + \log(\phi_k^\top R^T \phi_k). \tag{14}$$

$$L_{MAP}^U = \sum_{u=1}^U C_{ku}^{KU} \log \Omega_{u|k} + \log(\Omega_k^\top R^U \Omega_k). \tag{15}$$

Optimizing (14) with respect $\phi_{w|k}$ subject to the constraints $\sum_{t=1}^T \phi_{t|k} = 1$, we can obtain the following fixed point update:

$$\phi_{t|k} \leftarrow \frac{1}{C_{kt}^{KT} + 2} \left(C_{kt}^{KT} + 2 \frac{\phi_{t|k} \sum_{i=1}^T R_{it}^T \phi_{i|k}}{\phi_k^\top R^T \phi_k} \right). \tag{16}$$

Similarly, we can get the update for $\phi_{u|k}$ as follows:

$$\Omega_{u|k} \leftarrow \frac{1}{C_{ku}^{KU} + 2} \left(C_{ku}^{KU} + 2 \frac{\Omega_{u|k} \sum_{i=1}^W R_{iu}^U \Omega_{i|k}}{\Omega_k^\top R^U \Omega_k} \right). \tag{17}$$

We no longer have neat conjugate priors for $\phi_{t|k}$ and $\Omega_{u|k}$. Thus, at the end of each major cycle of Gibbs sampling, $\phi_{t|k}$ and $\Omega_{u|k}$ are re-estimated according to (16) and (17). Then, in the next iteration, the C_{kt}^{KT} and C_{ku}^{KU} in (9) are replaced as follows:

$$C_{kt}^{KT} = \phi_{t|k} \cdot C_{kt}^{KT}; \quad C_{ku}^{KU} = \Omega_{u|k} \cdot C_{ku}^{KU}. \tag{18}$$

6.2 Offline query-topic relation derivation

We proceed to discuss the method of deriving the relations between search queries and the topics. After the burn-in period of the Gibbs sampling discussed in Section 6.1, we record the topic assignments of each session. When the sampling procedure is completed, the i th session is associated with a search topic vector S_i^{topic} in which each entry is the times that the i th session has been assigned to the corresponding topic. Through the normalization on each row vector of S_i^{topic} , each entry in S_i^{topic} represents the probability that the i th session belongs to the corresponding topic. Then we can obtain the query-topic matrix Z by multiplying the query-session matrix B^S with S^{topic} . The procedure of deriving query-topic relation is summarized in Algorithm 2. This process can be done offline for the whole query log. Once the query-topic relations are obtained, they can be stored for efficient online query intent mining, which will be discussed in the next subsection.

Algorithm 2 Query-Topic Relation Derivation

Input: session-topic matrix S^{topic} , which is initialized as all zeros; query-session matrix B^S ; the number of iterations G ; the number of iterations for burn-in period I

Output: query-topic matrix Z

- 1: counter=0;
 - 2: **repeat**
 - 3: **if** $z_i = j$ **and** counter > $(G - I)$ **then**
 - 4: $S_{ij}^{topic} \leftarrow S_{ij}^{topic} + 1$;
 - 5: **end if**
 - 6: counter=counter+1;
 - 7: **until** iterates G times
 - 8: Compute the diagonal degree matrix D by $D_{ii} \leftarrow \sum_j S_{ij}^{topic}$;
 - 9: Compute session-topic matrix by $S^{topic} \leftarrow D^{-1} S^{topic}$;
 - 10: Computer query-topic matrix by $Z \leftarrow B^S S^{topic}$;
-

6.3 Online query intent mining

Based on the query-topic matrix Z obtained from Algorithm 2, we construct the topic-based query affinity matrix W^{topic} as follows:

$$W^{topic} = Z \Lambda^{-1} Z^\top, \tag{19}$$

where $\Lambda_{kk} = \sum_i Z_{ik}$. Then, the un-normalized graph Laplacian is computed as follows:

$$L_t = I - W^{topic}. \tag{20}$$

We define $A \in \mathbb{R}^{\mathbb{T} \times \mathbb{C}}$ as the topic-intent matrix, where \mathbb{T} is the number of topics and \mathbb{C} is the number of query intents. In A , each row represents a topic and each column represents an intent. Then we obtain Z_l by only keeping the entries corresponding to manually labeled queries in Z . Similarly, we obtain F_l by truncating F^0 in the same way. The regularization framework for the topic-based intent mining is formalized as follows:

$$\min_A \frac{1}{2} \|Z_l A - F_l\|_F^2 + \frac{\mu}{2} \sum_{j=1}^c (Z a_j)^\top L_t (Z a_j). \tag{21}$$

The notation $\|\cdot\|_F$ in (21) is the Frobenius norm.¹ In (21), the first term and the second term define the fitting constraint and the smoothness constraint respectively. This regularization framework is different from that discussed in Section 5 in that its optimization objective is to obtain the optimal topic-intent matrix A^* rather than the query-topic matrix F^* . With simple algebra, we obtain the optimal solution given by:

$$A^* = \left(Z_l^\top Z_l + \mu Z^\top Z - \mu \left(Z^\top Z \right) \Lambda^{-1} \left(Z^\top Z \right) \right)^{-1} Z_l^\top F_l. \quad (22)$$

Once the estimated topic-intent matrix A^* is obtained, we can straightforwardly get the class label for each query by:

$$F^* = Z A^*. \quad (23)$$

Finally, we assign each query the intent of the highest probability and further constrain that the highest probability should be higher than a threshold T_h .

7 Complexity analysis

In this section, we analyze the time and space complexity of the proposed frameworks. For ROF, we need to compute the inverse of an $N \times N$ matrix to obtain F^* . The time complexity is usually $O(N^3)$, since exactly solving the equivalent large linear systems is still challenging. The space complexity of Result-Oriented Framework is $O(N^2)$ because the inverse matrix is usually dense. For TOF, the time complexity of the Gibbs sampling is $O(GK(\hat{T}^\# + \hat{U}^\#))$, where G is the number of Gibbs iterations. The space complexity is $O(K(M + U^\# + S^\# + T^\#))$. As a session usually contains few queries and a query typically contains one or two terms and raises limited URL clickthrough, $\hat{U}^\#$, $S^\#$ and $\hat{T}^\#$ are nearly linear in \hat{N} . The long-tail phenomenon [33] in web search shows that "popular" search queries actually make up less than 30% of the overall searches performed on the web, the remaining are millions of queries that might be only searched a few times or even only once. Since \hat{N} is roughly linear in N and $K \ll N$, the time and space complexities of the sampling procedure are roughly linear in the number of unique queries N . As for the online computing, the time complexity of TOF is $O(K^3 + K^2 N_l + K N C)$ and the space complexity is $O(K N + N_l C)$. As $K \ll N$ and $C \ll N$, the online time and space complexities of TOF are nearly linear in N (Figure 2).

8 Experiments

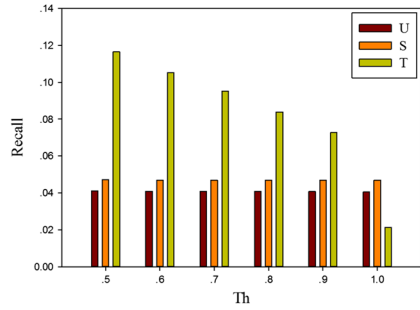
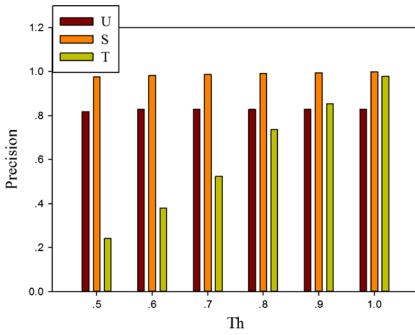
In this section, we gauge the performance of the proposed frameworks and compare them with the state-of-the-art methods. In Section 8.1, we describe the experimental setup. Then we study the sensitivity of the parameters of ROF and TOF in Sections 8.2 and 8.3. Finally, we make a systematical comparison of different query intent mining methods in Sections 8.4 and 8.5.

8.1 Experiment setup

We utilize the query log from Yahoo search engine for the experiments.² The query log contains 506,515 search queries of 2,158 search engine users during a period of four months.

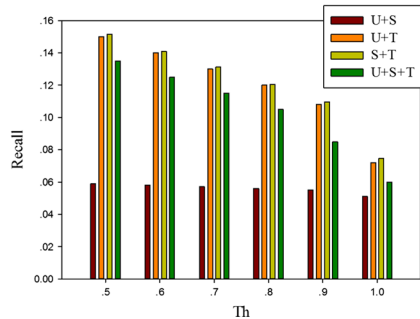
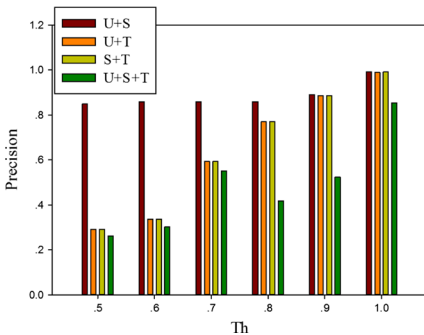
¹<http://mathworld.wolfram.com/FrobeniusNorm.html>

²The work was done when the first author visiting Yahoo Labs



(a) The effect of T_h on the precision of each single dimension

(b) The effect of T_h on the recall of each single dimension



(c) The effect of T_h on the precision of Result-Oriented Framework (Maximum Confidence Strategy)

(d) The effect of T_h on the recall of Result-Oriented Framework (Maximum Confidence Strategy)

Figure 2 Parameter sensitivity of single dimensions and Result-Oriented Framework (ROF)

The intents of the search queries are manually labeled by five human judges according to 546 ODP³ categories and conflicts of the labeled results are resolved by majority voting. A subset containing 5,000 users are utilized as the validation data, a subset containing 5,000 users are heldout as the testing data and the remaining are considered as the training data. In order to clarify the evaluation procedure, we present an overview of the evaluation workflow in Figure 3. Each training set TS_i contains a random number (say 100) of *positive* queries (QP_{TS_i}) which are randomly chosen from the queries that have a particular intent τ , and a random number (say 100) of *negative* queries (QN_{TS_i}) which are randomly chosen from the queries with intents other than τ (i.e., $\bar{\tau}$). To test whether a particular intent mining method, denoted by IM , can successfully distinguish queries with the desirable intent τ from the other queries, we first run IM using a particular training set TS_i to obtain a set of possible queries with intent τ (i.e., $QP_{TS_i}^{IM}$). Then, we compare the results $QP_{TS_i}^{IM}$ against the ground truth, which is the manually labeled query intents, using standard precision and recall measures. Finally, we repeat the mining phase for $TS_1, TS_2, TS_3, \dots, TS_{10,000}$ and

³<http://www.dmoz.org/>

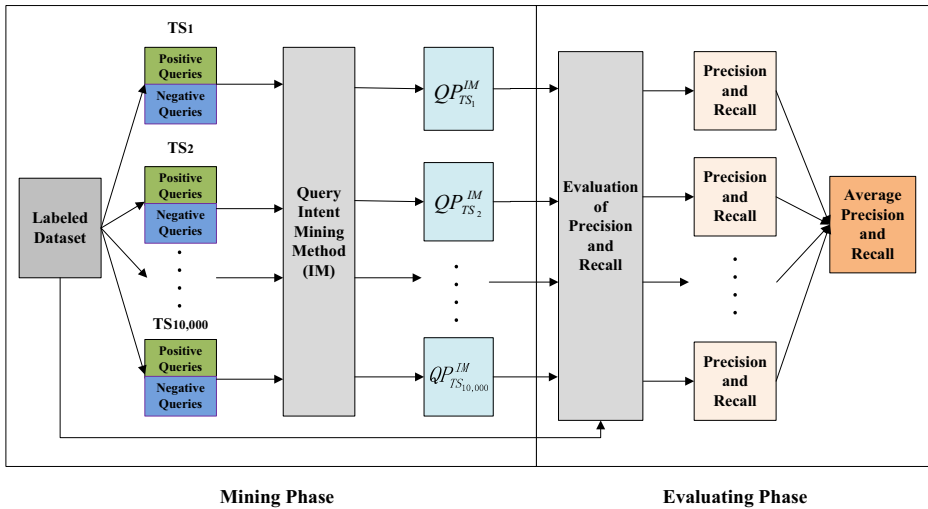


Figure 3 Overview of evaluation workflow

obtain the average precision and recall measures over all the training sets on all methods for the purpose of evaluation and comparison.

8.2 Parameter sensitivity analysis of ROF

We now study the sensitivity of the parameters in ROF on the validation dataset. Previous work [22] shows that (2) demonstrates similarly good performance when μ varies between 1 and 9. Thus, we choose $\mu = 5$ as the default μ value for (2). When ROF is applied to each dimension, it reduces to applying Algorithm 1 to a single dimension and uses the intermediate result from a single dimension as the final result. As the result obtained from each individual dimension serves as the basis of ROF, we first evaluate T_h 's effect on the intent mining performance of the URL, the session and the term dimensions. From the results shown in Figure 2a and b, we observe that the URL and the session dimensions are not sensitive to T_h . Their precision and recall stay roughly the same value when T_h varies from 0.5 to 1.0. On the contrary, the precision and recall of the term dimension change dramatically when T_h varies from 0.9 to 1.0. The result is consistent with our intuition that terms are usually ambiguous and the commonality of search intent can only be guaranteed by using a high threshold. Therefore, the term dimension needs high T_h to achieve high precision. Furthermore, when T_h is low the term dimension usually results in high recall. We also observe that the performance of the session dimension dominates that of the URL dimension while the precision or recall of the term dimension can also outperform those of the URL dimension by tuning T_h . The reason is that the session usually contains very coherent information and the search queries in the same session typically share the same query intent.

We then evaluate the performance of the *Maximum Confidence* strategy against different T_h . The results of the four different dimensional combinations (U+S), (U+T), (S+T) and (U+S+T) are shown in Figure 2c and d. We observe that the precision and recall of (U+S) are very stable when T_h varies from 0.5 to 0.9. However, When T_h increases from 0.9 to 1.0, there is a significant increase in precision but a slight decrease in recall. The other three

combinations are sensitive to T_h and demonstrate similar trends: when T_h increases, their precisions steadily become higher while their recalls steadily become lower. When T_h is 1.0, the (S+T) combination yields the highest precision and recall over all combinations. We also observe that by combining the information of different dimensions, some combinations achieve higher precision and some combinations achieve higher recall comparing to using a single dimension. The result verifies our assumption that the extra query relations obtained from different dimensions have the potential to improve the performance of query intent mining.

The *Majority Vote* strategy does not need the tuning of T_h . The only parameter used in this strategy is the number of votes. As we have three dimensions in total, the vote threshold is straightforwardly set to be 2, meaning that a query is labeled as the intent class that receives at least two votes. This strategy obtains a high precision with a fairly good recall. Both the results of the *Maximum Confidence* and *Majority Vote* validate our assumption that integrating more information has the potential to boost the performance of query intent mining. Therefore, our approach paves the way for designing far better methods for query intent mining than the existing approaches.

8.3 Parameter sensitivity analysis of TOF

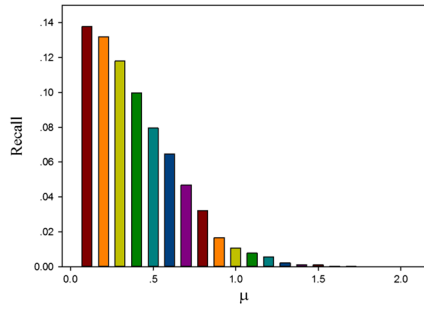
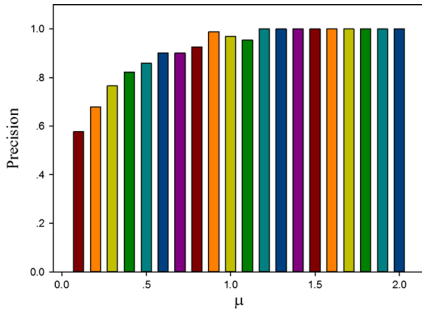
8.3.1 Evaluation of QLTM

We compare QLTM with some counterparts. In LDA, the default hyperparameter settings suggested by Griffiths and Steyvers [15] demonstrate very good performance. Therefore, we also choose $50/K$ as the default value of α in our experiments. Since we have no prior knowledge of the topical distributions of terms and URLs, we set both β and δ to 0.01. Empirically, we find that the QLTM demonstrates very good performance with the parameter settings discussed above. Parameter tuning of the topic model is well studied in the field of probabilistic topic modeling and is beyond the scope of this paper, interested readers may refer to [37] to find out a detailed discussion on this subject. After an extensive survey, we select five models as the baselines: LDA [4], PTM1 [10], PTM2 [10], DSTM [19] and RSTM [19] (Figure 4).

We utilize a held-out dataset that contains 10,000 search queries to evaluate the models' capability of predicting unseen data. Perplexity is a standard measure of evaluating the generalization performance of a probabilistic model [29]. It is monotonically decreasing in the likelihood of the held-out data. Therefore, a lower perplexity indicates better generalization performance. The result of perplexity comparison is presented in Figure 5, from which we observe that the QLTM demonstrates much better capability in predicting unseen data comparing to the baselines. For example, when the number of search topics is set to 1000, the perplexity of LDA, PTM1, PTM2, DSTM and RSTM are 18945, 21220, 20251, 12768 and 13575. QLTM significantly reduces the perplexity to 7594. Different from DSTM and RSTM which only utilize the URL information partially, QLTM takes full advantage of query terms, URLs as well as the frequent patterns in web search and thus provides better fit for the latent structure of query log data. Therefore, the resultant topic space can be considered as an effective representation of the information in query log.

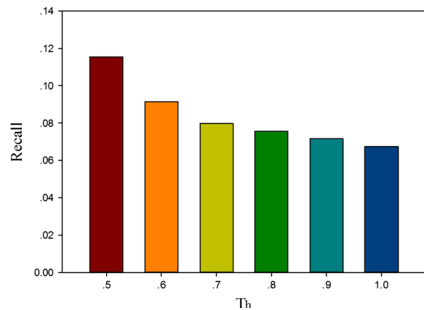
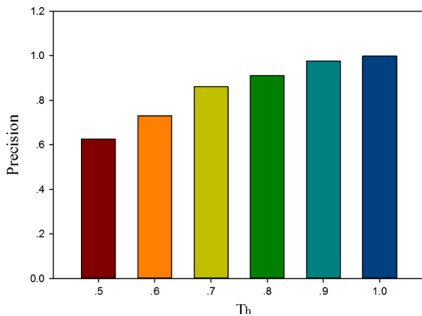
8.3.2 Evaluation of query intent mining

We now analyze the parameter sensitivity of the TOF on the validation dataset. We first evaluate the effect of μ on this framework by fixing T_h at a relatively high value of 0.7. The



(a) The effect of μ on the precision of TOF

(b) The effect of μ on the recall of TOF



(c) The effect of T_h on the precision of TOF

(d) The effect of T_h on the recall of TOF

Figure 4 Parameter sensitivity of Topic-Oriented Framework (TOF)

sensitivity study of μ is presented in Figure 4a and b. Due to the variation of the optimization objective, the settings of μ are quite different from that in (2). As μ becomes higher, the precision steadily increases while the recall dramatically decreases. When $\mu = 1.0$, the recall is close to zero. Therefore, we choose 0.5 as the default value for μ . By using this parameter setting, the framework demonstrates high precision and reasonable recall. We then study the sensitivity of T_h of the regularization framework and the results are shown in Figure 4c and d. The precision increases steadily as T_h increases while the recall decreases when T_h increases. When T_h is set to 0.7, the framework strikes a good balance between precision and recall. The result shows that using the latent topic space is effective to enhance both the precision and recall of query intent mining.

8.4 Result comparison

After determining the parameter settings of *ROF* and *TOF* in Sections 8.2 and 8.3, we now compare the performance of the following methods on the test dataset:

- *CG*: The click graph based method proposed in [22]. Essentially, the method equals to applying *Result-Oriented Framework* to the URL dimension.
- *GS*: The graph summarization based method in [11].
- *ROF-S*: *Result-Oriented Framework* on the session dimension.

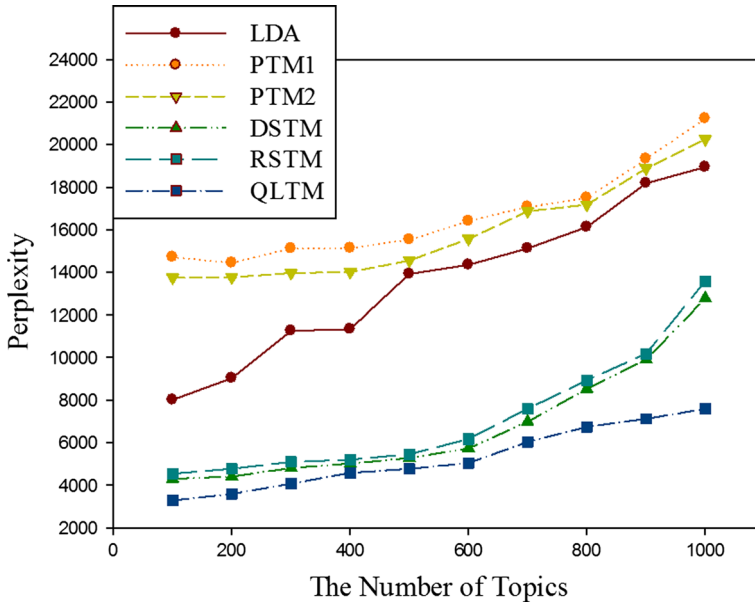
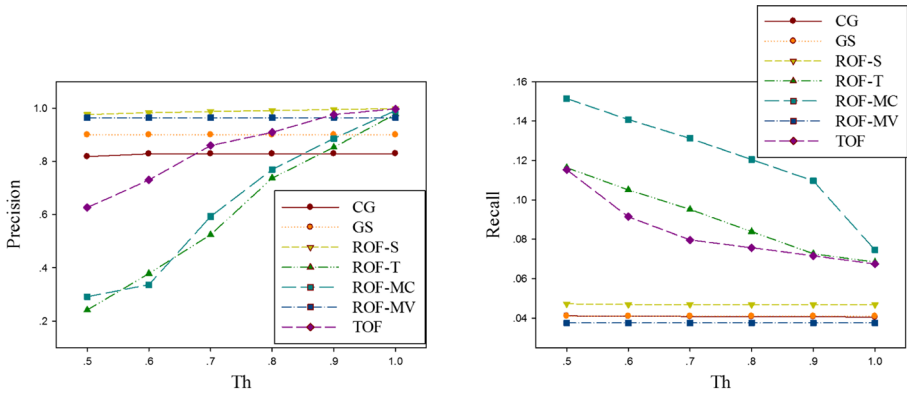


Figure 5 Perplexity comparison

- *ROF-T*: *Result-Oriented Framework* on the term dimension.
- *ROF-MC*: *Result-Oriented Framework* with *Maximum Confidence* strategy. In this method, the (S+T) combination demonstrates the best performance and thus it is used in the comparison.
- *ROF-MV*: *Result-Oriented Framework* with *Majority Vote* strategy.
- *TOF*: *Topic-Oriented Framework*.

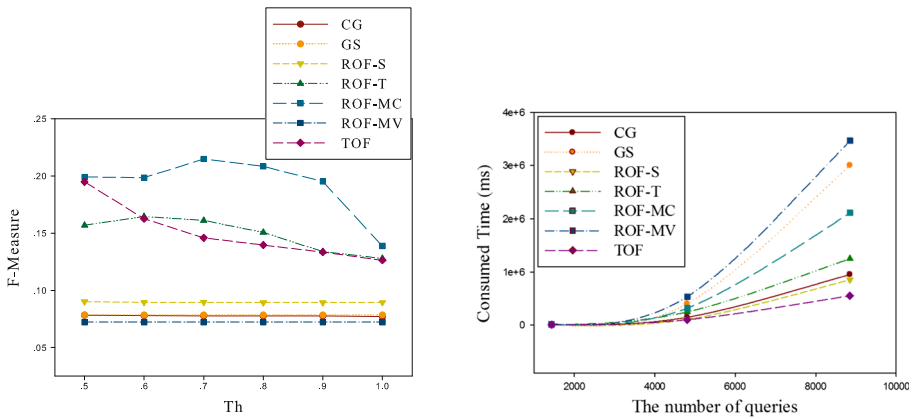
The results are presented in Figure 6a, b and c. We observe that these methods can be divided into two categories in term of whether they are sensitive to T_h . *CG*, *GS*, *ROF-S* and *ROF-MV* demonstrate relatively stable precisions and recalls when T_h varies. On the contrary, the performance of *ROF-T*, *ROF-MC*, and *TOF* is sensitive to T_h and their performance can be tuned by T_h .

Among the methods that are not sensitive to T_h , *ROF-MV* yields similar recall as *CG* and *GS*. However, the precision of *ROF-MV* is much higher than that of *CG* and *GS*. The precision improvement demonstrates the effectiveness of the voting phase of *Majority Vote* strategy, which successfully filters out some false-positive results by combining the voting results from the three dimensions. Another observation is that the performance of *ROF-S* dominates those of *CG*, *GS* and *ROF-MV* in terms of both precision and recall. Methods that are not sensitive to T_h have stable precision/recall performance. Thus they are not preferred when a set of results with a particular precision/recall setting (e.g., less results with high accuracy, more results with lower accuracy, etc) is needed. *ROF-T*, *ROF-MC* and *TOF* are more flexible and their performance can be flexibly tuned with T_h . Notably, *ROF-MC* yields the highest precision when T_h is high and the recall of *ROF-MC* is always the best across all the methods. These results verify the effectiveness of *Maximum Confidence* Strategy in ROF. On the other hand, the performance of *ROF-T* is, in general, worse than that of *CG* and *GS* due to the inherent ambiguity in query terms. *TOF* yields very good precision when



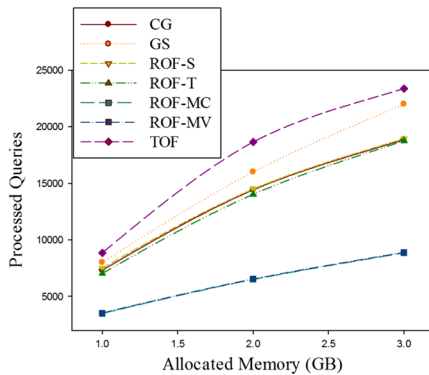
(a) Comparison of Precision

(b) Comparison of Recall



(c) Comparison of F-Measure

(d) Comparison of Scalability (Time)



(e) Comparison of Scalability (Space)

Figure 6 Comparison of different query intent mining methods

T_h varies from 0.5 to 1.0. Although the performance of *TOF* does not dominate that of *CG* and *GS*, it achieves higher precision than *CG* and *GS* when T_h is set to be larger than 0.8.

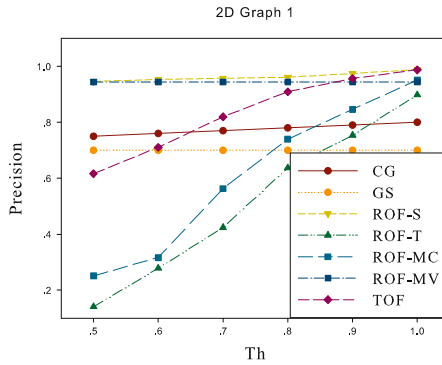
Meanwhile, the recall of *TOF* is significantly higher than that of *CG* and *GS* when T_h varies. Furthermore, we will show in Section 8.5 that *TOF* is better than the other methods in terms of scalability. In terms of F-measure, *ROF-T*, *ROF-MC* and *TOF* constantly outperform the other method when T_h varies, showing that they strike a good balance between precision and recall. This result again verify the superiority of the proposed methods in query intent mining. Moreover, in terms of F-measure, *ROF* demonstrates superior performance over *TOF*. However, we shall see that the strength of *TOF* lies in its efficiency and scalability.

Astute reader may find that the recall of the methods under study are all relatively low. To the best of our knowledge, previous work usually only considers the precision and the present work is the first one to present evaluation results of the recall of different query intent mining methods. In the scenario of semi-supervised query intent mining, a large amount of training examples can still be derived even the recall is low since the number of queries in query log is huge. Hence, low recall does not necessarily affect the practical use of these methods. However, as we have shown, the recall of our methods is higher than the state-of-the-art methods, indicating that our methods can utilizes a smaller query log to derive roughly the same amount of training examples as the other methods using a larger query log. The advantage of utilizing a smaller query log potentially enables our methods even better performance in terms of efficiency.

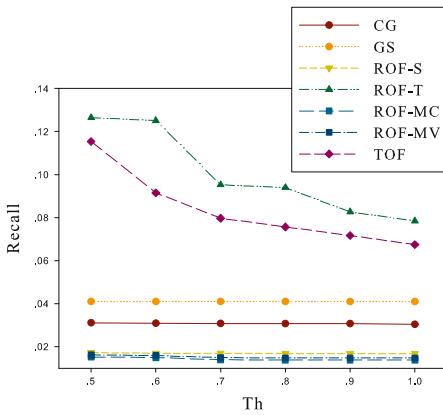
One challenging issue in query intent mining is how to tackle with the tail search queries, which only appear very few times in query log. In order to gauge the performance of our proposed approaches, we evaluate them on a dataset containing 500 tail search queries. The experimental results are shown in Figure 7. We observe that all the methods demonstrate relatively worse performance on the dataset only containing tail queries, because these methods usually have fewer information to mine the latent intent of the tail queries. However, as shown in Figure 7a, the *ROF* and *TOF* still show superior performance in terms of precision of query intent mining. Notably, in Figure 7b, *TOF* keeps very high recall on tail queries, showing that utilizing probabilistic topic model can effectively tackle the problems of tail queries. The F-measure is shown in Figure 7c, we can see that *TOF* and *ROF-T* significantly outperforms the baselines. This result shows that the proposed methods are well-suited to the scenario of query intent mining of tail search queries. Hence, they achieve the state-of-the-art performance.

8.5 Scalability comparison

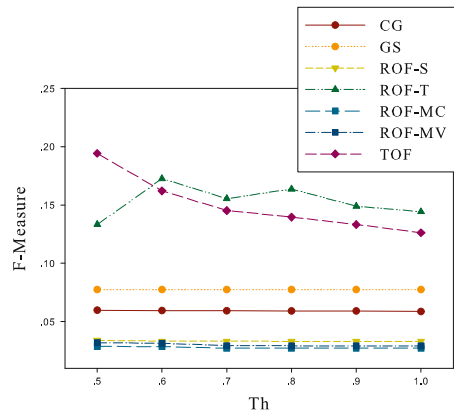
Section 7 theoretically analyzes the complexity of the proposed frameworks. We now empirically evaluate the scalability of them on a 2.8GHz CPU with 4GB memory. We first evaluate the consumed time against the number of queries and the experimental result is shown in Figure 6d. *TOF* always takes the least time to get the results. For example, when the size of the dataset is about 1,000 queries, *CG* takes 4738ms to get the result and the time consumed by *TOF* is 3998ms. *ROF-MV* and *ROF-MC* typically consume more time than *CG* because of the multi-dimensional computations and *GS* is also slower than *CG*. More importantly, the time consumption of *TOF* increases modestly with the increase of the number of queries. The result verifies the efficiency of the topic-based intent mining in *TOF*. The low-dimensional topic representation effectively reduces the online computation time and thus *TOF* demonstrates superior scalability over the other approaches. In order to evaluate the space consumption of the proposed frameworks, we allocate different amounts of memory to the methods under study and estimate how many queries each method is able to process. The result is shown in Figure 6e. The *ROF-MC* and *ROF-MV* consume more memory than other methods, due to the storage of the provisional results of the three query



(a) Comparison of Precision



(b) Comparison of Recall

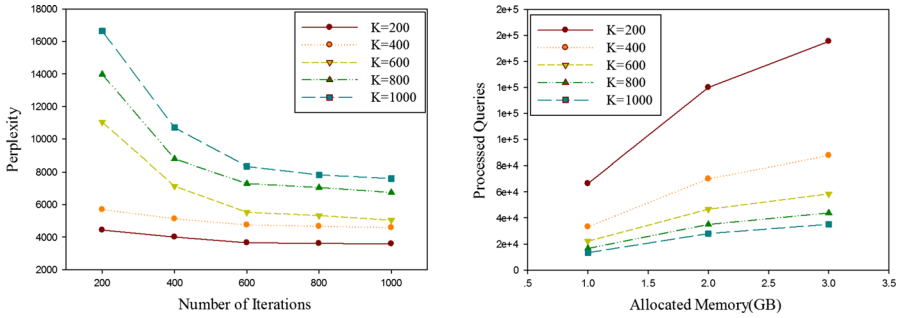


(c) Comparison of F-Measure

Figure 7 Comparison of different query intent mining methods on tail search queries

log dimensions. Among all the methods under study, TOF demonstrates the lowest memory consumption. For example, when 1.0 GB memory is allocated, *CG*, *GS*, *ROF-MV*, and *ROF-MC* run out of memory quickly when processing a dataset containing 9,000 queries and only *TOF* can successfully process this dataset. The results verify that *TOF* methods have much better scalability in terms of both time and space compared to the other methods.

We have already shown that TOF achieves the best performance in terms of both online time and memory consumption, which is usually the focus of web applications. In fact, QLTM also demonstrates very good offline performance. Figure 8a presents the perplexity value after each major Gibbs iteration of QLTM. It shows that, for a fixed number of topics (e.g., K varies from 200 to 1000), the convergence can be quickly achieved less than 500 iterations. Similar results can also be observed when varying the number of topics K to other values. Page limitations preclude a full description of all these results in this paper. As we discussed in Section 7, the time of each Gibbs sampling depends on the value of K and the statistics of the query log. In our experiments, each iteration is very efficient and usually only takes several seconds. It is worth emphasizing that, the offline computation of QLTM only needs to be done for one time and the obtained results can be stored for future use of mining different query intents of different queries. Another advantage of the offline



(a) Perplexity After Each Iteration of QLTM (b) Offline Memory Consumption of QLTM

Figure 8 Offline scalability analysis of QLTM

phase of QLTM lies in its low requirement for memory. As shown in Figure 8b, as K varies from 200 to 1000, the offline phase of QLTM can consistently process much more queries than the baselines when a fixed memory is allocated. The good scalability of the offline phase of QLTM again verifies that TOF is a good candidate when scalability is the priority in query intent mining.

9 Conclusion

In this paper, we propose two new frameworks to mine the intents of web search queries by exploiting the multi-dimensional information in the search engine query log. We utilize the URL dimension, the session dimension and the term dimension to collectively capture the latent query relations in query log. Based on the three dimensions, we first propose the *Result-Oriented Framework* (ROF) that infers query intents by combining the provisional results from different dimensions. ROF is easy to implement in the multi-dimensional scenario and significantly improves the quality of the mining results. In order to reduce the computational cost, we further propose the *Topic-Oriented Framework* (TOF) that uses a generative model that derives a small set of topics to represent query relations. Then query intents are inferred based on the query affinity matrices reconstructed by the topics. We conduct a wide spectrum of experiments to study the parameter sensitivity and effectiveness of the two frameworks. Experimental results demonstrate that the proposed frameworks significantly outperform the state-of-the-art methods with respect to different metrics.

References

1. Arguello, J., Diaz, F., Callan, J., Crespo, J.F.: Sources of evidence for vertical selection. SIGIR (2009)
2. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. SIGIR (1998)
3. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. SIGKDD (2000)
4. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. NIPS (2002)
5. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. CIKM (2008)
6. Broder, A.: A taxonomy of web search. SIGIR forum (2002)
7. Broder, A.Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., Zhang, T.: Robust classification of rare queries using web knowledge. SIGIR (2007)

8. Calderon-Benavides, L., Gonzalez-Caro, C., Baeza-Yates, R.: Towards a deeper understanding of the user's query intent. SIGIR Workshop (2010)
9. Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., Li, H.: 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, pp. 875–883, ACM (2008)
10. Carman, M.J., Crestani, F., Harvey, M., Baillie, M.: Towards query log based personalization using topic models. CIKM (2010)
11. Celikyilmaz, A., Hakkani-Tur, D., Tur, G.: Leveraging web query logs to learn user intent via bayesian discrete latent variable model. ICML (2011)
12. Craswell, N., Szummer, M.: Random walks on the click graph. SIGIR (2007)
13. Dang, V., Xue, X., Croft, W.B.: Inferring query aspects from reformulations using clustering. CIKM (2011)
14. Deng, H., Lyu, M.R., King, I.: A generalized co-hits algorithm and its application to bipartite graphs. SIGKDD (2009)
15. Griffiths, T.L., Steyvers, M.: Finding scientific topics. NAS (2004)
16. Han, J., Wang, J., Lu, Y., Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. ICDM (2002)
17. Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., Zheng, Q.: Mining query subtopics from search log data. SIGIR (2012)
18. Jiang, D., Leung, K.W.T., Ng, W.: Context-aware search personalization with concept preference. CIKM (2011)
19. Jiang, D., Vosecky, J., Leung, K.W.-T., Ng, W.: G-WSTD: a framework for geographic web search topic discovery. CIKM (2012)
20. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. WSDM (2011)
21. Lee, U., Liu, Z., Cho, J.: Automatic identification of user goals in web search. In: Proceedings of the 14th international conference on World Wide Web, pp. 391–400, ACM (2005)
22. Li, X., Wang, Y.Y., Acero, A.: Learning query intent from regularized click graphs. SIGIR (2008)
23. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval (2008)
24. Pantel, P., Lin, T., Gamon, M.: Mining entity types from query logs via user intent modeling. ACL (2012)
25. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. ACL (1993)
26. Poblete, B., Castillo, C., Gionis, A.: Dr. searcher and mr. browser: a unified hyperlink-click graph. CIKM (2008)
27. Qian, Y., Sakai, T., Ye, J., Zheng, Q., Li, C.: Dynamic query intent mining from a search log stream. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 1205–1208, ACM (2013)
28. Radlinski, F., Szummer, M., Craswell, N.: Inferring query intent from reformulations and clicks. WWW (2010)
29. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. UAI (2004)
30. Sadikov, E., Madhavan, J., Wang, L., Halevy, A.: Clustering query refinements by user intent. WWW (2010)
31. Shen, D., Pan, R., Sun, J.T., Pan, J.J., Wu, K., Yin, J., Yang, Q.: Query enrichment for web-query classification. TOIS (2006)
32. Shen, D., Sun, J.T., Yang, Q., Chen, Z.: Building bridges for web query classification. SIGIR (2006)
33. URL <http://www.seomoz.org/beginners-guide-to-seo/keywordresearch>
34. Wang, C.-J., Chen, H.-H.: Intent mining in search query logs for automatic search script generation. Knowl. Inf. Syst. **39**(3), 513–542 (2014)
35. Wang, X., Zhai, C.: Mining term association patterns from search logs for effective query reformulation. CIKM (2008)
36. Wallach, H.M.: Topic modeling: beyond bag-of-words. ICML (2006)
37. Wallach, H.M.: Structured topic models for language. Unpublished doctoral dissertation, Univ.of Cambridge (2008)
38. Wen, J.-R., Nie, J.-Y., Zhang, H.-J.: Query clustering using content words and user feedback. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 442–443, ACM (2001)
39. Yang, D., Shen, D.-R., Yu, G., Kou, Y., Nie, T.-Z.: Query intent disambiguation of keyword-based semantic entity search in dataspace. J. Comput. Sci. Technol. **28**(2), 382–393 (2013)