Front. Comput. Sci., 2020, 14(3): 143602
https://doi.org/10.1007/s11704-018-7283-x

RESEARCH ARTICLE

Personalized query suggestion diversification in information retrieval*

Wanyu CHEN[1], Fei CAI (✉)[1], Honghui CHEN[1], Maarten DE RIJKE[2]

1 Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
2 Informatics Institute, University of Amsterdam, Amsterdam, 1098XH, The Netherlands

Abstract Query suggestions help users refine their queries after they input an initial query. Previous work on query suggestion has mainly concentrated on approaches that are similarity-based or context-based, developing models that either focus on adapting to a specific user (personalization) or on diversifying query aspects in order to maximize the probability of the user being satisfied (diversification). We consider the task of generating query suggestions that are both personalized and diversified. We propose a personalized query suggestion diversification (PQSD) model, where a user's long-term search behavior is injected into a basic greedy query suggestion diversification model that considers a user's search context in their current session. Query aspects are identified through clicked documents based on the open directory project (ODP) with a latent dirichlet allocation (LDA) topic model. We quantify the improvement of our proposed PQSD model against a state-of-the-art baseline using the public america online (AOL) query log and show that it beats the baseline in terms of metrics used in query suggestion ranking and diversification. The experimental results show that PQSD achieves its best performance when only queries with clicked documents are taken as search context rather than all queries, especially when more query suggestions are returned in the list.

Keywords query suggestion, personalization, query suggestion diversification

1 Introduction

Modern search engines offer query suggestions to help users formulate a good query and thus to get their intended search results to address their information needs. Both Web search engines such as Baidu, Bing, Google, Yahoo! and Yandex and domain specific search engines such as Amazon (product search), Bloomberg (news) and ScienceDirect (academic publications) provide query suggestions to improve their system's usability. By predicting a user's search intent, a search engine recommends queries that reflect the user's information needs based on his inputs.

Previous work on query suggestion mainly focuses on recommending semantically related queries in response to a user's input query [2]. Such strategies cannot handle queries with uncertain search aspects, especially for users with diverse search intents. To alleviate the aforementioned problem, two categories of approaches have been introduced to complement conventional query suggestion methods: *diversification* and *personalization*. Intuitively, these two additions

Received August 11, 2017; accepted November 2, 2017

E-mail: {wanyuchen, caifei}@nudt.edu.cn; chhh0808@gmail.com; derijke@uva.nl

*A preliminary version of this paper is published in the proceedings of SIGIR 2017 [1]. In this extension, we (1) examine the impact on the model performance introduced by the trade-off parameter λ_2 which controls the contribution of personalization and diversification in our PQSD model via manually changing it from 0 to 1 with an interval 0.1; (2) investigate the sensitivity of our PQSD model to the number of query suggestions N, as a larger N simply increases the probability of including the ground truth in query suggestion list; and (3) include more related work and provide more detailed analyses of the approach and experimental results.

may appear to be orthogonal or even opposed to each other. Diversification has been injected into query suggestion systems [3,4] with a probabilistic model or with bipartite graphs while personalization is often incorporated into a query suggestion system by mining a user's past query behavior [5,6].

Regarding existing models for diversifying query suggestions, personal information of users has not been well-explored so far. However, we hypothesize that diversity and personalization can enhance each other when combined. Let us illustrate this by an example. Assume that a user submits "eclipse" to a search engine to find information about the software named Eclipse for Java Development Kit. Diversification aims to return a list of suggestions that covers as many facets of the input query as possible. For instance, in this case, diversification may suggest a list containing queries such as "Java Eclipse", "Eclipse song of C.N. Blue", "Car of Eclipse". However, this list of query suggestions may disappoint a user with a software engineering background if the query suggestion "Eclipse song of C.N. Blue" or "Car of Eclipse" is ranked higher than "Java Eclipse". In contrast, personalization strives to suggest query suggestions that are a good match to the user's past search history. Thus, when a software engineer submits "eclipse" to a search engine to find some information about the song named "Eclipse of C.N.Blue," a personalized query suggestion scenario will primarily focus on recommending queries about "Java Eclipse," which would be unsatisfactory.

From the above example, it seems that diversification can be helpful to handle a user's preferences but the topics covered in a list of query suggestions may be broad, resulting in dissatisfaction for a specific user. Personalization, on the other hand, can provide possible query suggestions related to the user's long-term preferences but it may be insensitive to changes in a user's preferences. If used excessively it may even cause redundancy in a list of query suggestions. Thus, in this paper, we take the advantages of both personalization and diversification to propose a personalized query suggestion diversification (PQSD) model, where diversification helps to generate multiple-aspect queries to increase the likelihood of suggested queries being clicked and personalization ensures that the suggested queries are close to a user's specific search intent.

The proposed PQSD model consists of two major stages. In the first stage, we develop a greedy query suggestion diversification model where a user's search context, consisting of queries and clicks, is considered to generate a diversified ranked list of queries; to this end, we use co-occurrences as well as semantic similarity between queries. In the second stage, we inject a user's long-term search behavior information into the model proposed in the first step with Bayes' rule. To determine a query's aspects,[1)] we collect documents that were shown and clicked in response to a query based on the search logs. After that, we extract descriptions of those documents based on the open directory project (ODP). Then, we incorporate latent dirichlet allocation (LDA) [7] to model the topic distribution of document descriptions. By doing so, we can generate a query distribution over topics via clicked documents.

For evaluation purposes, we compare the performance of PQSD against state-of-the-art query suggestion baselines on the public america online (AOL) query log dataset [8]. In particular, in addition to different personalization strategies with either only clicked queries or all queries in the search context, we also zoom in on the trade-off parameter that controls the contribution of personalization and diversification in our model. We also investigate the sensitivity of our model to the number of query suggestions. The results show the effectiveness of our PQSD model in terms of query suggestion ranking and diversification. In particular, the PQSD model gains an improvement of around 1.35% and 6.39% in terms of MRR and α-nDCG, respectively, over a competitive baseline [4].

Our contributions in this paper can be summarized as follows:

1) We tackle the challenge of query suggestion in a novel way by considering both diversification and personalization.
2) We propose a model for PQSD that incorporates a user's short-term search context in their current session and their long-term search history to detect their search interests.
3) We examine the performance of PQSD under different search context selection strategies and analyze the impact of different trade-off values controlling the personalization and diversification components on the query suggestion performance of our model. We find that PQSD yields better performance when the search context consists of queries with clicked documents rather than all queries, especially when more query suggestions are returned in the list.

We describe related work in Section 2. The details of the personalized query suggestion diversification model, PQSD, are described in Section 3. Section 4 presents our experimen-

[1)] In this paper, we use the terms "aspect" and "topic" interchangeably tal setup. In Section 5, we report and discuss our results. We conclude in Section 6, where we also suggest future research directions.

2 Related work

In recent years, a significant amount of work has gone into methods for obtaining a better understanding of queries submitted by users of a search engine and for improving the quality of the queries that users submit. Prominent examples of the latter include query auto completion [9–11] and query suggestion. Query suggestion is known to be useful for improving the user's search satisfaction [6, 12]. However, there are still some limitations in enhancing the performance of query suggestion lists only using relevance-oriented query suggestion methods [13]. In particular, they cannot handle queries with uncertain search aspects or suggest queries for a specific user. Thus, research has explored several strategies to incorporate either diversification or personalization into query suggestions [2, 4, 14, 15]. In this section, we summarize related work on diversified query suggestion and personalized query suggestion, respectively.

2.1 Personalized query suggestions

Personalized query suggestion methods acquire knowledge of a user's search history in order to reduce the uncertainty of the input query. Many publications are devoted to personalized query suggestion [14, 16–19]. Some provide a list of personalized query suggestions based on information clicked on by a user; here, query log data has been widely used [19]. Verberne et al. [18] implement a method for query suggestion that generates candidate follow-up queries from the documents clicked by the user. This is a potentially effective method for query suggestion, but it heavily depends on user behavior. Based on a user's conceptual profiles, Sharma and Mangla [17] propose a personalized concept-based clustering technique that makes use of click through data and the concept relationship graph mined from web-snippets.

A query-URL bipartite graph can be constructed from click data with one type of vertices corresponding to queries and another type corresponding to URLs. There are also personalized query suggestion methods that use the click graph representing the information flow in query logs with a Markov random walk model [20, 21]. Ma et al. [22] develop a two-level query recommendation method based on two bipartite graphs (user-query and query-URL bipartite graphs) extracted from click data. Li et al. [16] use the connectivity of a query-URL bipartite graph through a novel two-phrase algorithm to recommend relevant queries that can improve the effectiveness of personalized query recommendation. Mei et al. [23] propose a personalized query suggestion method by employing hitting time and creating pseudo query nodes in a click graph.

The personalization component in our approach is different from the work just described as we not only make use of a user's short-term search behavior to predict their search intent in the current session, but also integrate their long-term search history to reduce the uncertainty in the query suggestion list. In addition, we also test different strategies for the personalization when considering all queries or only clicked queries.

2.2 Diversified query suggestions

Modern Web search engines return their query suggestions to a large number of users. As Web search is essentially dynamic and a user's preferences change over time, diversification can help to handle those uncertain changes and generate multiple-aspect queries to increase the likelihood of at least one suggested query being clicked.

Ma et al. [4] propose an approach to query suggestion diversification based on a Markov random walk model on a query-URL bipartite graph that can generate result lists with reduced semantic redundancy. The hitting time $h(q_j \mid q_i)$ in this approach is the expected number of steps used to reach a query vertex q_j from a starting vertex q_i in a bipartite graph. In [4], given an input query q_0, queries q_j with the smallest hitting times $h(q_j \mid q_0)$ are recommended. The weakness of the hitting time approach is that query graphs are huge, which may cause problems in terms of time complexity. Another drawback it has concerns the sparseness problem. Typically, either a depth-first search or a breadth-first search on the query graph [20] is executed to obtain a reduced graph for the execution of the hitting time algorithm. Yang et al. [2] propose a post-ranking framework that aims at maximizing the diversity of the original search results as well as solving the complexity problem. In addition to those methods based on query graphs, Li et al. [3] propose a probabilistic model to recommended queries to avoid redundancy in terms of the concepts covered by suggested queries.

The ambition to combine diversity and personalization opens a rich area for research, one that has barely been explored to date. Vallet and Castells [19] develop a generalization of existing diversification approaches for search results, by adding a personalization component. Their framework suggests that the combination of diversification and personalization achieves competitive performance, improving over the baselines—plain diversification and plain personalization—in terms of both diversity and accuracy measures for search results. Liang et al. [24] deal with the problem of personalized diversification of search results with a supervised learning strategy that also enhances the performance of both plain diversification and plain personalization algorithms. To the best of our knowledge, only few publications study the problem of combining diversification and personalization for query suggestion.

Unlike previous publications that focus on diversification, we propose an explicit approach to obtain query suggestion lists that combines the advantages of both diversification and personalization to improve the performance for query suggestion. In Chen et al. [1], we introduce the PQSD model and quantify the improvement of PQSD against a state-of-the-art baseline. In this extension, we add the following. First, we examine the impact on the model performance introduced by the trade-off parameter λ_2 that controls the contribution of personalization and diversification to the performance of PQSD. Second, we investigate the sensitivity of PQSD model to the number of query suggestions N, as an increased value of N simply increases the probability of including the ground truth in query suggestion list. Third, we cover more related work and provide more detailed analyses of the approach and experimental results.

3 Approach

In this section, we first formally describe the problem of query suggestion diversification and propose a greedy query suggestion diversification model where a user's search context, e.g., queries and clicks, is considered to generate a diversified ranked list of queries in Section 3.1. Then we inject a user's long-term search history to get our proposed PQSD model in Section 3.2. We finally give the generation process of query distribution over topics in Section 3.3.

3.1 Greedy query suggestion diversification

Our method for query suggestion diversification assumes that an initial list of query suggestion candidates R_I produced for the user's query q_0 with length $|R_I| = L_I$ is given. We use a relevant term suggestion method [25] to generate this initial ranked list of queries.

First of all, we simplify the problem of query suggestion diversification. The aim of query suggestion diversification is to satisfy the average user who enters the query q_0 by finding at least one acceptable query suggestion among the top N query suggestions returned. This can be achieved by maximizing the following function:

$$P(R_S \mid q_0, S_C) = 1 - \prod_{q_c \in R_S} (1 - P(q_c \mid q_0, S_C)), \qquad (1)$$

where S_C denotes the search context in a given session of a user who inputs the initial query q_0 and R_S is a ranked list of queries that contains the top N query suggestion candidates to be returned. Obviously, we have $R_S \subseteq R_I$ with $|R_S| = N$, such that $N \leqslant L_I$.

Intuitively, the probability $P(q_c \mid q_0, S_C)$ in Eq. (1) denotes the likelihood that the suggested query candidate q_c satisfies a user who enters query q_0. With the assumption of query independence, the right-hand side of Eq. (1) denotes the probability that at least one query suggestion can satisfy the user. We further interpolate Eq. (1) at the aspect level and thus we have

$$P(R_S \mid q_0, S_C) = \sum_a \left(1 - \prod_{q_c \in R_S} (1 - P(q_c \mid q_0, a, S_C))\right), \qquad (2)$$

where a ranges over possible aspects.

To maximize the objective in Eq. (2), we propose a natural greedy algorithm for generating a diverse ranking of query suggestions. We follow a greedy selection process as follows:

$$q^\star \leftarrow \underset{q_c \in R_I \setminus R_S}{\operatorname{argmax}} \sum_a P(q_c \mid q_0, a, S_C) \prod_{q_s \in R_S} (1 - P(q_s \mid a, q_0, S_C)), \qquad (3)$$

which guarantees that a suggested query that is the most different from previously selected query suggestions in R_S is selected at each step. Thus, it can minimize the redundancy of the ranked list of query suggestions by iteratively filling the list R_S until $|R_S| = N$.

The expression $P(q_c \mid q_0, a, S_C)$ in Eq. (3) is the probability that a query candidate q_c addresses the query aspect a given the input query q_0 and the session context S_C. We estimate this probability based on the following two parts after normalization, with a trade-off λ_1 ($0 \leqslant \lambda_1 \leqslant 1$) controlling the contribution of each part [19]:

$$P(q_c \mid q_0, a, S_C) \leftarrow \lambda_1 P(q_c \mid q_0) + (1 - \lambda_1) P(q_c \mid a, S_C). \qquad (4)$$

Here, $P(q_c \mid q_0)$ denotes the probability that a suggested query q_c is relevant to the input query q_0, which can be estimated by the semantic similarity S_{q_0,q_c} between q_c and q_0, which is weighted by the normalized co-occurrence count C_{q_c,q_0} of q_c and q_0 in search sessions as:

$$P(q_c \mid q_0) \leftarrow C_{q_c,q_0} \cdot S_{q_0,q_c}. \qquad (5)$$

Intuitively, a higher co-occurrence of two queries q_c and q_0 in search sessions would result in a higher relevance probability of q_c and q_0. Following [25], C_{q_c,q_0} can be estimated by

$$C_{q_c,q_0} = \frac{co_{q_c,q_0}}{f_{q_0} + f_{q_c} - co_{q_c,q_0}}, \tag{6}$$

where f_{q_0} and f_{q_c} denote the number of search sessions containing query q_0 and q_c, respectively; co_{q_c,q_0} indicates the number of search sessions containing both query q_c and q_0.

For calculating S_{q_0,q_c}, we take the cosine similarity between two queries, represented by the average of the cosine similarity between query terms w returned by the word2vec model [26] learnt from the query logs, excluding stop words:

$$S_{q_0,q_c} \leftarrow cos(q_0, q_c) = \frac{1}{W} \sum_{w_k \in q_0} \sum_{w_j \in q_c} \cos(w_k, w_j), \tag{7}$$

where $W = |q_0| \cdot |q_c|$ and $|q|$ is the number of query terms in query q.

Turning to the right-hand side of Eq. (4), we make the query independence assumption [27] and decompose $P(q_c \mid a, S_C)$ to obtain:

$$P(q_c \mid q_0, a, S_C) \leftarrow \lambda_1 P(q_c, q_0) + (1 - \lambda_1) \prod_{q_t \in S_C} P(q_c \mid a, q_t). \tag{8}$$

The probability $P(q_c \mid a, q_t)$ in Eq. (8) can be estimated by the distance between query suggestion q_0 and query q_t in the search context given the aspect a. As queries that are submitted within a short temporal interval are bound to share common query aspects [27], we estimate the probability $P(q_c \mid a, q_t)$ as:

$$P(q_c \mid a, q_t) \leftarrow \theta_t \times \left(1 - \frac{|v_{q_c}(a) - v_{q_t}(a)|}{\sqrt{\sum_{i=1}^{M} (v_{q_c}(a_i) - v_{q_t}(a_i))^2}}\right), \tag{9}$$

where $\theta_t = \frac{1}{D(q_t)+1}$ and $D(q_t)$ refers to the position interval between previous query q_t and the last query q_T in the search context S_C; for example, $\theta_T = 1$ for the last query in the search context. Furthermore, M denotes the number of aspects of a query and $v_{q_c}(a_i)$ denotes the relevance of query q_c to its ith aspect. This explains how the term $P(q_c \mid q_0, a, S_C)$ in Eq. (3) can be estimated.

Next, for calculating $P(q_s \mid q_0, a, S_C)$ in Eq. (3), which denotes the probability of query suggestions that have been chosen in the list R_R addressing query aspect a given the search context S_C and input query q_0, based on the query independence assumption we can simplify $P(q_s \mid q_0, a, S_C)$ in Eq. (3) as:

$$P(q_s \mid q_0, a, S_C) \leftarrow P(q_s \mid a, S_C) = \prod_{q_t \in S_C} P(q_s \mid a, q_t), \tag{10}$$

where $P(q_s \mid a, q_t)$ is computed analogously to $P(q_c \mid a, q_t)$ in Eq. (9).

3.2 Personalized query suggestion diversification

In this section, we generalize the greedy selection rule to a personalized version by considering a user u's long-term search history so that $q^\star$ becomes:

$$q^\star \leftarrow \operatorname*{argmax}_{q_c \in R_I \setminus R_S} \sum_a P(q_c \mid q_0, a, S_C, u) \prod_{q_s \in R_S} (1 - P(q_s \mid a, q_0, S_C, u)). \tag{11}$$

Let us explain the model in more detail. For calculating $P(q_c \mid q_0, a, S_C, u)$, the first term on the right-hand side of Eq. (11), we use Bayes' rule:

$$P(q_c \mid q_0, a, S_C, u) = \frac{P(q_c) P(a, u, q_0, S_C \mid q_c)}{P(a, u, q_0, S_C)}. \tag{12}$$

We rewrite the term $P(a, u, q_0, S_C \mid q_c)$, which can be regarded as the combination of diversification and personalization, as:

$$P(a, u, q_0, S_C \mid q_c) \leftarrow \lambda_2 P(a, q_0, S_C \mid q_c) + (1 - \lambda_2) P(u, q_0, S_C \mid q_c), \tag{13}$$

where λ_2 ($0 \leqslant \lambda_2 \leqslant 1$) in Eq. (13) is a tradeoff controlling the contributions of diversification and personalization, respectively. Before producing the final score $P(a, u, q_0, S_C \mid q_c)$, we normalize the scores of $P(a, q_0, S_C \mid q_c)$ and $P(u, q_0, S_C \mid q_c)$, respectively. Based on Bayes' rule, $P(a, q_0, S_C \mid q_c)$ and $P(u, q_0, S_C \mid q_c)$ can be interpolated as

$$P(a, q_0, S_C \mid q_c) = \frac{P(q_c \mid a, q_0, S_C) P(a, q_0, S_C)}{P(q_c)} \tag{14}$$

and

$$P(u, q_0, S_C \mid q_c) = \frac{P(q_c \mid u, q_0, S_C) P(u, q_0, S_C)}{P(q_c)}, \tag{15}$$

respectively. The term $P(q_c \mid a, q_0, S_C)$ in Eq. (14) can be calculated following Eq. (8). Following the independence assumption used in Web search [19], we approximate $P(q_c \mid u, q_0, S_C)$ in Eq. (15) as

$$P(q_c \mid u, q_0, S_C) \propto \prod_{q_t \in S_C} P(q_c \mid u) P(q_c \mid q_0) P(q_c \mid q_t), \tag{16}$$

where $P(q_c \mid u)$ denotes the probability of suggesting q_c to user u according to their long-term search history and can be estimated as:

$$P(q_c \mid u) \leftarrow \frac{\sum_{q \in Q(u)} S_{q_c,q}}{|Q(u)|}, \tag{17}$$

where $Q(u)$ are all queries that user u has submitted and $|Q(u)|$ is the size of $Q(u)$. In addition, $S_{q_c,q}$ returns the semantic similarity between two queries like Eq. (7).

Similarly, for $P(q_s \mid a, q_0, S_C, u)$, the second term on the right-hand side of Eq. (11), based on the query independence assumption mentioned above and Bayes' rule, we can get the diversification and personalization components as follows:

$$P(a, q_0, S_C \mid q_s) = \frac{P(q_s \mid a, q_0, S_C)P(a, q_0, S_C)}{P(q_s)} \tag{18}$$

and

$$P(u, q_0, S_C \mid q_s) = \frac{P(q_s \mid u, q_0, S_C)P(u, q_0, S_C)}{P(q_s)}, \tag{19}$$

where $P(q_s \mid a, q_0, S_C)$ in Eq. (18) can be realized as Eq. (10), and $P(q_s \mid u, q_0, S_C)$ in Eq. (19) can be derived in the same way as $P(q_c \mid u, q_0, S_C)$ in Eq. (16).

We have now introduced the main process of our personalized query suggestion diversification model. Clearly, as shown in Algorithm 1, we first initialize the query suggestion list R_S with $q^\star$ having the maximum value of $P(q_c \mid q_0, a, S_C, u)$ from step 2 to 6. Then, with a greedy selection strategy from step 8 to 15, we iteratively fill the list R_S until $|R_S| = N$. In step 10 and 12, we guarantee that a newly suggested query added into R_S is maximally different from previously selected query suggestions in R_S and is relevant to the input query q_0. In the following section, we show how to generate the query distribution over topics in detail.

Algorithm 1 PQSD

Input: Input query q_0, an initial query suggestion list R_I, size of returned query suggestion list: N, search context S_C, long-term search history of a user u

Output: A reranked query suggestion list R_S;

1: $R_S = \emptyset$
2: **for** each candidate $q_c \in R_I$ **do**
3:     $FirstQuery(q_c) \leftarrow P(q_c \mid q_0, a, S_C, u)$; %% the first query suggestion
4: **end for**
5: $q^\star \leftarrow \arg\max_{q_c \in R_I} FirstQuery(q_c)$
6: $R_S \leftarrow R_S \cup \{q^\star\}$
7: $R_I \leftarrow R_I \backslash \{q^\star\}$
8: **for** $|R_S| \leqslant N$ **do**
9:     **for** $q_c \in R_I$ **do**
10:         $s(q_c) \leftarrow \sum_a P(q_c \mid q_0, a, S_C, u) \prod_{q_s \in R_S} (1 - P(q_s \mid a, q_0, S_C, u))$
11:     **end for**
12:     $q^\star \leftarrow \arg\max_{q_c} s(q_c)$
13:     $R_S \leftarrow R_S \cup \{q^\star\}$
14:     $R_I \leftarrow R_I \backslash \{q^\star\}$
15: **end for**
16: **return** R_S

3.3 Generating query distribution over topics

In the PQSD model, a key problem is how to represent queries over topics. As queries are usually short, it makes sense to use clicked documents to generate their topic distribution rather than using the queries directly [27]. In our method, we generate query distribution through three steps.

First, we extract clicked documents from the query log and collect the corresponding description texts in ODP for each URL. Specifically, we use the first two levels in a URL as the matching context. The clickthrough data is produced from the search behavior of real searchers and has been proved effective for estimating the relevance of a document to the corresponding query [28].

The second step is generating the topic distribution of documents using LDA. LDA has been shown to be a highly effective unsupervised learning methodology for finding distinct topics in document collections. It is a generative process that models each document as a mixture of topics. Each topic contains several words and corresponds to a multinomial distribution over those words. Then LDA can learn the document-topic and topic-word distribution after training and return the topic distribution of each document and the word distribution of each topic [7].

After that, we finally obtain a query q's topic distribution as:

$$v_q = \sum_{d \in D(q)} v_d \times f(q, d), \tag{20}$$

where $D(q)$ is the set of documents clicked in response to query q, v_d denotes the topic distribution of document d, which is vectorized using LDA, and $f(q, d)$ indicates the number of clicks on d after submitting q.

For queries without clicked documents, we generate the query distribution from similar queries that have been vectorized as semantically related queries (or words) often express similar search topics [29]. We find the most similar vectorized query q_{vector} for a query q_{nc} without clicks by

$$q_{vector} \leftarrow \underset{q_v \in Q_v}{\mathrm{argmax}} \cos(q_{nc}, q_v), \tag{21}$$

where Q_v is a set of vectorized queries. We take the cosine similarity between two queries as in Eq. (7).

The details are shown in Algorithm 2: we select the most similar query for q_{nc} (line 4), from which we obtain the vector of topic distribution that are finally assigned to the input query q_{nc} as aspect labels (line 5).

Algorithm 2 Dealing with query q_{nc} without clicks

Input: A query q_{nc} without click information, a set of vectorized queries Q_v with their vectors V

Output: Vector of q_{nc}: $v_{q_{nc}}$;

```
for each query q_v ∈ Q_v do
    score(q_v) = cos(q_nc, q_v)    %% semantic similarity
end for
q_vector ← argmax_{q_v∈Q_v} score(q_v)
v_{q_nc} ← v_{q_vector} ∈ V
return v_{q_nc} to q_nc
```

4 Experimental setup

We start by providing an overview of the query suggestion models to be discussed in this paper and lists the research questions that guide our experiments. Then we describe the dataset and give details about our evaluation metrics as well as the ground truth. We conclude the section by specifying the settings of the parameters in our experiments.

4.1 Model summary and research questions

Table 1 lists the models to be discussed: two state-of-the-art baselines, two models considering either diversification or personalization, and four flavors of approaches that we introduce in this paper: PQSD models with four combination strategies of user's selecting search context:

- a user's current search context, with two options:
  AS all preceding queries in current search session vs.
  CS only the clicked queries in current search session,
- and a user's long-term search history, again with two options:
  AL all preceding queries in user's search history, vs.
  CL only the clicked queries in user's search history.

The research questions guiding our experiments are:

RQ1 Is the PQSD model able to beat state-of-the-art query suggestion models in terms of query suggestion ranking and diversification?

RQ2 What is the impact on the query suggestion diversification performance of PQSD of the choice of search context, i.e., choosing all queries (AS and AL) or only queries with clicks (CS and CL)?

RQ3 How does the trade-off parameter between diversification and personalization (as encoded in λ_2) impact the performance of our PQSD model in terms of query suggestion ranking and diversification?

RQ4 Is the performance of our PQSD model sensitive to the number of query suggestions N?

4.2 Dataset and evaluation metrics

We use the AOL query log [8] in our experiments and preprocess the dataset following [31]. The AOL queries were sampled between March 1st, 2006 and May 31st, 2006. For the preprocessing of the data, we only keep those frequent well-formatted English queries, which appear more than 4 times and only contain characters "a", "b", . . . , "z" as well as space. In addition, we split the queries into sessions by 30 minutes of inactivity and sessions with at least two queries are kept. To obtain our training and test sets, we remove queries for which the ground truth is not included in the top fifteen query suggestion candidates returned by a co-occurrence method [25].

We notice that users often submit several queries before clicking a URL. When a user submits a query that is followed by clicking a URL, we call this query a *clicked query*. Intuitively, the user may be more satisfied with a clicked query than with queries without clicks. Thus we remove the sessions without clicked queries in the preprocessing. Table 2

Table 1 An overview of models discussed in the paper

Model	Description	Source
MMR	A query suggestion diversification approach based on Maximal Marginal Relevance (MMR).	[30]
DQS	A diversification-oriented query suggestion model based on Markov random walk and hitting time analysis on the query-URL bipartite graph.	[4]
D-QS	A query suggestion approach that only considers diversification purpose.	This paper
P-QS	A query suggestion approach that only considers personalization purpose.	This paper
$PQSD_{AL+AS}$	Personalized diversification query suggestion model incorporating all queries in a user's long-term search history and in the current session.	This paper
$PQSD_{AL+CS}$	Personalized diversification query suggestion model incorporating all queries in a user's long-term search history and only queries with clicks in the current session.	This paper
$PQSD_{CL+AS}$	Personalized diversification query suggestion model incorporating only queries with clicks in a user's long-term search history and all preceding queries in the current session.	This paper
$PQSD_{CL+CS}$	Personalized diversification query suggestion model incorporating only queries with clicks in a user's long-term search history and in the current session.	This paper

details the statistics of the dataset used.

Table 2 Dataset statistics

Variables	Training	Test
# Queries	7,256,569	2,628,284
# Unique queries	746,796	373,397
# Sessions	1,428,962	714,481
# Users	220,946	110,473
# Average queries with clicks per session	4.37	4.35
# Average queries with clicks per user	28.87	28.91

To evaluate the effectiveness of query suggestion ranking, Mean Reciprocal Rank (MRR) [32] is a standard measure. Let q be a query the query set Q associated with a list of query suggestion candidates R_S and assume that the user submitted q' as input; then, the Reciprocal Rank (RR) is computed as:

$$RR = \begin{cases} \frac{1}{\text{rank of } q' \text{ in } R_S}, & \text{if } q' \in R_S, \\ 0, & \text{else.} \end{cases} \tag{22}$$

MRR is computed as the mean of RR for all queries in Q.

As for diversification, we use the α-nDCG metric [33], which extends the traditional nDCG metric [34] in the following way for aspect-specific rankings:

$$\alpha\text{-}nDCG@N = Z_N \sum_{i=1}^{N} \frac{\sum_{a \in A_p} g_{i|a} (1-\alpha)^{\sum_{j=1}^{i-1} g_{j|a}}}{\log_2(i+1)}. \tag{23}$$

In Eq. (23), a denotes a topic in the set of query topics A_p, $g_{i|a}$ means the topic-specific gain of the ith query given topic a. And Z_N is a normalization constant to ensure that the best query suggestion list can achieve α-nDCG = 1. The parameter α is a trade-off controlling the weights of both relevance and diversity that is commonly set as $\alpha = 0.5$, thus treating them equally.

For generating the ground truth, i.e., the relevance of a query q to an aspect a, we follow [35] and use a 5-grade scale (perfect = 4, excellent = 3, good = 2, fair = 1, and bad = 0) as:

$$rel_{q,a} \leftarrow \min(\lfloor v_q(a) \rfloor, 4). \tag{24}$$

We use MRR and α-nDCG to measure the ranking and diversification performance of query suggestions. Statistical significance of differences between the performance of two approaches is tested using a t-test, which is denoted using $^\blacktriangle/^\blacktriangledown$ for $\alpha = .01$, or $^\vartriangle/^\triangledown$ for $\alpha = .05$.

4.3 Parameter setup

For the parameters in our experiments, we use the following settings. Following [19], we fix $\lambda_1 = 0.5$. In the LDA model, following [36], we set the number of topics $M = 100$, and the distribution parameters $\alpha = 0.5$ and $\beta = 0.1$.

Recall that λ_2 in Eq. (14) controls the contribution of personalization and diversification components in the PQSD models. We aim to analyze the impact of it on the performance of our model by manually changing it from 0 to 1 with steps of 0.1. We set $\lambda_2 = 0.5$ to give equal weight to diversification and personalization when comparing the performance between our models with the baselines.

As for the number of query suggestions N, we set $N = 10$ when comparing the performance between our models with the baseline models, which is commonly used [2]. In experiments aimed at assessing the impact of parameter tuning, we investigate the sensitivity of the PQSD model to N in terms of MRR and α-nDCG.

5 Results and discussion

We begin by comparing the performance of all models mentioned above in terms of precision and diversification of query rankings. We then detail the effect of different choices for search context. After that we analyze the effect of the parameter λ_2 in our proposed PQSD model. Finally, we examine how the models perform when more (or fewer) query suggestions are returned by varying the cutoff N.

5.1 Performance of query suggestion models

To answer **RQ1**, we examine the query suggestion performance of all presented models and include the results in Table 3.

Table 3 Performance of query suggestion models. The results produced by the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences (PQSD models vs. best baseline) determined by a t-test ($^\blacktriangle/^\blacktriangledown$ for $\alpha = .01$, or $^\vartriangle/^\triangledown$ for $\alpha = .05$)

Models	MRR@10	α-nDCG@10
MMR	.6611	.7021
DQS	<u>.6672</u>	<u>.7152</u>
D-QS	.6698	.7401
P-QS	.6685	.7276
PQSD_{AL+AS}	$.6726^\vartriangle$	$.7461^\vartriangle$
PQSD_{CL+AS}	$.6763^\blacktriangle$	$.7644^\blacktriangle$
PQSD_{AL+CS}	$.6756^\blacktriangle$	$.7686^\blacktriangle$
PQSD_{CL+CS}	$\mathbf{.6807}^\blacktriangle$	$\mathbf{.7791}^\blacktriangle$

The DQS model achieves a better performance than the MMR model in terms of MRR@10 and α-nDCG@10. Hence, we only use DQS as the baseline for comparisons in latter experiments. DQS shows a minor improvement in terms of MRR@10 over MMR (<1.0%) and a somewhat bigger improvement in terns of α-nDCG@10 over MMR (<1.9%).

For the models that consider either diversification or personalization, they both have better performance than the DQS approach. In particular, the D-QS model performs better than P-QS in terms of α-nDCG@10 and has a slightly higher value of MRR@10 than the P-QS model. However, they both lose against the PQSD model in terms of MRR@10 and α-nDCG@10, which indicates that the combination of diversification and personalization does help to improve query suggestion ranking and diversification performance.

Regarding the PQSD models, whatever type of search context is considered, PQSD achieves a better performance than the DQS baseline, resulting in MRR@10 improvements ranging from 0.8% to 2.0% and α-nDCG@10 improvements ranging from 4.3% to 8.9%. The fact that improvements in α-nDCG@10 are higher than the improvements in MRR@10 can be explained by the fact that in some cases, redundant query suggestions ranked lower than the final submitted query are removed from the original query suggestion list; this does not affect the reciprocal rank score but does result in improved diversity scores.

We can see from Table 3 that PQSD$_{CL+CS}$ achieves the best performance. Significant improvements against the baseline in terms of MRR@10 and α-nDCG@10 are observed for all PQSD models at the $\alpha = .01$ level except for PQSD$_{AL+AS}$, for which we observe significant improvements at the $\alpha = .05$ level. Hence, the content of the search context does affect the performance of our PQSD model, which motivates us to conduct a further investigation to answer **RQ2**.

5.2 Effect of different personalization strategies

For **RQ2** we fix the search context by using either all previous queries or only queries with clicks in the current session as well as the user's long-term search history. In general, PQSD achieves a better performance when it incorporates queries with clicks as search context than when using all previous queries. e.g., as shown in Table 3, PQSD$_{CL+AS}$ beats PQSD$_{AL+AS}$ in terms of both metrics. Similar results can be found when comparing PQSD$_{CL+CS}$ to PQSD$_{AL+CS}$. Hence, queries with clicks more accurately express a user's search intent, which is helpful for query suggestion personalization; the use of all queries as search context for personalization brings noise when detecting a user's real search intent.

Results of the PQSD models and the baseline at different query positions (in a session) are shown in Fig. 1. As shown in Fig. 1(a), as the search context becomes richer, the performance in terms of MRR@10 of all query suggestion models improves, e.g., at a late query position in a session (> 4), PQSD$_{CL+CS}$ improves MRR@10 over earlier query positions (= 2). In addition, as indicated by the results of the PQSD models at the start of a session (query position = 1), when a user's short-term search context in the current session is unavailable, PQSD achieves negligible improvements over the baseline, especially for PQSD$_{AL+AS}$ and PQSD$_{AL+CS}$.

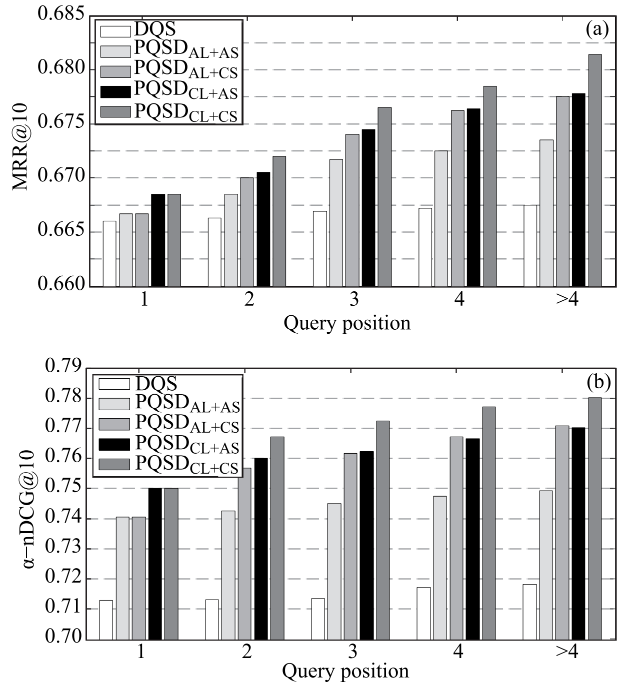


Fig. 1 Performance of PQSD models and the baseline at different query positions in a session. (a) Performance in terms of MRR@10; (b) performance in terms of α-nDCG@10

Regarding the evaluation of diversity, similar results can be found in Fig. 1(b) when reporting the performance of the query suggestion models in terms of α-nDCG@10. PQSD achieves larger improvements over the baseline in terms of α-nDCG@10 than in terms of MRR@10 at each query position, which is consistent with the findings reported in Table 3. To sum up, search contexts consisting of queries with clicks, whether in a user's long-term or short-term search history, can help generate more accurate and diversified query suggestion rankings.

5.3 Effect of the trade-off parameter λ_2

Next, we turn to **RQ3** and conduct a parameter sensitivity analysis of our PQSD models. We examine the performance of our PQSD models in terms of MRR@10 and α-nDCG@10 by gradually changing the parameters λ_2 from 0 to 1 with an interval 0.1. We plot the results in Fig. 2.

For any value of λ_2, PQSD$_{CL+CS}$ always performs best among the four models in terms of both MRR@10 and α-

nDCG@10. Another interesting finding that can be observed is that the PQSD$_{CL+AS}$ model loses against the PQSD$_{AL+CS}$ model in terms of α-nDCG@10. However, it outperforms the PQSD$_{AL+CS}$ model in terms of MRR@10. This indicates that a user's long-term search history can help to yield a better MRR@10 score especially with clicked information, while the search context in the current session with clicked queries is more helpful to improve the performance of our model in terms of α-nDCG@10.

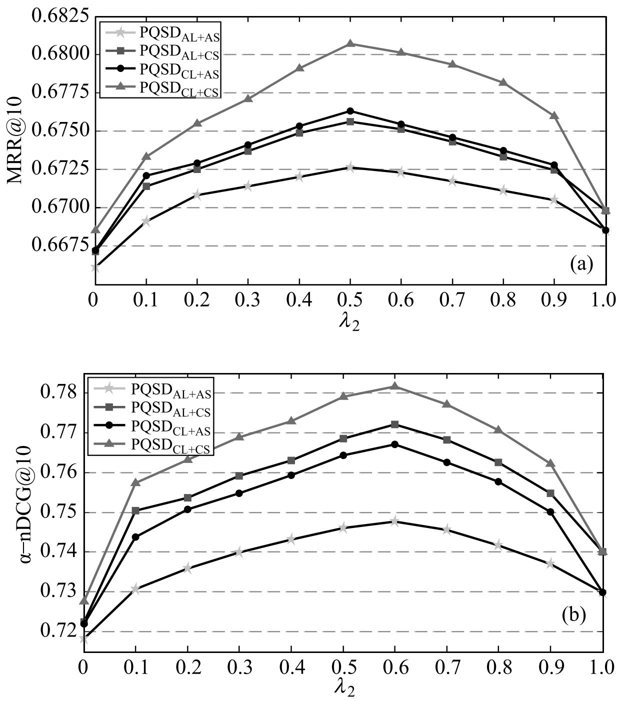


Fig. 2 Effect on performance of PQSD models in terms of MRR@10 and α-nDCG@10 by changing the trade-off parameter λ_2, tested on the AOL log. (a) Performance in terms of MRR@10; (b) performance in terms of α-nDCG@10

In particular, as shown in Fig. 2(a), we can see that the MRR@10 scores of all PQSD models increase consistently when λ_2 varies from 0 to 0.5; after that, the MRR@10 scores go down when λ_2 changes from 0.5 to 1. In addition, for any PQSD model, if it only focuses on personalization, i.e., $\lambda_2 = 0$, its performance is relatively worse than model that combine diversification and personalization for query suggestion, i.e., with values of λ_2 strictly in between 0 and 1. Specifically, a noticeable increase is observed when λ_2 changes from 0 to 0.1 in terms of MRR@10 performance, which means that integrating diversification does help to improve the ranking accuracy for query suggestion in our models.

Regarding query suggestion diversification, as shown in Fig. 2(b), for all PQSD models, their peak performance appears near $\lambda_2 = 0.6$. A sharp increase is observed when λ_2 changes from 0 to 0.1 in terms of α-nDCG@10, e.g., there is a 4.1% improvement for the PQSD$_{CL+CS}$ model which is the most significant fluctuation in Fig. 2(b). This shows that the greedy selection diversification model does help to generate multiple-aspect queries. In addition, when λ_2 changes from 0.6 to 1, the α-nDCG@10 scores of four PQSD models monotonically decline. This indicates that the personalization component in our PQSD model has a positive influence on the performance of our model in terms of α-nDCG@10.

From the observations in Fig. 2, we can conclude that: 1) our PQSD model with a combination of diversification and personalization shows better performance for query suggestion than a model that incorporates either personalization or diversification but not both; 2) λ_2 has a bigger influence on α-nDCG@10 than on MRR@10; for instance, in Fig. 2(b) we see that for the PQSD$_{CL+CS}$ model, there is a 1.8% improvement from the smallest value, i.e., $\lambda_2 = 0$ to the biggest, i.e., $\lambda_2 = 0.5$ in term of MRR@10; however, regarding the value of α-nDCG@10, the improvement is around 7.4% from the smallest ($\lambda_2 = 0$) to the biggest ($\lambda_2 = 0.6$).

5.4 Zooming in on the cut-off *N*

For research question **RQ4**, we examine the performance of our four PQSD models and the baseline model when less (or more) query suggestions are returned by varying the cut-off N from 5 to 15. We show the MRR and α-nDCG scores in Fig. 3 as tested on the america online (AOL) log, as before.

The overall performance in terms of MRR and α-nDCG increases when more query suggestions are returned for reranking. A large value of N increases the probability of including a user's intended query, i.e., the ground truth, in the query suggestion list. In addition, the same result can be found in Fig. 3 as we observe in Fig. 2, i.e., the MRR value of PQSD$_{CL+AS}$ is better than PQSD$_{AL+CS}$; however, in terms of α-nDCG, PQSD$_{CL+AS}$ shows worse performance than PQSD$_{AL+CS}$. More specifically, for a specific number of query suggestions, our PQSD models beat the baseline in terms of both MRR and α-nDCG. This indicates that the combination of personalization and diversification in the PQSD models has a positive effect on pushing the ground truth up in the list of query suggestions. As shown in Fig. 3(a), the best result is returned by the PQSD$_{CL+CS}$ model. Similar results can be found when comparing those models in terms of α-nDCG, as shown in Fig. 3(b).

With an increase in the number of query suggestions, the MRR improvements achieved by our PQSD models over the baseline are further magnified, as shown in Fig. 3(a). For instance, PQSD$_{CL+CS}$ model presents a 1.4% MRR improvement over the baseline at $N = 5$, a 2.0% improvement at $N = 10$, and a 3.2% improvement at $N = 15$.

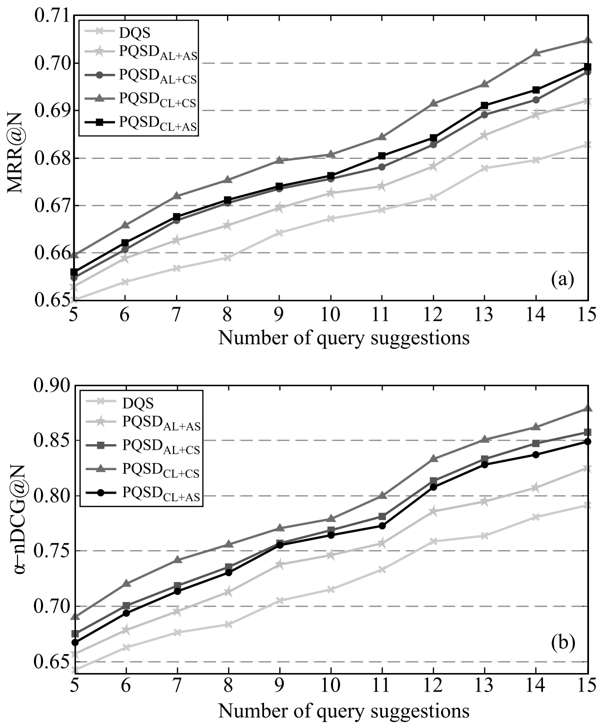


Fig. 3 Effect on performance of five models in terms of MRR and α-nDCG when more (or less) query suggestion candidates are returned, tested on the AOL log. (a) Performance in terms of MRR@N; (b) performance in terms of α-nDCG@N

Regarding query diversification, the improvements of the PQSD models are more significant in terms of α-nDCG ($N = 5$, 10 and 15) than MRR, as indicated by the relative improvements over the baseline. For instance, in Fig. 3(b), PQSD_{CL+CS} shows a 7.4% improvement over the baseline in terms of α-nDCG at cutoff $N = 5$, a 8.9% improvement at $N = 10$ and a 10.3% improvement at $N = 15$. This can be attributed to the fact that when more candidates are returned, more query redundancy is introduced into the list of query suggestions, leaving a relatively larger room for our PQSD models to improve the performance against the baseline in terms of α-nDCG.

6 Conclusions and future work

We have dealt with the task of combining personalization and diversification of query suggestions. We have proposed a personalized query suggestion diversification model, PQSD, based on a greedy selection algorithm that incorporates a user's previous queries as search context for personalization.

Our experimental results show that: 1) the combination of diversification and personalization does help boost the query suggestion performance in terms of precision and diversification of query rankings; 2) a variant of our PQSD model using queries with clicks achieves the best performance in terms of query ranking accuracy and diversification; 3) the advantages of our PQSD model over the baseline become more prominent when more query suggestions are returned.

Together, our findings make an important step beyond prior work on query suggestion. Prior to our work, the combination of personalization and diversification had already given rise to improvements of query auto completion. Now, query suggestion methods can be personalized as well as diversified too, allowing us to help users formulate their information needs in a more effective manner.

As to limitations of this work, we have implemented our PQSD model through injecting a user's long-term search history into a basic greedy query suggestion diversification model. There are many other strong signals for personalization which we do not consider, such as user profiles and time sensitivity. Also, we only examine our models on the AOL dataset, where we generate the relevance labels automatically. We should test our PQSD model on other datasets.

As future work, we plan to evaluate our models on other datasets so as to verify their effectiveness. We would like to investigate the merits of Web search result diversification [12, 37] on the task of query suggestion diversification. And we want to investigate other personalization strategies such as user profiles or behavior-based personalization, which has been shown to help improve effectiveness [38, 39]. We also want to have a closer look at the effect of different topic numbers have on the performance of our models. Can we expand the combination of personalization and diversification to other scenarios, with different modes of interaction?

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. 61702526), the National Advanced Research Project (6141B0801010b), Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Criteo Faculty Research Award program, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research PhD program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs (612.001.116, HOR-11-10, CI-14-25, 652.-002.001, 612.001.551, 652.001.003), and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

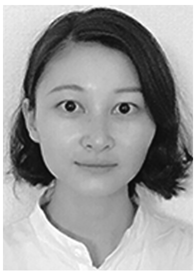
References

1. Chen W Y, Cai F, Chen H H, De Rijke M. Personalized query suggestion diversification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017, 817–820

2. Yang S, Zhou D Y, He L W. Post-ranking query suggestion by diversifying search results. In: Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2011, 815–824
3. Li R R, Kao B, Bi B, Cheng R, Lo E. DQR: a probabilistic approach to diversified query recommendation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. 2012, 16–25
4. Ma H, Lyu M R, King I. Diversifying query suggestion results. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. 2010, 1399–1404
5. Zhang Z Y, Nasraoui O. Mining search engine query logs for query recommendation. In: Proceedings of the 15th International Conference on World Wide Web. 2006, 1039–1040
6. Cao H H, Jiang D X, Pei J, He Q, Liao Z, Chen E H, 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. 2008, 875–883
7. Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3(4): 993–1022
8. Pass G, Chowdhury A, Torgeson C. A picture of search. In: Proceedings of the 1st International Conference on Scalable Information Systems. 2006, 1–7
9. Cai F, De Rijke M. A survey of query auto completion in information retrieval. Foundations and Trends in Information Retrieval, 2016, 10(4): 273–363
10. Cai F, Liang S S, De Rijke M. Prefix-adaptive and time-sensitive personalized query auto completion. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2452–2466
11. Cai F, De Rijke M. Learning from homologous queries and semantically related terms for query auto completion. Information Processing and Management, 2016, 52(4): 628–643
12. Santos R L T, Peng J, Macdonald C, Ounis I. Explicit search result diversification through sub-queries. In: Proceedings of the 32nd European Conference on Information Retrieval. 2010, 87–99
13. Al-otaibi S, Ykhlef M. Hybrid immunizing solution for job recommender system. Frontiers of Computer Science, 2017, 11(3): 511–527
14. Kharitonov E, Macdonald C, Serdyukov P, Ounis I. Intent models for contextualising and diversifying query suggestions. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management. 2013, 2303–2308
15. Ziegler C N, McNee S M, Konstan J A, Lausen G. Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web. 2005, 22–32
16. Li L, Yang Z L, Liu L, Kitsuregawa M. Query-URL bipartite based approach to personalized query recommendation. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence. 2008, 1189–1194
17. Sharma S, Mangla N. Obtaining personalized and accurate query suggestion by using agglomerative clustering algorithm and P-QC method. International Journal of Engineering Research and Technology, 2012, 1(5): 28–35
18. Verberne S, Sappelli M, Järvelin K, Kraaij W. User simulations for interactive search: evaluating personalized query suggestion. In: Proceedings of the 2015 European Conference on Information Retrieval. 2015, 678–690
19. Vallet D, Castells P. Personalized diversification of search results. In: Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2012, 841–850
20. Craswell N, Szummer M. Random walks on the click graph. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2007, 239–246
21. Cui J W, Liu H Y, Yan J, Ji L, Jin R M, He J, Guo Y Q, Chen Z, Du X Y. Multi-view random walk framework for search task discovery from click-through log. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. 2011, 135–140
22. Ma H, Yang H X, King I, R. Lyu M. Learning latent semantic relations from clickthrough data for query suggestion. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. 2008, 709–718
23. Mei Q Z, Zhou D, Church K. Query suggestion using hitting time. In: Proceedings of the 17th ACM International Conference on Information and Knowledge Management. 2008, 469–478
24. Liang S S, Cai F, Ren Z C, de Rijke M. Efficient structured learning for personalized diversification. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(11): 2958–2973
25. Huang C K, Chien L F, Oyang Y J. Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society for Information Science and Technology, 2003, 54(7): 638–649
26. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of Workshop at International Conference on Learning Representations. 2013, 1–13
27. Cai F, Ridho R, De Rijke M. Diversifying query auto-completion. ACM Transactions on Information Systems, 2016, 34(4): 1–33
28. Joachims T. Optimizing search engines using clickthrough data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002, 133–142
29. Bollegala D, Matsuo Y, Ishizuka M. Measuring semantic similarity between words using Web search engines. In: Proceedings of the 16th International Conference on World Wide Web. 2007, 757–766
30. Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1998, 335–336
31. Guo J F, Cheng X Q, Xu G, Zhu X F. Intent-aware query similarity. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. 2011, 259–268
32. Shah C, Croft W B. Evaluating high accuracy retrieval techniques. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2004, 2–9
33. Clarke C L A, Kolla M, V. Cormack G, Vechtomova O, Ashkan A, Büttcher S, MacKinnon I. Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2008, 659–666
34. Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems, 2002, 20(4): 422–446
35. Chapelle O, Metzler D, Zhang Y, Grinspan P. Expected reciprocal rank

for graded relevance. In: Proceedings of the 18th ACM International Conference on Information and Knowledge Management. 2009, 621–630

36. Asuncion A, Welling M, Smyth P, Teh W Y. On smoothing and inference for topic models. In: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence. 2009, 27–34
37. Agrawal R, Gollapudi S, Halverson A, Ieong S. Diversifying search results. In: Proceedings of the 2009 International Conference on Web Search and Data Mining. 2009, 5–14
38. Cai F, Wang S Q, De Rijke M. Behavior-based personalization in Web search. Journal of the Association for Information Science and Technology, 2017, 68(4): 855–868
39. Sepliarskai A, Radlinski F, De Rijke M. Simple personalized search based on long-term behavioral signals. In: Proceedings of the 39th European Conference on Information Retrieval. 2017, 95–107



Wanyu Chen is a master student at the National University of Defense Technology, China. Her research interests include in query suggestion and information retrieval. She got her bachelor degree at the National University of Defense Technology majoring in System Engineering, China in 2015. She has published a SIGIR paper in 2017.

Fei Cai is an assistant professor at the National University of Defense Technology, China. He got his Doctor degree on Computer Science from the University of Amsterdam, The Netherlands under the supervision of Prof. Maarten de Rijke. His research interests include information retrieval and query formulation. He has several papers published in SIGIR, CIKM, FnTIR, TOIS, TKDE, etc. In addition, he serves as a PC member for CIKM and WSDM as well as a reviewer for SIGIR, WWW, WSDM, CIKM, TKDE, IPM, JASIST, etc.

Honghui Chen is a professor at the National University of Defense Technology, China. He got his Doctor degree on Operational Research from the National University of Defense Technology, China in 2007. His research interests include information system and information retrieval. He has published serval papers at SIGIR, IPM and other top journals.

Maarten de Rijke is a professor of computer science in the Informatics Institute at the University of Amsterdam, The Netherlands. He is a member of the Royal Netherlands Academy of Arts and Sciences. His research focus is on intelligent information access, with projects on self-learning search engines and semantic search. He is the Editor-in-Chief of ACM Transactions on Information Systems and of Foundations and Trends in Information Retrieval. De Rijke has published over 700 papers.