Integrated Personalized and Diversified Search Based on Search Logs

Jiongnan Liu, Zhicheng Dou, *Member, IEEE*, Jian-Yun Nie, *Member, IEEE*, and Ji-Rong Wen, *Senior Member, IEEE*

***Abstract*— Personalized search and search result diversification are two possible solutions to cope with the query ambiguity problem in search engines. In most existing studies, they have been investigated separately, but intuitively, they address the problem from two complementary perspectives and should be combined. Some recent work tried to combine them by restricting result diversification to the subtopics corresponding to the user's personal profile. However, diversification can be required even when the subtopics are outside the user's profile. In this paper, we propose a more general approach to integrate them based on users' implicit feedback in query logs. The proposed approach PER+DIV aggregates a document's novelty score and personal relevance score dynamically according to how much the query falls into the user's interests. To train the model based on user clicks in the logs, we consider user click as a result of both personal relevance and result diversity and a new method is proposed to isolate and model these two factors. To evaluate the model, we design several diversified and personalized metrics in addition to the traditional click-based metrics. Experimental results on a large-scale query log dataset show that the proposed integrated method significantly outperforms the existing personalization and diversification approaches.**

***Index Terms*—Integration, personalized search, search result diversification.**

I. Introduction

STUDIES have shown that many queries issued to search engines by users are broad or ambiguous [1], [2]. Different users may intend to retrieve different information with the same query. For example, the query "apple" may be used to search for information about `Apple Inc.` or the `fruit apple`. To provide better search results for ambiguous queries, two main approaches have been proposed: search result diversification [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] and personalized search [14], [15], [16], [17], [18], [19], [20], [21], [22]. Search result diversification tries to provide a list of documents covering all subtopics related to the query so that all users can find relevant documents from the top ranked results. On the other hand, personalized search aims to directly identify the user's personalized intent. It creates a user profile through her search history and returns a ranked list corresponding to her interests. Both approaches try to solve the same problem (i.e. query ambiguity) from different perspectives. However, they have been mostly studied separately in the past.

There are a few exceptions, which proposed approaches to combine personalization and diversification. For example, Radlinski et al. [23] proposed to use similar queries for a specific user to address the diversification problem. Vallet et al. [24] and Liang et al. [25] proposed methods for personalized diversification of results using probability estimation and structured learning. However, all these approaches focused on the problem of personalized diversification, i.e. to make result diversification more consistent with the user's interests. In particular, one first determines the subtopics corresponding to the user's interests, and the results are selected to cover these subtopics. While such approaches can be useful in some circumstances (the user is only interested in documents related to her interests), in a general search context, the search intents of a user are much broader than her known interests - users frequently explore new topics in search. For such search queries, the above approaches may wrongly bias the results toward user's interests for any query, even when they are unrelated. In addition, most of these approaches require the subtopics of queries to be determined in advance, which may not be possible in large-scale real search engines, making them hard to be applied.

Instead of framing diversification within personalization, we consider personalization and diversification as two complementary ingredients that we can incorporate in general search when appropriate. Intuitively, personalization and diversification play different roles in search. Personalized search assumes that we know the user's interests well, thus the search results for a query within user's interests can be tuned toward these interests. On the other hand, diversification does not assume any knowledge about the user's interests. It ranks documents according to their

Manuscript received 8 November 2022; revised 12 May 2023; accepted 18 June 2023. Date of publication 30 June 2023; date of current version 11 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62272467 and 61872370, in part by Beijing Outstanding Young Scientist Program under Grant BJJWZYJH012019100020098, in part by the Fundamental Research Funds for the Central Universities, in part by the fund for building world-class universities (disciplines) of Renmin University of China, in part by the Research Funds of Renmin University of China under Grant 22XNKJ34, in part by the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China, Public Computing Cloud, Renmin University of China, and in part by Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE. Recommended for acceptance by R. Chi-Wing Wong. *(Corresponding author: Zhicheng Dou.)*

Jiongnan Liu, Zhicheng Dou, and Ji-Rong Wen are with the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China, and also with the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, Beijing 100872, China (e-mail: liujn@ruc.edu.cn; dou@ruc.edu.cn; jrwen@ruc.edu.cn).

Jian-Yun Nie is with DIRO, University de Montreal, Montreal, Quebec H3C 3J7, Canada (e-mail: nie@iro.umontreal.ca).

Digital Object Identifier 10.1109/TKDE.2023.3291006

TABLE I
THE USER'S CLICK BEHAVIOR

Result	Label	Reason
d_1	$\surd$	related to user's interests
d_2	$\times$	related to user's interests but too similar to d_1
d_3	$\times$	not related to user's interests
d_4	$\surd$	related to user's interests
d_5	$\times$	not related to user's interests

differences or their coverage of subtopics of the query. We see here that **personalization and diversification apply in different contexts**, depending on how much we know about the user and how much the current query is related to her interests. When we have rich histories to build reliable profiles, personalization may be appropriate. Otherwise, when we know little about the user's interests or the query is not related to them, diversification may be a safer solution.

In our opinion, **search result diversification and personalized search are not opposite, but can be complementary to each other**. Personalized search results should also be diversified and a user may have intrinsic diverse needs for some information. For example, a beginner programmer issuing a query on "java" may need diversified documents about "java tutorial", "java JDK" etc. even though we know that the query refers to "java programming language" based on the user's personal profile. This example illustrates a case where we need personalization and diversification simultaneously, rather than selectively using one of them. In this paper, we propose a method to do it.

Different from the existing personalized search result diversification approaches requiring pre-processed subtopic information, we try to integrate personalized search and search result diversification to directly improve top result satisfaction **based on large-scale search logs** without subtopics clearly identified. A critical problem, when exploiting search logs, is to understand how personalization and diversification have affected users' behaviors and what roles they played. We consider that users' behaviors are stimulated by several factors such as personal relevance and document redundancy. A user could click on a document if it is relevant to her personal interests and is novel (compared with other clicked documents). We show some typical examples in Table I, where the satisfied results are marked with $\surd$ while unsatisfied results are marked with $\times$. Document d_1 is related to both the query and the user interests and results in a satisfied click. Document d_2 is also related but it is too similar to d_1. The user cannot obtain any additional useful information from this document, thus she may ignore this document. d_3 and d_5 are relevant to the query but irrelevant to the user interests. d_4 covers what the user wants to search and is diverse compared to the document d_1, leading to another satisfied click. The ideal ranking list in this case should be $\{d_1, d_4, d_3, d_5, d_2\}$, and we cannot obtain the ideal ranking with the sole use of personalized search or search result diversification. Personalized search will rank d_2 at the second position and search result diversification will not consider user interests. Therefore, to better optimize search results towards user satisfaction, we need to combine them to combine personal relevance and document diversity.

To implement the above idea, we propose a **PER**sonalization + **DIV**ersification (PER+DIV) framework that trains an integrated ranking model based on query logs. **First**, we design a common hierarchical transformer structure with shared parameters to represent a query and its candidate documents by interacting them together. **Then**, we use two parallel modules: personalization and diversification, to measure personal relevance and document redundancy respectively. For the personalization module, the interest vectors calculated by the hierarchical transformer are used to build user profiles and to refine the query representation. As for diversification, we measure the result diversity by calculating the similarity matrices through neural tensor network. **Finally**, we use the similarity between the current query and user profile to estimate the extent to which the results should be personalized, which also serves as a parameter to combine the obtained personalization and diversification scores. **Furthermore**, as a user's click can be regarded as a merged signal of relevance and diversity, we design two different training methods relying on user's clicks - unified and separate methods, to capture and leverage the personalization and diversification signals either implicitly or explicitly. In the unified method, we use LambdaRank [26] to train an integrated ranking model using click as the label while considering the two factors as latent. In the separate method, we derive the corresponding relevance labels for personalization and diversification explicitly from search logs, and design a multitask structure to jointly optimize the personalization and diversification losses. We also use these labels to build personalized and diversified metrics.

Experimental results on a large-scale commercial dataset show that both proposed methods can significantly outperform the existing personalization and diversification approaches.

The main contribution of our work is three-fold:

1) We propose a new framework integrating personalized search and search result diversification aiming at improving the overall user satisfaction based on large-scale search logs.
2) To better train our model and understand user behaviors, we propose two different strategies to handle the two factors in clicks in unified and separate manners.
3) We experimentally verify the effectiveness of the proposed method on a large real user log from both the personal relevance and document redundancy.

The rest of the paper is organized as follows. We introduce related works in Section II. Our PER+DIV framework for integrated personalized and diversified search is described in Section III. We describe the training and optimizing process in Section IV. We present experimental settings and analyze results in Sections V and VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

A. Personalized Search

To deal with ambiguous and broad queries, many personalized search approaches have been proposed. These approaches combine the historical information with the query to capture the user intent. Different search results can be produced for different

users according to their interests. Early approaches in personalized search are mainly based on personal features extracted from query logs [15], [27] or based on topic models [28]. In recent years, more advanced approaches applying deep learning and neural network techniques have been proposed.

1) Personalized Search Based on User Profile: As we discussed earlier, personalized search tries to extract user interests from search histories. Several approaches have been proposed to construct user profiles. Ge et al. [16] proposed HRNN that uses the hierarchical RNN technique to capture user long-term and short-term interests. Then, it conducted the attention technique to refine both interests using the query terms. Lu et al. [17] introduced generative adversarial network(GAN) into personalization and proposed PSGAN based on HRNN. Ma et al. [18] replaced the traditional RNN with time-aware RNN to incorporate temporal information in personalized search. Yao et al. [29] adopted reinforcement learning methods to mimic user behaviors to capture user preferences. Zhou et al. [22] utilized memory network to enlarge the capacity of model to build more detailed user profiles. With a user profile obtained by above approaches, document ranking can be provided based on the combination of the document-profile and document-query similarities.

2) Context-Aware Personalized Search: Recently, several context-aware approaches have been proposed to improve the performance of personalized search. To better understand user intent, these approaches combine the profile and query together to refine the query representation itself. Yao et al. [19] used the personalized embedding to construct a personalized word embedding table for each user to rebuild the query representation. Zhou et al. [20] proposed HTPS, applying transformer to conduct query disambiguation. Deng et al. [30] focused on the multiple positive and negative feedback provided by the users and improved HTPS. PSSL [21] further improved HTPS by introducing several self-supervised learning tasks. Recently, Some researchers [31] try to find similar users from social network to augment the representation of current users.

B. Search Result Diversification

In parallel to personalized search, search result diversification is another approach to cope with the query ambiguity problem in information retrieval. It aims to make the top results cover as many subtopics of the query as possible so that users with different intents can likely find documents corresponding to their interests. According to whether the model relies on a set of subtopics, existing approaches can be divided into explicit approaches (with subtopics) and implicit approaches (no subtopics). Several recent approaches such as DESA [9] and DVGAN [10] have combined explicit and implicit approaches. DVGAN introduced generative adversarial network into diversification and DESA adopted transformer in the interaction between documents and subtopics.

1) Explicit Approaches: Explicit search result diversification approaches explicitly use subtopics as inputs for modeling diversity. They usually select a document relevant to subtopics that have not been well covered before. It infers that the document should be more related to the subtopics which are not covered by previous ranked documents. In order to evaluate the coverage of subtopics at each step, different approaches use different methods to assess the subtopic distribution. xQuAD [4] and PM2 [32] are the representative unsupervised explicit approaches. xQuAD defined the distribution of subtopics as the possibility of the previous chosen documents not including them. PM2 counted the number of documents relevant to subtopics to get the distribution of subtopics. Many approaches have been derived from these two representatives by using hierarchical information (HPM2 and HxQuAD [33]) or using term level information (TPM2 and TxQuAD [34]). Jiang et al. [8] proposed DSSA that introduced deep learning into explicit approaches and used RNN and attention techniques to model subtopic coverage. However, explicit approaches need the subtopic and document-subtopic relevance information of each query, which takes lots of time to annotate and is hard to apply under real search situations. Therefore, we mainly utilize implicit diversification method in our framework.

2) Implicit Approaches: Different from explicit approaches, implicit approaches consider document novelty in diversifying process. They score a document according to whether the document is different from the selected document set. MMR [3] evaluated document novelty by calculating the similarity between the candidate document and the documents already selected. R-LTR [6] and PAMM [7] introduced the Plackett-Luce model [35] and calculated the similarity from different elements such as title. Both of them require manually defined dissimilarity features between documents. In contrast, NTN [36] modeled document similarity by using neural tensor network and distributed representation of documents to avoid feature engineering. Recently, some attention-based approaches [12], [13] have been proposed. Daletor [12] applied metric learning to directly optimize the diversification metrics. Graph4DIV [13] constructed a document graph and devised GNN to calculate diversification score. It used BERT to classify whether two documents belong to same subtopic and built edges between documents.

3) Personalized Diversification Approaches: To improve the diversification performance by enhancing the users' information, several personalized search result diversification approaches have been proposed. Radlinski and Dumais [23] solved this problem by finding similar follow-up queries to construct subtopics. Vallet et al. [24] optimized xQuAD by adding the user variance into its score function. Liang et al. [25] introduced structure learning method to deal with the problem. It is, however, important to stress that personalized search result diversification is different from our model. The former mainly focused on determining the specific subtopics for users and used diversification approaches to cover them. Most of these approaches require the subtopics of queries determined in advance. In contrast, we directly optimize the user satisfaction (click-based metric) based on large-scale search logs without labelling of subtopics. Personalization and diversification are two latent aspects involved in the process rather than the final objectives. In other words, our approach can determine dynamically the extent to which the results need to be personalized or diversified according to what we learn from the click data.

TABLE II
NOTATIONS IN OUR FRAMEWORK

Name	Description
q, U	The current query and the user
D, d	The candidate document set for current query q, $d \in D$
$\mathcal{H}^s, \mathcal{H}^l$	The short-term and long-term query history
q_i^s	The i-th query in the current session
$d_{i_k}^s$	The k-th candidate document for query q_i^s
h_i^s	The i-th short-term historical vector
t_k^d, t_k^q	The k-th term in d, q
T_d, T_q	The term list of d and q, $T_d = [t_1^d, \cdots, t_M^d]$
$\phi(\cdot), \psi(\cdot)$	MLP layers
Trm$(\cdot)$	Transformer layers

C. Transformer

Attention mechanism is widely used in the area of NLP [37] and IR. Many branches of attention mechanism have been developed in recent years, such as self-attention, multi-head attention, and etc. Transformer [38] uses self-attention and multi-head attention in machine translation tasks in order to encode the sequence by itself and achieves great success. It is shown that attention mechanism fits in extracting information from sequences as an alternative of RNN and CNN. Qin et al. [9] introduced attention into search result diversification using transformer. Zhou et al. [20] applied transformer structure to conduct fine-grained personalization. In our paper, we utilize the attention-based transformer structure to extract user preferences from historical sequences and measure document novelty in the personalization and diversification module respectively.

III. THE PER+DIV FRAMEWORK

In this section, we describe our approach to leverage both personalized search and diversified search for improving search quality and user satisfaction. As we discussed in Section I, users' click behavior can result from both personal relevance and content novelty of the document. To integrate both, we grade the personalization and diversification scores for the document simultaneously and combine them in the final document ranking.

We start with the problem formulation. The notations used in this paper are listed in Table II. Given the user U, the query q and the candidate document set D, we need to retrieve a ranking list that is both relevant to user interests and diverse between documents. Our PER+DIV framework tries to solve this problem by calculating a score $f(d|q, U, D)$ for each document d based on both personalization and diversification and using it to re-rank the results. Suppose that for each user U, her historical data $\mathcal{H}$ can be divided into short-term history $\mathcal{H}^s$ and long-term history $\mathcal{H}^l$. The short-term history $\mathcal{H}^s = \{\{q_1^s, D_1^s\}, \ldots, \{q_i^s, D_i^s\}, \ldots, \{q_{|s|}^s, D_{|s|}^s\}\}$ denotes the queries and their corresponding candidate documents in the current session. The candidate document set $D_i^s = \{d_{i_1}^s, \ldots, d_{i_m}^s\}$ denotes all the candidate documents for query q_i^s. The long-term history $\mathcal{H}^l = \{\{q_1^l, D_1^l\}, \ldots, \{q_i^l, D_i^l\}, \ldots, \{q_{|l|}^l, D_{|l|}^l\}\}$ denotes the queries and corresponding candidate documents in earlier sessions of query history. For the convenience of presentation, we assume the same size m for all the candidate document sets D_i^s, D_i^l, and D, i.e., $|D_i^s| = |D_i^l| = |D| = m$. We use padding with zero vectors to get the required m elements when necessary.

As we discussed before, the user's click behaviors can result from both personal relevance and document novelty. Hence we need to consider both sides of information and calculate the weight for the personalization and diversification module. The score function is as follows:

$$f(d|q, U, D) = \lambda(q, U) S^{\text{per}}(d|q, U) + (1 - \lambda(q, U)) S^{\text{div}}(d|D), \quad (1)$$

where $\lambda(q, U)$ denotes the weight attributed to personalization; $S^{\text{per}}(d|q, U)$ and $S^{\text{div}}(d|D)$ denote respectively the personalization score mainly focusing on the relevance to user interests and diversification score measuring novelty for document d. The whole structure of our framework is shown in Fig. 1. We will introduce the details and analyze the complexity of our model in the remaining parts of this section.

A. Query and Document Representations

To build document and query representations, we first use word2vec [39] to generate word embeddings for all terms in history. After building the word embeddings, we consider two initial representations for documents and queries in our PER+DIV framework:

1) For query q and document d, we add the word embedding of their corresponding terms respectively to obtain the initial embedding vector q^0 and d^0, which presents the initial and original meaning of query and document:
$$q^0 = \sum_i t_i^q, \quad d^0 = \sum_i t_i^d. \quad (2)$$
2) However, as the meaning of words may change in different contexts, we need to embed the word with contexts. Since the transformer encoder structure achieves great performances in many areas and previous models such as HTPS [20] have introduced it into personalization, we apply it to interact and aggregate the word embeddings to obtain the contextualized representations. More specifically, we conduct the term-level transformer Trm^{term} to obtain context-aware word embedding and add them together to calculate integrated query and document representation q^{int} and d^{int}:
$$q^{\text{int}} = \sum \text{Trm}^{\text{term}}(t_1^q, \ldots, t_M^q),$$
$$d^{\text{int}} = \sum \text{Trm}^{\text{term}}(t_1^d, \ldots, t_M^d). \quad (3)$$

So far, we have obtained two initial representations for the query and document. However, previous results have shown that queries issued to search engines are usually very short and ambiguous [1]. To better understand the intent of a query, we rely on its search results to boost the representation of queries. Furthermore, the content of a document can also be enriched by incorporating an interaction among search results. Inspired by HTPS [20], we design a hierarchical transformer [38] structure to model the query and its candidate documents.

In the first level, we mix and integrate the semantic information between the terms in queries and documents to refine their representations. Given query q and candidate document d

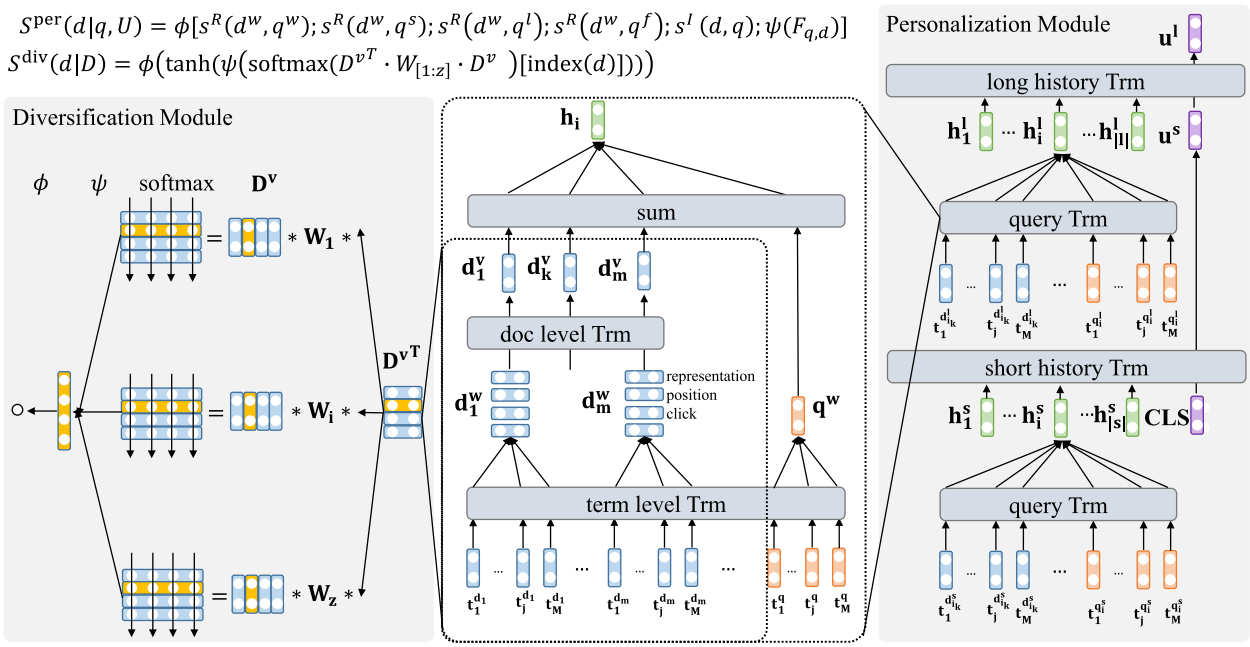


Fig. 1. The main structure of our PER+DIV framework.

in D (recall that $|D| = m$), we first concatenate the list of term embeddings of all words in all candidate documents T_{d_k} ($T_{d_k} = [t_1^{d_k}, \ldots, t_M^{d_k}]$) and the query itself T_q ($T_q = [t_1^q, \ldots, t_M^q]$) together to form a term embedding list. Noticed that due to the length limitation of the transformer encoder, we only reserve the word embeddings of terms in the document's titles in the list. We argue that term-level integration is helpful in refining query and document representation as it introduces more context information.

We apply the first term-level transformer to make interactions between documents and query, i.e.,

$$\tilde{T}_q, \tilde{T}_{d_1}, \ldots, \tilde{T}_{d_m} = \text{Trm}^{\text{term}}(T_q, T_{d_1}, \ldots, T_{d_m}).$$

Note that $\tilde{T}_{d_k}$ and $\tilde{T}_q$ are still term embedding lists ($\tilde{T}_{d_k} = [\tilde{t}_1^{d_k}, \ldots, \tilde{t}_M^{d_k}]$ and $\tilde{T}_q = [\tilde{t}_1^q, \ldots, \tilde{t}_M^q]$). Then we slice the term embedding list and calculate the context-enriched representation d_k^w and q^w for document d_k and query q by summing their own term embedding vectors, i.e.,

$$d_k^w = \sum \tilde{T}_{d_k} = \sum_j \tilde{t}_j^{d_k}, \tag{4}$$

$$q^w = \sum \tilde{T}_q = \sum_j \tilde{t}_j^q. \tag{5}$$

However, by using only the first-level term-level transformer encoder, the model ignores some non-semantic information of the documents such as the click signals and the displayed positions. Different from the existing personalized search approaches mostly only use the clicked documents to capture user's interests, we argue that in our case, the unclicked documents can also provide useful information to help model user interests. To illustrate it, we show an example in Table III, where A, B, C, D, E denote diverse subtopics of the query. If we only use the clicked documents to calculate the historical vector, the three document ranking lists will have the same representation, but they actually differ in user's behaviors. The reason that the documents A_2, A_3, A_4 in Ranking 2 are unclicked is probably due to the redundancy between documents and this is different from the reason for C_1, D_1, E_1 in Ranking 3.

TABLE III
EXAMPLE SEARCH RESULTS AND CORRESPONDING CLICKS

	Position	1	2	3	4	5
Ranking 1	Document	A_1	B_1	C_1	D_1	E_1
	Click	1	1	0	0	0
Ranking 2	Document	A_1	A_2	A_3	A_4	B_1
	Click	1	0	0	0	1
Ranking 3	Document	A_1	C_1	D_1	E_1	B_1
	Click	1	0	0	0	1

To distinguish different rankings with the same clicked documents as in Table III and to embrace more non-semantic information into user modeling, we design a second document-level transformer structure to build document and search representations. In particular, inspired by BERT [37], we add position embedding and click embedding to enhance document representations. Furthermore, as we only use the terms in titles in the first term-level transformer, the embedding vectors may be inaccurate and noisy. To better represent documents, we also enhance the distributed embedding of documents on the document contents [1]. Finally, the refined document representation is calculated through the second document-level transformer:

$$\tilde{D}^w = D^w + D^{\text{pos}} + D^{\text{clk}} + D^{\text{rep}},$$

$$D^v = \text{Trm}^{\text{doc}}(\tilde{D}^w), \tag{6}$$

[1]We simply use the doc2vec method in our experiments. However, it can be easily replaced by other methods such as BERT representation.

where $D^v = [d_1^v, \ldots, d_k^v, \ldots, d_m^v]$ is the list of refined document representations; $D^w = [d_1^w, \ldots, d_k^w, \ldots, d_m^w]$ is calculated by (5); D^{pos} denotes the position embedding; D^{clk} denotes the click embedding and D^{rep} denotes the distributed embedding by doc2vec.

B. Personalization Module

In our framework, the personalization module is adopted to evaluate the document's relevance to the user interests. In this module, we design a transformer-based structure to extract the user profile for measuring the personal relevance, since it achieves significant improvements [20], [21] over the traditional RNN-based encoder [16], [18].

We first aggregate the query and document representations together to capture the representation of a historical search and click behavior. We take the i-th short-term query q_i^s as an example. Its representation h_i^s is calculated by:

$$h_i^s = \sum_j d_j^v + q_i^w, \tag{7}$$

where d_j^v is the document representation calculated by (6), and q_i^w is the interactive representation of q_i based on (5). Because we introduce the position and click signals in the second document-level transformer in 6, the built historical search representation h_i^s can better represent user interests from both the positive (clicked documents) and negative (unclicked documents) feedback.

Then we concatenate these vectors together to form the short-term historical vector list $H^s = [h_1^s, \ldots, h_i^s, \ldots, h_{|s|}^s]$ and long-term historical vector list $H^l = [h_1^l, \ldots, h_i^l, \ldots, h_{|l|}^l]$. Short-term historical vectors may contain more information about user intent in the current query because the queries are in the same session. Long-term interests are more stable and reflect the general interests of the user. They can help refine short-term interests. Following existing approaches [16], [18], [20], we consider that users' long-term and short-term interests may have hierarchical structures. We conduct a hierarchical transformer to capture the final user profile as follows:

1) First, we need to capture user interests in the current session since the user intents in current queries may be derived from and stimulated by the search behaviors in current sessions. We construct the short-term user interests in this session to help refine the current query representations. Since the user interests may continuously evolve in the search and browsing flow during the current session, we apply the transformer encoder to integrate the historical search representation h_i^s to build the short-term profile. More specifically, we add the "[CLS]" token to the end of short-term historical vector list H^s and apply the position-aware transformer to capture the short-term user profile vector u^s, i.e,

$$u^s = \text{Trm}^{\text{short}}([H^s, \text{CLS}] + [H^s, \text{CLS}]^{\text{pos}})[|s|+1], \tag{8}$$

where u^s is captured by slicing the last embedding vector, which corresponds to the "[CLS]" token.

2) However, utilizing only the users' behaviors in the current sessions may be inadequate for capturing user preferences, we need to enhance users' overall interests during all their histories to reflect their general preferences and to tune their short-term interests. However, as we only need to build the overall long-term interests, it is not necessary to extract the interest vectors for each session. Therefore, we simply develop a transformer to integrate users' searching representations across all histories. In particular, we add u^s to the end of long-term historical vector list H^l and apply the long-term transformer on the concatenated list to obtain the representation:

$$u^l = \text{Trm}^{\text{long}}([H^l, u^s] + [H^l, u^s]^{\text{pos}})[|l|+1], \tag{9}$$

where u^l is obtained the same way as u^s.

3) Finally, having constructed the long-term user profile vector u^l, the short-term user profile vector u^s, and the integrated query vector q^{int}, following previous approaches as HTPS [20] and PEPS [19], we apply gate functions to aggregate them into the final refined query representation:

$$\begin{aligned} \text{gate}(x,y) &= zx + (1-z)y, \quad z = \sigma(\phi([x;y])), \\ u^f &= \text{gate}(u^s, u^l), \quad q^s = \text{gate}(q^{\text{int}}, u^s), \\ q^l &= \text{gate}(q^{\text{int}}, u^l), \quad q^f = \text{gate}(q^s, q^l), \end{aligned} \tag{10}$$

where u^f denotes the final profile vector; q^s, q^l denote the refined query representation enhanced short-term profile and long-term profile; q^f denotes the final refined query representation.

Previous approaches [19], [20], [21] have shown that calculating the similarities between the scored document representations and multiple query representations mixed with different history interests can be beneficial for the personalization performance. Since we have already obtained the initial, integrated, and refined query representation in (3) and (10), we can calculate several representation-based similarities between vectors by similarity function s^R following previous approaches. In this paper, we adopt cosine similarity as s^R function, but it can be any other function such as euclidean distance and dot product. Inspired by PEPS [19], we also apply K-NRM [40] to obtain interaction-based similarity s^I between the initial and integrated query and document vectors to better model the ad hoc similarity between them. Then we use an MLP layer to integrate these similarities. Finally, personalization score is calculated by:

$$\begin{aligned} S^{\text{per}}(d|q,U) = \phi([&s^I(d^0,q^0), s^I(d^{\text{int}},q^{\text{int}}), s^R(d^{\text{int}},q^{\text{int}}), \\ &s^R(d^{\text{int}},q^s), s^R(d^{\text{int}},q^l), s^R(d^{\text{int}},q^f), \psi(\mathcal{F}_{q,d})]), \end{aligned} \tag{11}$$

where $\mathcal{F}_{q,d}$ denotes the feature vector, ϕ, ψ denotes MLP layers.

C. Diversification Module

The diversification module in PER+DIV framework focuses on improving the novelty of candidate documents and the diversity of results. As it is hard to capture the subtopics in real search engines, we measure the diversification score in an implicit manner.

To model the diversity of search results, we use the same method introduced in Section III-A to conduct interactions between current candidate documents. Following existing methods such as DESA [9], we do not take the current query into consideration while modeling document novelty. Thus we omit the query part in (5), i.e., we set $T_q = \emptyset$ and only conduct interaction between documents. We use a second level transformer with shared parameters in (6) to conduct interactions between document representations and obtain the interactive document representation matrix D^v. As we do not have the click information for the candidate documents of the current query, we regard them all as clicked ones.

Previous approaches such as DESA [9] and Daletor [12] mostly slice the document d's embedding vector from the representation matrix D^v and apply an MLP layer to it to calculate the diversification score. We believe that this simple method cannot explicitly model the dissimilarity between documents. A more convincing and natural way is to compare all the candidate documents to measure their uniqueness. Thus, in our PER+DIV framework, we adopt the Neural Tensor Network method, which computes document similarities in its model to directly capture document novelty. However, different from the origin NTN [36] method that only measures the dissimilarity between current document and previously selected documents, we calculate the dissimilarities among all candidate documents.

To capture document dissimilarity in different aspects, we adapt NTN method and apply z trainable weight matrices $W \in \mathbb{R}^{\alpha \times \alpha}$ in our model, where α denotes the embedding length. Given a matrix W_i, we calculate the multiple-perspective similarities between documents using the above representation matrix D^v:

$$S_i = \text{softmax}(D^{v\top} \cdot W_i \cdot D^v), \tag{12}$$

where $S_i \in \mathbb{R}^{m \times m}$ denotes the similarity matrix between documents and the softmax function is done on the row for normalization. This similarity calculation method can be regarded as a general dot product method. If we set W_i is an identity matrix, then it is the dot product similarity between documents. When W_i changes, the similarity calculation can focus on different dimensions of document representations. As we devise z trainable matrices, we can evaluate the similarities among documents from different perspectives. Therefore we can get z similarity matrices $S_{[1:z]}$ by applying the (12) z times:

$$S_{[1:z]} = \text{softmax}(D^{v\top} \cdot W_{[1:z]} \cdot D^v).$$

To capture the novelty of the scoring document d, we need to slice z similarity vectors $s_{[1:z]} = S_{[1:z]}[\text{index}(d)] \in \mathbb{R}^{m \times z}$ from the whole similarity tensor. Then we aggregate the similarity vectors to obtain the document novelty in one aspect. The aggregation function can be the sum, average, etc. We use linear combination and tanh$(\cdot)$ function to align the range of cosine similarity in personalization module, as it yields the best performance:

$$\xi = \tanh(\psi(s_{[1:z]})),$$

where $\xi \in \mathbb{R}^z$ refers to the document novelty in z aspects. Finally, we use an MLP layer to capture the final diversification score, i.e.

$$S^{\text{div}}(d|D) = \phi(\xi) = \phi(\tanh(\psi(s_{[1:z]}))). \tag{13}$$

D. Combination

To better integrate personal relevance and document novelty, the combined weight between personalization and diversification score should be determined according to the query and user. Intuitively, we should emphasize the personal relevance part if the current query is highly relevant to user interests. Otherwise, if current query has little to do with user interests, we should provide more diversified results to cover user potential intents as much as possible. To implement this idea, we calculate the similarity between the final profile vector u^f and the integrated query representation q^{int} and use it to determine the combined weight in (1) between personalization and diversification score:

$$\lambda(q, U) = s^R(u^f, q^{\text{int}}).$$

In this paper we use cosine similarity for s^R and more complex combinations can be explored in future work.

E. Time Complexity

As we described in the above part, the PER+DIV framework can be divided into the common hierarchical transformer module, the personalization module, and the diversification module. We will analyze their complexity respectively. Preliminarily, we assume that the embedding length is α for all vectors, and the inner hidden size in transformer's FFN layer is β.

First, in the hierarchical transformer module, the length of the whole term embedding list is $(m+1)M$ for one query, where M denotes the maximum length among the query and the title of documents, m denotes the maximum number of candidate documents. Therefore, the overall time complexity of the hierarchical transformer module is $\mathcal{O}(m^2\, M^2\alpha + mM\alpha\beta)$ for one query.

Second, in the personalization module, there are two one-layer transformers, so the overall complexity is $\mathcal{O}(|s|^2\alpha + |s|\alpha\beta + |l|^2\alpha + |l|\alpha\beta) = \mathcal{O}((|s|^2 + |l|^2)\alpha + (|s| + |l|)\alpha\beta)$, where $|s|$ and $|l|$ denotes the length of user short history $\mathcal{H}^s$ and long history $\mathcal{H}^l$.

Third, in the diversification module, the time complexity of the NTN module is $\mathcal{O}(z(m^2\alpha + m\alpha^2))$, corresponding to the two matrix multiplication operations in (12), where z is the number of trainable matrices W_i.

In summary, the overall time complexity of PER+DIV is $\mathcal{O}((|s| + |l|)(m^2\, M^2\alpha + mM\alpha\beta) + (|s|^2 + |l|^2)\alpha + (|s| + |l|)\alpha\beta + z(m^2\alpha + m\alpha^2))$. Noticed that the complexity of the diversification module is relatively low compared to the personalization module and the hierarchical transformer module. Therefore, the time complexity of PER+DIV is comparable to the HTPS method, which also adopts hierarchical transformer structure to construct user profiles.

IV. Training and Optimization

As we discussed before, users' behaviors can be affected by several factors such as personal relevance and document novelty. Thus, it is intuitive to model click in two different ways: regard as a unified one and separate it into different elements. As a result, we put forward two methods to train our model. We will introduce the details of two training methods in the following parts.

A. The Unified Method

In this method, we regard the user's click as a unified whole. Following existing personalization methods [16], [18], [20], we use the LambdaRank [26] algorithm to train our model in pairwise loss. Given query q and its candidate set D, we sample a positive (satisfied clicked) document and a negative (unclicked) document to form the training pair $S = \langle d_i, d_j \rangle$. We train our model by maximizing the score margin between the positive and negative samples. We infer that the probability $\widehat{p_{ij}}$ in which d_i is more likely to be clicked by users than d_j is calculated by the σ function:

$$\widehat{p_{ij}} = \sigma(f(d_i) - f(d_j)) = \frac{1}{(1 + \exp(f(d_j) - f(d_i)))}, \quad (14)$$

where $f(d)$ is an abbreviation of $f(d|q, U, D)$. Given the predicted probability and the true label p_{ij}, the loss is calculated by the weighted cross entropy function as follows:

$$\begin{aligned} \mathcal{L}^{\text{unified}} &= \mathcal{CE}(p_{ij}, \widehat{p_{ij}}) \\ &= -\sum |\Delta_{ij}|(p_{ij} \log(\widehat{p_{ij}}) + (1 - p_{ij}) \log(1 - \widehat{p_{ij}})), \end{aligned} \quad (15)$$

where Δ_{ij} denotes the metric change such as MAP when swapping the position of d_i and d_j in the ranking list.

B. The Separate Method

As we stated above, the reason why users click a document can be affected by both its relevance to user's interests and document novelty. We also measure personalization and diversification scores respectively in our model. However, this makes it difficult to train the model using clicks as mixed signals as in the unified training method. Thus, we propose another training method, which aims to separate the personalization and diversification in click behavior and train each module respectively.

First, we estimate which additional documents would have been clicked if the user did not consider their novelty. To do this, we calculate the similarity between each unclicked document to the clicked documents.[2] If the average similarity is higher than a threshold τ, we consider that this document should have been clicked only due to personalization, and label it as *pseudo click*. So, the loss of personalization module is calculated by the same score function in (14) and (15) but we remove the pseudo clicked documents from negative samples and add them into positive ones. As we only consider personalization here, the predicted probability is also calculated by the personalization score only, i.e.,

$$\widehat{p_{ij}^{\text{per}}} = \sigma(S^{\text{per}}(d_i) - S^{\text{per}}(d_j))$$

$$\mathcal{L}^{\text{per}} = \mathcal{CE}(p_{ij}^{\text{per}}, \widehat{p_{ij}^{\text{per}}}).$$

Then, we design the diversification loss. The reason why users didn't click these pseudo clicked documents may be that they are redundant. In other words, the clicked documents are more diversified than those pseudo clicked ones. Similar to (14) and (15), we design the diversification loss but only use the pseudo documents as negative documents, not all the unclicked documents. The predicted probability is also calculated by S^{div}:

$$\widehat{p_{ij}^{\text{div}}} = \sigma(S^{\text{div}}(d_i) - S^{\text{div}}(d_j))$$

$$\mathcal{L}^{\text{div}} = \mathcal{CE}(p_{ij}^{\text{div}}, \widehat{p_{ij}^{\text{div}}}).$$

We show the formation of positive and negative samples of different loss function in Table IV. The final loss of separate method is the combination of $\mathcal{L}^{\text{per}}$ and $\mathcal{L}^{\text{div}}$:

$$\mathcal{L}^{\text{separate}} = \mathcal{L}^{\text{per}} + \mu \mathcal{L}^{\text{div}}, \quad (16)$$

where μ is a hyper-parameter.

TABLE IV
The Formation of Positive and Negative Samples in Loss Function. $\surd$: Positive, $\times$: Negative, $\circ$: Neither

Loss	Clicked Doc.	Pseudo Clicked Doc.	Unclicked Doc.
$\mathcal{L}^{\text{unified}}$	$\surd$	$\times$	$\times$
$\mathcal{L}^{\text{per}}$	$\surd$	$\surd$	$\times$
$\mathcal{L}^{\text{div}}$	$\surd$	$\times$	$\circ$

TABLE V
Basic Statistics of the Dataset

Item	Value	Item	Value
# Train queries	188,267	# Days	58
# Test queries	41,261	# Users	5,317
Avg. # click per train query	1.19	# Docs	681,512
Avg. # click per test query	1.20	Avg. query length	3.25
Avg. # doc per train query	6.54	Avg. session length	2.63
Avg. # doc per test query	5.22	Avg. doc length	988.7

V. Experimental Setup

A. Datasets

There are some public datasets for personalized search such as the AOL dataset and the WEBIS dataset. However, the candidate documents of queries in AOL are generated by vanilla retrieval methods such as BM25 and are not provided by real search engines. Users may not have seen them in real situations. As we focus on the user behaviors in this paper, such a pseudo dataset is not appropriate. Similarly, the WEBIS dataset also lacks the original ranking results, which is also unsuitable for our experiments.

Therefore, we conduct experiments on a search log dataset from a commercial search engine. The basic statistics of this commercial dataset are shown in Table V. The searches in the dataset date from 1^{st} Jan. 2013 to 28^{th} Feb. 2013. We regard

[2]We calculate the similarities between documents mentioned in the rest of this paper using their doc2vec representation in default of further description.

the first four weeks as the user's history and do our experiments on the last two weeks. We extract the document from the html source of web pages. Following [16], we use the 30 minutes of user inactivity as the boundary to divide sessions. We view the document click with more than 30 seconds dwelling time as a satisfied click, which eliminates the effects of ranking bias as much as possible. For each user, we divide the training, validation and test set by the sessions with 4:1:1 ratio.

B. Baselines

1) Adhoc Ranking Models:

Org. We directly use the original ranking as the baseline in the commercial dataset.

K-NRM [40]. We take K-NRM as the adhoc search baseline. K-NRM is a kernel-based neural ranking model. It uses k kernels to calculate the interaction-based similarity between document and query. We take $k = 11$ and use the same LambdaRank algorithm to train K-NRM.

2) Personalized Search Models:

SLTB [15]. SLTB is a feature-based personalized search model. It extracts 102 features from the query history for each covering topic feature, time feature and etc.

P-Click [14]: P-Click assumes users will click the same document that most users clicked for the same query before. It ranks the documents based on the number of historical clicks made by the same user.

HRNN [16]. HRNN uses hierarchical RNN to construct user long-term and short-term user profiles and adopt attention mechanism to integrate the profiles and current query. The personalization score is calculated by query-document matching, profile-document matching and feature-based score.

HTPS [20]. HTPS is one of the state-of-the-art context-aware personalized search baseline. It uses hierarchical transformer to disambiguate the query and designs a language model predicting next query to help training.

PEPS [19]. PEPS is another state-of-the-art context-aware personalized baseline. It constructs personalized word embedding for each user and rebuilds the query representation by his/her unique embedding matrix.

3) Search Result Diversification Models:

MMR [3]. MMR is the representative unsupervised search result diversification baseline. It scores the document by the linear combination of document-query relevance and document-document dissimilarity. We tune the combination rate $\lambda = 0.5$ and 0.7 in our experiments.

ORG+MMR: It uses $1/\sqrt{r_i}$ as the relevance score in MRR [41], given the fact that the original ranking quality is quite good, but we don't have the ranking score available.

DESA-IM [9]. DESA-IM is a transformer-based implicit model. We use the implicit part in DESA to build this model. It conducts transformer encoder to do the interaction between candidate documents. We use pairwise cross entropy loss function to train this model.

4) Personalized and Diversified Search Models:

PEPS+MMR. PEPS+MMR is a simple pipeline model to integrate personalization and diversification by replacing the relevance score in MMR by the score calculated by PEPS model. We use $\lambda = 0.7$ in PEPS+MMR baseline.

C. Implementation Details

Our model PER+DIV [3] is trained via both the unified method PER+DIV(u) and separate method PER+DIV(s). The size m of candidate document sets is set to 50. The word embedding size α is 100. The inner length in transformer FFN β is 256. The number of heads in all transformers is 6. The number of layers in transformers is selected from 2. The number of kernels in KNRM is 11. For the diversification module, we use $z = 4$ as it yields the best performance. We adopt $\tau = 0.8$ for cosine similarity and $\mu = 0.5$ for the separate method. For other models, we use the configuration in their papers to conduct experiments.

For all of the supervised methods, we tune the learning rate r from 10^{-7} to 10^{-1} and adopt the Adam optimizer to train the model.

D. Evaluation Metrics

We use three kinds of evaluation metrics to evaluate models: unified metrics, personalized metrics, and diversified metrics.

1) Unified Metrics: We use the metrics based on users' satisfied clicks such as MAP, MRR and P@1 as our unified metrics. The calculation of these metrics is as follows:

$$\text{MAP} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{c_i}\sum_{j=1}^{c_i}\frac{j}{p_i^j},$$

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{p_i^1},$$

$$\text{P@1} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{[p_i^1=1]},$$

where N is the number of queries, c_i is the number of user clicks in query i, p_i^j is the position of j-th click of query i.

2) Personalized Metrics: Because we cannot obtain users' real intents or interests, we need to design pseudo judgments to measure personalization performance. We replace user click label with the union of the real satisfied click and the pseudo click we design in Section IV and calculate the MAP as personalized metric "P-MAP". (Personalized MAP). Because these additional pseudo clicked documents are very similar to the original clicked documents, users should click them if document redundancy is not considered.

3) Diversified Metrics: Evaluating diversity is very hard in our experiments because we do not have human-created intent-aware relevance labels for queries in the log. Hence we mine subtopics for queries and assess the relevance between documents and subtopics. We apply two methods to extract the subtopics.

In the first way, following [23], we regard the extension queries of a query q in the query corpus as the subtopics of q. We

[3]https://github.com/rucliujn/PER-DIV

TABLE VI
BASIC STATISTICS OF THE CONSTRUCTED SUBTOPICS

Metric	ERR-IA^1	ERR-IA^2
#queries with at least one subtopic (Q1)	9,009	20,759
Avg. #subtopics in Q1	5.34	2.34
#queries with more than one subtopic (Q2)	5,059	2,731
Avg. #subtopics in Q2	9.52	17.79

assume that a document is relevant to a subtopic if and only if it also occurs in the candidate documents of the extension query. We ignore the documents that occur for more than 10 queries (for example, google.com) as they are likely navigational and useless for diversification. Note that we don't use click to estimate the relevance of subtopics because click behavior is too sparse and is affected by other's personal relevance.

In the second way, we regard a clicked document and documents similar to it as a virtual subtopic. A document is similar to another one if and only if their cosine similarity is larger than τ.[2] Specifically, if two clicked documents are similar, their corresponding virtual subtopics will also be merged into a bigger subtopic. The basic statistics of the constructed subtopics are shown in Table VI.

Having constructed these subtopics, we use ERR-IA@5 [42] as diversified metrics. We label the metrics applying the subtopics built by the first and second way as ERR-IA^1@5 and ERR-IA^2@5. The calculation of ERR-IA is as follows:

$$\mathrm{PP}_r^i = \prod_{k=1}^{r-1}(1 - R_k^i)R_r^i,$$

$$\mathrm{ERR-IA@}K = \frac{\sum_{i=1}^{l}\sum_{r=1}^{K}\phi(r)\cdot \mathrm{PP}_r^i}{l},$$

where l, n is the number of subtopics and the length of the ranking list, PP_r^i denotes the probability of the user stops her browser at position r for subtopic i, $\phi(r)$ is any function about r. Noticed that we provide the ERR-IA calculation of one query. The metric for the whole dataset is the average value of ERR-IA of all queries.

VI. EXPERIMENTAL RESULTS

We present the overall results in the Section VI-A and do additional personalization and diversification result analysis in the Sections VI-B and VI-C. We also conduct ablation studies in the Section VI-D.

A. Overall Results

The whole results are shown in Table. VII and we can make the following observations:

1) *Compared with all the personalized search baselines, our PER+DIV(u) model achieves significant improvements in all metrics:* The relative improvement over PEPS, the best context-aware personalized search baseline, is up to 1.4%, 1.3%, 1.4% and 1.4% in terms of MAP, ERR-IA^1@5, ERR-IA^2@5 and P-MAP. This indicates that users' click behavior is not only affected by personal interests but also document novelty. Combining personalization and diversification properly will improve all the metrics.
2) *Compared with the search result diversification baselines, our PER+DIV(u) model has significant improvements in both unified and personalized metrics:* The relative improvement over DESA-IM, one implicit search result diversification baseline, is up to 49.1% in terms of MAP and up to 34.9% in terms of P-MAP. For the diversified metrics, our PER+DIV(u) model only has a 2.2% relative decrease with the MMR($\lambda = 0.5$) in ERR-IA^1@5 but outperforms it in ERR-IA^2@5. This indicates that only adopting diversification cannot improve user satisfaction. We need to leverage user's search history to provide better rankings.
3) *Compared with the pipeline unified method PEPS+MMR, our PER+DIV(u) model achieves significant improvements in all metrics:* The relative improvement over PEPS+MMR, is up to 2.7%, 0.4%, 4.6% and 5.5% in terms of MAP, ERR-IA^1@5, ERR-IA^2@5 and P-MAP. The results of PEPS+MMR are also lower than PEPS in terms of unified metrics. This shows that a simple pipeline combination cannot integrate diversification into personalization.
4) *Our separate model PER+DIV(s) is worse than the unified trained model PER+DIV(u) in unified metrics but is comparable in diversified and personalized metrics:* The relative setback below PER+DIV(u), is about 2.3% and 0.4% in MAP and ERR-IA^1@5 and the improvement over it is up to 0.9% in P-MAP. The reason why it outperforms PER+DIV(u) in the last metric may be that the evaluation of P-MAP is close to the separate training method. Moreover, PER+DIV(s) also outperforms all the user profile based personalized search baselines in all metrics. This indicates that our separate model also improves the satisfaction of retrieved rankings. A possible reason why it does not outperform PER+DIV(u) in unified metrics may be that we don't have the real data only reflecting personalization or diversification. The pseudo training data we constructed may have biased training pairs.
5) *Our proposed model PER+DIV(u) and PER+DIV(s) outperform all baselines in ERR-IA^2@5:* For ERR-IA^2@5, we need to stress that it is actually a personalized diversity metric according to its definition. It synthesizes personal relevance and document novelty to evaluate results because it uses click documents as seeds to construct subtopics. We can notice that it basically has the same trend with the unified metrics but has subtle differences. It evaluates unique subtopic coverage according to corresponding user interests. The improvements of PER+DIV models in ERR-IA^2@5 demonstrate that our integrated approach also performs well in personalized diversification scenarios, indicating our model is a general and flexible method.

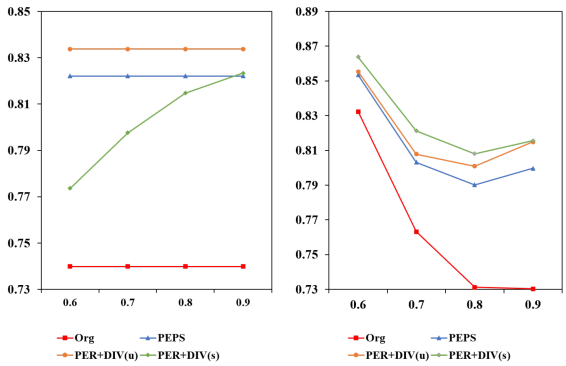
TABLE VII
OVERALL PERFORMANCES OF MODELS

Evaluation Metrics		Unified						Diversified (ERR-IA@5)		Personalized
Task	Model	MAP		MRR		P@1		ERR-IA^1	ERR-IA^2	P-MAP
Adhoc Search	Orginal	.7399	-10.0%	.7506	-9.8%	.6162	-15.9%	.4320	.6050	.7312
	K-NRM	.4916	-40.2%	.5001	-39.9%	.2849	-60.7%	.4300	.5162	.6215
Personalized Search	User profile based methods									
	P-Click	.7509	-9.7%	.7634	-8.3%	.6260	-13.7%	.4325	.6127	.7400
	SLTB	.7921	-3.6%	.7998	-3.9%	.6901	-4.8%	.4333	.6449	.7812
	HRNN	.8065	-1.9%	.8191	-1.6%	.7127	-1.7%	.4344	.6497	.7921
	Context-aware methods									
	HTPS	.8220	-0.0%	.8318	-0.0%	.7286	+0.5%	.4321	.6512	.7897
	PEPS	.8221	-	.8321	-	.7251	-	.4327	.6520	.7902
Search Result Diversification	Implicit methods									
	MMR(λ=0.7)	.4249	-48.3%	.4339	-47.9%	.2047	-71.8%	.4466	.3851	.4879
	MMR(λ=0.5)	.4212	-48.8%	.4304	-48.3%	.2044	-71.8%	**.4482**	.3795	.4755
	ORG+MMR(λ=0.7)	.7398	-10.0%	.7505	-9.8%	.6162	-15.9%	.4327	.6041	.7287
	ORG+MMR(λ=0.5)	.7389	-10.1%	.7499	-9.9%	.6162	-15.9%	.4346	.5999	.7189
	DESA-IM	.5591	-32.0%	.5734	-31.1%	.4093	-43.6%	.4454	.4843	.5936
Unified Methods	Pipeline method									
	PEPS+MMR	.8122	-1.2%	.8234	-1.0%	.7251	0.0%	.4366	.6325	.7591
	Our methods									
	PER+DIV(s)	.8147	-0.9%	.8262	-0.7%	.7164	-1.2%	.4365	**.6616**†	**.8079**†
	PER+DIV(u)	**.8338**†	+1.4%	**.8434**†	+1.4%	**.7414**†	+2.2%	.4383*	.6614†	.8009†

"†" Indicates the model outperforms all baselines significantly with paired T-test at $p < 0.05$ level. "⋆" Indicates the model outperforms all non-diversified baselines significantly with paired T-test at $p < 0.05$ level. For all the unified metrics, we show the relative performances compared with PEPS. The best results are shown in bold.

TABLE VIII
THE NUMBER OF DOCUMENTS CHANGED FROM UNCLICK LABEL TO PSEUDO CLICK LABEL

τ	Pseudo Clicked Doc.	Unclicked Doc.
0.6	63092	154205
0.7	35911	181386
0.8	14588	202709
0.9	5126	212171



(a) The MAP change with τ (b) The P-MAP change with τ

Fig. 2. The personalization results analysis with τ. (a) The MAP change with τ. (b) The P-MAP change with τ.

B. Additional Personalization Result Analysis

In this section, we tune the threshold τ in Section IV-B to do additional personalization result analysis and verify the effectiveness of our model. Firstly, we display the basic statistics that how many unclicked documents are labelled as pseudo clicked documents in the test dataset adopting different τ in Table VIII. Then we compare the unified MAP results and pure MAP results of Original, PEPS, PER+DIV(u) and PER+DIV(s). The results are shown in Fig. 2.

For PEPS and PER+DIV(u), we use the user's click as a mixed label to train our model, thus the results of MAP keep unchanged in these two models. In PER+DIV(s), the loss function is related to the threshold τ and the results of both metrics are changed with τ. The results of MAP show that with τ increasing, the number of pseudo clicked documents decreases, the performance of our PER+DIV(s) model becomes closer to PER+DIV(u). The results show that the simple labeling of pseudo clicks based on cosine similarity in PER+DIV(s) may be harmful to the final performance. We need a more accurate way to mark the click only considering personalization. For P-MAP results, we can observe that our PER+DIV(s) model outperforms PER+DIV(u) regardless of τ. The results show that the separate training methods can improve personalization performance. Furthermore, the improvement over PEPS in P-MAP verifies the benefits of enhancing the unclicked documents.

C. Additional Diversification Result Analysis

For diversified metrics, we design two heuristic ways to construct the subtopics and the judgements for documents. Our integrated framework PER+DIV achieves promising performance in ERR-IA. However, these metrics evaluate diversity by subtopic coverage but our model measures it by document similarity, which may lead to biased results. Although the main metrics to evaluate diversification are based on subtopics, there exists an evaluation metric based on similarity. The metric $S@k$ is calculated as follows:

$$S@k = 1 - \frac{\sum_{i,j,i\neq j} w_{ij} s^R(d_i, d_j)}{\sum_{i,j,i\neq j} w_{ij}}$$

$$w_{ij} = \frac{K - \text{avg}(i,j)}{K}.$$

To better measure the diversity of results, we also evaluate our model using S@k metrics. The results are shown in

TABLE IX
PERFORMANCE OF MODELS IN S@K AND ERR-IA@5

Model	ERR-IA^1	ERR-IA^2	S@3	S@5	S@10
Original	.4320	.6050	.3696	.3802	.3870
PEPS	.4327	.6520	.3711	.3806	.3871
PEPS+MMR	.4366	.6325	**.4391**	**.4224**	**.4004**
PER+DIV(u)	**.4383**	**.6614**	.3775	.3846	.3882

TABLE X
PERFORMANCE OF MODELS IN ABLATION STUDY

Model	MAP	ERR-IA^1	P-MAP
PER+DIV(u)	**.8338**	.4383	.8009
w/o doc Trm	.8074 (-3.17%)	.4352 (-0.71%)	.7806 (-2.53%)
w/o per Trm	.8293 (-0.53%)	.4334 (-1.12%)	.7969(-0.49%)
w/o NTN	.8277 (-0.73%)	.4334 (-1.12%)	.7930 (-0.99%)
w/o DIV	.8250 (-1.06%)	.4342 (-0.94%)	.7912 (-1.21%)
w/o PER	.4600 (-44.83%)	**.4469** (+1.94%)	.5996(-25.13%)
w/o COMB	.8276 (-0.74%)	.4393 (+0.23%)	.7930(-0.99%)
w/o INT	.8313 (-0.30%)	.4367 (-0.36%)	**.8049**(+0.50%)
w BERT	.8322 (-0.19%)	.4367 (-0.36%)	.8032(+0.29%)

Table IX. These results show that PEPS+MMR achieves the best performance in S@k as it uses the pipeline way to rerank search results and MMR directly uses the document similarity to model diversification. Except for PEPS+MMR, our PER+DIV(u) outperforms PEPS by 1.7% relative improvement over PEPS in terms of S@3. The results of both subtopic (ERR-IA) and non-subtopic (S) metrics show that the diversification module of our model actually has positive effects on the final rankings and can improve the diversity of top results.

D. Ablation Study

In this section, we conduct an ablation study of the main modules in PER+DIV to verify their effectiveness. All these ablation models are trained in the unified method. These models are shown as follows:

w/o. doc Trm. We remove the document-level transformer $\mathrm{Trm}^{\mathrm{doc}}$ in Section III-A.

w/o. per Trm. We remove the two transformer structure $\mathrm{Trm}^{\mathrm{short}}$ and $\mathrm{Trm}^{\mathrm{long}}$ in the personalization module in Section III-B.

w/o. NTN. We remove the neural tensor network from the diversification module and calculate diversification score using the same way in SetRank [43] and DESA [9]:

$$S^{\mathrm{div}'}(d|D) = \phi(D^v[\mathrm{index}(d)]). \tag{17}$$

w/o. DIV, w/o. PER. We remove one of the diversification / personalization modules from the PER+DIV.

w/o. COMB. We remove the combination module measuring the weight $\lambda(q, U)$ in our framework and calculate the score by simply adding S^{per} and S^{div} together.

w/o. INT. We remove the interaction-based score $s^I(d^0, q^0), s^I(d^{\mathrm{int}}, q^{\mathrm{int}})$ in the final personalization score calculation in 11.

w. BERT. Since pre-trained language models have achieved great performance in other information retrieval tasks, in this ablation model, we try to incorporate BERT into our PER+DIV framework. However, due to the limitations of BERT's input length and long user histories, we can only use BERT to help calculate the relevance between current queries and scoring documents. More specifically, we add a relevance score $S^{\mathrm{BERT}}(d, q)$ calculated by BERT-based cross-encoder in 11.

The ablation results are shown in Table X. All these ablation models underperform our PER+DIV(u) model. Only using diversification module leads to the worst results in ablation models, which indicates that it is necessary to enhance click history. The w/o. DIV results using only the personalization score are lower than that of the unified model, but outperform all other baselines. This demonstrates the usefulness of considering the unclicked documents is useful in improving results satisfaction. The results w/o. NTN model show that the designed structure in diversification module actually benefits model performance. The SetRank way in (17) does not model dissimilarity explicitly. The low results of PER+DIV w/o. doc Trm may be explained by its failure to capture the click and position information in term-level transformer. Removing the personal-level transformers in PER+DIV w/o. per Trm also leads to a decrease in performance, especially in the personalization metric P-MAP, which indicates the effectiveness of the designed personalization module in modeling personal relevance. The results of w/o. COMB show that a vanilla summing strategy cannot achieve the best performance in the unified metrics. We need to take user's search history and current query into consideration to better estimate the combining weight. From the slightly decreased results of PER+DIV w/o. INT compared to the PER+DIV model, we can conclude that the main improvements of our proposed framework come from the designed hierarchical-transformer-based representation module, the personalization module, and the diversification module. The higher results on P-MAP of it may be due to the fact that we remove two ad-hoc interaction-based scores thus making the ranking list more personalized. The results of the BERT-enhanced model PER+DIV w. BERT show only comparable results to the original PER+DIV model. We state that the reason may be that the main point of our problem is to integrate personalization and diversification together, simply improving the adhoc relevance will not significantly improve the performances.

VII. CONCLUSION

In this paper, we proposed an integrated personalized and diversified framework PER+DIV to enhance both personal relevance and document novelty. We adopted a hierarchical transformer structure to extract information from historical logs and current candidates and to calculate the personalization and diversification scores of a document. These scores are then integrated through a combined weight estimated according to the similarity of query and user profile. We put forward two different training methods regarding user's click in mixed and separate ways respectively to better train our model. Experimental results showed that our model can significantly outperform all personalized search and search result diversification baselines in unified metrics. This paper shows that personalization and result diversification are two complementary approaches dealing with ambiguous queries that can be combined. Our work can be applied in web search situations to provide more satisfying results

for users. However, due to the high complexity of the PER+DIV framework, it can only be utilized in the final re-ranking stage.

There is still potential for improvement in combining personalization and diversification in the ranking area. User click behavior in web search situations is quite noisy and can be caused by a variety of elements, including both personal interests and document novelty. Therefore, it may be inaccurate to use them to evaluate personalization and diversification simultaneously. A better way may be to construct a new dataset that contains accurate signals for both sides.

REFERENCES

[1] C. Silverstein, M. R. Henzinger, H. Marais, and M. Moricz, "Analysis of a very large web search engine query log," *ACM SIGIR Forum*, vol. 33, no. 1, pp. 6–12, 1999.

[2] Y. Yano, Y. Tagami, and A. Tajima, "Quantifying query ambiguity with topic distributions," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1877–1880.

[3] J. G. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1998, pp. 335–336.

[4] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *Proc. 19th Int. Conf. World Wide Web*, New York, NY, USA, 2010, pp. 881–890.

[5] R. L. Santos, "Explicit web search result diversification," *ACM SIGIR Forum*, vol. 47, no. 1, pp. 67–68, Jun. 2012.

[6] Y. Zhu, Y. Lan, J. Guo, X. Cheng, and S. Niu, "Learning for search result diversification," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2014, pp. 293–302.

[7] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Learning maximal marginal relevance model via directly optimizing diversity evaluation measures," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2015, pp. 113–122.

[8] Z. Jiang, J.-R. Wen, Z. Dou, W. X. Zhao, J.-Y. Nie, and M. Yue, "Learning to diversify search results via subtopic attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2017, pp. 545–554.

[9] X. Qin, Z. Dou, and J.-R. Wen, "Diversifying search results using self-attention network," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2020, pp. 1265–1274.

[10] J. Liu, Z. Dou, X. Wang, S. Lu, and J.-R. Wen, "DVGAN: A minimax game for search result diversification combining explicit and implicit features," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2020, pp. 479–488.

[11] S. Yigit-Sert, I. S. Altingovde, C. Macdonald, I. Ounis, and Özgür Ulusoy, "Supervised approaches for explicit search result diversification," *Inf. Process. Manage.*, vol. 57, no. 6, 2020, Art. no. 102356.

[12] L. Yan, Z. Qin, R. K. Pasumarthi, X. Wang, and M. Bendersky, "Diversification-aware learning to rank using distributed representation," in *Proc. Web Conf.*, 2021, pp. 127–136.

[13] Z. Su, Z. Dou, Y. Zhu, X. Qin, and J. Wen, "Modeling intent graph for search result diversification," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 736–746.

[14] Z. Dou, R. Song, and J.-R. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proc. 16th Int. Conf. World Wide Web*, New York, NY, USA, 2007, pp. 581–590.

[15] P. N. Bennett et al., "Modeling the impact of short-and long-term behavior on search personalization," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2012, pp. 185–194.

[16] S. Ge, Z. Dou, Z. Jiang, J.-Y. Nie, and J.-R. Wen, "Personalizing search results using hierarchical RNN with query-aware attention," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2018, pp. 347–356.

[17] S. Lu, Z. Dou, X. Jun, J.-Y. Nie, and J.-R. Wen, "PSGAN: A minimax game for personalized search with limited and noisy click data," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2019, pp. 555–564.

[18] Z. Ma, Z. Dou, G. Bian, and J.-R. Wen, "PSTIE: Time information enhanced personalized search," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*New York, NY, USA, 2020, pp. 1075–1084.

[19] J. Yao, Z. Dou, and J.-R. Wen, "Employing personal word embeddings for personalized search," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2020, pp. 1359–1368.

[20] Y. Zhou, Z. Dou, and J.-R. Wen, "Encoding history with context-aware representation learning for personalized search," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2020, pp. 1111–1120.

[21] Y. Zhou, Z. Dou, Y. Zhu, and J. Wen, "PSSL: Self-supervised learning for personalized search with contrastive sampling," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2749–2758.

[22] Y. Zhou, Z. Dou, and J.-R. Wen, "Enhancing re-finding behavior with external memories for personalized search," in *Proc. 13th Int. Conf. Web Search Data Mining*, New York, NY, USA, 2020, pp. 789–797.

[23] F. Radlinski and S. Dumais, "Improving personalized web search using result diversification," in *Proc. 29th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2006, pp. 691–692.

[24] D. Vallet and P. Castells, "Personalized diversification of search results," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2012, pp. 841–850.

[25] S. Liang, Z. Ren, and M. de Rijke, "Personalized search result diversification via structured learning," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2014, pp. 751–760.

[26] C. Burges et al., "Learning to rank using gradient descent," in *Proc. 22nd Int. Conf. Mach. Learn.*, New York, NY, USA, 2005, pp. 89–96.

[27] M. Volkovs, "Context models for web search personalization," 2015, *arXiv:1502.00527*.

[28] M. J. Carman, F. Crestani, M. Harvey, and M. Baillie, "Towards query log based personalization using topic models," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2010, pp. 1849–1852.

[29] J. Yao, Z. Dou, J. Xu, and J. Wen, "RLPS: A reinforcement learning–based framework for personalized search," *ACM Trans. Inf. Syst.*, vol. 39, no. 3, pp. 1–29, 2021.

[30] C. Deng, Y. Zhou, and Z. Dou, "Improving personalized search with dual-feedback network," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 210–218.

[31] Y. Zhou, Z. Dou, B. Wei, R. Xie, and J. Wen, "Group based personalized search by integrating search behaviour and friend network," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 92–101.

[32] V. Dang and W. B. Croft, "Diversity by proportionality: An election-based approach to search result diversification," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2012, pp. 65–74.

[33] S. Hu, Z. Dou, X. Wang, T. Sakai, and J.-R. Wen, "Search result diversification based on hierarchical intents," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2015, pp. 63–72, doi: 10.1145/2806416.2806455.

[34] C. L. Clarke, M. Kolla, and O. Vechtomova, "An effectiveness measure for ambiguous and underspecified queries," in *Proc. 2nd Int. Conf. Theory Inf. Retrieval: Adv. Inf. Retrieval Theory*, 2009, pp. 188–199, doi: 10.1007/978-3-642-04417-5_17.

[35] J. I. Marden, *Analyzing and Modeling Rank Data*. Boca Raton, Florida, USA: CRC Press, 1996.

[36] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Modeling document novelty with neural tensor network for search result diversification," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2016, pp. 395–404.

[37] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[38] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[40] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2017, pp. 55–64.

[41] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen, "Multi-dimensional search result diversification," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, 2011, pp. 475–484.

[42] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2009, pp. 621–630.

[43] L. Pang, J. Xu, Q. Ai, Y. Lan, X. Cheng, and J. Wen, "SetRank: Learning a permutation-invariant ranking model for information retrieval," in *Proc. 43th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2020, pp. 499–508.

Jiongnan Liu received the BE degree in computer science and technology in 2017 from the Renmin University of China, Beijing, China, where he is currently working toward the PhD degree with the Gaoling School of Artificial Intelligence. His research interests include search result diversification, personalized search, and product search.

Zhicheng Dou (Member, IEEE) received the BS and PhD degrees in computer science and technology from the Nankai University, Tianjin, China, in 2003 and 2008, respectively. He is currently a professor with the Renmin University of China, Beijing, China. From July 2008 to September 2014, he was with Microsoft Research Asia. His current research interests are information retrieval, natural language processing, and big data analysis. He was the recipient of the Best Paper Runner-Up Award from SIGIR 2013, and Best Paper Award from AIRS 2012. He was the program co-chair of the short paper track for SIGIR 2019. His homepage is http://playbigdata.ruc.edu.cn/dou.

Jian-Yun Nie (Member, IEEE) is currently a professor with the University of Montreal, Montreal, ON, Canada. He has been an invited professor and researcher with several universities and companies. He has authored or coauthored more than 150 papers in information retrieval and natural language processing in journals and conferences. He was the general co-chair of the ACM SIGIR Conference in 2011. He is currently on the editorial board of seven international journals.

Ji-Rong Wen (Senior Member, IEEE) received the BS and MS degrees from the Renmin University of China, Beijing, China, and the Ph.D. degree from the Chinese Academy of Science, Beijing, in 1999. He is currently a professor with the Renmin University of China. From 2000 to 2014, he was a senior researcher and research manager with Microsoft Research. His main research interests include web data management, information retrieval (especially web IR), and data mining.