Knowledge and Information Systems (2024) 66:5581–5604
https://doi.org/10.1007/s10115-024-02138-y

REGULAR PAPER

How to personalize and whether to personalize? Candidate documents decide

Wenhan Liu[1] · Yujia Zhou[1] · Yutao Zhu[1] · Zhicheng Dou[1]

Received: 19 June 2023 / Revised: 14 April 2024 / Accepted: 3 May 2024 /
Published online: 27 May 2024

Abstract

Personalized search plays an important role in satisfying users' information needs owing to its ability to build user profiles based on users' search histories. Most of the existing personalized methods built dynamic user profiles by emphasizing query-related historical behaviors rather than treating each historical behavior equally. Sometimes, the ambiguity and short nature of the query make it difficult to understand the potential query intent exactly, and the query-centric user profiles built in these cases will be biased and inaccurate. In this work, we propose to leverage candidate documents, which contain richer information than the short query text, to help understand the query intent more accurately and improve the quality of user profiles afterward. Specifically, we intend to better understand the query intent through candidate documents, so that more relevant user behaviors from history can be selected to build more accurate user profiles. Moreover, by analyzing the differences between candidate documents, we can better control the degree of personalization on the ranking of results. This controlled personalization approach is also expected to further improve the stability of personalized search as blind personalization may harm the ranking results. We conduct extensive experiments on two datasets, and the results show that our model significantly outperforms competitive baselines, which confirms the benefit of utilizing candidate documents for personalized web search.

Keywords Personalized search · User profile · Query intent understanding · Candidate documents

✉ Zhicheng Dou
dou@ruc.edu.cn

Wenhan Liu
lwh@ruc.edu.cn

Yujia Zhou
zhouyujia@ruc.edu.cn

Yutao Zhu
yutaozhu94@gmail.com

[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

1 Introduction

Search engines play an important role in the process of obtaining information in our daily lives. However, for the same query, existing search engines usually return the same result to different users, which hardly meets the needs of different people. For example, for the query "Apple," a computer enthusiast tends to seek information related to "Apple computer" while a farmer might prefer "Apple fruit." Personalized search has been proposed to cope with this problem by re-ranking candidate documents based on user interests. Traditional personalized search studies [1–6] mainly focused on extracting human-designed personalized features from users' query logs to predict users' intents. In recent years, with the rapid development of deep learning, many neural models [7–11] have been proposed to build user profiles using neural networks to improve the personalization quality.

Existing personalized neural models mainly built user profiles by exploiting personalized signals, especially query-related search behaviors, from users' query logs. For example, HRNN [12] used the attention mechanism to highlight query-related historical behaviors to build dynamic user profiles. Along this line, we notice that the quality of such a "query-centric" user profile is highly dependent on the representation of the current query. However, due to the ambiguity and short nature of the query [13, 14], it is often difficult to create accurate query representation to reflect the potential intents of users, leading to an inaccurate and biased user profile. For example (see Fig. 1A), a programmer is interested in the new programming language Go and just issues a single-term query "go." Since the word "go" is quite general and its topic is very broad, it is hard to guarantee its representation can cover all related intents. In the case that the programming-related subtopic is ignored in the representation, we have no way to find relevant user histories and build correct profiles.

To alleviate the above problem, we attempt to enhance the representation of the query with its retrieved candidate documents, thereby improving the quality of the user profile. Candidate documents under a query are often regarded as a summary of the intent corresponding to the query [15–17]. Compared with a short query text, they can provide richer information for better understanding the potential query intents for personalization. Let us analyze this

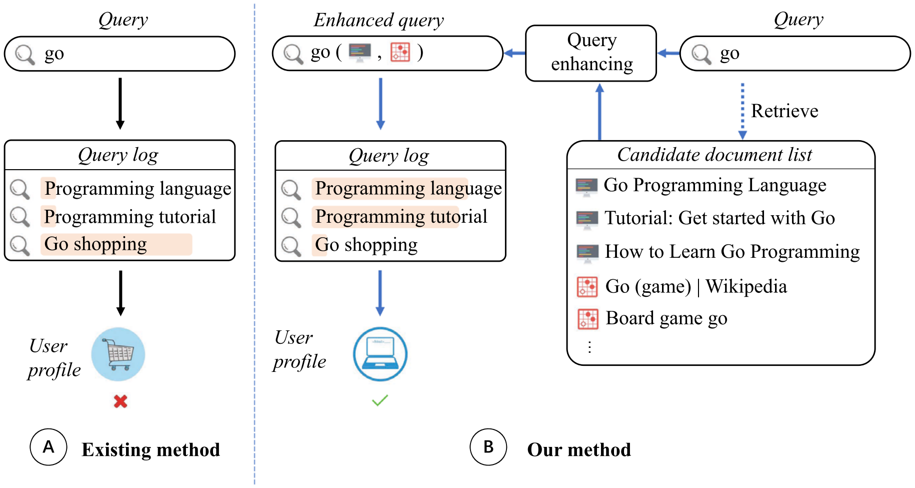

Fig. 1 **A** Existing user profile building method. **B** Our candidate documents-enhanced personalization method

benefit from two aspects: how to personalize and whether to personalize. For the first aspect, similar to some query classification studies [18, 19], using candidate documents can provide richer information on potential query intents to enhance the query representation, to extract more comprehensive and useful behaviors from user history logs (see Fig. 1B). For the "go" example, we mentioned above the information in the candidate documents indicates that "go" potentially has multiple intents: go programming language, board game go, and so on. Therefore, we can use this information to enrich the representation of the query "go." With the enriched query representation, more useful historical behaviors (especially programming-related ones) can be identified to build a more accurate user profile. For the second aspect, the semantic difference between candidate documents indicates whether personalization should be considered in document ranking. Intuitively, if candidate documents are semantically similar to one another or cover-related topics (which often occurs when the query intent is very clear), then there is no need to personalize the results. For example, the candidate documents of the query "go programming tutorial" may include "Go Programming Language Tutorials," "How to Learn Go Programming," and "Getting Started with Go programming." Due to the small semantic differences between these documents, personalizing the ranking of them is unnecessary. Blind personalization may even have side effects [20, 21] for such queries. This indicates that there should be a stronger correlation between the level of personalization and the semantic differences between documents. Consequently, we intend to adjust the degree of personalization by measuring the semantic difference across candidate documents.

Although candidate documents can provide rich query intent information, depending blindly on them all to improve query representation may introduce additional issues with topic distribution bias. For example, when multiple topics exist in candidate documents, their corresponding documents may vary greatly in number. As a result, the query representation neglects the user's real interest in minor subtopics and is dominated by major subtopics with more documents. To address this problem, we propose to select a diverse set of candidate documents to improve the diversity of the query representation, so that it can cover as many potential subtopics as possible.

Specifically, we propose a documents-enhanced personalized search framework (DEPS), which mines the semantic information of candidate documents to improve the personalization from two aspects. For the first aspect, we use a diverse document set selected from all the candidate documents to enhance the representation of the current query based on Transformer, to make the query representation cover the potential user's intents. Then, we use the enhanced query representation to highlight relevant historical behaviors based on the attention mechanism to build the user profile. For the second aspect, we design a difference-aware self-attention mechanism that helps to measure the semantic difference between the candidate documents, and use the difference to control the weight of personalization in the final ranking score.

We conduct extensive experiments with two widely used datasets for personalization, namely the AOL dataset and a query log dataset from a commercial search engine. Experimental results show that our model outperforms all existing baseline models and achieves a favorable trade-off between improved effectiveness and increased time latency. The results also confirm the benefit of utilizing candidate documents for either refining user profiles or controlling the degree of personalization.

The main contributions of this paper are summarized as follows: (1) We are the first to propose improving personalized search from the perspective of candidate documents, mainly following two questions: how to personalize and whether to personalize. (2) We utilize a diverse document set to enhance the query representation, so as to build a more accurate user profile. (3) We propose dynamically adjusting the degree of personalization according to the semantic difference between the candidate documents.

The rest of the paper is organized as follows. We first summarize the previous studies that are related to our paper in Sect. 2 and introduce the proposed method in Sect. 3. Then, experimental settings are described in Sect. 4, and the results are analyzed in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related work

2.1 Personalized search

Due to the ability of personalized search to meet the personalized information needs of different users, various personalization-related studies have been conducted. Some traditional models used click-based features to calculate the relevance score of the candidate document. Dou et al. [20] proposed the P-Click model to predict user's intent by counting the click on the same document in the search history. Teevan et al. [3] also extracted these click-based features from query logs to forecast users' future navigational behaviors. Apart from that, some studies [22, 23] tried to apply topic-based features extracted from the documents to model the users' interests. Some other studies used feature engineering to improve the quality of personalized search. They extracted click-based features, query entropy, and other features from the current query and the user's query log. Then, the learning-to-rank algorithm LambdaMART [24] is used to combine these features to train the ranking model.

In recent years, deep learning has been widely applied in information retrieval due to its powerful representation learning capability. For personalized search, it is usually applied to predict users' interests [8, 10–12, 25–27]. Song et al. [2] used an adaptive ranking model to build dynamic user profiles. Li et al. [28] used the semantic features of in-session contexts to improve the ranking results. In addition, many studies apply various network structures in personalized search. Ge et al. [12] used hierarchical recurrent neural networks with the attention mechanism to model the user interests. Ma et al. [25] proposed a fine-grained time-enhanced model based on LSTM to model a more accurate user profile. Zhou et al. [10] used the context of history to learn a better semantic representation of the current query. Deng et al. [29] applied a dual-feedback network that incorporated users' positive/negative behavior to better understand the user's intent. These methods extensively used the attention mechanism to filter users' historical behaviors based on the current query to build "query-centric" user profiles. Since the user history contains rich information, filtering historical behaviors through the current query has been proved to be an effective method. In this work, in addition to the current query, we further consider using its corresponding candidate documents for more accurate historical behavior selection. We believe that this method can help build more accurate and stable user profiles.

2.2 Pseudo-Relevance Feedback

Pseudo-Relevance Feedback is a technique used in the field of information retrieval to improve the results of a search query. The basic idea behind Pseudo-Relevance Feedback is to use the top-ranked documents to update the query language model and improve the ranking results. It has been applied in many retrieval models [30–32]. For example, Zhai and Lafferty [31] extracted topic information from feedback documents to improve text retrieval task. Ai et al.

[32] proposed to use the top retrieved documents to learn a deep listwise context model for learning-to-rank task. In this paper, we propose to utilize the candidate documents as a kind of Pseudo-Relevance Feedback in personalized search task. Different from aforementioned works that use all the candidate documents as feedback, we propose to select a diverse set from the candidate documents to avoid the problem of topic distribution bias.

2.3 Modeling interaction of documents for ranking

Recently, modeling the interaction of the candidate documents has been proved to be effective for ranking in IR. Some studies [33, 34] revealed that the inter-relationship between candidate documents helps model the query–document relevance. Ai et al. [32, 35] take multiple documents as the input of the scoring function and predict their ranking scores together. Besides, some researchers [36, 37] tried to capture the cross-document comparative information based on the self-attention mechanism to re-rank the documents. Qin et al. [38] proposed a supervised diversification framework that uses self-attention to model the interactions between all candidate documents globally in diversified search. The success of these studies shows that the difference between candidate documents reveals some ranking signals. Inspired by these works, we propose to design a difference-aware self-attention mechanism to better capture the semantic differences between candidate documents for personalized document ranking.

3 Proposed method: DEPS

Personalized search mainly improves ranking results by modeling user profiles based on users' search logs. As we stated in Sect. 1, the shortness and ambiguity of the query make its representation fail to reflect the potential intents of users, leading to the deviation of the user profile. Besides, the personalization incompatible with document semantic differences may degrade the ranking quality. In this paper, we propose to leverage the semantic information hidden in the candidate documents to address these two issues. Specifically, we use a diverse document set selected from the candidate documents to enhance the query representation. With the query representation enhanced, a more accurate user profile can be built based on the attention mechanism. Furthermore, we devise an attention-based method to measure the semantic difference between the documents and adjust the degree of personalization in the final ranking.

To begin with, we formulate the problem with notations (listed in Table 1). The search history H records the user's historical behaviors, including query requests and corresponding click actions. We represent the user's search history as a sequence $H = \{q_1, d_{1,1}^{+}, \ldots, q_{t-1}, d_{t-1,1}^{+}, \ldots\}$, where t is the current time and $d_{i,j}^{+}$ refers to the j-th clicked document under query q_i. Given the current query q, we use $D = \{d_1, d_2, \ldots, d_N\}$ to represent the corresponding candidate documents retrieved by the search engine. Our task is to score each candidate document based on the current query and the user's search history. Different from previous studies in which the candidate documents are only used for similarity matching, we attempt to mine the semantic information and their relationships hidden in the candidate document list to improve the ranking results. In this paper, in addition to the current query and user's search history, we also use the candidate documents as additional data to calculate the final personalized score. The final score of the i-th candidate document can be computed as:

$$\text{score}(d_i|q, H, D) = \phi\left(\text{Pscore}\left(d_i|q, H, D\right), \text{Ascore}(d_i|q)\right), \tag{1}$$

Table 1 Notations in our approaches

Symbol	Explanation
q_i	User query at time step i
$d^{+}_{i,j}$	The j-th clicked document under q_i
q	Current user query
D	Candidate document set of q
d_i	The i-th candidate document of q
D^{div}	Diverse document set selected from D
w^{p}_i	Personalized score weight of d_i
$\mathbf{q}^{\text{e}}$	Enhanced representation of q
$\mathbf{U}^{\text{e}}$	Enhanced user profile
$\mathbf{d}^{\text{agg}}_i$	Aggregated semantic representation of d_i
Pscore	Personalized score
Ascore	Ad hoc score

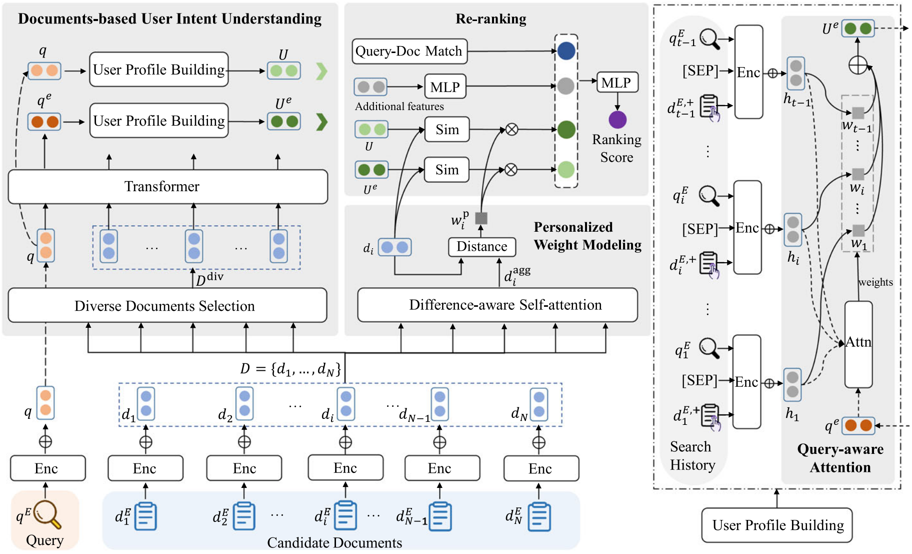

Fig. 2 Architecture of DEPS. Given the current candidate documents, a diverse document set is selected from them to enhance the topic coverage of the current query based on Transformer, so that a better user profile can be built with the enhanced query. Then, a difference-aware self-attention is designed to help measure the semantic difference between candidate documents and calculate a personalized score weight for each document. Finally, the ranking score is calculated with the assistance of several other features

where $\text{Pscore}(d_i|q, H, D)$ represents the personalized score of the i-th document and $\text{Ascore}(d_i|q)$ is the ad hoc score between the query and the i-th document. The function $\phi(\cdot)$ is a multilayer perceptron (MLP) using $\tanh(\cdot)$ as the activate function. The structure of our model is shown in Fig. 2. Next, we will introduce each part of our model in detail.

3.1 How to personalize: documents-based user intent understanding

As we discussed in Sect. 1, some queries lack semantic information due to their shortness and ambiguity, which hinders us from understanding the potential query intents. To cope with this problem, we try to select a diverse document set from candidate documents to enhance the potential topic coverage of the query representation. Then, we use the enhanced query representation to filter historical behaviors to better understand user's intent based on the attention mechanism. Specifically, we divide the whole process into four parts: (1) query/document encoding, (2) diverse documents selection, (3) documents-enhanced query representation, and (4) user profile building. We will introduce the details of each part in the following.

3.1.1 Query/document encoding

To get the embedding of the current query and corresponding candidate documents, we initialize a global word embedding matrix $\mathbf{M} \in \mathbb{R}^{|V| \times m}$, where $|V|$ is the vocabulary size and m represents the dimension of word vector. we use this matrix to convert each word in the query and documents into vectors.

For the current time t, we use $\mathbf{q}^E \in \mathbb{R}^{|q| \times m}$ and $\{\mathbf{d}_1^E, \ldots, \mathbf{d}_i^E, \ldots, \mathbf{d}_N^E\}$ $(\mathbf{d}_i^E \in \mathbb{R}^{|d_i| \times m})$ to represent the embeddings of the current query and the corresponding candidate documents. Then, we intend to learn their context-aware representations with Transformer [39] based on the entire text, which is defined as:

$$\mathbf{q} = \mathrm{Trm}^{\mathrm{sum}}(\mathbf{q}^E), \tag{2}$$

$$\mathbf{d}_i = \mathrm{Trm}^{\mathrm{sum}}(\mathbf{d}_i^E), \tag{3}$$

where $\mathrm{Trm}^{\mathrm{sum}}(\cdot)$ means the sum of outputs of Transformer. The obtained context-aware representations of candidate documents are denoted as $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$.

3.1.2 Diverse documents selection

As we stated in Sect. 1, enhancing the query representation with all candidate documents could make it ignore the minor subtopics in which the user's real intent may lie. For example, for the query "apple," its top result list contains significantly more results about the subtopic "Apple company" than the subtopic "apple fruit." If each result contributes equally to enhancing the query representation, the resulting query representation will be overwhelmed by "Apple company" because of the extreme imbalance in the result number, and it is hard to accurately capture the topic about "apple fruit." When a fruit farmer issues the query "apple," the biased query representation will likely fail to identify relevant information about "apple fruit" in the user's search history, and it will yield an inaccurate user profile.

Therefore, it is necessary to reduce the redundancy of candidate documents to balance the number of documents corresponding to different subtopics, so that the enhanced query representation can cover each subtopic more accurately. This will somewhat benefit different users who have diverse information needs. In this section, we attempt to select diverse documents from the candidate documents based on the Maximal Marginal Relevance (MMR) algorithm [40].

MMR is a greedy algorithm that strives to reduce document redundancy while maintaining query–document relevance in the search result diversification task. Its algorithm for selecting a document can be formulated as follows:

$$\mathbf{d}_s = \arg \max_{\mathbf{d}_i \in \mathbf{D} \backslash \mathbf{D}^{\text{div}}} \left[\lambda \, \text{Sim} \left(\mathbf{d}_i, \mathbf{q} \right) - (1 - \lambda) \max_{\mathbf{d}_j \in \mathbf{D}^{\text{div}}} \text{Sim} \left(\mathbf{d}_i, \mathbf{d}_j \right) \right], \tag{4}$$

where $\mathbf{D}$ is the candidate document set; $\mathbf{D}^{\text{div}}$ is the diverse document set already selected from $\mathbf{D}$; $\mathbf{D} \backslash \mathbf{D}^{\text{div}}$ represents the set difference, i.e., the documents that have not yet been selected from $\mathbf{D}$; $\text{Sim}(\cdot, \cdot)$ is the similarity metric used in matching and is implemented as the cosine similarity in this work; λ is used to adjust the query–document relevance and document diversity of the selected document set. To ensure the diversity of the selected document set, we set a document similarity threshold θ. When the minimum similarity between selected documents $\mathbf{D}^{\text{div}}$ and unselected documents $\mathbf{D} \backslash \mathbf{D}^{\text{div}}$ exceeds θ, we stop the document selection. We tune the parameter λ and document similarity threshold θ by grid search and finally set them as 0.3 and 0.75, respectively, in this paper. The overall process of our diverse document selection is summarized as Algorithm 1.

Algorithm 1 Diverse documents selection based on MMR

```
Input: candidate document set $\mathbf{D}$; the document similarity threshold $\theta$.
Output: diverse document set $\mathbf{D}^{\text{div}}$.
$\mathbf{D}^{\text{div}} \leftarrow \{\mathbf{d}_1\}$ // initialize $\mathbf{D}^{\text{div}}$ with $\mathbf{d}_1$
while $\mathbf{D} \backslash \mathbf{D}^{\text{div}}$ do
    $\mathbf{d}_s = \arg\max_{\mathbf{d}_i \in \mathbf{D} \backslash \mathbf{D}^{\text{div}}} \left[ \lambda \, \text{Sim}(\mathbf{d}_i, \mathbf{q}) - (1 - \lambda) \max_{\mathbf{d}_j \in \mathbf{D}^{\text{div}}} \text{Sim}(\mathbf{d}_i, \mathbf{d}_j) \right]$
    if $\max_{\mathbf{d}_j \in \mathbf{D}^{\text{div}}} \text{Sim}(\mathbf{d}_s, \mathbf{d}_j) > \theta$ then
        return $\mathbf{D}^{\text{div}}$
    end if
    $\mathbf{D}^{\text{div}} \leftarrow \mathbf{D}^{\text{div}} \cup \{\mathbf{d}_s\}$
end while
return $\mathbf{D}^{\text{div}}$
```

After the selection, we have obtained the diverse document set $\mathbf{D}^{\text{div}}$ which will be used to enhance the query representation in the next stage.

3.1.3 Documents-enhanced query representation

As we have discussed in Sect. 1, the ambiguity and short nature of the query hinder it from accurately representing users' potential intents. In this part, we intend to use the diverse document set to enhance the query representation based on Transformer, so that the query representation can more accurately reflect the potential intents of users. We put the query representation $\mathbf{q}$ and the diverse document set $\mathbf{D}^{\text{div}} = \{\mathbf{d}_1^{\text{div}}, ..., \mathbf{d}_n^{\text{div}}\}$ as the input of the Transformer.

$$\mathbf{q}^{\text{e}} = \text{Trm}^{\text{f}}([\mathbf{q}, \mathbf{D}^{\text{div}}]), \tag{5}$$

where $\text{Trm}^{\text{f}}(\cdot)$ means merely taking the output in the first position; $\mathbf{q}^{\text{e}}$ represents the enhanced query representation and will be used to build the user profile in the following section.

3.1.4 User profile building

Now that we have obtained the enhanced query representation, we attempt to use it to filter the historical search behaviors to build an accurate user profile. Ge et al. [12] revealed that different search behaviors could contribute differently to building user profiles and their weights should be determined by their relevance to the current query. In this part, we design a user profile building module based on the query-aware attention mechanism. The details are as follows:

Formally, for each query in the search history, we concatenate the word embeddings of the query and corresponding clicked documents, with "[SEP]" token as the separator. Then we feed them into Transformer and sum the outputs together to get the representation of the i-th historical behavior, which is denoted as $\mathbf{h}_i$.

$$\mathbf{h}_i = \mathrm{Trm}^{\mathrm{sum}}\left(\mathbf{q}_i^E, [\mathrm{SEP}], \mathbf{d}_{i,1}^{E,+}, [\mathrm{SEP}], \ldots, \mathbf{d}_{i,C}^{E,+}\right), \tag{6}$$

where C refers to the number of clicked documents under q_i.

Then we calculate weights $\{w_1, \ldots, w_{t-1}\}$ for each historical behaviors vector in $\{\mathbf{h}_1, \ldots, \mathbf{h}_{t-1}\}$:

$$x_i = \mathbf{q}^e \cdot \mathbf{h}_i, \tag{7}$$

$$w_i = \frac{\exp(x_i)}{\sum_{j=1}^{t-1} \exp(x_j)}. \tag{8}$$

Then the enhanced user profile $\mathbf{U}^{\mathrm{e}}$ can be computed by a weighted linear combination of $\{\mathbf{h}_1, \ldots, \mathbf{h}_{t-1}\}$:

$$\mathbf{U}^{\mathrm{e}} = \sum_{i=1}^{t-1} w_i \mathbf{h}_i. \tag{9}$$

Compared with the enhanced query representation $\mathbf{q}^{\mathrm{e}}$, we believe that the original query representation $\mathbf{q}$ still contains some useful information. Thus, we also use $\mathbf{q}$ to build an original user profile $\mathbf{U}$ in the same way as $\mathbf{U}^{\mathrm{e}}$, and the two user profiles will contribute to computing the final personalized score in Sect. 3.3.

3.2 Whether to personalize: personalized weight modeling

As we stated in Sect. 1, the degree of personalization should be adaptive to the semantic difference between the candidate documents. In this section, for each candidate document, we intend to measure its semantic difference from other candidate documents, thereby adjusting the effect of personalization on its ranking. Specifically, we design a difference-aware self-attention mechanism (denoted as DifAttn) that takes the representations of each document as the input and aggregates the semantic representations from other documents based on the Euclidean distance (see Fig. 3). Then we compute a personalized weight based on the semantic difference between each document and the corresponding aggregated semantics. The details of the implementation are as follows:

3.2.1 Difference-aware self-attention mechanism

The traditional self-attention mechanism mainly transmits similar information in the sequence through the dot product. In this part, to capture the documents with greater semantic differences from each document, we replace the dot product function with an Euclidean

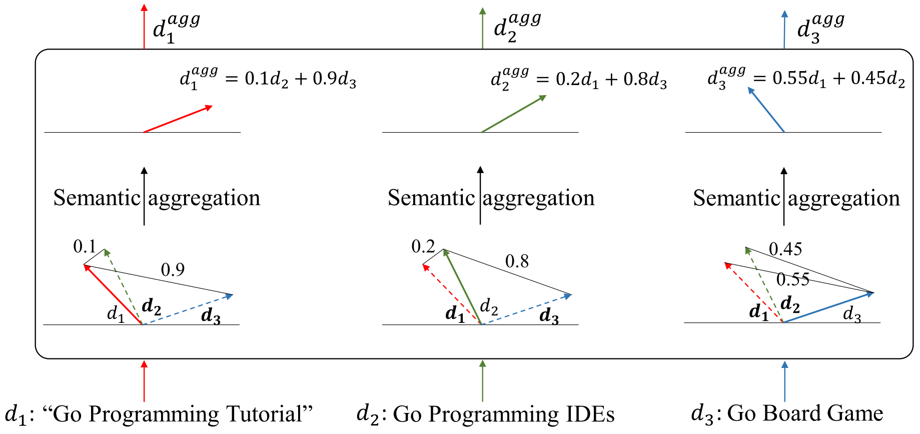

Fig. 3 An example of using DifAttn for semantic aggregation

distance-based function $f(\cdot, \cdot)$.

$$\text{DifAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{f(\mathbf{Q}, \mathbf{K})}{\sqrt{d/h}}\right)\mathbf{V}, \tag{10}$$

$$f(\mathbf{Q}, \mathbf{K}) = \begin{bmatrix} \|\mathbf{Q}_1 - \mathbf{K}_1\|_2 & \cdots & \|\mathbf{Q}_1 - \mathbf{K}_{N_k}\|_2 \\ \vdots & \ddots & \vdots \\ \|\mathbf{Q}_{N_q} - \mathbf{K}_1\|_2 & \cdots & \|\mathbf{Q}_{N_q} - \mathbf{K}_{N_k}\|_2 \end{bmatrix}, \tag{11}$$

where $\mathbf{Q} \in \mathbb{R}^{N_q \times E}$, $\mathbf{K} \in \mathbb{R}^{N_k \times E}$ and $\mathbf{V} \in \mathbb{R}^{N_k \times E}$ denote the query, key, and value matrices of the attention mechanism. Following by previous studies [36, 39], we use the multi-head self-attention (denoted as MS) to learn multiple aspects of different documents. The MS will first project the inputs into h subspaces with the dimension $\hat{\mathbf{E}} = \mathbf{E}/h$ and employ the DifAttn$(\cdot)$ for each head. Then the final output is obtained by concatenating each head.

$$\begin{aligned} \mathbf{D}^{\text{agg}} &= \text{MS}(\mathbf{D}) = [\text{head}_1, \ldots, \text{head}_h]\mathbf{W}^O, \\ \text{head}_i &= \text{DifAttn}(\mathbf{D}\mathbf{W}_i^Q, \mathbf{D}\mathbf{W}_i^K, \mathbf{D}\mathbf{W}_i^V), \end{aligned} \tag{12}$$

where $\mathbf{D}^{\text{agg}} = \{\mathbf{d}_1^{\text{agg}}, \ldots, \mathbf{d}_N^{\text{agg}}\}$ are the aggregated semantic representation. $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ are the representations of all the candidate documents obtained in Sect. 3.1.1 and the projection matrices of each head $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$, and $\mathbf{W}^O$ are parameters learned during the training. To remove each document's attention to itself when computing semantic differences, we mask out (setting to -∞) its attention value in the input of the softmax.

3.2.2 Personalized score weight

As we discussed in Sect. 1, the more significant the semantic difference between documents, the greater the degree of personalization should be. In this part, we propose computing a weight for each document to adjust its final personalized score. Formally, for the i-th candidate document, its personalized weight w_i^{p} is calculated by the Euclidean distance between $\mathbf{d}_i$ and $\mathbf{d}_i^{\text{agg}}$. Then we use sigmoid$(\cdot)$ to make w_i^{p} between 0 and 1.

$$w_i^{\text{p}} = \sigma(\|\mathbf{d}_i - \mathbf{d}_i^{\text{agg}}\|_2). \tag{13}$$

3.3 Re-ranking

The final ranking score of each candidate document consists of two parts. For the personalized relevance, we compute the similarity between the document representation d_i and user profiles obtained in Sect. 3.1 and multiply them by the personalized weight w_i^{p}:

$$\mathrm{Pscore}(d_i|q, H, D) = \phi\left(\mathrm{Sim}(\mathbf{U}^{\mathrm{e}}, \mathbf{d}_i) \cdot w_i^{\mathrm{p}}, \mathrm{Sim}(\mathbf{U}, \mathbf{d}_i) \cdot w_i^{\mathrm{p}}\right), \tag{14}$$

where $\mathbf{U}^{\mathrm{e}}$ and $\mathbf{U}$ represent the user profiles built by the enhanced query and the original query, respectively, in Sect. 3.1.4. For the ad hoc relevance, we divide it into two parts: (1) we consider the interaction-based and representation-based similarity between the query and document matching of the original query $\mathbf{q}^E$ and document $\mathbf{d}_i^E$, and their context-aware representations $\mathbf{q}$ and $\mathbf{d}_i$; (2) we follow [10] and extract some additional features f_{q,d_i} for each candidate document, including clicks features, topic features, and some neural matching features. These features are also fed into MLP to compute a relevance score:

$$\mathrm{Ascore}(d_i|q) = \phi\left(s^{\mathrm{I}}(\mathbf{q}^E, \mathbf{d}_i^E), s^{\mathrm{R}}(\mathbf{q}, \mathbf{d}_i), \phi(f_{q,d_i})\right), \tag{15}$$

where $s^{\mathrm{I}}(\cdot)$ and $s^{\mathrm{R}}(\cdot)$ are implemented as KNRM model and cosine similarity, respectively, and $\phi(\cdot)$ is implemented as multilayer perceptron.

We adopt pairwise learning-to-rank algorithm LambdaRank [41] to train our model. We construct a training pair with a positive sample (the clicked document) and a negative sample (the unclicked document). Given a positive sample d_i and a negative sample d_j, the probability that d_i is more relevant than d_j is computed as

$$P_{ij} = \frac{1}{1 + e^{-(\mathrm{score}(d_i) - \mathrm{score}(d_j))}}, \tag{16}$$

where score$(\cdot)$ calculates the final score of the document. The final loss function is defined as the weighted cross entropy between the ground truth $\overline{P}_{ij}$ and predicted probability P_{ij}:

$$\mathcal{L} = \left|\lambda_{ij}\right| \left(-\overline{P}_{ij} \log\left(P_{ij}\right) - \overline{P}_{ji} \log\left(P_{ji}\right)\right), \tag{17}$$

where the weight λ_{ij} is the change value when swapping the position of d_i and d_j.

4 Experimental setup

4.1 Dataset

We use AOL search log [42] and the dataset from a commercial search engine (abbreviated as Commercial dataset in the following) to conduct our experiments. Table 2 shows the detailed statistics of both datasets.

AOL dataset is a publicly available dataset that includes three months (from March 1, 2006, to May 31, 2006) of user query and click data. Since the dataset only contains the documents that the user clicked on, we select the candidate documents from the top documents recalled by BM25 algorithm [43]. Following [44], we split the query log into sessions whose boundaries are decided by the similarity between two consecutive queries. Each piece of data contains an anonymous user ID, a query text, the time when the query is issued, a candidate document, and a click tag. Since the personalized search relies on the search history, we divide the whole

Table 2 Statistics of two datasets

Type	AOL dataset	Commercial dataset
# Days	91	58
# Users	110,439	33,204
# Queries	736,454	2,665,625
# Sessions	279,930	654,776
Average Query Length	2.87	3.25
Average Session Length	2.55	2.63
Average # Click per Query	1.11	0.46

dataset into two parts: historical data and experimental data. Specifically, the first five weeks of data correspond to historical data which contributes to the personalized search. The last eight weeks' data are considered experimental data which are further divided into training data and test data at a ratio of 5:1. For each query, we sample 5 candidate documents for training and 50 candidate documents for testing following [45, 46]. We only use the document title to calculate the relevance between the query and the document and remove users who did not have historical or training data.

Commercial dataset contains search logs from January to February 2013 without applying personalization technology. Each piece of data contains a user ID, a query text, the time when the query is issued, the URLs of the top-20 retrieved documents, the click label of each URL, and the corresponding dwell time. The dataset differs from the AOL dataset in a few ways. Firstly, the candidate documents are directly retrieved by the search engine, making the original ranking quality much higher than BM25. Secondly, we crawl the content of the document according to its URL to represent the document, which makes the document representation more accurate than just using the document title. Lastly, this dataset contains the click dwell time, so we regard the click whose dwell time is more than 30 s or the last click as a satisfied click. We regard 30 min of inactivity as the boundary, based on which we segment the search log into different sessions.

4.2 Baselines

For AOL dataset, the original rankings are generated based on the classical BM25 algorithm. For the Commercial dataset, the original ranking results are directly returned by a commercial search engine. In addition to the original rankings, we also compare our model with several ad hoc search baselines and personalized search baselines. The details of these baselines are listed as follows:

KNRM [47]. KNRM is a kernel-based neural ranking model. It builds a word-level similarity matrix between query and document and uses a kernel pooling technique to extract multi-level soft match signals from it. Then, a learning-to-rank algorithm is used to map these features into the final ranking score.

Conv-KNRM [48]. Conv-KNRM is proposed based on the KNRM model. It first utilizes convolutional neural networks to model n-gram soft matches for ad hoc search. The kernel pooling and learning-to-rank algorithm are applied to calculate the final ranking score.

BERT [49]. This model matches a query and a document based on the pre-trained BERT model. It takes the concatenated query–document sequence as the input and regards the features of “[CLS]” token as the matching signals.

P-Click [20]. P-Click re-ranks the candidate documents based on the number of times the user clicked on the same document in the search history, which is inspired by the user’s re-finding behaviors during the search process.

HRNN [12]. It employs a hierarchical recurrent neural network to model the sequential information underlying user history and dynamically generates the user profile based on a query-aware attention mechanism. Then, it re-ranks the candidate documents based on their relevance with the short-term and long-term user profiles.

PSGAN [7]. PSGAN is a personalized framework for overcoming the problem of noisy training data based on a generative adversarial network. It can generate queries that better match users’ search intents and select better document pairs for modeling user interests. We use the trained discriminator for the re-ranking task.

RPMN [11]. This study proposes to construct memory networks (MN) to identify complex re-finding behavior. It can build a fine-grained user model dynamically based on current information and use the model to re-rank the documents.

HTPS [10]. This model applies a hierarchical Transformer to encode the search history and disambiguate the user’s query in multiple stages. Besides, a personalized language model is designed to predict the user intent accurately.

PEPS [8]. The PEPS model uses historical data to train a personalized word embedding for each user. It proposes to improve the performance of personalized search based on better data representation instead of the user profile.

4.3 Evaluation metrics

For the AOL dataset and Commercial dataset, we regard the clicked documents and satisfied documents as relevant documents and label the others as irrelevant. We apply three common evaluation metrics to evaluate the models: mean average precise (MAP), mean reciprocal rank (MRR), and precision@1 (P@1). Because users’ click behavior may be influenced by the original order and some relevant documents may be ignored due to their low rankings, we use a more credible metric P-improve [7] as the fourth evaluation metric to measure the ranking results more objectively. We calculate P-improve as the ratio of increased correct pairs compared with the original ranking results. The more detailed explanation can be referred to in [7]. Since the candidate documents in the AOL dataset are recalled by BM25 and are not presented to users, we only use this metric on the Commercial dataset whose candidate documents are directly retrieved by the search engine. These evaluation metrics are widely used in the field of personalized search. Both the evaluation metrics and datasets we mentioned in Sect. 4.1 align with those used in the baselines, ensuring a fair comparison with these baselines.

4.4 Implementation details

We implement our model with Pytorch and carry out a series of experiments to determine the parameters of the model. The word embedding is set as 100. As for the Transformer, the hidden size is 512 and the number of heads in the multi-head attention mechanism is set as 8. The whole model is optimized by Adam, with a batch size of 32 and a learning rate of 9e−5.

5 Results and analysis

5.1 Overall performance comparison

The overall results of different models on the two datasets are shown in Table 3. It can be observed that:

(1) The comparison of our model and baselines. Our DEPS model outperforms all the baseline models on the two datasets. DEPS shows significant improvements in terms of all the evaluation metrics with paired t-test at $p < 0.05$ level, especially compared with competitive baselines PEPS and RPMN. Specifically, when compared to the best baseline model PEPS on the AOL dataset, our model achieves a 5.6% increase in MAP. Moreover, it exceeds RPMN by 1.0% on the Commercial dataset, further proving its superiority in performance. The reason for the improvement reduction on the Commercial dataset is that the original quality of the Commercial dataset is much higher than the AOL dataset, and it is difficult to improve the results. Therefore, the P-improve on which our model DEPS increases by 5.5% is more creditable. The significant performance improvement in the two datasets proves that making use of the semantic information of candidate documents is effective for improving search quality.

(2) The comparison of different datasets. Compared with the AOL dataset, the Commercial dataset has a much higher origin ranking quality, which makes the ad hoc search baselines perform worse than the original ranking. On the AOL dataset, with rich interactive matching signals between the query and the document, the model HTPS and PEPS outperform RPMN significantly, while on the Commercial dataset, RPMN performs better. This proves that the AOL dataset mainly evaluates the methods of modeling user interests and query–document matching, while the Commercial dataset focuses on the model's capability of capturing personalized signals. Our model outperforms PEPS and RPMN significantly on both datasets, which further confirms the robustness of the our proposed DEPS model.

In summary, the experimental results indicate that the candidate documents can further improve personalization by enhancing query representation and adjusting the personalized scores. For a more detailed analysis of our model, we conduct a series of supplementary experiments: ablation studies and experiments on different query sets.

5.2 Ablation experiments

Our DEPS model includes several main components: the enhanced query representation q^e, the personalized weight w^p, and the diverse documents selection module. To analyze the role of each part, we conduct several ablation experiments on two datasets. The details of the ablation models are as follows:

DEPS w/o. EQR We abandon the enhanced query representation q^e and only use the original query representation q to build the user profile and compute the personalized score.

DEPS w/o. PW We discard the personalized weight and calculate the personalized scores only by matching the user profiles to the documents.

Table 3 Performance comparison of all models on the AOL dataset and commercial dataset

Dataset	Model	MAP	(%)	MRR	(%)	P@1	(%)	P-improve	(%)
AOL	Ori	.2504	−64.9	.2596	−64.2	.1534	−75.6	–	–
	KNRM	.4291	−39.8	.4391	−39.5	.2704	−56.9	–	–
	Conv-KNRM	.4738	−33.5	.4849	−33.2	.3266	−48.0	–	–
	BERT	.5033	−29.4	.5135	−29.3	.3552	−43.4	–	–
	P-Click	.4224	−40.7	.4298	40.8	.3788	−39.7	-	-
	HRNN	.5423	−23.9	.5545	−23.6	.4854	−22.7	–	–
	PSGAN	.5480	−23.1	.5601	−22.8	.4892	−22.1	–	–
	RPMN	.5926	−16.9	.6049	−16.7	.5322	−15.2	–	–
	HTPS	.7091	−0.5	.7251	−0.1	.6268	−0.2	–	–
	PEPS	.7127	–	.7258	–	.6279	–	–	–
	DEPS (Ours)	**.7527**†	+5.6	**.7655**†	+5.5	**.6698**†	+6.7	–	–
Commercial	Ori	.7399	−10.2	.7506	−10.0	.6162	−15.6	–	–
	KNRM	.4916	−40.3	.5001	−40.1	.2849	−61.0	.0655	−75.3
	Conv-KNRM	.5872	−28.7	.5977	−28.4	.4188	−42.7	.1422	−46.5
	BERT	.6232	−24.4	.6326	−24.2	.4475	−38.7	.1778	−33.1
	P-Click	.7509	−8.8	.7634	−8.5	.6260	−14.3	.0611	−77.0%
	HRNN	.8065	−2.1	.8191	−1.8	.7127	−2.4	.2404	−9.5
	PSGAN	.8135	−1.3	.8234	−1.3	.7174	−1.8	.2489	−6.3
	RPMN	.8238	–	.8342	–	.7305	–	.2656	–
	HTPS	.8224	−0.2	.8324	−0.2	.7286	−0.3	.2552	−3.9
	PEPS	.8221	−0.2	.8321	−0.3	.7251	−0.7	.2545	−4.2
	DEPS (Ours)	**.8322**†	+1.0	**.8423**†	+1.0	**.7394**†	+1.2	**.2802**	+5.5

Note that since the candidate documents of the AOL dataset are not presented to users, it is not suitable to use the P-improving metric on the AOL dataset. The percentage represents improvements based on the SOTA baseline. "†" indicates the model outperforms all baselines significantly with paired t-test at $p < 0.05$ level. The best results are in bold

DEPS w/o. DDS We remove the diverse documents selection module and use all the candidate documents to enhance the query.

The experimental results are shown in Table 4. Removing any component of our model will damage the results on different datasets. Specifically, abandoning the enhanced query representation q^e (EQR) causes the most decline in each metric, which confirms that the candidate documents can effectively enhance query representation. Besides, without the personalized weight w^{p} (PW), the MAP, MRR, and P@1 metrics exhibit declines of 0.8%, 0.7%, and 0.9%, respectively, on the AOL dataset. This highlights the significance of adjusting the personalization degree according to the semantic difference among candidate documents, thereby proving the effectiveness of our approach. Furthermore, considering the time costs of the diverse documents selection module (DDS), we attempt to remove it and test the performance of our model. The results show that the gap between our model and the SOTA baseline is still large even if the model loses 0.4% in MAP on the AOL dataset.

Effectiveness and time latency discussion In this part, we intend to discuss the trade-off between effectiveness and increased time latency of our model. To achieve this, we measure the time latency with average query latency for the three main modules of our model. For DEPS w/o. DDS, DEPS w/o. EQR, and DEPS w/o. PW, we remove the Diverse Documents Selection module, Transformer module, and Difference-aware Self-attention module in our model (see Fig. 2), respectively. We conduct each latency measurement on a single GPU (16 G NVIDIA Tesla V100) and use the same batch size for all experiments to guarantee a fair comparison. The experimental results are shown in Table 4. Following the removal of each module, our experiments show a time latency reduction of several milliseconds. This reduction indicates that the latency introduced by each module is indeed moderate. Based on the effectiveness improvement in each module, we believe that the time latency they introduce is acceptable. Besides, the time latency on the AOL dataset is higher compared to the commercial dataset. This discrepancy can be attributed to the larger number of candidate documents associated with each query in the AOL dataset. Even so, our model maintains a low time latency on the AOL dataset, demonstrating its potential for online applications. Overall, we believe that our model achieves a favorable trade-off between effectiveness and time latency.

5.3 Performance on different query sets

To further explore how our model improves the ranking results, we divide the test data into different subsets based on two different scenarios and compare the improvement in metric MAP on different models on the AOL dataset. The details are as follows.

Ambiguous and non-ambiguous queries In this part, we investigate the model performance on ambiguous and unambiguous queries, respectively. For an ambiguous query (such as "Apple"), it usually has multiple subtopics and different users may have different intents. As we discussed in Sect. 1, ambiguous queries have more potential for personalization, and applying personalization to non-ambiguous queries may hurt the search quality. The query ambiguity is measured by click entropy. As we mentioned in Sect. 1, applying personalization to queries with low click entropy may hurt the search quality. Thus, we compute the click entropy of all queries and divide the whole query set into an unambiguous query set (click entropy < 1) and an ambiguous query set (click entropy$\geq$1).

We choose PSGAN, HTPS, and PEPS as the baselines, and the experimental results are shown in Fig. 4. The delta MAP represents the improvement relative to the original ranking.

Table 4 Experimental results of ablation models on AOL dataset and Commercial dataset

Dataset	Model	La (ms/q)	MAP	(%)	MRR	(%)	P@1	(%)	P-improve	(%)
AOL	w/o.EQR	17.48	.7355	−2.3	.7484	−2.2	.6471	−3.4	–	–
	w/o.PW	18.05	.7470	−0.8	.7600	−0.7	.6638	−0.9	–	–
	w/o.DDS	18.87	.7494	−0.4	.7621	−0.4	.6635	−0.9	–	–
	DEPS	20.89	**.7527**	–	**.7655**	–	**.6698**	–	–	–
Commercial	w/o.EQR	2.17	.8227	−1.1	.8333	−1.0	.7278	−1.4	.2655	−5.5
	w/o.PW	2.32	.8282	−0.4	.8388	−0.3	.7356	−0.4	.2736	−2.6
	w/o.DDS	2.06	.8310	−0.1	.8414	−0.1	.7384	−0.1	.2787	−0.5
	DEPS	2.42	**.8322**	–	**.8423**	–	**.7394**	–	**.2802**	–

La (ms/q) represents the average query latency, in milliseconds

The best results on each dataset are marked in bold

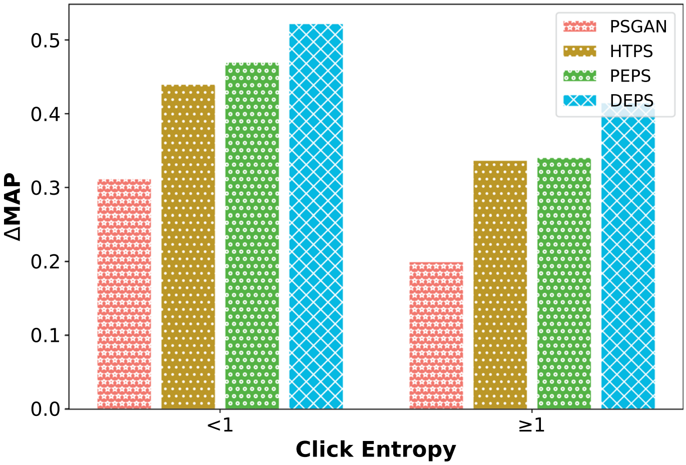

Fig. 4 Experimental results on unambiguous query set (click entropy $<$ 1) and ambiguous query set (click entropy $\geq$ 1)

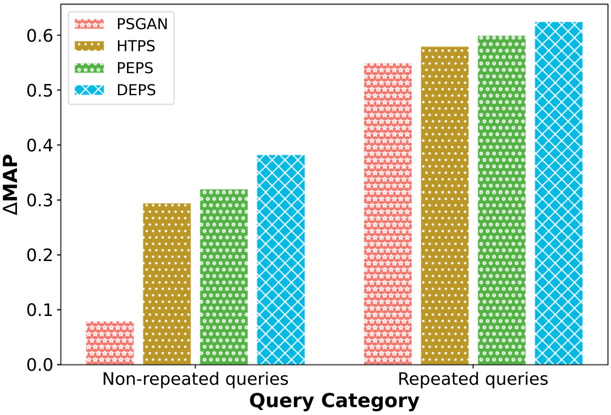

Fig. 5 Experimental results on repeated query set and non-repeated query set

We can see that all the models improve the MAP on both query sets, which shows that proper personalization is effective for both kinds of queries. Besides, our model outperforms the best baseline PEPS, especially on the ambiguous query set. This indicates that candidate documents can enrich the potential subtopics of the current query and the quality of query-centric user profiles can also be improved.

Repeated and non-repeated queries We also categorize the query set into repeated and non-repeated queries. For repeated queries, a more accurate user profile can be built based on the click behaviors on the same query in the past. But for the non-repeated queries, there is no identical historical search behavior to refer to, which has greater difficulty in predicting user intent. The experimental results are shown in Fig. 5.

The results indicate that all models have better performance on the repeated queries. This demonstrates that most personalized models can improve personalization by capturing the user's re-finding behaviors. Besides, our DEPS outperforms all the models on both

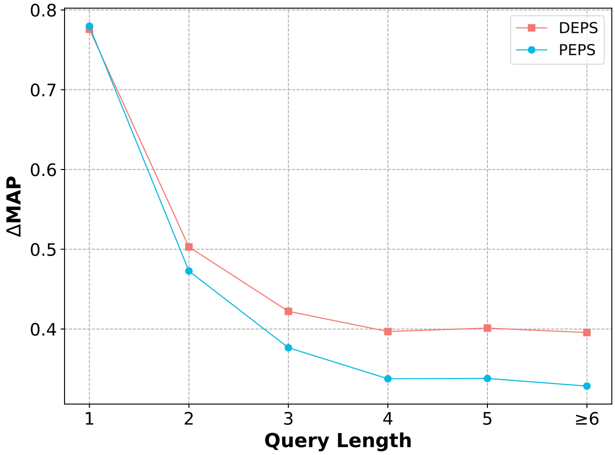

Fig. 6 Experimental results on different query lengths

query sets and the improvements on the non-repeated queries are more obvious. This phenomenon means that our model can better improve the ranking results by mining the semantic information hidden in the candidate list when facing new queries.

Queries of different lengths According to our statistics, about 45.5% of issued queries contain only one or two words. The shorter the query is, the less intent information the query representation contains. As we stated in Sect. 1, candidate documents provide sufficient information that reveals the potential query intents. To further demonstrate the effects of our model, it is worthwhile to test our model on queries of different lengths.

We choose PEPS as our baseline model, and the comparison result is shown in Fig. 6. We observe that our model performs better on short queries, especially those with a length of 1 or 2. This is because short queries often lack semantic information, so by enhancing their semantics with candidate documents, they can get more personalized improvements. Another observation is that our model DEPS outperforms the baseline model PEPS on almost all lengths of queries, which further demonstrates the effectiveness of our model.

5.4 Generalization and scalability

The use of candidate documents to enhance query understanding and user profile accuracy is applicable across various domains, as it does not rely on domain-specific features but on the inherent content and semantic relationships within the documents. This allows our method to be generalized to different domains. Besides, our model's architecture is based on Transformer, so it can process large sequences of data, ensuring that our approach can handle extensive search logs. Furthermore, our model can also achieve a favorable trade-off between improved effectiveness and increased time latency when the number of candidate documents increases. Therefore, our method also possesses good scalability.

5.5 Limitations and future work

Our method incorporates modeling of the query's retrieved candidate documents, which introduces additional time latency. As the number of candidate documents increases, there is a corresponding increase in time latency. To mitigate this issue, we intend to design more sophisticated methods for efficiently modeling candidate documents. Besides, to address the issue of topic distribution bias while enhancing the query representation with candidate documents, we designed a diverse document selection module, which is still a rule-based method. In the future, we intend to explore some deep learning-based methods for incorporating candidate documents into the personalization process more effectively. Furthermore, considering the rapid development of large language models (LLMs), using LLMs to analyze the search history and build more accurate user profiles is also a promising direction. The application of LLMs may also be extended to the interpretation of candidate documents themselves, providing a deeper semantic analysis that can further refine the personalization process.

6 Conclusion

In this work, based on the candidate documents, we designed a personalized search framework that explores two questions worth considering in the field of personalization: how to personalize and whether to personalize. For the first question, we proposed using candidate documents to broaden the topic coverage of the current query; hence, more accurate user profile can be built based on the enhanced query. For the second question, we designed a difference-aware self-attention mechanism to capture the semantic difference between candidate documents and calculate a personalized weight to adjust the final personalized score for each document. Our experiments confirmed the effectiveness of our framework for personalized search.

Author Contributions Wenhan Liu proposed the method, conducted the experiments, and wrote manuscripts. Yujia Zhou and Yutao Zhu revised the manuscript. Zhicheng Dou provided the experimental resources and revised the manuscript. All authors reviewed the manuscript.

Funding This work was supported by National Natural Science Foundation of China No. 62272467 and the fund for building world-class universities (disciplines) of Renmin University of China.

Data availability statement The data and codes used in this paper are available at: https://github.com/8421BCD/DEPS.

Declarations

Conflict of interest The authors declare that they have no conflict of interest regarding the publication of this paper.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Cai F, Liang S, De Rijke M (2014). Personalized document re-ranking based on bayesian probabilistic matrix factorization. In: Proceedings of the SIGIR’2014, pp 835–838. ACM
2. Song Y, Wang H, He X (2014) Adapting deep ranknet for personalized search. In: WSDM’2014. ACM, pp 83–92

3. Teevan J, Liebling DJ, Ravichandran Geetha G (2011) Understanding and predicting personal navigation. In: WSDM'2011. ACM, pp 85–94
4. Harvey M, Crestani F, Carman MJ (2013) Building user profiles from topic models for personalised search. In: CIKM'2013, pp 2309–2314. ACM
5. Vu T, Song D, Willis A, Tran S.N, Li J (2014). Improving search personalisation with dynamic group formation. In: SIGIR'2014, pp 951–954
6. Vu T, Willis A, Tran S.N, Song D (2015) Temporal latent topic user profiles for search personalisation. In: ECIR'2015. Springer, pp 605–616
7. Lu S, Dou Z, Jun X, Nie J-Y, Wen J-R (2019). Psgan: a minimax game for personalized search with limited and noisy click data. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. SIGIR'19
8. Yao J, Dou Z, Wen J-R (2020) Employing personal word embeddings for personalized search. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 1359–1368
9. Yao J, Dou Z, Xu J, Wen J-R (2020) RLPer: a reinforcement learning model for personalized search. In: Proceedings of the web conference 2020, pp 2298–2308
10. Zhou Y, Dou Z, Wen J-R (2020) Encoding history with context-aware representation learning for personalized search. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 1111–1120
11. Zhou Y, Dou Z, Wen J-R (2020) Enhancing re-finding behavior with external memories for personalized search. In: Proceedings of the 13th international conference on web search and data mining, pp 789–797
12. Ge S, Dou Z, Jiang Z, Nie J-Y, Wen J-R (2018). Personalizing search results using hierarchical rnn with query-aware attention. In: Proceedings of the 27th ACM international conference on information and knowledge management. CIKM '18
13. Silverstein C, Marais H, Henzinger M, Moricz M (1999). Analysis of a very large web search engine query log. In: ACM SIGIR forum, 33, 6–12. ACM
14. Cronen-Townsend S, Croft WB (2002) Quantifying query ambiguity. In: Proceedings of the second international conference on human language technology research. Morgan Kaufmann Publishers Inc, pp 104–109
15. Zamani H, Dadashkarimi J, Shakery A, Croft WB (2016) Pseudo-relevance feedback based on matrix factorization. In: Proceedings of the 25th ACM international on conference on information and knowledge management, pp 1483–1492
16. Lv Y, Zhai C (2010) Positional relevance model for pseudo-relevance feedback. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval, pp 579–586
17. Tao T, Zhai C (2006) Regularized estimation of mixture models for robust pseudo-relevance feedback. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, pp 162–169
18. Broder A.Z, Fontoura M, Gabrilovich E, Joshi A, Josifovski V, Zhang T (2007) Robust classification of rare queries using web knowledge. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, pp 231–238
19. Shen D, Pan R, Sun J-T, Pan JJ, Wu K, Yin J, Yang Q (2005) Q2c@ust: our winning solution to query classification in kddcup 2005. ACM SIGKDD Explor Newsl 7(2):100–110
20. Dou Z, Song R, Wen J-R (2007) A large-scale evaluation and analysis of personalized search strategies. In: WWW'2007, pp 581–590. ACM
21. Dou Z, Song R, Wen J-R, Yuan X (2008) Evaluating the effectiveness of personalized web search. IEEE Trans Knowl Data Eng 21(8):1178–1190
22. Volkovs M (2015) Context models for web search personalization. arXiv preprint arXiv:1502.00527
23. Wang H, He X, Chang M.-W, Song Y, White RW, Chu W (2013) Personalized ranking model adaptation for web search. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, pp 323–332
24. Wu Q, Burges CJ, Svore KM, Gao J (2008) Ranking, boosting, and model adaptation. Technical report, MSR-TR-2008-109
25. Ma Z, Dou Z, Bian G, Wen J-R (2020) Pstie: time information enhanced personalized search. In: Proceedings of the 29th ACM international conference on information and knowledge management, pp 1075–1084
26. Qian H, Li X, Zhong H, Guo Y, Ma Y, Zhu Y, Liu Z, Dou Z, Wen J-R (2021) Pchatbot: a large-scale dataset for personalized chatbot. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 2470–2477

27. Zhou Y, Dou Z, Wei B, Xie R, Wen J-R (2021) Group based personalized search by integrating search behaviour and friend network
28. Li X, Guo C, Chu W, Wang Y-Y, Shavlik J (2014) Deep learning powered in-session contextual ranking using clickthrough data. In: NIPS'2014
29. Deng C, Zhou Y, Dou Z (2022). Improving personalized search with dual-feedback network. In: Proceedings of the fifteenth ACM international conference on web search and data mining, pp 210–218
30. Lavrenko V, Croft WB (2017) Relevance-based language models. In: ACM SIGIR forum, 51, 260–267. ACM New York, NY, USA
31. Zhai C, Lafferty J (2001) Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the tenth international conference on information and knowledge management, pp 403–410
32. Ai Q, Bi K, Guo J, Croft WB (2018) Learning a deep listwise context model for ranking refinement. In: The 41st international ACM SIGIR conference on research and development in information retrieval, pp 135–144
33. Scholer F, Turpin A, Sanderson M (2011) Quantifying test collection quality based on the consistency of relevance judgements. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, pp 1063–1072
34. Yang Z (2017) Relevance judgments: preferences, scores and ties. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 1373–1373
35. Ai Q, Wang X, Golbandi N, Bendersky M, Najork M (2019). Learning groupwise scoring functions using deep neural networks
36. Pang L, Xu J, Ai Q, Lan Y, Cheng X, Wen J (2020) Setrank: learning a permutation-invariant ranking model for information retrieval. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 499–508
37. Pasumarthi RK, Wang X, Bendersky M, Najork M (2019). Self-attentive document interaction networks for permutation equivariant ranking. arXiv preprint arXiv:1910.09676
38. Qin X, Dou Z, Wen J-R (2020). Diversifying search results using self-attention network. In: Proceedings of the 29th ACM international conference on information and knowledge management, pp 1265–1274
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A.N, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
40. Carbonell J, Goldstein J (1998) 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, pp 335–336
41. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender GN (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on machine learning (ICML-05), pp 89–96
42. Pass G, Chowdhury A, Torgeson C (2006) A picture of search. In: InfoScale, 152:1
43. Robertson S, Zaragoza H. et al (2009) The probabilistic relevance framework: Bm25 and beyond. Found Trends® Inf Retrieval 3(4):333–389
44. Ahmad WU, Chang K-W, Wang H (2018) Multi-task learning for document ranking and query suggestion
45. Ahmad WU, Chang K-W, Wang H (2019) Context attentive document ranking and query suggestion. arXiv preprint arXiv:1906.02329
46. Huang J, Zhang W, Sun Y, Wang H, Liu T (2018). Improving entity recommendation with search log and multi-task learning. In: IJCAI, pp 4107–4114
47. Xiong C, Dai Z, Callan J, Liu Z, Power R (2017) End-to-end neural ad-hoc ranking with kernel pooling. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 55–64
48. Dai Z, Xiong C, Callan J, Liu Z (2018) Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Proceedings of the eleventh ACM international conference on web search and data mining, pp 126–134. ACM
49. Qiao Y, Xiong C, Liu Z, Liu Z (2019) Understanding the behaviors of bert in ranking. arXiv preprint arXiv:1904.07531

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Wenhan Liu is currently a second-year Ph.D. student advised by Prof. Zhicheng Dou at the Gaoling School of Artificial Intelligence, Renmin University of China. He received his B.S. degree in computer science and technology from Shandong University in 2022. His research interests include large language models for information retrieval, search clarification and personalized search.

Yujia Zhou received the BE degree in computer science and technology from the School of Information, Renmin University of China, in 2019. And he is studying for Ph.D. in the School of Information, Renmin University of China. He won the best student paper award in CCIR 2018. His research interests include information retrieval, personalized search, deep learning and data mining.

Yutao Zhu received the B.S. and M.S. degree from Renmin University of China, and the Ph.D. degree from the University of Montreal. He is currently a postdoc at Renmin University of China. His current research interests are large language models and information retrieval. He received the Best Paper Award from CCIR 2021 and the Google Scholarship for UdeM in 2019. He served as the PC member of several top-tier conferences, such as NeurIPS, ACL, SIGIR, SIGKDD, WWW, AAAI and EMNLP.

Zhicheng Dou (Member, IEEE) received the B.S. and Ph.D. 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.