Long Short-term Session Search: Joint Personalized Reranking and Next Query Prediction

Cheng, Q.; Ren, Z.; Lin, Y.; Ren, P.; Chen, Z.; Liu, X.; de Rijke, M.

DOI
10.1145/3442381.3449941

Publication date
2021

Document Version
Final published version

Published in
The Web Conference 2021

Long Short-Term Session Search: Joint Personalized Reranking and Next Query Prediction

Qiannan Cheng[1] Zhaochun Ren[1,*] Yujie Lin[1] Pengjie Ren[1]
Zhumin Chen[1] Xiangyuan Liu[1] Maarten de Rijke[2]
[1]Shandong University, Qingdao, China
[2]University of Amsterdam & Ahold Delhaize, Amsterdam, The Netherlands
chengqiannan@mail.sdu.edu.cn,zhaochun.ren@sdu.edu.cn,yu.jie.lin@outlook.com,renpengjie@sdu.edu.cn
chenzhumin@sdu.edu.cn,liuxiangyuan@mail.sdu.edu.cn,m.derijke@uva.nl

ABSTRACT

Document reranking (DR) and next query prediction (NQP) are two core tasks in session search. They are often driven by the same search intent and, hence, it is natural to jointly optimize both tasks. So far, most models proposed for jointly optimizing DR and NQP have focused on users' short-term intent in an ongoing search session. Because of this limitation, these models fail to account for users' long-term intent as captured in their historical search sessions. In contrast, we consider a personalized mechanism for learning a user's profile from their long-term and short-term behavior to simultaneously enhance the performance of DR and NQP in an ongoing search session.

We propose a personalized session search model, called **Lo**ng **s**hort-**t**erm session search, **Net**work (LostNet), that jointly learns to rerank documents for the current query and predict the next query. LostNet consists of three modules: (1) a hierarchical session-based attention mechanism, (2) a personalized multi-hop memory network, and (3) joint learning of DR and NQP. The hierarchical session-based attention mechanism tracks the fine-grained short-term intent in an ongoing session. The personalized multi-hop memory network tracks a user's dynamic profile information from their prior search sessions so as to infer their personal search intent. Jointly learning of DR and NQP is aimed at simultaneously reranking documents and predicting the next query based on outputs from the above two modules. We conduct experiments on two large-scale session search benchmark datasets. The results show that LostNet achieves significant improvements over state-of-the-art baselines.

ACM Reference Format:
Qiannan Cheng, Zhaochun Ren, Yujie Lin, Pengjie Ren, Zhumin Chen, Xiangyuan Liu, and Maarten de Rijke. 2021. Long Short-Term Session Search: Joint Personalized Reranking and Next Query Prediction. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449941

*Zhaochun Ren is the corresponding author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449941

1 INTRODUCTION

A *search session* refers to a short-term interval with multiple search interactions [3, 13, 24, 26]. Search sessions play an important role for understanding users' information needs. During a search session, a user may interact in different ways. Interactions with the ranked list of results presented to them result in clicks and skips. In addition, users may repeatedly revise a query (e.g., adding terms or removing terms) so as to clarify their information need. These two types of behavior are driven by the same underlying search intent. To explore the underlying search intent behind these types of behavior, document reranking (DR) and next query prediction (NQP) have attracted significant research efforts in research on session search [see, e.g., 11, 36, 37]. It is known that DR and NQP may benefit from each other [1]. Accordingly, recent work has proposed to jointly optimize these two tasks [1, 2]. Two advantages have been verified in these multi-task solutions: (1) they can learn more useful representations by leveraging training signals from related tasks; and (2) they can improve the generalization ability by benefiting from the regularization effect. However, existing approaches to this joint optimization problem only consider a user's short-term search intent in an ongoing session; the user's long-term personal preferences as exhibited during previous sessions are neglected. During multi-session search, a user plays different roles in different search scenarios [5]. Moreover, similar queries may express different intents as different users have different characteristics. Therefore, indiscriminately returning the same results to all users may lead to inaccurate search results [9, 47, 52].

Intuitively, a personalized retrieval strategy is able to alleviate the problems identified above by providing more personalized results. Two kinds of personalized information exists in long-term behaviors [5, 8, 17, 58]: (1) similar past sessions; and (2) long-term search roles and intents. Existing personalized retrieval approaches using long-term search logs mostly focus on optimizing a ranked list of documents using search logs, neglecting the NQP task [5, 15, 17]. For example, Ge et al. [17] leverage a hierarchical recurrent neural network with an attention mechanism to capture sequential information for dynamic profiling, whereas Zhou et al. [58] utilize memories to enhance user re-finding behavior in personalized search. However, existing personalization approaches still have several shortcomings: (1) it is hard to capture long-term search intent; (2) a lack of fine-grained historical information reduces search performance; and (3) low efficiency limits the potential applicability in real-world settings.

To be able to simultaneously improve the ranked list of documents and next query prediction in a manner that overcomes the shortcomings listed above, we focus on a joint learning mechanism that uses personalized profiling features from a user's short- and long-term behavior. The idea is illustrated in Fig. 1, with a user who issues multiple queries and who has interacted with the search engine during multiple previous sessions. By learning both short-term and long-term user profiling features, our goal is to infer a user's personalized search intent so as to simultaneously rerank documents and predict the next query.

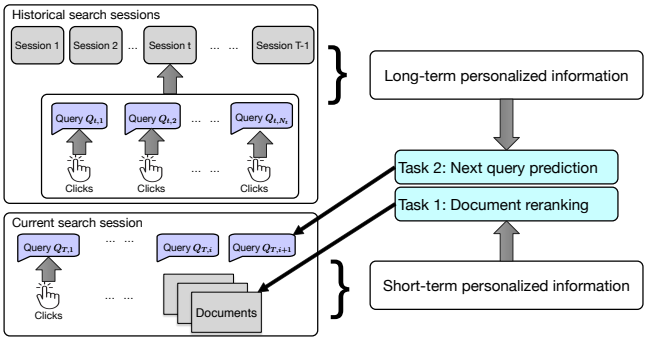


Figure 1: Personalized joint learning of document reranking and next query prediction.

We operationalize this idea by employing External Memory Networks (EMNs) [49, 55] to model a user's long-term profiling information with an external memory module. Because they allow for long-term preservation, instant updates, and effective operations, EMNs have shown promising performance in many sequential prediction tasks, such as question answering [28], task-oriented dialogue systems [6], and sequential recommendation [12]. EMNs contain a memory matrix to store the states separately in memory slots, which enable long-term preservation and instant updates; they also support complex and proper operations on the matrix to leverage previous information.

Specifically, we propose a personalized session search model, called **Lo**ng **s**hort-**t**erm session search **Net**work (LostNet), that jointly learns to rerank documents for the current query and predict the next query. LostNet is organized into three main steps: (1) modeling session-level search intent, (2) personalized search intent tracking, and (3) joint learning of document ranking and query recommendation. We propose a hierarchical session-based attention mechanism to infer the session-level intent. Then, a multi-hop external memory network learns the user's long-term profile. And, finally, a joint learning framework is applied to optimize the reranked list of documents and the query prediction results. We conduct experiments on two large-scale benchmark datasets to evaluate the effectiveness of LostNet. The results confirm the positive effect of capturing the user's long-term search intent in enhancing both DR and NQP. We find that LostNet achieves significant improvements over state-of-the-art baselines.

Our contributions can be summarized as follows: (1) a personalized approach to jointly optimize document reranking and next query prediction; (2) a multi-hop memory network to learn the user's long-term profile; (3) a hierarchical session-based attention mechanism to model fine-grained session-level intent; and (4) extensive experiments to verify the effectiveness of LostNet, the long short-term session search network.

2 RELATED WORK

We discuss related work on neural information retrieval, search result personalization, and memory-augmented neural networks.

2.1 Neural information retrieval

Deep neural networks have been successfully applied to information retrieval to optimize the ranked list of documents [29, 37, 39]. The representation-based models [see, e.g., 19, 22, 45] utilize deep neural networks to build representation vectors of queries and documents for similarity matching. The interaction-based models [18, 40, 57] build local interactions between queries and documents to learn hierarchical matching patterns. Several approaches have also been proposed by combing above two categories [23, 38].

As another important task in information retrieval, the task of *next query prediction* (*query suggestion*) focuses on predicting the next query in a search session [36]. Several deep learning approaches have been proposed [11, 25, 36, 56]. HRED [48] uses a Hierarchical Recurrent Neural Network (HRNN) to encode a sequence of queries in the same session for next query generation. ACG [14] augments the standard seq2seq model with query-aware attention and a copy mechanism to capture a query's importance and repeat words from the session context. However, these models only optimize either document ranking or query suggestion, which ignores the mutual reinforcement between related retrieval tasks.

Multi-task learning use shared representations to learn multiple tasks in parallel and to improve each task by leveraging the domain-specific information shared in all related tasks [10]. Neural multi-task learning methods have been applied to ad-hoc retrieval [1, 21, 31]. Particularly relevant to us is the work by Ahmad et al. [1], who propose a neural framework that can jointly learn document reranking and next query prediction for capturing the latent intent embedded in a user's search behavior.

Unlike existing approaches, we model a *personalized* mechanism for learning a user's profile from their *short*- and *long*-term behavior to simultaneously enhance the performance of DR and NQP.

2.2 Search result personalization

Returning the same search results to all users limits the usefulness of search engines. Personalization promises to improve a search engine's usefulness by providing more personalized results for individual users [34]. Traditional personalized search methods profile a user by extracting click-based and topic-based features from a user's search history [9, 15, 46, 50, 52]. To tackle sparsity and incompleteness in user profiling, deep neural networks have been applied to automatically learn personalized features in search result personalization. Song et al. [47] propose a RankNet model based on the personal search history and result preference. Li et al. [30] generate deep semantic features from in-session contexts, and integrate them into the current ranking model. Ge et al. [17] utilize a hierarchical recurrent neural network to model the sequential information hidden in historical query-click data. Zhou et al. [58]

utilize memories to support the identifcation of complex re-finding behavior for personalized search.

Unlike previous studies that focus on learning a user's profile to optimize a ranked list of documents, we share personalized search intent across related retrieval tasks, i.e., document reranking as well as next query prediction.

2.3 Memory-augmented neural networks

External Memory Networks (EMNs) are composed of two components: a memory bank that stores the latent states and a controller that performs read/write operations to this bank [55]. With the power to process sequential data effectively, EMNs have been successfully applied to recommender systems. Chen et al. [12] leverage the external memory matrix to store and manipulate users' historical records, which improves the performance of sequential recommendation. Ebesu et al. [16] exploits various memory states to model user-item interactions in recommendation. Huang et al. [20] propose a knowledge enhanced sequential recommender that incorporates knowledge base information to capture attribute-level user preferences. Wang et al. [54] investigate collaborative neighborhood information by means of an external memory module to enhance session-based recommendation. However, EMNs have rarely been applied to session search.

In this paper, we design a personalized multi-hop memory network in *LostNet* to learn the user's long-term profile. Unlike the state-of-the-art personalized session search method [17], ours has the advantage of long-term memorization, fine-grained profiling and high efficiency.

3 PROBLEM FORMULATION

Before introducing our personalized reranking and query suggestion framework, we first introduce notions and concepts used in this paper. Following Ge et al. [17], a *SAT-checked document* is a document that received positive feedback (e.g., clicks) from the user. Given a query Q, we assume that there is a set of SAT-checked documents D. At time t, a search session S_t consists of N_t turns of search interactions (e.g., query and clicks) from the user u. The search session S_t is represented as a sequence of query-SAT checked documents pairs, i.e., $S_t = [(Q_{t,1}, D_{t,1}), (Q_{t,2}, D_{t,2}), \ldots, (Q_{t,N_t}, D_{t,N_t})]$. At the current time T, in a session-level search scenario, a user u releases a query $Q_{T,i}$ at the i-th turn. Then the search engine provides a list of top-M ranked candidate documents, i.e., $D^{can}_{T,i} = \{D^{can}_{T,i,1}, D^{can}_{T,i,2}, \ldots, D^{can}_{T,i,M}\}$. Moreover, for the user u there is a sequence of historical search sessions $\mathcal{L}_u = [S_1, S_2, \ldots, S_{T-1}]$ during the past $T-1$ timesteps.

Next, we formalize our research target in this paper. Given user u's query click log $\mathcal{L}_u$ and current session S_T, our target is to establish a model to infer user u's personalized search intent $I_{T,i}$ at the i-th turn in S_T, where $I_{T,i}$ is represented as a vector with dimensionality d_s. We aim to jointly predict the query $Q_{T,i+1}$ at the $(i+1)$-st turn, and the probability of click $C_{T,i,m}$ for each candidate document $D^{can}_{T,i,m}$, $1 \le m \le M$.

4 METHOD

In this section, we describe the long short-term session search network, abbreviated as LostNet. We first provide an overview of LostNet. We then describe the architecture to model personalized search intent. Finally, we detail the joint learning process of document reranking and next query prediction.

4.1 Overview

For an issued query (e.g., "job hunting preparation") in the current session, without considering long-term information, the search results will be about "preparation for different potential jobs." However, given queries (e.g., "java runtime environment") and SAT documents (e.g., "dynamic programming algorithm") in previous sessions, it becomes evident about the user's accurate intent, i.e., "looking for an algorithm engineer position." Shown by this example, we argue that the system can provide accurate search results by by inferring the user's long-term search intent.

We introduce the *long short-term session search network* (**LostNet**) to infer the personalized search intent for jointly optimizing document reranking and next query prediction. Fig. 2 provides an overview of the personalized search process in LostNet. There are three main ingredients: (A) session-level search intent tracking (see §4.2); (B) personalized search intent tracking (see §4.3); and (C) document reranking and next query prediction (see §4.4). Along with LostNet, for (A) we propose a hierarchical session-based attention mechanism to infer session-level search intent. Then, for (B) we propose a multi-hop memory network to learn the user's long-term profile. Finally, we provide an optimized learning procedure in LostNet for (C), where we apply a multi-task learning strategy to jointly optimize document reranking and next query prediction.

4.2 (A) Tracking session-level search intent

We start by detailing how we model session-level search intent within a session. Unlike previous methods, our method applies a hierarchical session-based attention mechanism with recurrent neural network (RNN) to learn the fine-grained session-level intent representation. This mechanism is composed of two hierarchical components: text (i.e., query and document) representation and session-level intent representation.

4.2.1 Text representation. Within a session S_t, $1 \le t \le T$, there is a query-click sequence, i.e., $S_t = [(Q_{t,1}, D_{t,1}), \ldots, (Q_{t,N_t}, D_{t,N_t})]$, where N_t is the number of turns in S_t. We consider each query and document as a variable length word sequence. Given a J-length sequence of words $\left[w_1, w_2, \ldots, w_J\right]$, the model first embeds each word w_j, $1 \le j \le J$, into a d_e-dimensional embedding x_j by looking up in an embedding matrix $E \in R^{|V| \times d_e}$, i.e., $x_j = E(w_j)$. To better understand the content presented in the text, we apply bidirectional long short-term memories (LSTMs) (i.e., one forward and one backward LSTM) to encode the sequence of words in both queries and documents. As shown in Eq. 1, for each x_j we obtain a hidden representation h_j by concatenating the hidden states generated by BiLSTMs:

$$\overrightarrow{h_j} = \text{LSTM}(\overrightarrow{h}_{j-1}, x_j),\ \overleftarrow{h_j} = \text{LSTM}(\overleftarrow{h}_{j+1}, x_j),\ h_j = \left[\overrightarrow{h_j}, \overleftarrow{h_j}\right], \quad (1)$$

Thus, we obtain a sequence of J hidden representations $[h_1, h_2, \ldots, h_J]$, $h_j \in R^{2d_h}$, where d_h refers to the dimension of the forward and backward LSTM hidden unit. To detect the unbiased weight of each word in a text, we present an inner-attention mechanism [53]

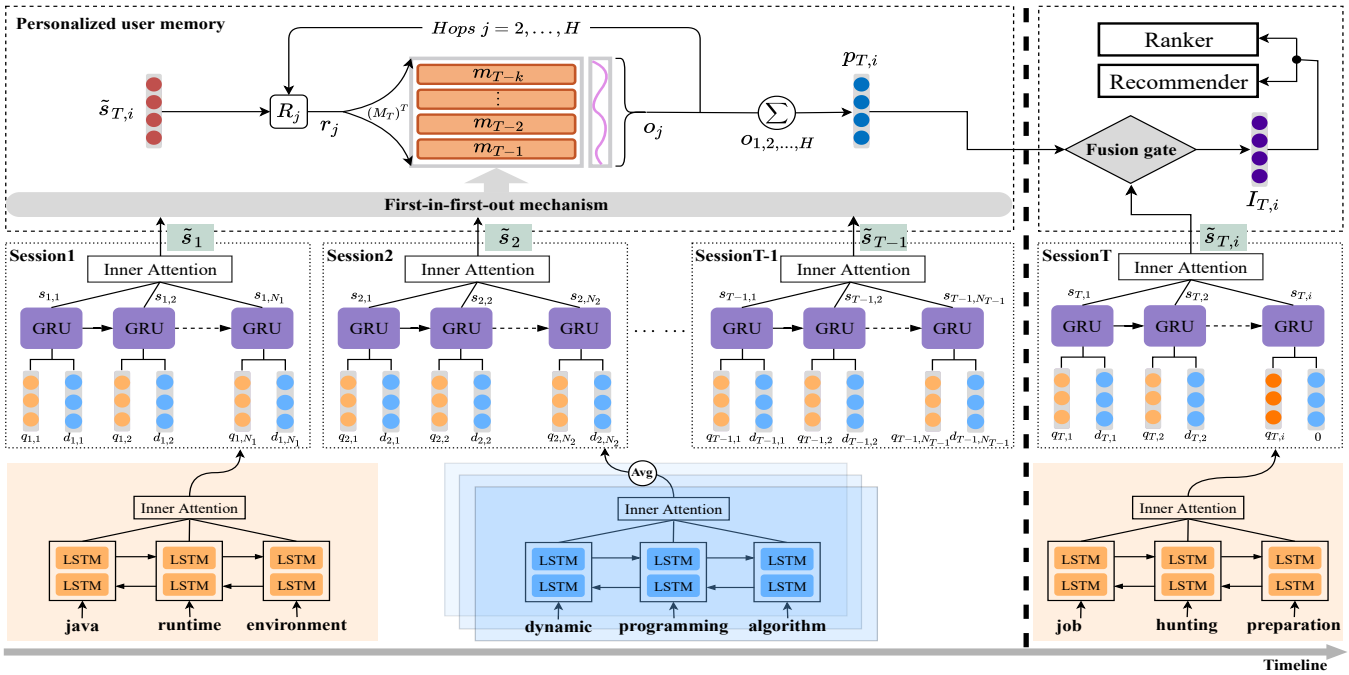


Figure 2: Overview of our long short-term session search network (LostNet). Queries (e.g., "java runtime environment") and SAT-documents (e.g., "dynamic programming algorithm") exist in past sessions. For a query in current session, LostNet infers the personalized search intent based on the hierarchical session-based attention mechanism and the personalized memory network. On this basis, LostNet jointly reranks candidate documents and predicts the next query.

to generate a fixed-length representation $\pi \in R^{2d_h}$, so we have:

$$\pi = \sum_{j=1}^{J} \alpha_j^h \cdot h_j, \quad \alpha_j^h = \frac{\exp(v_1^T \tanh(W_1^\alpha h_j + b_1^\alpha))}{\sum_{l=1}^{J} \exp(v_1^T \tanh(W_1^\alpha h_l + b_1^\alpha))}, \tag{2}$$

where α_j^h is the attention weight for h_j, $1 \le j \le J$, $\tanh(\cdot)$ refers to an element-wise tangent function, and $v_1 \in R^{d_a}$, $W_1^\alpha \in R^{d_a \times 2d_h}$, and $b_1^\alpha \in R^{d_a}$ are parameters of our inner-attention mechanism.

During a search session S_t, at the n-th turn, after applying the inner-attention procedure, we represent the query $Q_{t,n}$ as a distributional vector $q_{t,n} \in R^{2d_h}$. We write $d_{t,n} \in R^{2d_h}$ to indicate the average distributional vector of all clicked documents under $Q_{t,n}$. Therefore, after this process, we have an embedding sequence representation of S_t as follows:

$$s_t = \left[(q_{t,1}, d_{t,1}), (q_{t,2}, d_{t,2}), \ldots, (q_{t,N_t}, d_{t,N_t})\right]. \tag{3}$$

4.2.2 Session-level intent representation. As the bottom layer of our session-based attention mechanism, we get the within-session query/document embedding representation via an inner-attention layer, i.e., a fixed-length latent representation of each query and its clicked documents. The upper layer of our session-based attention mechanism concerns the session-level intent representation.

Given a search session S_t, we obtain its within-session representation s_t as a sequence of embedding representations of queries and clicked documents. To accurately model the temporal information hidden in such a query chain, we apply a GRU-based solution to form the vector representation of s_t. Shown in purple in Fig. 2, at the n-th interaction turn within session s_t, the GRU encoder takes the $(n-1)$-th turn's hidden state and the current turn's query-click pair as the input, so we have:

$$s_{t,n} = \text{GRU}(s_{t,n-1}, [q_{t,n}, d_{t,n}]), \; s_{t,n} \in R^{d_s}, \tag{4}$$

where $s_{t,n}$ refers to the hidden representation at the n-th turn in session s_t, whereas $s_{t,0}$ is initialized by a zero vector, the parameters in GRU are shared across all sessions. Thereafter, we have a variable length sequence of hidden representations based on GRU encoders, i.e., $[s_{t,1}, s_{t,2}, \ldots, s_{t,N_t}]$. As with the within-session embedding representation, we still apply the inner-attention mechanism to achieve focus of a session and infer a fixed-length intent representation. Therefore, we obtain an attentive vector $\tilde{s}_t \in R^{d_s}$. At the i-th turn during S_T, as there is no corresponding click for the current query $Q_{T,i}$, we assume that $D_{T,i} = \varnothing$, so that we have $d_{T,i} = \mathbf{0}$. Therefore, we obtain a dynamic short-term search intent representation $\tilde{s}_{T,i}$ for the current session S_T. Moreover, we have $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{T-1}$ for previous sessions $S_1, S_2, \ldots, S_{T-1}$, which will be applied to build the user's long-term profile representation.

4.3 (B) Tracking personalized search intent

We propose a new memory augmented neural network to track the user's personalized search intent. Our method can be divided into two main components: a *personalized memory encoder* (PME) and a *fusion gating layer* (FGL). The PME encodes information from previous sessions into a user profile representation, while the FGL module selectively combines information from the current session and the PME for inferring personalized search intent.

4.3.1 Personalized memory encoder. To explicitly store and memorize information, memory networks apply an external memory

as a large array of slots. Given a user u, to address their long-term search interest, we apply a external memory module with k matrix slots to store the hidden representations of the last k sessions. We denote the personalized memory matrix as M_T, so we have $M_T = \{m_{T-k}, m_{T-k+1}, \ldots, m_t, \ldots, m_{T-1}\}$, $m_t \in R^{d_s}$, $M_T \in R^{d_s \times k}$. Moreover, the PME uses H hops reading operations with soft attention and first-in-first-out writing operations to access the memory matrix M_T.

Read operation. In fact, not all historical sessions are useful for predictions in an ongoing session. To discriminate previous behaviors for building an effective user profile, we design multi-hop reading mechanism to scrutinize previous sessions and highlight important part dynamically according to present information need. Given a user u, we assume there are already i turns of query reformulations during the current session S_T. We write $r_j \in R^{d_s}$ for a reading vector to retrieve the memory matrix M_T at the j-th hop ($1 \le j \le H$). To derive the output, o_j, at the j-th hop, we let r_j read matrix M_T as follows:

$$o_j \leftarrow \text{READ}(M_T, r_j). \tag{5}$$

For r_j, we initially set r_1 as the short-term search intent representation $\tilde{s}_{T,i}$ of the current session. For the j-th hop, $1 < j \le H$, we use both the output from the last hop o_{j-1} and $\tilde{s}_{T,i}$ to derive an updated reading vector r_j:

$$r_j = R_j(\tilde{s}_{T,i} + o_{j-1}), \tag{6}$$

where $R_j \in R^{d_s \times d_s}$ is a learned parameter matrix, which will be updated on each hop j. To specify the *READ* operation, we present an attentive combination of memory slots, which is shown in Eq. 7:

$$\begin{aligned} z_{j,t} &= \frac{\exp(\beta \cdot (r_j)^T m_t)}{\sum_{l=T-k}^{T-1} \exp(\beta \cdot (r_j)^T m_l)}, \forall t \in \{T-k, \ldots, T-1\} \\ o_j &= \sum_{t=T-k}^{T-1} z_{j,t} \cdot m_t, \ o_j \in R^{d_s}, \end{aligned} \tag{7}$$

where $z_{j,t}$ refers to the attention weight of the t-th slot in memory matrix M_T at the j-th hop, and β is the strength parameter. After the read operation, our model is able to combine new evidence into the reading vector r_j, which will guide to focus on and retrieve more relevant personalized information in later accesses. After H hops, we combine all outputs of read operations to generate the long-term user profile, i.e., $p_{T,i} = \sum_{j=1}^{H} \alpha_j o_j$, where $p_{T,i} \in R^{d_s}$, and $\alpha_1, \alpha_2, \ldots, \alpha_H \in R$ are learned weight parameters to recognize the importance of the reading outputs at each hop j.

Write operation. At the beginning, we initialize the memory matrix as $\varnothing$. We adopt a first-in-first-out mechanism to update the memory matrix, which stores the user u's latest k sessions. When we write the memory, the earliest session is removed from the memory matrix, and the new one is added into the queue. When the memory matrix is not yet full, the session is directly added without removing any existing session.

Efficiency analysis. We compare the efficiency of our memory module to the RNN-based layers in our problem. In our algorithm, we define the session-level calculation as one basic operation. In terms of computational complexity, for each issued query, the RNN-based layers have to recalculate k previous sessions' information, i.e., $O(k)$. Such low efficiency makes them not appropriate for broad real-world applications. In contrast, our PME contains a memory matrix to store the long-term states in memory slots separately, which enables long-term information update instantly. Thus it will decrease the time complexity to $O(1)$ by using space-time tradeoffs. Moreover, our memory module connects all sessions with a constant number of sequentially executed operations, whereas the RNN-based layer requires $O(k)$ sequential operations [51].

4.3.2 Fusion gating layer. To balance the impact of the short-term search intent and long-term user profile, we use a *fusion gating mechanism* to infer the personalized search intent $I_{T,i} \in R^{d_s}$ at the i-th turn in the ongoing session S_T.

$$I_{T,i} = f_{T,i} \cdot \tilde{s}_{T,i} + (1 - f_{T,i}) \cdot p_{T,i}, \tag{8}$$

where the gate $f_{T,i} \in R^{d_s}$ is derived by $\sigma(W^f \tilde{s}_{T,i} + V^f p_{T,i})$, whereas W^f, V^f refers to the $d_s \times d_s$ parameter matrices. The personalized intent representation, $I_{T,i}$, will be used to enhance the joint learning of document reranking and next query prediction.

4.4 (C) Joint learning

For the user u, at the i-th turn in session S_T, we have the current query $Q_{T,i}$ and a personalized search intent representation $I_{T,i}$. We aim to derive the clicked probability $C_{T,i,m}$ of the candidate document $D^{can}_{T,i,m}$, $(1 \le m \le M)$ using a sigmod function over $I_{T,i}$ and $Q_{T,i}$; and we apply a word-level attentive mechanism to predict words in the next query $Q_{T,i+1}$. Details of each component are given next.

4.4.1 Document reranking. Our goal is to rerank candidate documents in $D^{can}_{T,i}$ according to their relevance w.r.t. $Q_{T,i}$ and $I_{T,i}$. Using text representation procedure in Section 4.2.1, we obtain the embedding representation $q_{T,i}$ and $d^{can}_{T,i,m}$ for the query $Q_{T,i}$ and a candidate document $D^{can}_{T,i,m}$, respectively. We first concatenate $q_{T,i}$ with $I_{T,i}$ via a non-linear transformation, and then apply a sigmod function to infer the click probability of $d^{can}_{T,i,m}$, i.e., $C_{T,i,m}$:

$$\begin{aligned} C_{T,i,m} &= P(d^{can}_{T,i,m} | q_{T,i}, I_{T,i}) \\ &= \sigma\left((d^{can}_{T,i,m})^T \tanh(W^P [q_{T,i}; I_{T,i}] + b^P)\right), \end{aligned} \tag{9}$$

where $1 \le m \le M$, and $W^P \in R^{2d_h \times (2d_h + d_s)}$, $b^P \in R^{2d_h}$ are parameters of our reranking component. Thereafter, candidate documents are re-ordered according to these click probabilities.

4.4.2 Next query prediction. To predict the next query $Q_{T,i+1}$, we calculate the prediction probability of $Q_{T,i+1}$ given $Q_{T,i}$ and $I_{T,i}$. By decomposing $P(Q_{T,i+1} | Q_{T,i}, I_{T,i})$ into multiplications of a series of probabilities over adjacent words in $Q_{T,i+1}$, we have:

$$P(Q_{T,i+1} | Q_{T,i}, I_{T,i}) = \prod_{y=1}^{|Q_{T,i+1}|} P(w_y | w_{1:y-1}, Q_{T.i}, I_{T,i}). \tag{10}$$

We estimate this probability by using an LSTM-based decoding procedure. We set $W^{h_0} \in R^{d_p \times d_s}$ and $b^{h_0} \in R^{d_p}$ as decoder parameters, and initialize the hidden state h_0^{dec} as $\tanh(W^{h_0} I_{T,i} + b^{h_0})$. We use the following procedure to calculate the rest hidden states:

$$h_y^{dec} = LSTM(h_{y-1}^{dec}, x_{y-1}), \forall y \in \{1, 2, \ldots, |Q_{T,i+1}|\}, \tag{11}$$

where $h_y^{dec} \in R^{d_p}$, and x_{y-1} is the embedding of the generated word w_{y-1}. Considering that users often retain words from their last query [2], we apply an attentive method to enhance the influence of the current query $Q_{T,i}$ in predicting the words in $Q_{T,i+1}$. Specifically, we predict the y-th word in the next query $Q_{T,i+1}$ based on a word-level attentive vector $c_y \in R^{2d_h}$ that encodes the words in the current query $Q_{T,i}$ with respect to the y-th hidden state of the decoder:

$$c_y = \sum_{j=1}^{|Q_{T,i}|} \frac{\exp((h_y^{dec})^T W^a h_j^{enc})}{\sum_{l=1}^{|Q_{T,i}|} \exp((h_y^{dec})^T W^a h_l^{enc})} \cdot h_j^{enc}, \tag{12}$$

where h_j^{enc} is the j-th BiLSTM hidden representation when encoding $Q_{T,i}$, and $W^a \in R^{d_p \times 2d_h}$ is a parameter matrix. We use c_y to update the decoder hidden state $\tilde{h}_y^{dec} = \tanh(W^c[c_y; h_y^{dec}])$, where $\tilde{h}_y^{dec} \in R^{d_p}$ and $W^c \in R^{d_p \times (2d_h + d_p)}$. Finally, we generate the y-th word w_y in $Q_{T,i+1}$ based on the following probability distribution over the vocabulary V:

$$P(w_y | w_{1:y-1}, Q_{T,i}, I_{T,i}) = \text{softmax}(W^{gen} \tilde{h}_y^{dec}). \tag{13}$$

4.4.3 Joint optimization. For an issued query, LostNet reranks candidate documents and predicts the next query given historical search sessions and the current session of the user. Therefore, the training objective of LostNet consists of two terms. The first term is the binary cross entropy loss for the document reranking:

$$\mathcal{L}_{rank} = -\frac{1}{M} \sum_{m=1}^{M} \left[\bar{C}_{T,i,m} \log(C_{T,i,m}) + (1 - \bar{C}_{T,i,m}) \log(1 - C_{T,i,m}) \right] \tag{14}$$

where $\bar{C}_{T,i,m} \in \{0, 1\}$ represents a binary click label for $D_{T,i,m}^{can}$. The second term is the negative log-likelihood loss for next query prediction, which is shown in Eq. 15:

$$\mathcal{L}_{pre} = -\sum_{y=1}^{|Q_{T,i+1}|} \log P(w_y | w_{1:y-1}, Q_{T.i}, I_{T,i}). \tag{15}$$

The final objective is the sum of L_{rank} and L_{pre} over all queries.

5 EXPERIMENTAL SETUP

We address the following research questions to guide our experiments: (RQ1) How does LostNet perform on document reranking and next query prediction? Does it outperform the baselines? (See §6.1) (RQ2) What is the effect of long-term personalized information? (See §6.2) (RQ3) How do the settings of PME influence the performance? (See §6.3) (RQ4) What is the effect of our joint learning? How well does LostNet perform with single task learning instead of joint learning? (See §6.4)

5.1 Datasets

We employ two benchmark datasets in our experiments: the AOL search log [42][1] and SogouQ [33].[2]

- **AOL**: The AOL search log is an English language dataset containing real users' query click data from 1st March, 2006 to 31st May, 2006 [42]. Since different clicks of a single query correspond to multiple lines in the log, we merge consecutive identical queries issued by the same user and aggregate their corresponding clicked documents. Following Ahmad et al. [1, 2], we remove all non-alphanumeric characters from the queries, and only use document titles as the content. Like Jansen et al. [24], we use a 30-minute period between interactions as the session boundary, and only consider sessions with more than two queries in the experiments.
- **SogouQ**: SogouQ is a Chinese web search log dataset that includes about one month (June 2008) of queries and user clicks from the Sogou search engine [33]. We first extract titles of clicked URLs from SogouT[3] which is an internet corpus containing 130 million original web pages [32]. As the log lacks exact timestamps for issued queries, we follow [2, 26] to set the boundaries between sessions based on the similarity between two consecutive queries. Since SogouQ only provides one month search log, we keep sessions with only one query to complement the number of historical sessions for users.

In order to simulate the memory's ability to dynamically store historical sessions of a user, we divide users in the data into 8:1:1 as training, validation, and test set respectively. Following [17], we also use a heuristic method to filter out spam users. Table 1 shows detailed statistics for the two datasets.

Table 1: Dataset statistics.

Datasets	Training users	Validation users	Test users
AOL	127,620	15,953	15,953
SogouQ	54,161	6,770	6,771

5.2 Baselines and comparisons

We write *LostNet* for the overall process as described in Section 4, which includes components (A), (B), and (C). We write *LostNet-Short* for the model that only considers components (A) and (C), so it uses a hierarchical session-based attention mechanism to model the short-term search intent for the joint learning of DR and NQP. To evaluate the effectiveness of LostNet, we compare it with a range of methods for document reranking and next query prediction.

For document reranking, our baselines include both classical and neural retrieval models. We consider *BM25* [44] as a baseline for classical retrieval methods. To compare with neural ranking models, we consider baselines categorized in representation-focused, interaction-focused, and combinations of both. We consider *DSSM* [22], *CLSM* [45], and *ARC-I* [19] as baselines for representation-focused models. We also consider *ARC-II* [19] and *DRMM* [18] as baselines for interaction-focused models. As combinations of representation- and interaction-focused models, we consider *DUET* [38] and *Match Tensor* [23].

To assess the performance of personalized document ranking, our baselines also include two state-of-the-art personalized ranking methods, *HRNN+QA* [17] and *RPMN* [58]. HRNN+QA extracts user profiles using a query-aware attention model, whereas RPMN applies external memories to enhance user re-finding behavior in personalized search.

[1] http://www.cim.mcgill.ca/~dudek/206/Logs/AOL-user-ct-collection/
[2] https://www.sogou.com/labs/resource/q.php
[3] https://www.sogou.com/labs/resource/t.php

To assess the performance of next query prediction, we consider *HRED-qs* [48], a hierarchical recurrent encoder-decoder model, as a baseline in our experiments. Our baselines also include recent work using sequence2sequence (seq2seq) models: *Seq2seq* [4] and *Seq2seq+Attn.* [35].

To evaluate the joint learning performance, we also compare LostNet with session search models using multi-task learning, i.e., *M-NSRF* [1], *M-MATCH Tensor* [1], and *CARS* [2].

5.3 Implementation details

We implement LostNet with Tensorflow and carry out experiments on a Geforce RTX 2080 Ti GPU. We follow Ahmad et al. [1] to limit the maximum available number of clicked documents per query to 5, and set the maximum allowable length of query and document (only the title) to 10 and 20 respectively. We use 300-dimensional word vectors trained with GloVe [43] to initialize the word embedding matrix E. We use a Gaussian distribution with a mean of 0 and standard deviation of 0.01 to randomly initialize model parameters. We use mini-batch SGD with Adam [27] for end-to-end training. The batch size is selected from $\{8, 16, 32, 64\}$ to fit in single GPU memory and the learning rate is empirically set to 10^{-4}. In LostNet, the number of hidden neurons in each of its encoders and decoders is determined by grid search based on its performance on the validation set. Considering the balance between efficiency and result quality, we set d_h, d_s, d_a, d_p in the sentence encoder, session-level GRU, session-level attention and next query prediction to 256, 512, 256, and 512, respectively. We vary the number of slots and reading hops in the user personalized memory from $\{2, 4, 8, 16, 32\}$ and $\{1, 3, 5\}$, respectively, to study their effect. We adopt an early stopping strategy with a patience of 3 epochs and select the model that achieves the minimum loss on the validation set.

5.4 Evaluation metrics

For the document reranking task, we employ three widely-used evaluation metrics, i.e., mean average precision (MAP), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG) computed at positions one, three, five and ten. Following Ahmad et al. [1], as search logs only contain clicked documents for each query, we construct candidate documents by selecting top-ranked documents using BM25 [44]. We sample 50 candidate documents per query in the test set, and 5 candidates per query for training and validation sets.

To assess the next query prediction performance, we consider the users' next submitted query as the ground-truth [48]. We conduct evaluations in terms of two aspects: the abilities of generation and discrimination. For evaluating the ability of generation, We use BLEU scores [41] as the evaluation metrics, which have been widely applied in many text generation tasks [7]. In addition, we also report MRR scores to evaluate the discriminative ability in next query prediction, where we use a co-occurrence based suggestion model to generate candidates by following [48].

6 RESULTS AND ANALYSIS

6.1 Overall performance (RQ1)

We start by addressing **RQ1** and evaluate the overall performance of document reranking and next query prediction.

6.1.1 Document reranking. Table 2 lists the document reranking performance of all models. We find that the neural retrieval models achieve better performance than BM25 in terms of most of the evaluation metrics, which indicates that lexical ranking models suffer from the vocabulary gap problem. Among the neural ranking models, we find that the interaction-focused models, and combinations of representation- and interaction-focused models outperform representation-focused models. For both datasets, we observe RPMN outperforms most of other baselines in the experiments, which verifies the effectiveness of the personalization. Among personalized search models, HRNN+QA achieves sub-optimal performance, which suggests that a hierarchical RNN method has difficulty performing long-term memorization and mining valid information. Moreover, the original ranking position of candidate documents is used as a additional feature in HRNN+QA. However, such kind of feature is inaccessible in these two benchmark datasets. RPMN outperforms HRNN+QA in terms of all metrics, which suggests external memories are helpful for personalized multi-session search. As expected, we see multi-task learning methods exhibit an improvement over other baselines. Specifically, we see that M-Match Tensor achieves a 1.25% (1.32%) increase over Match Tensor in terms of MAP (average NDCG) on AOL dataset, which indicates that incorporating a query prediction component is helpful for document reranking. CARS performs the best among all multi-task learning baselines, which confirms the validity of modeling the associated clicks for document reranking.

Table 2 shows that LostNet significantly outperforms all baselines in terms of all evaluation metrics on both datasets. In terms of MAP, LostNet obtains a 4.82% and 6.68% increase over the best-performing baseline CARS on the two datasets, respectively. Thus we conclude that leveraging multi-session personalized information with memory networks is beneficial for document reranking.

6.1.2 Next query prediction. We evaluate the models in two ways: (a) by identifying users' next query from a list of candidate queries, and (b) by generating users' next query [2]. Table 3 shows the results for the two datasets. As expected, seq2seq performs worst. However, the attention mechanism significantly enhances seq2seq in terms of all evaluation metrics. Seq2seq+Attn. even obtains the best performance among all baselines in terms of both BLEU-1 and MRR metrics on the SogouQ dataset, which confirms the value of applying word-level attention on current query. MNSRF outperforms HRED-qs in terms of all BLUE scores and MRR by incorporating the document reranking component, which indicates that multi-task learning also enhances the next query prediction. LostNet outperforms all baselines with significant margins in terms of both BLEU and MRR. This shows the advantage of our learned dynamic user profiles and their utility to the query prediction task.

6.2 Impact of long-term search intent (RQ2)

To address **RQ2**, we compare the performance of LostNet and its variant, LostNet-Short, on the document reranking and next query

Table 2: Performance comparison of document reranking models on two datasets. Boldface indicates leading results. Significant improvements over the best baseline results are marked with * (paired t-test, p-value < 0.01).

		AOL search log						SogouQ					
				NDCG						NDCG			
Model type	Model name	MAP	MRR	@1	@3	@5	@10	MAP	MRR	@1	@3	@5	@10
Classical	BM25	0.2520	0.2593	0.1546	0.2024	0.2306	0.2745	0.3067	0.3219	0.2004	0.2680	0.3029	0.3500
Represent. Focused	DSSM	0.2657	0.2696	0.1753	0.2214	0.2572	0.2901	0.2861	0.2992	0.1476	0.2326	0.2805	0.3524
	CLSM	0.3973	0.4285	0.3082	0.3744	0.3998	0.4354	0.2903	0.3026	0.1507	0.2381	0.2871	0.3540
	ARC-I	0.4952	0.5266	0.3845	0.4697	0.5149	0.5620	0.4420	0.4676	0.3502	0.4166	0.4501	0.4901
Interact. Focused	DRMM	0.3386	0.3685	0.2543	0.3024	0.3353	0.3796	0.2350	0.2515	0.1601	0.2000	0.2252	0.2573
	ARC-II	0.5786	0.6102	0.4925	0.5613	0.5972	0.6340	0.4558	0.4806	0.3634	0.4303	0.4652	0.5041
Represent. & Interact.	DUET	0.5635	0.6056	0.5149	0.5444	0.5724	0.6108	0.5233	0.5551	0.4767	0.5031	0.5270	0.5565
	Match Tensor	0.6796	0.7124	0.6402	0.6687	0.6905	0.7175	0.5672	0.5990	0.5215	0.5490	0.5713	0.6010
Personalized	HRNN+QA	0.3697	0.4283	0.3924	0.4671	0.4925	0.5197	0.3876	0.5108	0.2921	0.3195	0.3405	0.3764
	RPMN	0.5674	0.5739	0.4807	0.5539	0.5739	0.5981	0.6006	0.6097	0.5007	0.5922	0.6160	0.6404
Multi-task	M-NSRF	0.6856	0.7194	0.6484	0.6759	0.6978	0.7225	0.5798	0.6098	0.5217	0.5630	0.5881	0.6189
	M-Match Tensor	0.6881	0.7212	0.6512	0.6771	0.6989	0.7255	0.5865	0.6167	0.5301	0.5681	0.5946	0.6270
	CARS	0.7426	0.7598	0.6482	0.7448	0.7653	0.7859	0.6295	0.6547	0.5657	0.5983	0.6345	0.6779
Ours	LostNet-Short	0.7603*	0.7830*	0.7214*	0.7535*	0.7721*	0.7916*	0.6579*	0.6799*	0.5993*	0.6425*	0.6662*	0.6944*
	LostNet	**0.7784***	**0.8001***	**0.7397***	**0.7732***	**0.7908***	**0.8092***	**0.6716***	**0.6945***	**0.6131***	**0.6590***	**0.6811***	**0.7078***

Table 3: Performance comparison of next query prediction models on two datasets. Boldface indicates leading results. Significant improvements over the best baseline results are marked with * (paired t-test, p-value < 0.01).

	AOL search log (English)					SogouQ (Chinese)				
	BLEU					BLEU				
Model	1	2	3	4	MRR	1	2	3	4	MRR
Single-task learning										
Seq2seq	4.1	0.0	0.0	0.0	0.2204	2.0	0.3	0.0	0.0	0.4185
Seq2seq+Attn.	20.8	8.3	3.5	1.3	0.3172	44.5	19.6	8.8	3.2	0.5870
HRED-qs	20.8	8.3	3.7	1.5	0.3320	38.4	15.3	7.9	3.4	0.5216
Multi-task learning										
M-Match Tensor	2.3	0.4	0.1	0.0	–	5.0	0.9	0.2	0.0	–
M-NSRF	22.0	9.3	4.5	2.0	0.3501	41.9	18.0	9.3	5.1	0.5747
CARS	10.9	6.1	3.4	1.7	0.3230	28.9	19.7	9.2	5.8	0.5521
Personalized multi-task learning										
LostNet-Short	30.2*	14.1*	**8.4***	**4.0***	0.3781*	49.2*	23.7*	11.3*	6.6*	0.6168*
LostNet	**32.1***	**14.3***	8.1*	3.3*	**0.4283***	**49.7***	**23.7***	**11.5***	**6.8***	**0.6262***

prediction tasks. LostNet-Short refers to LostNet without the long-term search intent, that uses an hierarchical session-based attention mechanism to process the sequential interactions in an ongoing session for joint learning of DR and NQP. As shown in Table 2, without modeling information about past sessions, LostNet suffers a significant performance drop in the document reranking task. Specifically, LostNet obtains improvements over LostNet-Short of 2.38% and 2.08% in MAP on the two datasets, respectively. With regard to average NDCG score, the relative improvements over LostNet-Short are 2.45% and 2.25%, respectively. This indicates that it is useful to combine short-term search interest and a user's long-term search profile for better document reranking. We also report their query prediction performance in Table 3. We observe that LostNet outperforms LostNet-Short in most cases. In terms of average BLEU score, LostNet achieves a 1.94% (0.99%) increase over LostNet-Short on the AOL (SogouQ) dataset; in terms of MRR, LostNet offers improvements of up to 13.28% and 1.52%, respectively. We conclude that modeling long-term search interactions helps to learn a more precise user profile, and consequently yield better reranking and query prediction results.

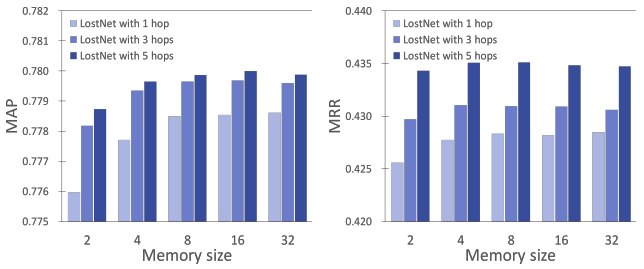
6.3 Setting the personalized memory encoder (RQ3)

To address **RQ3**, we analyze the influence of two parameters of the personalized memory encoder, the number of hops H and the memory size k. We report our results in Fig. 3 for (a) document reranking, and (b) next query prediction on the AOL dataset. We observe that: (1) LostNet with more hops achieves better performance, which indicates the effectiveness of a multi-hop mechanism during dynamic profile learning. (2) The performance of LostNet in terms of MAP increases with the growth of the memory size; and the query prediction performance measured by MRR has a roughly similar trend. This is because more historical session information is used for predicting the next click and query. (3) The performance of LostNet improves rapidly when the memory size is fewer than 8. With the increase of the memory size, the performance gain gradually diminishes in both tasks. Hence, it is reasonable to adopt a first-in-first-out mechanism to maintain the latest k sessions in user memory matrix, as a more recent session should contribute more than an older one to the current search.

The discussions above show that accurately inferring personalzied search intent depends on the degree to which we are able to incorporate previous session information. We conclude that our proposed PME is able to effectively exploit evidence from historical search activities and yield better personalization.

Table 4: Ablation on joint learning for performance analysis of LostNet.

	AOL search log (English)									SogouQ (Chinese)								
		NDCG				BLEU					NDCG				BLEU			
LostNet variant	MAP	@1	@3	@5	@10	1	2	3	4	MAP	@1	@3	@5	@10	1	2	3	4
w/o Recommender	0.7713	0.7363	0.7647	0.7817	0.8010	–	–	–	–	0.6292	0.5805	0.6129	0.6344	0.6595	–	–	–	–
w/o Ranker	–	–	–	–	–	31.6	13.3	7.4	3.0	–	–	–	–	–	**49.8**	23.7	10.8	6.3
LostNet	**0.7784**	**0.7397**	**0.7732**	**0.7908**	**0.8092**	**32.1**	**14.3**	**8.1**	**3.3**	**0.6716**	**0.6131**	**0.6590**	**0.6811**	**0.7078**	49.7	**23.7**	**11.5**	**6.8**



(a) Evaluation on document reranking. (b) Evaluation on query suggestion.

Figure 3: Performance comparison of LostNet with different numbers of hops and slots in user personalized memory component on the AOL dataset.

6.4 Effect of joint learning (RQ4)

To answer **RQ4**, we turn off the document reranking and query prediction components in LostNet, one at a time. The empirical results on the two datasets are summarized in Table 4. We first train LostNet without query prediction to purely focus on document reranking. We find a significant decrease in reranking performance, which indicates the utility of supervision signals from the query recommender to the ranker. Then, when the document ranker is disabled, we observe that the prediction performance of LostNet also receives a consistent drop in terms of all BLEU scores on the AOL dataset. This shows that a regularization effect of document reranking task improve the generalization ability of query prediction task.

As to the SogouQ dataset, LostNet with joint learning achieves a 6.74% increase over the condition without prediction in terms of MAP on the reranking performance, and it achieves a 1.21% increase over the condition where it is learns without the reranker in terms of the average BLUE score on prediction performance. By leveraging the training signals of the two objectives, our joint learning framework learns more effective shared representations, which is suitable for two related tasks. Hence, we conclude that joint learning of these two tasks for our proposed personalized approach mutually benefits both tasks.

7 CONCLUSION

We have considered the task of personalized joint learning of document reranking and next query prediction. Previous work on this joint learning task has mostly neglected to use a user's long-term information. In our work, we have identified three main challenges: (1) the difficulty of capturing long-term search intent, (2) the lack of fine-grained information about session-level search intent, and (3) low efficiency during profiling information from previous sessions. We have proposed a long short-term session search network (LostNet) to address these challenges. LostNet is composed of three components: (1) session-level intent tracking, (2) personalized search intent tracking, and (3) joint learning of document reranking and query recommendation. In our experiments, we have demonstrated the effectiveness of LostNet, finding significant improvements over state-of-the-art baselines for both document reranking and next query prediction on two benchmark datasets.

Limitations of our work concern the fact that we do not use collaborative information in neighborhood sessions [54] and the lack of external knowledge in LostNet. As to future work, we plan to apply other users' sessions so as to enhance the current session search. Also, our solution can be transferred to broader domains related to sequential interactions between users and a system. And it would be interesting to study LostNet in an online setting. Lastly, we would like to integrate external domain knowledge to further improve the performance of LostNet.

REPRODUCIBILITY

To facilitate reproducibility of the results in this paper we share data and code at https://github.com/QiannanCheng/LostNet.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China with grant No. 2020YFB1406704, the Natural Science Foundation of China (61972234, 61902219, 61672324, 61672322, 62072279), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Fundamental Research Funds of Shandong University, the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R. China (COGOSC-20190003). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-task Learning for Document Ranking and Query Suggestion. In *ICLR*.
[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. *arXiv preprint arXiv:1906.02329* (2019).
[3] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query Recommendation using Query Logs in Search Engines. In *EDBT*. 588–596.
[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473* (2014).
[5] Paul N Bennett, Ryen W White, Wei Chu, Susan T Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the Impact of Short- and Long-term Behavior on Search Personalization. In *SIGIR*. 185–194.
[6] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning End-to-end Goal-oriented Dialog. *arXiv preprint arXiv:1605.07683* (2016).

[7] Fei Cai and Maarten de Rijke. 2016. A Survey of Query Auto Completion in Information Retrieval. *Foundations and Trends in Information Retrieval* 10, 4 (2016), 273–363.

[8] Fei Cai, Shangsong Liang, and Maarten de Rijke. 2014. Time-sensitive Personalized Query Auto-completion. In *CIKM*. 1599–1608.

[9] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards Query Log based Personalization using Topic Models. In *CIKM*. 1849–1852.

[10] Rich Caruana. 1997. Multitask Learning. *Machine learning* 28, 1 (1997), 41–75.

[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *SIGIR*. 1093–1096.

[12] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *WSDM*. 108–116.

[13] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search Engines: Information Retrieval in Practice*. Vol. 520. Addison-Wesley Reading.

[14] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-based Query Suggestion. In *CIKM*. 1747–1756.

[15] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A Large-scale Evaluation and Analysis of Personalized Search Strategies. In *WWW*. 581–590.

[16] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *SIGIR*. 515–524.

[17] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In *CIKM*. 347–356.

[18] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM*. 55–64.

[19] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NeurIPS*. 2042–2050.

[20] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware Multi-hop Reasoning Networks for Sequential Recommendation. In *WSDM*. 573–581.

[21] Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. 2018. Improving Entity Recommendation with Search Log and Multi-Task Learning.. In *IJCAI*. 4107–4114.

[22] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM*. 2333–2338.

[23] Aaron Jaech, Hetunandan Kamisetty, Eric Ringger, and Charlie Clarke. 2017. Match-tensor: A deep Relevance Model for Search. *arXiv preprint arXiv:1701.07795* (2017).

[24] Bernard J. Jansen, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a Session on Web Search Engines: Research Articles. *Journal of the American Society for Information Science and Technology* 58, 6 (2007), 862–871.

[25] Jyun-Yu Jiang and Wei Wang. 2018. RIN: Reformulation Inference Network for Context-Aware Query Suggestion. In *CIKM*. 197–206.

[26] Rosie Jones and Kristina Lisa Klinkner. 2008. Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs. In *CIKM*. 699–708.

[27] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *ICML*. 1378–1387.

[29] Hang Li and Jun Xu. 2014. Semantic Matching in Search. *Foundations and Trends in Information Retrieval* 7, 5 (2014), 343–469.

[30] Xiujun Li, Chenlei Guo, Wei Chu, Ye-Yi Wang, and Jude Shavlik. 2014. Deep Learning Powered In-session Contextual Ranking using Clickthrough Data. In *NeurIPS*.

[31] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation Learning Using Multi-task Deep Neural Networks for Semantic Classification and Information Retrieval. In *ACL*. 912–921.

[32] Yiqun Liu, Fei Chen, Weize Kong, Huijia Yu, Min Zhang, Shaoping Ma, and Liyun Ru. 2012. Identifying Web Spam with the Wisdom of the Crowds. *ACM Transactions on the Web* 6, 1 (2012), 2.

[33] Yiqun Liu, Junwei Miao, Min Zhang, Shaoping Ma, and Liyun Ru. 2011. How Do Users Describe their Information Need: Query Recommendation based on Snippet Click Model. *Expert Systems with Applications* 38, 11 (2011), 13847–13856.

[34] Shuqi Lu, Zhicheng Dou, Xu Jun, Jian-Yun Nie, and Ji-Rong Wen. 2019. PSGAN: A Minimax Game for Personalized Search with Limited and Noisy Click Data. In *SIGIR*. 555–564.

[35] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint arXiv:1508.04025* (2015).

[36] Bhaskar Mitra. 2015. Exploring Session Context using Distributed Representations of Queries and Reformulations. In *SIGIR*. 3–12.

[37] Bhaskar Mitra and Nick Craswell. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends in Information Retrieval* 13, 1 (2018), 1–126.

[38] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *WWW*. 1291–1299.

[39] Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Thanh Nguyen, Dan Xu, Byron C. Wallace, Maarten de Rijke, and Matthew Lease. 2018. Neural Information Retrieval: At the End of the Early Years. *Information Retrieval Journal* 21, 2–3 (2018), 111–182.

[40] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *AAAI*.

[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.

[42] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A Picture of Search. In *InfoScale*.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[44] Stephen Robertson, Hugo Zaragoza, et al. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[45] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-pooling Structure for Information Retrieval. In *CIKM*. 101–110.

[46] Ahu Sieg, Bamshad Mobasher, and Robin Burke. 2007. Web Search Personalization with Ontological User Profiles. In *CIKM*. 525–534.

[47] Yang Song, Hongning Wang, and Xiaodong He. 2014. Adapting Deep RankNet for Personalized Search. In *WSDM*. 83–92.

[48] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-decoder for Generative Context-aware Query Suggestion. In *CIKM*. 553–562.

[49] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end Memory Networks. In *NeurIPS*. 2440–2448.

[50] Jaime Teevan, Daniel J Liebling, and Gayathri Ravichandran Geetha. 2011. Understanding and Predicting Personal Navigation. In *WSDM*. 85–94.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[52] Thanh Vu, Alistair Willis, Son N Tran, and Dawei Song. 2015. Temporal Latent Topic User Profiles for Search Personalisation. In *ECIR*. 605–616.

[53] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner Attention based Recurrent Neural Networks for Answer Selection. In *ACL*. 1288–1297.

[54] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *SIGIR*. 345–354.

[55] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *arXiv preprint arXiv:1410.3916* (2014).

[56] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Query Suggestion with Feedback Memory Network. In *WWW*. 1563–1571.

[57] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR*. 55–64.

[58] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM*. 789–797.