Knowledge Enhanced Personalized Search

Shuqi Lu[1,3], Zhicheng Dou[2], Chenyan Xiong[5], Xiaojie Wang[6], and Ji-Rong Wen[3,4]
[1]School of Information, Renmin University of China, [5]Microsoft Research AI
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Key Laboratory of Data Engineering and Knowledge Engineering, MOE
[6]School of Computing and Information Systems, University of Melbourne
lusq@ruc.edu.cn,dou@ruc.edu.cn,Chenyan.Xiong@microsoft.com,Xiaojiew94@gmail.com,jirong.wen@gmail.com

ABSTRACT

This paper presents a knowledge graph enhanced personalized search model, KEPS. For each user and her queries, KEPS first conducts personalized entity linking on the queries and forms better intent representations; then it builds a knowledge enhanced profile for the user, using memory networks to store the predicted search intents and linked entities in her search history. The knowledge enhanced user profile and intent representation are then utilized by KEPS for better, knowledge enhanced, personalized search. Furthermore, after providing personalized search for each query, KEPS leverages user's feedback (click on documents) to post-adjust the entity linking on previous queries. This fixes previous linking errors and improves ranking quality for future queries. Experiments on the public AOL search log demonstrate the advantage of knowledge in personalized search: personalized entity linking better reflects user's search intent, the memory networks better maintain user's subtle preferences, and the post linking adjustment fixes some linking errors with the received feedback signals. The three components together lead to a significantly better ranking accuracy of KEPS.

KEYWORDS

Knowledge Base, Personalized Search, Entity-Oriented Search

ACM Reference Format:

Shuqi Lu, Zhicheng Dou, Chenyan Xiong, Xiaojie Wang, and Ji-Rong Wen. 2020. Knowledge Enhanced Personalized Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research & Development in Information Retrieval(SIGIR'20), July 25-30, 2020, Virtural Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401089

1 INTRODUCTION

Personalized search customizes document rankings for each user based on her individual interests in order to improve her search experiences. Previous research often personalizes the search results by matching documents with user profiles, which represent user's intents and are constructed using user's search history [16, 18, 22, 30, 34, 35]. On the other hand, recent research in entity-oriented search utilizes the explicit semantics, e.g. entities and relations from knowledge graphs, in search systems and effectively improves the text representation and ranking accuracy [14, 19, 21, 28, 40, 42, 43].

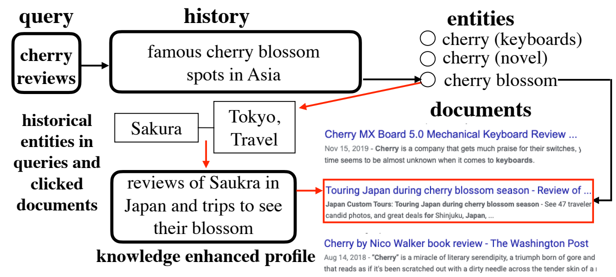


Figure 1: knowledge enhanced personalized search example

Both personalized search and entity-oriented search leverage information beyond the current query to better understand and satisfy user's information needs. Their advantages also naturally reinforce each other. One key challenge of entity-oriented search is the difficulty of query entity linking. Queries are often short and ambiguous[42], making query entity linking a challenging task: A recent study shows that state-of-the-art entity linking techniques only have 50% accuracy on web queries [41]. Using such noisy query entities in ranking often requires manual annotations [12] or soft linking/diversification [42]. Personalization provides a natural way to help resolve the ambiguity in query entity linking: For example, in Fig. 1, the query "cherry reviews" alone is ambiguous, but her search history shows she is referring to cherry the flower.

On the other hand, the explicit semantics from knowledge graphs provide a natural way to represent user's search intent and preference profile. The entity-based representations are intuitive, explainable, and incorporate *external knowledge* that is hard to capture by the word-based or embedding-based representations used in personalized search [2]. In Fig. 1, entities such as Sakura, Tokyo, Japan, and Travel may frequently appear in the user's search history, and intuitively form the "points of interests" for her profile. So it becomes obvious that the personalized information needs for the current query would be "reviews of Sakura in Japan and trips to see their blossom", which is much different from "reviews for cherry keyboards" or the novel.

This paper presents KEPS, "Knowledge Enhanced Personalized Search", a new model that brings together the advantages of entity-oriented search and personalized search. Given user's search history and her current query, KEPS first conducts *personalized entity linking* on the query, leveraging the rich information from her search

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401089

history to better disambiguate the entity linking, and to better represent the current search intent using these entities. Then the personalized entity annotations enable KEPS to construct entity enhanced user profiles, using a memory network that represents user's search preferences in the word-entity duet representation space [42]. KEPS then conducts personalized ranking to adapt document ranking to satisfy user's information need, using the personalized search intent and the knowledge enhanced user profiles.

KEPS also embraces the multi-round nature of personalized search: the system can be viewed as a personal assistant tailored for the individual user's interests, while also learns from the interactions with the user. During search, after the personalized ranking is provided and the interactive feedback is received from the user (i.e. click), KEPS leverages the feedback signals as extra information to revise the system's understanding of the user's preference. Specifically, this is achieved by *post-ranking entity linking adjustment*, where KEPS fixes previous linking errors according to user's click preference on documents. This helps to improve the ranking quality for future queries.

In our experiments on a recent contextualized search dataset [1], KEPS provides significantly more accurate search results than: previous ad-hoc ranking models, entity-oriented models, session-based models, and personalization models. Our analyses confirm that KEPS effectively brings together personalized search and entity-oriented search. Personalization effectively improves the entity linking accuracy, yielding significantly better results in ambiguous queries. Enhancing user's profiles with entities helps to capture more nuances in user preferences; the word-entity duet memory network is effective in modeling long-term history. The post search entity linking adjustment further improves the ranking accuracy on ambiguous queries, learning from the user's interactive feedback.

We also provide additional studies on the effectiveness of personalized search versus session-based search; the comparisons with the state-of-the-art session-based ranker [1] and KEPS's different variants demonstrate the advantage of KEPS in modeling and balancing the personalized signals from both user's session search and long-standing preferences.

2 RELATED WORK

Personalized Web Search. In addition to search result diversification [36, 37], personalized search is another way to address the problem of vauge queries to search engines. Personalized search has been widely studied for its ability to model users' preferences and adapt document rankings to users' query intent. Different from session-based search which utilizes short term search context, personalized search learns to model user's interest using long-interval search history. Early personalization methods [3, 4, 6, 7, 9, 13, 17, 23, 29, 31–35, 38, 39] manually extracted the rich click features and topic features according to user's historical searches and clicks, which are effective in personalized search. Specifically, some works [13, 32] studied personalized click features, and Dou et al. [13] proposed P-Click using click features to improve personalized ranking effect. Other works [3, 9, 17, 23, 29, 34, 35, 38] extracted the topics features from user's search history to predicted document relevance. Click features and topic features are combined and studied in some researches [4, 6, 33, 38, 39]. Bennett et al. [4] proposed SLTB to combine the two types of features using learning to rank.

Recently, deep learning has been applied in personalized search [16, 18, 22, 25, 30]. It significantly improved personalization by learning the effective representations of user profiles and other personalized features from user's history. Ge et al. HRNN [16] proposed to use a hierarchical RNN to model user's profile. PSGAN [22] proposed a generative adversarial network framework to promote the training of deep personalized models which further enhances the personalized effect.

Another challenge in personalized search is that many search logs are not publicly available [4, 16, 22, 25, 34]. In the released dataset from Yandex[1], the text contents of queries and documents have been encrypted, making it impossible to link them to entities. The Sogou dataset [20] contains only one month of user search log, limiting the effectiveness of personalized search which often uses a longer period to construct user profiles. Recently, the context search dataset constructed by Wasi et al. [1], which is based on the public AOL search log [26], makes it possible to study personalized search in the public domain. We conduct experiments on this dataset [1].

Entity-Oriented Search. There have been many attempts to integrate entity knowledge into ad-hoc web search. Some take entities contained in the query or document as a kind of relevance ranking features, such as term weight in queries according to entity descriptions [8, 12]. There are also some researches using entities as connections between the documents and queries for better matching. Liu et al. [19] and Xiong et al. [40] takes the entity as a latent space and learn query-document matching relevance through the latent space. Ensan et al. [14] used a probability model to model the semantic entity linking of documents and queries. Raviv et al. [28] used a language model to balance the entity-based and term-based information. Some researches also utilize entity representations. Xiong et al. [41] consider the bags of entity representations in search model, and the interaction between bags of word representations and bags of entity representations is also studied in [42]. Neural-based search model EDRM [21] study the interaction between word vectors and entity vectors. Our model KEPS is also a neural-based model using entity representations, but focuses on entity enhanced personalized search.

3 OUR APPROACH

The architecture of KEPS includes four components, as shown in Fig. 2: *personalized entity linking* for better query intent modeling; *user profile constructing* for user preference modeling according to the query intents; *personalized ranking* for document relevance modeling according to the query intents and profiles; *post-ranking entity linking adjustment* for adjusting entity linking probabilities of previous queries using user's feedback for personalized ranking of subsequent queries. These components are trained together using the user's click information.

3.1 Problem Definition

This section describes the notation and the specific task in this paper. We denote the search history of a user u by $\mathcal{H} = [S_1, ..., S_m]$, which

[1] https://www.kaggle.com/c/yandex-personalized-web-search-challenge/data

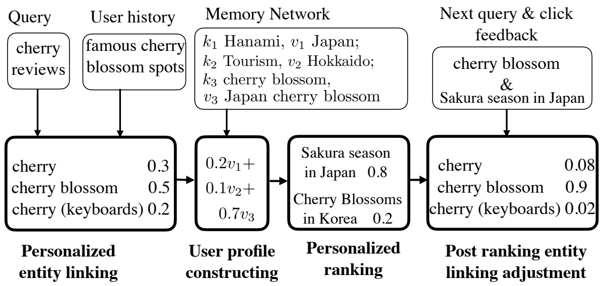


Figure 2: The KEPS framework.

consists of a series of sessions S. Each session contains pairs of a query and a list of candidate documents, i.e., for the h-th sesssion $S_h = [(q_1^h, \mathcal{D}_1^h), ..., (q_{x_h}^h, \mathcal{D}_{x_h}^h)]$, where x_h is the number of queries in the session. To simplify notation, we suppose the current session is the m-th session in which user has issued $t-1$ queries, i.e., $S_m = [(q_1, \mathcal{D}_1), ...(q_{t-1}, \mathcal{D}_{t-1})]$. Then when the user issues the t-th query q_t, our task is to rank the corresponding candidate document set $\mathcal{D}_t = [d_1, ..., d_{|\mathcal{D}_t|}]$ to satisfy user's search intent on query q_t according to the search history $\mathcal{H}$. This process is repeated until the current session ends.

We define the historical search sequence in the current session as short-term history Q_s, while that in previous sessions before the current session as long-term history Q_l following HRNN [16]:

$$Q_s = [q_1^s, ..., q_{|Q_s|}^s] = [q_1^m, ...q_{t-1}^m],$$
$$Q_l = [q_1^l, ..., q_{|Q_l|}^l] = [q_1^1, .., q_{x_1}^1, ..., q_1^{m-1}, ..., q_{x_{m-1}}^{m-1}].$$

Modeling short-term and long-term history separately is effective because history in the current session tends to reflect user's session search intent, while the previous history may reflect user's global interests [16, 22].

Suppose a query q has x entity mentions (text string in query that may refer to certain entity), we denote the candidate entity list for the query by $\mathcal{E} = [[e_{1,1}, ..., e_{1,n_1}], ..., [e_{x,1}, ..., e_{x,n_x}]]$ where n_x indicates the number of candidate entities for the x-th mention. Using Emb_e to represent the entity embedding layer, we then define the entity embedding for queries and documents, Emb_e^q and Emb_e^d, which incorporated with semantic information from knowledge graph. For query q and document d, we have:

$$\text{Emb}_e^q(q) = \sum_{i=1}^{x}\sum_{j=1}^{n_i} p_{i,j}\boldsymbol{e}_{i,j}\ , \text{Emb}_e^d(d) \quad = \sum_{e_i \text{ in } d} c_i * \boldsymbol{e}_i,$$

where $p_{i,j}$ is the link probability of the entity $e_{i,j}$ (see Sec. 3.2 for details), $\boldsymbol{e}_i$ is the embedding of entity e_i, and c_i denotes the frequency of entity. Let $\boldsymbol{w}_i$ be the embedding of word w_i, we define the text embedding for queries and documents, Emb_w^q and Emb_w^d:

$$\text{Emb}_w^q(q) = \sum_{w_i \text{ in } q} c_i * \boldsymbol{w}_i\ , \text{Emb}_w^d(d) \quad = \sum_{w_i \text{ in } d} c_i * \boldsymbol{w}_i.$$

3.2 Personalized Entity Linking

To leverage knowledge in modeling user's search intent, we conduct personalized entity linking for queries as shown in Fig. 3. This is because: with search history, we can improve the linking accuracy; with explicitly linked entities, we can better represent user's intent for subsequent user profile constructing and personalized ranking.

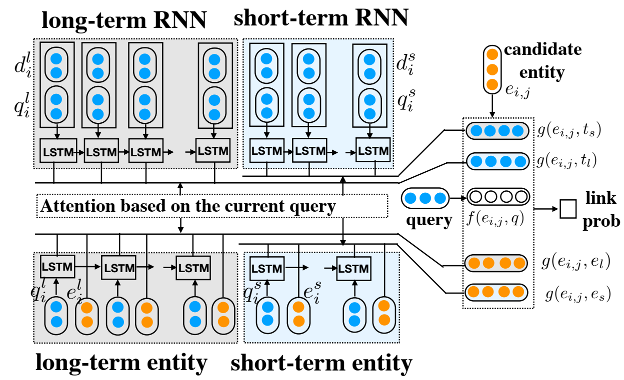


Figure 3: Structure of personalized entity linking.

We use soft-alignment and predict the link probability for all the candidate entities. The link probability of the j-th candidate entity of the i-th mention in the current query q is calculated by the relevance with query text and personalized history:

$$p_{i,j} = \mathcal{F}(e_{i,j}|q, \mathcal{H}) = \frac{\exp(\text{MLP}(f(e_{i,j}, q) \oplus f(e_{i,j}, \mathcal{H})))}{\sum_{k=1}^{n_i} \exp(\text{MLP}(f(e_{i,k}, q) \oplus f(e_{i,k}, \mathcal{H})))},$$

where $\text{MLP}(\cdot)$ denotes a multi-layer perceptron model, and $\oplus$ is the operation of vector concatenation. The relevance between entity $e_{i,j}$ and query q includes vector similarity and statistical features:

$$f(e_{i,j}, q) = \tanh(\boldsymbol{e}_{i,j}^{\mathsf{T}} * \text{MLP}(\boldsymbol{q})) \ \oplus \text{MLP}(\boldsymbol{l}_{i,j}),$$

where $\boldsymbol{q}$ is the text embedding of query q, and $\boldsymbol{l}_{i,j}$ is the statistical features including the popularity of the candidate entity and the linking scores given by TAGME [15].

Since user's historical search behavior reflects user's implicit preference, it is helpful for query disambiguation and entity linking. Specifically, we model user's search history to calculate the historical linking features $f(e_{i,j}, \mathcal{H})$ from two perspectives: mining the sequential information in user's search history to provide the basis for entity linking; detect similar historical queries and use associate entities in these queries to predict entity linking.

Sequential History Modeling. We adopt an LSTM layer and an attention mechanism based on the current query to mine user's interests according to her sequential search history. We concatenate the query vectors with its corresponding document vectors as input to feed the LSTM layer. When using the short-term history, we obtain a vector $\boldsymbol{t}_s$ representing user's short-term interests by:

$$\boldsymbol{h}_1^t, ..., \boldsymbol{h}_{|Q_s|}^t = \text{LSTM}([(\boldsymbol{q}_1^s, \boldsymbol{d}_1^s), ..., (\boldsymbol{q}_{|Q_s|}^s, \boldsymbol{d}_{|Q_s|}^s)]), \tag{1}$$

$$\boldsymbol{t}_s = \sum_{i=1}^{|Q_s|} \alpha_i \boldsymbol{h}_i^t \text{ and } \alpha_i = \text{softmax}(\boldsymbol{v}_t^{\mathsf{T}} * \text{MLP}(\boldsymbol{h}_i^t \oplus \boldsymbol{q})),$$

where $\boldsymbol{q}_i^s$ is the text embedding of query q_i^s, the corresponding document vector $\boldsymbol{d}_i^s$ is the average of text embedding of clicked documents corresponding to query q_i^s, and $\boldsymbol{v}_t$ is a parameter vector.

We obtain the long-term interests $\boldsymbol{t}_l$ by replacing $\boldsymbol{q}_i^s$ and $\boldsymbol{d}_i^s$ in Eqn. (1) with the query text embedding $\boldsymbol{q}_i^l$ in long-term history $\mathbf{Q}_l$ and its corresponding clicked document embedding.

Related Historical Entity Extracting. We apply another LSTM layer and an attention mechanism based on the current query to highlighting related historical queries and take the entities in these queries as related historical entities. Taking the historical query as input, when using short-term history we obtain the vector $\boldsymbol{e}_s$ representing related historical entity information by:

$$
\begin{aligned}
\boldsymbol{h}_1^e, ..., \boldsymbol{h}_{|Q_s|}^e &= \mathrm{LSTM}([\boldsymbol{q}_1^s, ..., \boldsymbol{q}_{|Q_s|}^s]), \\
\boldsymbol{e}_s &= \sum_{i=1}^{|Q_s|} \alpha_i \boldsymbol{e}_i^s \text{ and } \alpha_i = \mathrm{softmax}(\boldsymbol{v}_e^{\mathsf{T}} * \mathrm{MLP}([\boldsymbol{h}_i^e \oplus \boldsymbol{q}])),
\end{aligned}
\tag{2}
$$

where $\boldsymbol{q}_i^s$ and $\boldsymbol{e}_i^s$ are the text and entity embedding of query q_i^s.

We obtain a long-term historical entity representation $\boldsymbol{e}_l$ by replacing $\boldsymbol{q}_i^s$ and $\boldsymbol{e}_i^s$ in Eqn. (2) with $\boldsymbol{q}_i^l$ and $\boldsymbol{e}_i^l$ in long-term history.

So the personalized historical linking feature of the j-th entity $e_{i,j}$ of i-th mention is defined by the following four parts, where $g(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\boldsymbol{x}^T * \mathrm{MLP}(\boldsymbol{y}))$ denotes vector similarity:

$$
f(\boldsymbol{e}_{i,j}, \mathcal{H}) = g(\boldsymbol{e}_{i,j}, \boldsymbol{t}_s) \oplus g(\boldsymbol{e}_{i,j}, \boldsymbol{t}_l) \oplus g(\boldsymbol{e}_{i,j}, \boldsymbol{e}_s) \oplus g(\boldsymbol{e}_{i,j}, \boldsymbol{e}_l).
$$

3.3 User Profile Constructing

Using the search intent reflecting by personalized entity linking probabilities, we can retrieve related search history and obtain user's preference from the corresponding clicked documents. Here we use key-value memory networks [24] to store user's history. Further, as stated in Sec 1, entities in user's search history can be utilized to model the subtle preferences, so we proposed both entity memory networks and text memory networks for preference profiles constructing. The structure is shown in Fig 4.

Entity Memory Network treats the historical query entity vectors as key, while the clicked document entity vectors as value. Thus we record user's subtle preferences reflecting by her clicks under certain query intent in her history. For short-term history, we have:

$$
\begin{aligned}
\mathrm{K}_s &= [\boldsymbol{k}_1^s, ..., \boldsymbol{k}_{|Q_s|}^s] = [\boldsymbol{e}_1^s, ..., \boldsymbol{e}_{|Q_s|}^s], \\
\mathrm{V}_s &= [\boldsymbol{v}_1^s, ..., \boldsymbol{v}_{|Q_s|}^s],
\end{aligned}
\tag{3}
$$

where $\boldsymbol{e}_i^s$ is the entity embedding of q_i^s and $\boldsymbol{v}_i^s$ is the average of clicked document entity embedding associated with query q_i^s in short-term history.

Then we construct user profiles using the memory network. First, according to the entity embedding of the current query $\boldsymbol{e}_q = \mathrm{Emb}_e^q(q)$, which reflects user's search intent by the entity linking probabilities, we assign different attention weights to user's historical clicked entities. When using short-term history, we obtain a short-term entity preference profile $\boldsymbol{p}_s^e$ by:

$$
\boldsymbol{p}_s^e = \sum_{i=1}^{|Q_s|} \beta_i \boldsymbol{v}_i^s \text{ and } \beta_i = \mathrm{softmax}(\boldsymbol{k}_i^{s\mathsf{T}} * \boldsymbol{P}_e * \boldsymbol{e}_q). \tag{4}
$$

However using the entity vector we may only find the history slots related to entities in the current query. We further integrate $\boldsymbol{p}_s^e$ into $\boldsymbol{e}_q$ and read related history from the memory network again. Thus we can find history slots related both to the query and to $\boldsymbol{e}_q$ which represent some interest points of the user, and construct a comprehensive preference profile incorporating the user's wider interest. So we have:

$$
\begin{aligned}
\boldsymbol{e}_q' &= \boldsymbol{W}_e * \boldsymbol{p}_s^e + \boldsymbol{e}_q \\
\boldsymbol{p}_s^e &= \sum_{i=1}^{|Q_s|} \beta_i' \boldsymbol{v}_i^s, \ \beta_i' = \mathrm{softmax}(\boldsymbol{k}_i^{s\mathsf{T}} * \boldsymbol{P}_e * \boldsymbol{e}_q'),
\end{aligned}
\tag{5}
$$

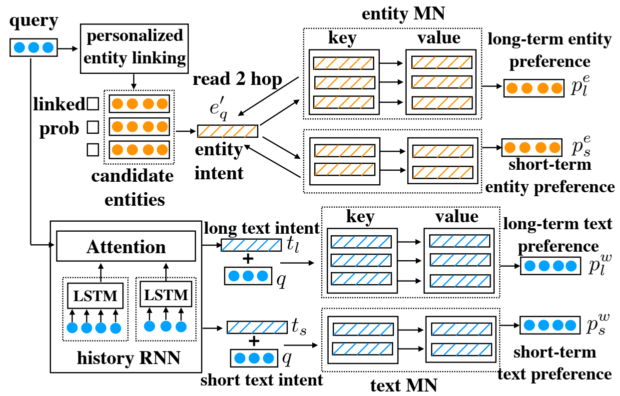


Figure 4: Structure of preference profile constructing.

where $\boldsymbol{W}_e$ and $\boldsymbol{P}_e$ are parameter matrices.

For long-term history, we obtain a long-term entity preference profile $\boldsymbol{p}_l^e$ by replacing $\boldsymbol{k}_i^s$ and $\boldsymbol{v}_i^s$ in Eqn. (3) with $\boldsymbol{e}_i^l$ and corresponding document entity vectors in long-term history. Then read related history as in Eqn. (4) and (5).

Text Memory Network treats the historical query vectors as key, while the average of corresponding clicked document vectors as value. So when in short-term history, we have:

$$
\begin{aligned}
\mathrm{K}_s &= [\boldsymbol{k}_1^s, ..., \boldsymbol{k}_{|Q_s|}^s] = [\boldsymbol{q}_1^s, ..., \boldsymbol{q}_{|Q_s|}^s] \\
\mathrm{V}_s &= [\boldsymbol{v}_1^s, ..., \boldsymbol{v}_{|Q_s|}^s],
\end{aligned}
\tag{6}
$$

where $\boldsymbol{q}_i^s$ is the text embedding of query q_i^s and $\boldsymbol{v}_i^s$ is the average of clicked document embedding associated with query q_i^s. Since the original query string may not completely reflect user's search intent, we splice the query vector $\boldsymbol{q} = \mathrm{Emb}_w^q(q)$ and $\boldsymbol{t}_s$ which reflects search intent and is learned in Sec. 3.2 as an intent vector. Then we read out related history according to the intent vector. Since the associations between text vectors (may be verbose) are not as strong as that between entities, we only read once here. So we obtain a short-term text preference profile $\boldsymbol{p}_s^w$ by:

$$
\begin{aligned}
\boldsymbol{q}' &= \boldsymbol{t}_s \oplus \boldsymbol{q} \\
\boldsymbol{p}_s^w &= \sum_{i=1}^{|Q_s|} \beta_i \boldsymbol{v}_i^s, \ \beta_i = \mathrm{softmax}(\boldsymbol{k}_i^{s\mathsf{T}} * \boldsymbol{P}_w * \boldsymbol{q}').
\end{aligned}
\tag{7}
$$

For long-term history, we read out a long-term personalized text preference profile $\boldsymbol{p}_l^w$ by replacing $\boldsymbol{k}_i^s$ and $\boldsymbol{v}_i^s$ in Eqn. (6) with $\boldsymbol{q}_i^l$ and the corresponding clicked document vectors in long-term history. Then read out related history as in Eqn. (7).

3.4 Personalized Ranking

With the predicted user's search intents in Sec 3.2 and the user's preference profiles constructed in Sec 3.3, we then calculate the relevance of candidate documents to conduct personalized ranking. So given the search history $\mathcal{H}$, we compute the relevance of a candidate document d for the query q by:

$$
\mathcal{F}(d|q, \mathcal{H}) = \mathrm{MLP}(f(d \oplus \mathcal{I}) \oplus f(d, \mathcal{P}) \oplus f(d, q)),
$$

where $\mathcal{I}$ and $\mathcal{P}$ are the predicted intents and preference profiles.

Intention Relevance computes the similarity between the candidate document vectors and the vectors $\boldsymbol{t}_l$, $\boldsymbol{t}_s$,$\boldsymbol{e}_q$ reflecting user's intent in Sec. 3.2 by $g(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\boldsymbol{x}^T * \text{MLP}(\boldsymbol{y}))$:

$$f(d, \mathcal{I}) = g(\boldsymbol{d}, \boldsymbol{t}_s) \oplus g(\boldsymbol{d}, \boldsymbol{t}_l) \oplus g(\boldsymbol{d}^e, \boldsymbol{e}_q) \oplus g(\boldsymbol{d}, \boldsymbol{e}_q)],$$

where $\boldsymbol{d}, \boldsymbol{d}^e$ is the text and entity embedding of document d respectively, and $\boldsymbol{e}_q$ is the entity embedding of the current query.

Preference Relevance is the relevance between the document vectors and profile vectors reflecting user's preference in Sec. 3.3:

$$f(d, \mathcal{P}) = g(\boldsymbol{d}, \boldsymbol{p}_s^w) \oplus g(\boldsymbol{d}, \boldsymbol{p}_l^w) \oplus g(\boldsymbol{d}^e, \boldsymbol{p}_s^e) \oplus g(\boldsymbol{d}^e, \boldsymbol{p}_l^e)].$$

Query relevance is the relevance between documents and the original query, including vector similarity, click features f_d as in [13], and interactive word-entity duet matching features f_m:

$$f(d, q) = g(\boldsymbol{d}, \boldsymbol{q}) \oplus \text{MLP}(f_d) \oplus f_m,$$

where $\boldsymbol{d}, \boldsymbol{q}$ is the text embedding of the query and document. We propose an entity-based matching component incorporating personalized information **PEDRM** to calculate the interactive features f_m. We use $\boldsymbol{e}^q, \boldsymbol{w}^q$ represent the embedding of entities and words in queries, and $\boldsymbol{e}^d$ and $\boldsymbol{w}^d$ represent that in documents. For the current query q with x mentions, we collect all the candidate entities and the corresponding link probabilities into a list:

$$\mathcal{E} = [e_1^q, ..., e_{|\mathcal{E}|}^q] = [e_{1,1}, ..., e_{1,n_1}, ..., e_{x,1}, ..., e_{x,n_x}],$$
$$\boldsymbol{p} = [p_1, ..., p_{|\mathcal{E}|}] = [p_{1,1}, ..., p_{1,n_1}, ..., p_{x,1}, ..., p_{x,n_x}].$$

PEDRM is based on EDRM (EDRM-CKNRM in [21]), the state-of-art ad-hoc model using interactive entity-based featuress. EDRM first constructs the interaction matrices between words and entities in queries and documents, and then uses kernel-pooling to extract the matching features:

$$\phi(\boldsymbol{M}) = \{K_1(\boldsymbol{M}), ..., K_K(\boldsymbol{M})\}, K_k(\boldsymbol{M}) = \sum_j \exp(-\frac{\boldsymbol{M}^{ij} - \mu_k}{2\delta_k^2}),$$
$$f_m = \text{MLP}(\phi(\boldsymbol{M}_{e,e}) \oplus \phi(\boldsymbol{M}_{e,w}) \oplus \phi(\boldsymbol{M}_{w,e}) \oplus \phi(\boldsymbol{M}_{w,w}))),$$

where $\oplus$ is concatenation operation, $\boldsymbol{M}_{e,e}$, $\boldsymbol{M}_{e,w}$, $\boldsymbol{M}_{w,e}$, $\boldsymbol{M}_{w,w}$ denote the interaction of $\boldsymbol{e}^q$-$\boldsymbol{e}^d$, $\boldsymbol{e}^q$-$\boldsymbol{w}^d$, $\boldsymbol{w}^q$-$\boldsymbol{e}^d$, $\boldsymbol{w}^q$-$\boldsymbol{w}^d$ respectively.

PEDRM further incorporates personalized entity linking probability into the interaction matrix of query entities as the weight reflecting user's intent. Furthermore, we also add an interaction matrix R between entity relation vector and the query vector:

$$\boldsymbol{M}_{e,e}^{ij} = p_i * \boldsymbol{M}_{e,e}^{ij}, \boldsymbol{M}_{e,w}^{ij} = p_i * \boldsymbol{M}_{e,w}^{ij},$$
$$\boldsymbol{R}_{i,j} = p_i * (\boldsymbol{r}_{i,j}^\top * \boldsymbol{W}_R * \boldsymbol{q}),$$

where $\boldsymbol{r}_{i,j} = \boldsymbol{e}_i^q - \boldsymbol{e}_j^d$ indicates the entity relation vector since we use TransE [5] to pre-train entity embedding. Adding the matrix R is because the relationship between entities may also reflect the relevance to the query. For example, when querying "Obama's wife", both "Michelle" and "U.S.A" are related to entity "Obama". But the relationship "isWife" between "Michelle" and "Obama" reflects that "Michelle" is more relevant to the query. So f_m is defined by:

$$f_m = \text{MLP}(\phi(\boldsymbol{M}_{e,e}) \oplus \phi(\boldsymbol{M}_{e,w}) \oplus \phi(\boldsymbol{M}_{w,e}) \oplus \phi(\boldsymbol{M}_{w,w}) \oplus \phi(\boldsymbol{R})).$$

3.5 Post-ranking Entity Linking Adjustment

After the ranking, we adjust the entity linking probabilities of historical queries within the current session using the current query and user's feedback. In the same session user's search intents are often consistent, so user's current querying and the click feedback are helpful to understand the previous ambiguous queries. For example, using the entity "software" in current query or user's click feedback, we have higher confidence that the previous ambiguous query "Java" refers to "java language". The adjusted results further contribute to constructing user's preference profile when the user search for "Books for programming" later.

When conduct post-ranking adjustment, our main idea is: firstly select the entity with the highest linking probability which reflects the user's intention; then use this entity to adjust the linking probabilities of entities associated with other mentions in the session.

Specifically, after personalized ranking, we first adjust the linking probabilities for the current query using the entities in the clicked document according to user's click feedback:

$$p_{i,j}^t = p_{i,j}^t + \boldsymbol{e}_{i,j}^{t\top} * \boldsymbol{W} * \boldsymbol{d}_e^t,$$
$$p_{i,j}^t = \frac{\exp(p_{i,j}^t)}{\sum_{k=1}^{n_i^t} \exp(p_{i,k}^t)}, \tag{8}$$

where $\boldsymbol{d}_e^t$ is the average of entity embedding of the current clicked documents. Here we use the superscript to identify the location of the query in the session, and the current is the t-th query. Then we find the entity in the candidate entity set $\mathcal{E}_t$ of the current query with the highest link probability $p = \max(p_{i,j}^t)$, for each $e_{i,j}^t$ in $\mathcal{E}_t$. If $p <= \delta$ (we set $\delta = 0.5$), we end the adjustment and are to rank the $t+1$-th query. δ here denotes a confidence score, which is to filter the linking results with low linking credibility.

If $p > \delta$, and assume the index of the mention associated with the selected entity is a, we then use the predicted entities of a-th mention $\boldsymbol{e}_a^t = \sum_{j=1}^{n_a^t} p_{a,j}^t * \boldsymbol{e}_{a,j}^t$ to adjust the link probabilities in other candidate entity set $\mathcal{E}_k, 1 <= k <= t-1$. According to the entity similarity and text similarity, we adjust the probabilities:

$$p_{i,j}^k = p_{i,j}^k + \text{MLP}(\boldsymbol{e}_a^{t\top} * \boldsymbol{W}_1 * \boldsymbol{e}_{i,j}^k \oplus \boldsymbol{q}^{t\top} * \boldsymbol{W}_2 * \boldsymbol{q}^k),$$

then normalize the link probabilities as Eqn. (8). Next we find the entity in $\mathcal{E}_k, 1 <= k <= t$ with the highest link probability $p = \max p_{i,j}^k$, for each $e_{i,j}^k$ in $\mathcal{E}_k, 1 <= k <= t$. If $p > \delta$ and the mention has not been selected before, then we repeat the above steps using the entities of the selected mention to adjust other link probabilities; else we end the adjustment and are to rank the $t+1$-th query. The adjustment will be stopped if there is no link probability larger than δ, or all the mentions have had been selected. So the maximum number of cycles in the adjustment is the number of mentions in the current session. We take a session as a training unit, and the matrix parameters $\boldsymbol{W}, \boldsymbol{W}_1, \boldsymbol{W}_2$ will be optimized when we reduce the training loss of the session.

3.6 Training

We take a session as a training unit, and use the pairwise loss:

$$l = \sum_u \sum_S \sum_{q \in S} \sum_{d^-, d^+ \in \mathcal{D}} \max(0, 1 - f(q, d^+, \mathcal{H}) + f(q, d^-, \mathcal{H})),$$

Table 1: Specific statistics of experimental data

Type	Train	Valid	Test
User Num	100,110	4,774	5,555
Query Num	598,812	66,879	70,763
Session Num	224,891	27,272	27,767
Avg Session Len	2.66	2.45	2.55
Avg History Len	50.56	64.67	68.18
Avg Click Num	1.11	1.09	1.12

where u denotes user, S denotes session, $\mathcal{H}$ is user's search history, q denotes query, and d^+ represents the positive documents while d^- represents others in the document list $\mathcal{D}$ associated with q.

4 EXPERIMENT SETUP

In this section, we describe the detailed experimental setups.

Dataset. The dataset we use is constructed by Wasi et al. [1] using the AOL search log, in which the candidate documents for each query are collected according to BM25 ranking from a document list. To our best knowledge and as discussed in Sec. 2, AOL search log is the only public dataset that can be used for knowledge enhanced personalized search. We split the search log following [1] to get background set, training set, test and valid set. Note the setting different from [1] is that we keep the background set to provide user's basic historical search sequences for user profile constructing and interest modeling. We divide user's queries into different sessions and the session boundaries are decided based on the differences between query vectors as the task boundaries in [1].

We use the entity titles in Wikipedia. The candidate entities for queries are collected using DEXTER [10] (note that entity linking for queries are done in Sec. 3.2). Since there is no entity annotation in documents in our data, we use TAGME [15] to link entities in Wikipedia to the document titles. And the specific statistics of our experimental data are shown in Tab. 1.

Baselines. Our Baselines include state-of-the-art personalized search methods and entity-based ad-hoc ranking models. Since the candidate documents are retrieved by BM25 score, we take BM25 ranking as original ranking.

`Ad-hoc baselines` include `Conv-KNRM` [11], a neural model using interactive features and `EDRM` (EDRM-CKNRM [21]), the state-of-art ad-hoc model using interactive entity-based features. We use these baselines to reflect the personalized improvement of our model. We also evaluate the effect of `PEDRM` we proposed in Sec 3.4, using TAGME [15] to conduct entity linking for queries and removing the presonalized linking probabilities for comparision.

`Personalization baselines` include the model using traditional features: `P-Click` [13] using click features and `SLTB` [4] using click features and topic features, which is the state-of-art personalization model using traditional features; and the model based on deep learning: `HRNN` (HRNN+QA [16]) using hierarchical RNN and `PSGAN` (we choose the document-selection based model [22]) using adversarial training. To make a fair comparison, we also add entity information to the baseline HRNN to construct an entity-based personalization baseline `HRNN-Entity`. We splice the entity representations and original text representations of documents and queries as new representations, and train HRNN taking them as input. The entity representations of queries and documents are calculated by the average sum of embedding of associated entity linked by TAGME [15]. Since context-aware search also uses part of user's history, we take the state-of-art context search model `CARS` [1] as a baseline to show the effect of personalization.

Implementation Details. The dimension of pre-trained word embedding and entity embedding are 50. The vocabulary size of entities and words is 723,073 and 124,056. We train the word embedding using GloVe [27], taking the query texts and document titles in the search log as training corpus. We train the entity embedding using TransE [5], taking the entities and relations extracted from our data as input. Both the word embedding and entity embedding are fixed in conv-KNRM, EDRM and PEDRM we proposed, but will be mapped into a new vector space by training a projection matrix.The new word vector dimension is 50 and the entity vector dimension is set to 128. All MLPs used in the experiments have one hidden layer. The hidden state of LSTM layers is all set to 100 dimensions. The inner vector in attention layers is all set to 100 dimensions. The setting of CNN layers and kernel functions in PEDRM are consistent with EDRM [21].

Evaluation Metrics. Following the previous work [16, 22], we use MAP, MRR, P@K (precision in the top k positions) and AR (average ranking position of relevant documents) to evaluate our model. We follow official TREC_Eval and break the order of documents with the same ranking scores randomly. We take the right clicked documents as relevant following [1].

5 EVALUATION RESULTS

5.1 Overall Performance

First, we compare KEPS with various comparison models. The overall results are shown in Tab. 2. We have the following findings:

(1) **KEPS outperforms all other baselines with significant improvement, showing the effectiveness of our knowledge enhanced personalization model on user's interests modeling and document ranking.** KEPS outperforms PSGAN by over 20% improvement. Compared with the personalization model added additional entity information HRNN-Entity, KEPS also has significant improvement. This shows that our model has a better effect not only because we introduce rich external knowledge, but also the model structure is very effective.

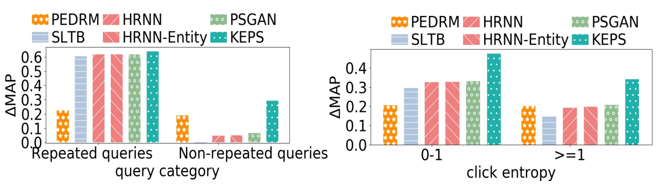
(2) Another finding is that all the personalization models perform better than the ad-hoc search models on MAP, MRR, P@1 and P@3, showing the effect of personalization. However, on the metric of AR and P@5, personalization baseline models underperform the ad-hoc search models. One possible reason is that personalization models may better capture the relevance features for the documents with many related click histories. But for the documents lacking history, they may not learn the matching features between queries and documents as well as the ad-hoc search models, thus lowers AR and P@5. But KEPS performs well on all the metrics including AR and P@5, showing the effectiveness of our model.

(3) PEDRM has further improvement based on EDRM, showing that the interactive matching between entity relations and queries we proposed is helpful to rank the documents.

(4) The results of CARS are lower than that reported in[1] is because we follow official TREC_ Eval and rank documents with

Table 2: Overall ranking effect of KEPS and other baselines. The relative percentages are calculated based on PSGAN. Best results are in bold. † indicates significant improvement over the ad-hoc baseline models Ori.Ranking, Conv-KNRM, EDRM, PEDRM. ‡ indicates significant improvement over the session-based model CARS. $*$ indicates significant improvement over the personalization models P-Click, SLTB, HRNN, HRNN-Entity and PSGAN. ($p < 0.05$ in two-tailed paired t-test).

Model	MAP		MRR		AR		Precision@1		Precision@3		Precision@5	
Ad-hoc Search models												
Ori. R (BM25)	.2501	-54.36%	.2583	-53.86%	17.1516	-67.06%	.1483	-69.69%	.2742	-52.06%	.3452	-42.31%
Conv-KNRM	.3810	-30.47%	.3910	-30.18%	9.4712	+7.75%	.2298	-53.03%	.4612	-19.37%	.5761	-6.17%
EDRM	.4116	-24.89%	.4219	-24.66%	8.8755	+13.55%	.2602	-46.81%	.4998	-12.62%	.6117	-0.37%
PEDRM	.4313	-21.30%	.4436	-20.79%	8.4612	+17.59%	.3019	-38.29%	.5327	-6.87%	.6290	+2.44%
Session Based model												
CARS	.4616†	-15.77%	.4734†	-15.46%	7.956†	+22.51%	.3219†	-34.20%	.5538†	-3.18%	.6568†	+6.97%
Personalization models												
P-Click	.4224	-22.92%	.4298	-23.25%	16.5264	-60.97%	.3788	-22.57%	.4150	-27.45%	.4445	-27.61%
SLTB	.5072†‡	- 7.45%	.5194†‡	-7.25%	13.9264	-35.64%	.4657†‡	-4.80%	.5203	-9.04%	.5451	-11.22%
HRNN	.5423†‡	-1.04%	.5545†‡	-0.98%	10.5523	-2.78%	.4854†‡	-0.78%	.5652	-1.19%	.6046	-1.53%
PSGAN	.5480†‡	-	.5600†‡	-	10.2670	-	.4892†‡	-	.5720†‡	-	.6140	-
Knowledge Enhanced Personalization models												
HRNN-Entity	.5444†‡	-0.66%	.5565†‡	-0.63%	10.4791	-2.07%	.4783†‡	-2.23%	.5676†‡	-0.77%	.6073	-1.09%
KEPS	**.6903**†‡*	+25.97%	**.7044**†‡*	+25.79%	**5.0645**†‡*	+50.67%	**.6124**†‡*	+25.18%	**.7578**†‡*	+32.48%	**.8118**†‡*	+32.21%



(a) (b)

Figure 5: Comparison of model effects on different queries: (a) repeated and non-repeated queries; (b) queries with different click entropy.

the same ranking scores randomly. If we use the evaluation script as in [1], the gain of KEPS is consistent, which reaches 0.78 on MAP.

These results show the effectiveness of KEPS. In the following experiments, we further study the detailed personalized effect.

5.2 Effectiveness on Different Scenarios

we further analyze the detailed personalization effect on two different scenarios: refinding gain and gain on queries with different click entropy. We compare KEPS with personalization baselines SLTB, HRNN, HRNN-Entity and PSGAN, and the ad-hoc search model with the best effect PEDRM (note that we remove the personalized probabilities from PEDRM) to show the difference.

Refinding Experiments. We count the improvement on MAP of models' ranking results over original ranking on repeated queries and non-repeated queries. The repeated queries are the queries that have been issued before by the user, while the non-repeated queries are issued for the first time. By exploring the model effect on these queries, we can know the model's perception of user's history and the ability to infer user's intent for the queries never seen before. The results are shown in Fig. 5a.

In Fig. 5a, all the personalization models perform better on repeated queries than non-repeated queries, while the effect of the ad-hoc search model is similar on both types of queries. This is because repeated queries indeed have more personalized features for personalization models. KEPS performs little better than personalization baselines on repeated queries, but much better on non-repeated queries. It also has a further improvement over the ad-hoc search model on non-repeated queries. This shows that KEPS can not only learn user's historical preference for repeated queries, but also infer user's preference for non-repeated queries. And it does not depend heavily on refinding information.

Click Entropy Experiments. Larger click entropy [13] indicates that the query tends to be informational and ambiguous query while less click entropy indicates the query is likely to be a navigational query. So more attention should be paid on the queries with larger click entropy to explore the personalized effect. We study the ranking effects of models on queries with click entropy < 1 and queries with click entropy >= 1. The results are shown in Fig. 5b.

From Fig. 5b, we still have that the effect of PEDRM is similar on both types of queries. However we find that different from stated in [16, 22], the personalization models perform better on queries with entropy less than 1. This may be because we count the improvement on MAP over the original ranking based on BM25, which is less efficient than the ranking from search engine used in [16, 22]. Compared with SLTB, we can still see HRNN, HRNN-Entity and PSGAN have more improvement on queries with click entropy no less than 1, which is consistent with the previous works [16, 22]. KEPS has significant improvement over the baselines on both two types of queries, and the gain on queries with less click entropy may partly come from the interactive matching component. But the proportion of KEPS's improvement over the personalization baselines is bigger on queries with larger click entropy. So focus on the significant improvement in queries with larger click entropy, we can see KEPS can better conduct personalization on the queries tend to be ambiguous. This meets our experimental expectations.

5.3 Ablation Studies

To analyze the contribution of the main components in KEPS, we experiment with the variations of KEPS. First, to analyze the effect of knowledge enhancement, we set the entity lists associated with queries and documents to empty; this leads `KEPS-noEntity`, which removes all the knowledge enhancement information from KEPS. To study the effect of personalized entity linking in Sec 3.2, memory networks modeling user profile in Sec 3.3, and post ranking entity linking adjustment in Sec 3.5, we remove these parts individually and construct the corresponding model `KEPS-noPSLink`, `KEPS-noMN` and `KEPS-noAdjust`. `KEPS-noPSLink`, directly using TAGME [15] to link entities on the queries.

We also conduct ablation studies on the three relevance modelings in Sec 3.4. First, removing the preference relevance from KEPS leads `KEPS-noMN`. Then since preference modeling is based on intention modeling, we remove both intention and preference relevance, and keep the query relevance to construct `KEPS-QR`. It can also be taken as a basic model to reflect the effect of user profile modeling since it only contains some traditional personalized features and a query-document matching component.

The results are shown in Tab. 3. The improvement from KEPS-noEntity compared with PSGAN is greater than that of KEPS compared with KEPS-noEntity. One possible reason is that both KEPS-noEntity and KEPS contain the component PEDRM which is effective in words matching and may partly make up for the role of entities. But the improvement of KEPS over KEPS-noEntity reflects that the model can further be improved by using entities.

Compared with KEPS, KESP-noLink, KEPS-noMN, and KEPS-noAdjust perform significantly worse in all metrics. This confirms the effectiveness of the three components. Compared with PSGAN, KESP-noLink improves by 22% on MAP and KEPS-noMN improves by 19%, while KEPS-noAdjust improves by 24% near to KEPS. This shows that personalized entity linking and the preference memory networks contribute more to the effect of KEPS. Post linking (KEPS-noAdjust vs. KEPS) does not contribute as much as other components; its effectiveness is restricted by the number of queries whose linking needs post adjustments. KEPS-noMN underperforms KEPS-noEntity shows that memory network effectively stores and models user preference in the entity and text profiles. Since the post-ranking adjustment mechanism mainly works on historical entities, the gap between KEPS-noAdjust and KEPS confirms the effect of personalized entity linking and entity enhanced user profiles.

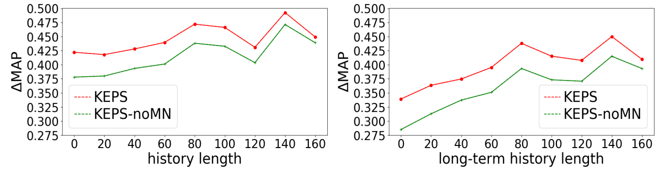
When only use query relevance modeling (KEPS-QR), the model effect declines the most. Adding intention relevance modeling (KEPS-noMN) leads to better ranking accuracy. KEPS performs the best with three relevance modeling components added. The three components all contribute significantly to KEPS's effectiveness.

5.4 Effect of Personalized Entity Linking

To analyze the personalized linking effect, we compare the average MAP improvement over original ranking of the personalization baselines and our models, KEPS-noPSLink and KEPS on ambiguous queries. Here we define the queries containing mentions with more than one candidate entities as ambiguous queries that need to be conducted entity linking. The results are shown in Tab. 4. The specific number of ambiguous queries in our data is in Tab. 5 to show the room for improvement. We can see that KEPS has better effect on both two types of queries than other baseline models but has more improvement on the ambiguous queries. Especially compared with KEPS-noPSLink, the improvement of KEPS on the ambiguous queries is 5.65% while on others is 3.61%, which confirms the effectiveness of personalized entity linking. Compared with other baselines, KEPS-noPSLink is also more effective on ambiguous queries, and this may be because TAGME also has a good effect on entity linking. This shows using entities to represent user's intent is effective especially on ambiguous queries in personalized search.

Further we compare the linking effect of personalized entity linking (PSLink) and TAGME. We randomly select 100 ambiguous queries in test data and get the ground-truth label of entity annotation by manual labeling by two people with the consistency coefficient Kappa =0.69. The numbers of correct linked entities of TAGME and our model KEPS (we select the entity with the highest probability as linked entity) in Tab. 5 show that with personalized information, we can improve the linking quality for queries.

These results show that personalized entity linking helps to better infer and represent user's intent, thus helps personalized document ranking.



(a) (b)

Figure 6: Effect of MN on queries with different length of history: (a) overall history; (b) long-term history.

5.5 Effect of Memory Network

Since memory networks are used to preserve user's preference in search history, we analyze the model performance on queries with different history length. We compare the effects of the length of all history and the length of long-term history on the model performance. To avoid the influence of short-term history, when studying the long-term history we focus on the first query in each session which has no short-term history. To balance the number of queries in each interval, we divide the queries with different history lengths into different groups at 20 intervals. We calculate the MAP improvement of KEPS and KEPS-noMN in Fig. 6a and 6b.

In both figures, the consistent trend is that the improvement of KEPS over KEPS-noMN decreases with the increase of history length when the length comes up to 120. This shows memory networks become inefficient when history is too long. A possible reason is that attention mechanism may become insensitive when there are too many irrelevant memory slots. Another finding is that, KEPS has a greater improvement over KEPS-noMN in Fig 6b than in Fig. 6a especially on the queries with history length larger than 120. This indicates that the memory network is more important when dealing with history spanning a long period time. This may be because the length of short-term history is generally short, the role of short-term memory network can be partly made up by the LSTM layer in KEPS. We also evaluate the model on queries with

Table 3: Performance of variants of KEPS. The relative percentages are calculated based on PSGAN. $*$ indicates significant improvement over PSGAN. $\diamond$ indicates significant improvement over KEPS-noEntity,KEPS-noPSLink, KEPS-noMN and KEPS-noAdjust. ($p < 0.05$ in two-tailed paired t-test)

Model	MAP		MRR		AR		Precision					
							P@1		P@3		P@5	
PSGAN	.5480	-	.5600	-	10.2670	-	.4892	-	.5720	-	.6140	-
KEPS-noEntity	.6618*	+20.77%	.6771*	+20.91%	5.6227*	+45.24%	.5868*	+19.95%	.7239*	+26.56%	.7805*	+27.12%
KEPS-noPSLink	.6700*	+22.27%	.6842*	+22.18%	5.4799*	+46.63%	.5929*	21.20%	.7320*	+27.97%	.7889*	+28.48%
KEPS-noMN	.6547*	+19.47%	.6691*	+19.48%	5.7821*	+43.68%	.5782*	+18.19%	.7129*	+24.63%	.7719*	+25.72%
KEPS-noAdjust	.6811*	+24.29%	.6942*	+23.96%	5.2180*	+49.18%	.6020*	+23.06%	.7456*	+30.35%	.8017*	+30.57%
KEPS-QR	.6481*	+18.27%	.6609*	+18.02%	5.6743*	+44.73%	.5637*	+15.23%	.7089*	+23.93%	.7748*	+26.19%
KEPS	**.6903**$^{*\diamond}$	+25.97%	**.7044**$^{*\diamond}$	+25.79%	**5.0645**$^{*\diamond}$	+50.67%	**.6124**$^{*\diamond}$	+25.18%	**.7578**$^{*\diamond}$	+32.48%	**.8118**$^{*\diamond}$	+32.21

Table 4: Δ MAP of different personalization models on ambiguous queries and other queries

Type	Amb ΔMAP		Non-amb ΔMAP	
HRNN	0.2696	-31.64%	0.3334	-28.27%
HRNN-Entity	0.2721	-31.01%	0.3347	-27.99%
PSGAN	0.2747	-30.35%	0.3373	-27.43%
KEPS-noPSLink	0.3944	-	0.4648	-
KEPS	0.4167	+5.65%	0.4816	+3.61%

Table 5: Linking Effect of PSLink and TAGME on 100 ranomly selected ambiguous queries

Type	Ambiguous queries	test mentions	TAGME	KEPS
Num	470,104	136	79	85

Table 6: Effect of memory network on short-term history

Model	KEPS	KEPS-noMN
ΔMAP	0.4016	0.3556

Table 7: ΔMAP of KEPS-noAdjust on ambiguous queries

Type	Amb ΔMAP		Non-amb ΔMAP	
KEPS-noAdjust	0.4086	-	0.4764	-
KEPS	0.4167	+1.98%	0.4816	+1.09%

only short-term history in Tab. 6 We can see memory networks also have certain advantages in dealing with short-term history.

These results show memory networks are effective especially in preserving user's long-interval historical preference in long-term history. But a suitable history length is important.

5.6 Effect of Post Ranking Entity Linking Adjustment

Post linking adjustment mechanism adjusts the entity link probabilities of historical queries to better model user's history. Since the ranking effect of KEPS-noAdjust has been studied in Sec. 5.3, we further analyze the improvement on ambiguous queries using the adjusted historical entity information in Tab. 7. We can see the improvement of KEPS over KEPS-noAdjust is larger on ambiguous queries. This shows that with the adjusted historical entities given by post linking adjustment, KEPS can better model user's historical interest and promote the ranking effect for ambiguous queries. This confirms the effectiveness of the post linking adjustment mechanism on modeling user's history.

Table 8: Effect of different history. $\star$, $\circ$, $\bullet$ indicates significant improvement over CARS*, LKEPS$^{\circ}$ and SKEPS$^{\bullet}$. ($p < 0.05$ in two-tailed paired t-test)

Model	MAP	MRR	AR	P@1	p@3	p@5
CARS	.4658	.4786	7.5876	.3239	.5538	.6568
SKEPS	.5920*	.6059*	6.3690*	.4902*	.6689*	.7419*
LKEPS	.6077$^{\star\bullet}$	.6202$^{\star\bullet}$	6.6839$^{\star\bullet}$	.5117$^{\star\bullet}$	.6760$^{\star\bullet}$	.7439
KEPS	.6903$^{\star\bullet\circ}$	.7044$^{\star\bullet\circ}$	5.0645$^{\star\bullet\circ}$	.6124$^{\star\bullet\circ}$	.7578$^{\star\bullet\circ}$	.8118$^{\star\bullet\circ}$

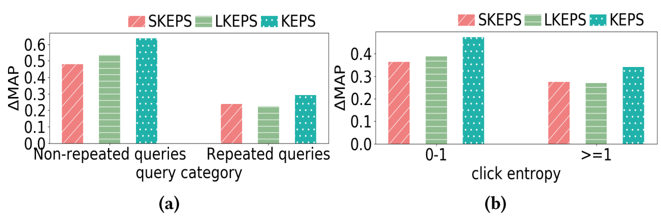


Figure 7: Effect of model using different history on different queries: (a) repeated and non-repeated queries; (b) queries with different click entropy.

5.7 Study of Long-term and Short-term History

Since KEPS takes sessions as training units and dynamically adjusts the linking probabilities for queries in short-term history, it treats the long-term and short-term history differently. So to further study the different roles of long-term and short-term history, we compare the effect of KEPS, SKEPS which is trained and tested only using short-term history and LKEPS which is trained and tested only using long-term history in Tab. 8. The model effect on different queries is shown in Fig. 7b and 7a. When only using short-term history, our task becomes similar to session-based search. So we also compare SKEPS with the state-of-art session-based search model CARS [1] to show our model effect in Tab. 8.

From Tab. 8 we can see, excluding either long-term history or short-term history, the effect of our model is obviously reduced. The decline of SKEPS is more obvious, showing that long-term history contains more personalized information. Another finding from Fig. 7b and 7a is that the improvement of LKEPS over SKEPS is mainly on repeated queries and the queries with click entropy less than 1. LKEPS even underperforms SKEPS on non-repeated queries. This shows that the contribution of long-term history mainly comes from the refinding information in user's long-interval history. The contribution of the short-term history mainly comes from the predicting of user's current search intention through the search context in the session. Further compared with CARS, SKEPS also has significant improvement, which confirms the ability of our model KEPS to model user's session search intent.

6 CONCLUSION

In this paper, we propose knowledge enhanced personalized model KEPS. KEPS first conducts personalized entity linking to model user's intent, then constructs user preference profiles using the linked entity and memory networks. After ranking we propose to use user's feedback signal to adjust the entity linking probabilities of historical queries, which helps to model user's interest for future queries. Experimental results confirmed the effectiveness of our model. In the future we plan to use Graph Neural Network to model the relation between entities to adjust linking probabilities.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China No. 2018YFC0830703, National Natural Science Foundation of China No. 61872370 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098. We thank Wasi et al. [1] for providing the data.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. *arXiv preprint arXiv:1906.02329* (2019).

[2] Krisztian Balog. 2018. *Entity-oriented search.* Springer.

[3] Paul N. Bennett, Krysta Marie Svore, and Susan T. Dumais. 2010. Classification-enhanced ranking. In *WWW.* ACM, 111–120.

[4] 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.* ACM, 185–194.

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems.* 2787–2795.

[6] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of ICML'2005.* ACM, 89–96.

[7] Fei Cai, Shangsong Liang, and Maarten de Rijke. 2014. Personalized document re-ranking based on Bayesian probabilistic matrix factorization. In *SIGIR.* 835–838.

[8] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR.* ACM, 243–250.

[9] Mark James Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards query log based personalization using topic models. In *CIKM.* ACM, 1849–1852.

[10] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Dexter: an open source framework for entity linking. In *ESAIR'13, Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval, co-located with CIKM 2013, San Francisco, CA, USA, October 28, 2013.* 17–20.

[11] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 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.* ACM, 126–134.

[12] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *SIGIR.* ACM, 365–374.

[13] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *WWW.* ACM, 581–590.

[14] Faezeh Ensan and Ebrahim Bagheri. 2017. Document Retrieval Model Through Semantic Linking. In *WSDM.* ACM, 181–190.

[15] Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management.* ACM, 1625–1628.

[16] 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.* ACM, 347–356.

[17] Morgan Harvey, Fabio Crestani, and Mark James Carman. 2013. Building user profiles from topic models for personalised search. In *CIKM.* ACM, 2309–2314.

[18] Xiujun Li, Chenlei Guo, Wei Chu, Ye-Yi Wang, and Jude Shavlik. 2014. Deep learning powered in-session contextual ranking using clickthrough data. In *NIPS'2014.*

[19] Xitong Liu and Hui Fang. 2015. Latent entity space: a novel retrieval approach for entity-bearing queries. *Inf. Retr. Journal* 18, 6 (2015), 473–503.

[20] 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.

[21] Zheng-Hao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *ACL (1).* Association for Computational Linguistics, 2395–2405.

[22] 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.* ACM, 555–564.

[23] Nicolaas Matthijs and Filip Radlinski. 2011. Personalizing web search using long term browsing history. In *WSDM.* ACM, 25–34.

[24] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* (2016).

[25] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2018. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. *CoRR* abs/1808.04122 (2018).

[26] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A Picture of Search. In *Proceedings of the 1st International Conference on Scalable Information Systems (InfoScale '06).* 1–es.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).* 1532–1543.

[28] Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document Retrieval Using Entity-Based Language Models. In *SIGIR.* ACM, 65–74.

[29] Ahu Sieg, Bamshad Mobasher, and Robin D. Burke. 2007. Web search personalization with ontological user profiles. In *CIKM.* ACM, 525–534.

[30] Yang Song, Hongning Wang, and Xiaodong He. 2014. Adapting deep ranknet for personalized search. In *WSDM'2014.* ACM, 83–92.

[31] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. 2008. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR.* ACM, 163–170.

[32] Jaime Teevan, Daniel J. Liebling, and Gayathri Ravichandran Geetha. 2011. Understanding and predicting personal navigation. In *WSDM.* ACM, 85–94.

[33] Maksims Volkovs. 2015. Context Models For Web Search Personalization. *CoRR* abs/1502.00527 (2015).

[34] Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search Personalization with Embeddings. In *ECIR (Lecture Notes in Computer Science),* Vol. 10193. 598–604.

[35] Thanh Tien Vu, Alistair Willis, Son Ngoc Tran, and Dawei Song. 2015. Temporal Latent Topic User Profiles for Search Personalisation. In *ECIR (Lecture Notes in Computer Science),* Vol. 9022. 605–616.

[36] Xiaojie Wang, Zhicheng Dou, Tetsuya Sakai, and Ji-Rong Wen. 2016. Evaluating search result diversity using intent hierarchies. In *SIGIR.* 415–424.

[37] Xiaojie Wang, Ji-Rong Wen, Zhicheng Dou, Tetsuya Sakai, and Rui Zhang. 2017. Search result diversity evaluation based on intent hierarchies. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (2017), 156–169.

[38] Ryen W. White, Wei Chu, Ahmed Hassan Awadallah, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *WWW.* ACM, 1411–1420.

[39] Qiang Wu, Chris JC Burges, Krysta M Svore, and Jianfeng Gao. 2008. *Ranking, boosting, and model adaptation.* Technical Report. Technical report, Microsoft Research.

[40] Chenyan Xiong and Jamie Callan. 2015. EsdRank: Connecting Query and Documents through External Semi-Structured Data. In *CIKM.* ACM, 951–960.

[41] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-Entities Representation for Ranking. In *ICTIR.* ACM, 181–184.

[42] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-Entity Duet Representations for Document Ranking. In *SIGIR.* ACM, 763–772.

[43] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard H. Hovy. 2017. JointSem: Combining Query Entity Linking and Entity based Document Ranking. In *CIKM.* ACM, 2391–2394.