Improving Personalized Search with Dual-Feedback Network

Chenlong Deng[2], Yujia Zhou[2,3], and Zhicheng Dou[1]*
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
{dengchenlong,*dou}@ruc.edu.cn

ABSTRACT

Personalized search improves the quality of search results by modeling historical user behavior. In recent years, many methods based on deep learning have greatly improved the performance of personalized search. However, most of the existing methods only focus on modeling positive user behavior signals, which leads to incomplete user interest modeling. At the same time, the user's search behavior hides much explicit or implicit feedback information. For example, clicking and staying for a certain period represents implicit positive feedback, and skipping behavior represents implicit negative feedback. Intuitively, this information can be utilized to construct a more complete and accurate user profile. In this paper, we propose a dual-feedback modeling framework, which integrates multi-granular user feedback information to model the user's current search intention. Specifically, we propose a feedback extraction network to refine the dual-feedback representation in multiple stages. For enhancing the user's real-time search quality, we design an additional dual-feedback feature gating module to capture the user's real-time feedback in the current session. We conducted a large number of experiments on two real-world datasets, and the experimental results show that our method can effectively improve the performance of personalized search.

CCS CONCEPTS

• **Information systems → Personalization**.

KEYWORDS

Personalized search; Dual-feedback network; User intention

ACM Reference Format:
Chenlong Deng, Yujia Zhou, Zhicheng Dou. 2022. Improving Personalized Search with Dual-Feedback Network. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498447

1 INTRODUCTION

The search engine has provided great assistance for our daily information acquisition needs. With the rapid growth of information on the Internet in recent decades, excellent search engines have played a more important role in alleviating information overload. However, with the simultaneous increase in the number of users, returning the same document list for the same query can no longer meet users' needs. The user's query intention is often affected by many factors, such as the user's preferences and his short-term interaction with the search results. Personalized search is considered to be one of the promising ways to solve these problems [9, 23]. It carries out special modeling of user interests, which provides personalized document rankings for different users.

In the past, traditional personalized search methods [3, 7, 24, 28–30] focused on extracting different aspects of personalized signals from query logs. Although most of their signals are manually designed, these researches also show that personalization can effectively improve the retrieval capabilities of search engines. In recent years, with the research on deep learning becoming popular again, many personalization methods [10, 17, 39, 41–45] based on deep learning have significantly improved the search accuracy to a new level.

Most of the existing works based on deep learning rely on the user's click history to model the user's interest or eliminate the ambiguity of the query [10, 17, 42]. However, click behavior is only a small part of the entire search interaction, and only using this information will lead to positive biased user profile. For example, in a query that the user has never issued, each candidate document matches the user's positive interest to the same degree. In this case, a method that only relies on positive interest modeling may make a blind ranking decision. Therefore, it is not enough to only rely on click behavior history to model the user's search intention.

Fortunately, in addition to the click behavior, there are many other interactive behaviors in the search log that can provide feedback. Behaviors such as skip and quick close contain a lot of information [15] that can help estimate user preferences, and can be utilized to create a more complete and accurate user profile. However, studies have shown that user behavior is often related to the origin display order of documents, which cause implicit feedback to contain a lot of noise [15]. For example, when the first item in the document list can meet all the user's needs, even if the following documents also meet the user's needs, the user may not click on the following documents after reading the first document. Therefore, these implicit feedback cannot be utilized directly, and it is important to design suitable module structures to capture different types of feedback.

Furthermore, previous researches have shown that long-term history and short-term history have different impact properties [3, 16, 32]. The search history in the current session (short-term history) can reflect the current query intention and information that the user has already obtained. For example, when the user has

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498447

searched for “The Matrix I” and “The Matrix II: Reloaded”, if the user searches for “The Matrix’s Sequel” at this time, it is probably that the user’s real search intention is “The Matrix III: Revolutions”. Therefore, in order to utilize short-term feedback, it is necessary to perform sequential and fine-grained modeling of user interaction behaviors.

In this paper, we propose a **P**ersonalized **S**earch model with **D**ual-**F**eedback **N**etwork (PSDFN). This model extracts different levels of dual-feedback from the query history to construct a more comprehensive search intention. It consists of two sub-modules: a long-term historical dual-feedback extraction module and a real-time dual-feedback extraction module. The former module digs out feedback signals from users’ long-term search logs, and then converts these signals into dual-feedback representation in multiple stages. The latter focuses on the details of the search interaction of the current session, trying to capture fine-grained topic information and sequence relationships from short-term history. Finally, with the mutual complementary information of the two sub-modules, we can obtain the dual intention representation of the current query, and utilize this representation to calculate the relevance of candidate documents to personalize the search results.

The main contributions of this paper are summarized as follows:

(1) We propose a feedback modeling framework, which models users’ positive/negative behavior and uses dual-feedback to improve the performance of personalized search.

(2) We design a hierarchical feedback extraction model, which can extract feedback information from levels such as query level and long-term historical level.

(3) We specially design the feedback feature gating module for real-time feedback, which can capture more fine-grained sequential information and assist the model to obtain more accurate user intentions.

2 RELATED WORK

2.1 Personalized Search

Personalized search improves the search quality of search engines by virtue of its characteristics that can meet the personalized needs of different users [4]. An important way to achieve personalization is to dig out available information from the user’s search history and behaviors, and then to model user preferences. Traditional research methods mostly use the method of extracting features from click data and document topics. This kind of method was widely studied at that time because of its high efficiency and stability. Dou et al. [9] proposed P-click and G-Click. These two methods are used to personalize the ranking of candidate documents according to individual-level and group-level click behaviors. In addition, there are some researchers who tried to focus on the topic features of the document. The Open Directory project (ODP) provides a large amount of classified web information, which can be used as an explicit topic feature of the document [2, 22, 33]. However, manual editing has brought huge labor costs and incompleteness of document categories. Research based on the latent topic space alleviates these problems to a certain extent [5, 13, 27, 28]. These methods [3, 26] are dedicated to learning implicit vector representations and using learning to rank methods to train models.

In recent years, deep learning has provided powerful representation learning capabilities for personalized search, which has been leveraged by researchers to improve the capabilities of models in many aspects. Some researches [10, 18] are devoted to modeling the potential interests of users, which leads to better user profiles and more precise personalization. However, the previous methods didn’t explore the user’s negative feedback, which makes the constructed user interest incomplete. Rather than just focusing on positive feedback, our framework focuses on integrating multiple user feedbacks to obtain comprehensive user interests.

2.2 Modelling Negative Feedback for Information Retrieval

Negative feedback is a special kind of feedback signal, which can solve some problems that positive feedback can’t solve due to different concerns about information. In the field of information retrieval, some frameworks are used to capture negative feedback signals. Wang et al. [31] modeled the unclicked documents under difficult queries, so that the unshown documents are re-ranked based on negative feedback. Hao et al. [12] proposed a method to automatically identify the type of document feedback, and used the Expectation-maximization algorithm to estimate the positive feedback and negative feedback based on the document type. Zhang and Wang [40] classified the preliminary screening documents according to their relevance, and then conducted a second retrieval based on the feedback. Recently, some frameworks have tried to apply negative feedback to recommendation systems. Xie et al. [36] used explicit positive and negative feedback to extract implicit feedback signals. Wu et al. [35] established a unified framework that jointly considered the user’s behavior sequence and the influence of multiple feedback signals. In this paper, we attempt to jointly utilize the positive and negative feedback in the search process to improve the user’s query intention representation.

3 PSDFN: OUR PROPOSED MODEL

Personalized search has shown great potential in improving the quality of search results. As we stated in Section 1, personalized search uses the user’s past search history to understand the query intention more accurately. Many queries have multiple intents in the text itself, one of which will be the user’s true search intention, which we call "positive intention". Correspondingly, the other intents of the query are called "negative intention". However, most of the existing personalized search models only use the click behavior to construct the user’s positive preference, and less consider how to extract additional information from the user’s negative behaviors. These shortcomings make these methods weak in the construction of the user’s complete profile. In this paper, we suggest learning different types of search intention representations of users under specific queries to model users’ complete search preferences. Specifically, we propose a hierarchical dual-feedback extraction model, which extracts dual feedback from long-term history and short-term history in multiple stages, thereby enhancing the comprehensiveness of current query understanding.

To start with, the problem can be formulated as follows. Suppose that for a user U, his historical search data is $H_u = \{S_1, S_2, ..., S_N\}$, where each element represents a separate session and S_N represents the current session. Each session contains a list of chronological queries and each query has its corresponding candidate document list, denoted as $S_k = \{\{q_k^1, d_{k,1}^1, d_{k,2}^1\}, ..., \{q_k^{n^k}, d_{k,1}^{n^k}, ...\}\}$ and n^k is the number of queries in the session S_k. Given a new query at this time and candidate documents $D = \{d_1, d_2, ...\}$ are returned by the search engine. Our task is to compute the relevance score of each candidate document based on the "query-document" relevance and the user's preference. The final probability that the document d will be clicked by user U is denoted as $p(d|q, U)$:

$$p(d|q, U) = \phi\left(p(d, q), p(d|q, U^+), p(d|q, U^-)\right),$$

where $p(d|q, U)$ consists of three parts: the first part represents the origin semantic relevance between the query and the candidate document. The second part represents the relevance between the user's positive intention and the candidate document. The third part represents the relevance between the user's negative intention and the candidate document. Meanwhile, ϕ represents an MLP (Multi-Layer Perceptron) with $ReLU$ and $tanh$ as the activation function, which is used to fuse these features to obtain the final predicting result. The whole architecture of our model is illustrated in Figure 1. Next, we will introduce the details of the three parts: (1) Long-term history dual-feedback (2) Real-time dual-feedback (3) Re-ranking.

3.1 Long-term History Dual-Feedback

As we stated in Section 1, previous researches mostly focus on the building of the user's positive behaviors, while neglecting other explicit or implicit negative feedback. In order to capture a more complete user search intention, we designed a multi-type feedback extraction model. The model simulates the user's past decision-making state, which assists in extracting the user's real positive intentions and negative intentions at the time. Specifically, we divide the process into four parts: (1) Query/Document Encoding (2) Sequence Interaction (3) Hierarchical Dual-Feedback Extraction (4) Intention Prediction.

3.1.1 Query/Document Encoding. For understanding the semantic information of queries and documents, we can explore two aspects: context and user behavior. Firstly, for each document, we calculate three parts of its embedding. The first part is word embedding, we obtain the pre-trained word vector from word2vec, and then dynamically update the word vector during training. The second part is position embedding, which is used to capture the sequential logic of contextual semantics. The third part is type embedding, which is designed to distinguish the behavior on different documents. Documents that are clicked and stayed for more than 30 seconds are regarded as satisfactory types. We take the last satisfactory document in the list of displayed documents as the pivot, the unclicked document above it is regarded as the skip type, and the unclicked document below it is regarded as the ignored type. Then, we map all document types to the same dimension as the document representation, thereby obtaining the type embedding. Thus, for each document $D = \{w_1, w_2, ..., w_m\}$, the encoding process of each document can be denoted as:

$$E_D = Avg(\mathrm{Trm}(E_D^w + E_D^p)) + E_D^t, \tag{1}$$

where $E_D^w \in \mathbb{R}^{m*d}$ is word embedding for document D, $E_D^p \in \mathbb{R}^{m*d}$ is its position embedding and $E_D^t \in \mathbb{R}^d$ is type embedding. And $\mathrm{Trm}(\cdot)$ is a Transformer encoder layer [25], which is to integrate the information from word embedding and position embedding. $Avg(\cdot)$ is an average function, which calculates the average representation of a series of vectors and regards it as the semantic presentation of this document. Next, we add the type embedding to the semantic representation, which generates the final presentation of the document. The query encoding process is similar to the above process, but eliminates the type embedding to maintain the independence of query. Finally, we add the query encoding to its candidate documents' encoding, which aims to model the search intention represented by documents under a specific query. So far, we obtain unified encoding of queries and documents for later extraction of user feedback information.

3.1.2 Sequence Interaction. As we have stated in section 1, the interactive behaviors in historical user data contains more comprehensive preference information. The process of a user using a search engine is often an interaction in the unit of "Query-Displayed document list". The interaction process between the user and displayed documents is very complicated, especially for unclicked documents that are full of noise. For example, if a user clicks on the first document but skips the second document, there are two possibilities: (1) One is simply that the second document does not meet the user's needs (2) The other one is that the first document is similar to the second document, and the user has already obtained the required information from the first document, so give up browsing the second document. Therefore, we need to model the real positive and negative preferences of users in the unit of "Query-Displayed document list".

Formally, for each query q_i in history, it has a candidate document list $D_i = \{D_{i,1}, D_{i,2}, ..., D_{i,m}\}$. We take the final representation of documents in section 3.1.1, denoted as $I_i = \{I_{i,1}, I_{i,2}, ..., I_{i,m}\}$. For capturing potential information from different types of documents in the displayed list, we design a heterogeneous transformer to further eliminate the noise of presentation:

$$I_i^w = \mathrm{Trm}^h(I_i + I_i^p), \tag{2}$$

where Trm^h is a heterogeneous transformer and I_i^p is position embedding for perceiving the order relation among different feedback.

3.1.3 Dual-Feedback Extraction. Next, we need to extract accurate feedback from these refined behaviors. First, we use the click action in the behavior sequence as positive feedback. Specifically, we take the average vector of clicked documents as the positive intention representation $F_i^{p'}$ for the i-th query. At the same time, the user's search intention is not only related to the query itself, but also closely related to the search process within the session. Therefore, we construct a session-level RNN, which calculates a series of revised positive intention representation based on historical search intention within the session. Formally, for the n-th query of the m-th session, the revised positive intention $F_{m,n}^p$ can be represented as:

$$F_{m,n}^p = \mathrm{RNNCell}(h_{m,n-1}, F_{m,n}^{p'}), \tag{3}$$

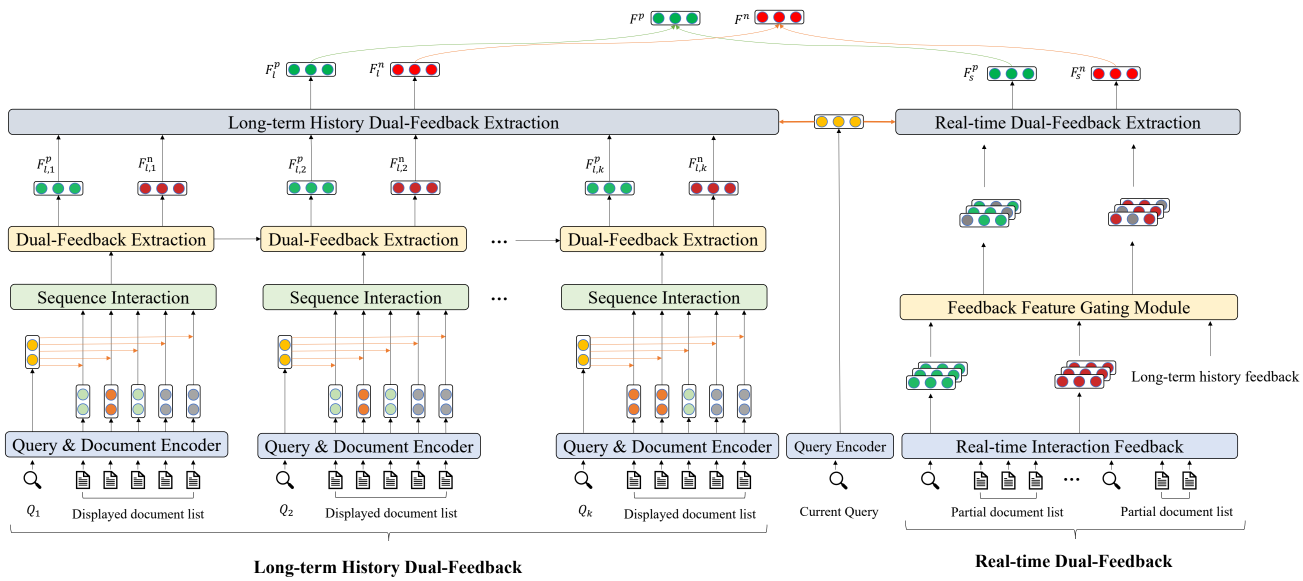


Figure 1: The architecture of our model PSDFN. Historical queries and document information are encoded in vector form, and then sequence interaction and dual-feedback extraction are performed in the query unit. Next, we generate historical-level feedback from the long-term dual-feedback based on the current query, and capture fine-grained topic information in the short-term feedback. Finally, the long-term and short-term feedback are fused and matched with the candidate documents, and the final personalized score is calculated with the assistance of additional features.

where $h_{m,n-1}$ is the latent vector at the previous point in time and $h_{m,0}$ is implemented as zero vector. The subscripts of $F^{p'}_{m,n}$ and $F^{p'}_{i}$ have a one-to-one correspondence according to the search time. RNNCell($\cdot$) can be vanilla RNN, LSTM [14], GRU [6] or other RNN variants [11]. For better blancing operating efficency and training time, we implement it as GRU in our model:

$$
\begin{aligned}
r_{m,n} &= \sigma(W_r \cdot F^{p'}_{m,n} + V_r \cdot h_{m,n-1} + b_r),\\
z_{m,n} &= \sigma(W_z \cdot F^{p'}_{m,n} + V_z \cdot h_{m,n-1} + b_z),\\
c_{m,n} &= \tanh(W_c \cdot F^{p'}_{m,n} + r_{m,n} \odot (V_c \cdot h_{m,n-1} + b_{cv}) + b_c),\\
h_{m,n} &= (1 - z_{m,n}) \odot h_{m,n-1} + z_{m,n} \odot c_{m,n},
\end{aligned}
\tag{4}
$$

where reset gate $r_{m,n}$ and update gate $z_{m,n}$ control information from previous states and current state, σ represents the sigmoid function, $W_r, V_r, W_z, V_z, W_c, V_c, b_{cv}, b_c$ are learnable and shared parameters updated with the training process. We take the hidden state output at each time point as the revised positive intention $F^{p}_{m,n}$, and then convert it to F^{p}_{i} corresponding to the i-th query through the one-to-one mapping relationship of the sequence.

Compared with clearer positive intentions, the negative intentions of users are more sparsely distributed. In other words, under a certain query, in addition to the user's real positive intention, the other intentions are all negative intentions of varying degrees. Therefore, our purpose is not to construct a negative user profile, but to construct the feedback that is most dissimilar to the user's search intention under a certain query as an "anchor". Inspired by self-supervised learning, we regard that the unknown negative intentions of the user can be obtained from the known positive intention. Thus, we design a negative signals extraction module to capture the negative feedback. Formally, for the non-positive feedback behavior subsequence $\{I_{i,i1}, I_{i,i2}, ..., I_{i,im}\}$ of the i-th query:

$$
\begin{aligned}
\alpha'_{i,k} &= 1 - \frac{\exp(I_{i,k} \cdot F^{p}_{i})}{\sum_{j=1}^{\mathrm{im}} \exp(I_{i,j} \cdot F^{p}_{i})},\\
\alpha_{i,k} &= \frac{\exp(\alpha'_{i,k})}{\sum_{j=1}^{\mathrm{im}} \exp(\alpha'_{i,j})},\\
F^{n}_{l,i} &= \sum_{k=1}^{\mathrm{im}} \alpha_{i,k} I_{i,k},
\end{aligned}
\tag{5}
$$

where $\alpha'_{i,k}$ is the weight of the k-th negative term, and $\alpha_{i,k}$ is the weight after applying softmax function to $a_{i,k}$. Finally, we obtain the negative feedback of the i-th query.

3.1.4 Intention Prediction. After getting the dual-feedback under each query in the long-term history, we use the feedback information to predict the real intention of the user under the current query. In order to capture different types of user intentions, we use two homogenous transformers to predict the user's real positive and negative intentions under the current query. Formally, given the historical feedback $F^{p}_{l} = \{F^{p}_{l,1}, F^{p}_{l,2}, ..., F^{P}_{l,h}\}$, $F^{n}_{l} = \{F^{n}_{l,1}, F^{n}_{l,2}, ..., F^{n}_{l,h}\}$, the process can be formulated as follows:

$$q_l^p = \mathrm{Trm}^l([F_l^p, q^c]),$$
$$q_l^n = \mathrm{Trm}^l([F_l^n, q^c]), \quad (6)$$

where q^c means the current query encoding output in section 3.1.1, Trm^l represents taking the output in the last position after transformer encoding.

3.2 Real-time Dual-Feedback

The long-term module proposed above can perceive the positive feedback and negative feedback of similar queries in the past, but does not pay attention to the interactive feedback between the user and search results in the current session. Just like "the Matrix" example we gave in section 1, the interactive information contained in the current session sometimes dominates the user's search intention more than the long-term history. Therefore, we aim to establish a fine-grained and short-term feedback dominant model to increase focus on recent interactions.

Given the short-term search interaction history in the current session, we intent to capture the sequential features of changes in content requirements. Firstly, with the assistance of Dual-Feedback extraction module stated in section 3.1, we obtain the positive feedback $F_s^p = \{F_{s,1}^p, F_{s,2}^p, ..., F_{s,\mathrm{sn}}^p\}$ and negative feedback $F_s^n = \{F_{s,1}^n, F_{s,2}^n, ..., F_{s,\mathrm{sn}}^n\}$ in the current session. Then, we take the positive and negative intentions obtained by the long-term module as the user's long-term preferences, and apply feature extraction operation to the short-term feedback information. The feature extraction operation can be formulated as:

$$F_s^{pf} = F_s^p \otimes (F_s^p W_1^p + q_l^p W_2^p + b_p),$$
$$F_s^{nf} = F_s^n \otimes (F_s^n W_1^n + q_l^n W_2^n + b_n), \quad (7)$$

where $F_s^p, F_s^n \in \mathbb{R}^{\mathrm{sn}*d}$ are short-term feedback, $\otimes$ represents element-wise vector multiplication, $W_1^p, W_2^p, W_1^n, W_2^n \in \mathbb{R}^{d*d}$ are learnable parameters and $b_p, b_n \in \mathbb{R}^d$ are learnable bias vectors.

Now that we have extracted fine-grained features from the feedback information to enhance the representation of short-term interactive information, we then try to use short-term feedback to predict the real intention of the current query.

$$q_s^p = \mathrm{Trm}^l([F_s^{pf}, q^c]).$$
$$q_s^n = \mathrm{Trm}^l([F_s^{nf}, q^c]). \quad (8)$$

At this point, we have obtained the query intention after a short-term enhancement of dual-feedback. Next, we will use long-term and short-term enhanced query intention to re-rank the list of candidate documents.

3.3 Re-ranking

Finally, we can calculate the final score of each candidate document through the multiple user query intention representation obtained above. For the matching of candidate documents, we follow HTPS model proposed by [42] to calculate the context-aware representation of the document, denoted as d^w. Next, we will introduce the calculation methods of each part.

For personalized matching relevance, we collected long-term and short-term dual-feedback enhanced query expressions, including: (1) long-term positive intention enhanced query q_l^p (2) short-term positive intention enhanced query q_s^p (3) long-term negative intention enhanced query q_l^n (4) short-term negative intention enhanced query q_s^n. We have:

$$p(d|q, U^+) = \phi(\mathrm{Sim}(q_l^p, d^w), \mathrm{Sim}(q_s^p, d^w)),$$
$$p(d|q, U^-) = \phi(\mathrm{Sim}(q_l^n, d^w), \mathrm{Sim}(q_s^n, d^w)),$$

where Sim is the similarity between the refined query vector and document vector, which is implemented as cosine similarity in our experiments.

For ad-hoc matching relevance, we divide it into three parts to calculate: In the first part, we follow [42] and extract some clicks and topic features $\mathcal{F}$ in candidate documents. In the second part, we consider the matching relationship between the original query and the document. In the third part, we follow the K-NRM model proposed by [37], which applies k kernels to capture matching information of different ranges:

$$p(d|q) = \phi\left(\phi(\mathcal{F}), \mathrm{Sim}(q^c, d^w), \mathrm{Sim}^I(q, d)\right),$$

where $\mathrm{Sim}^I(q, d)$ represents calculating matching relevance based on interactive methods K-NRM. Finally, we calculate the score of each candidate document, and re-rank the list of candidate documents according to the score.

In the training and optimization part, we chose the neural ranking algorithm LambdaRank to train the entire model, which is based on the pairwise method. Given two documents doc_i and doc_j, the probability that doc_i is more relevant than doc_j can be computed as follows:

$$P_{ij} = \frac{1}{1 + e^{-\sigma(s_i - s_j)}},$$

where σ is learnable parameter, s_i and s_j are relevance scores of doc_i and doc_j respectively. Next, we compute the real probability $\bar{P}_{ij}$ of each document:

$$\bar{P}_{ij} = \frac{1}{2}(1 + S_{ij}),$$

where S_{ij} is the real label of this pair of documents. If doc_i is more relevant than doc_j then $S_{ij} = +1$, if the correlation between doc_i and doc_j is equal then $S_{ij} = 0$, otherwise $S_{ij} = -1$. Finally, we use cross entropy to define the loss function:

$$\mathcal{L} = -\bar{P}_{ij} \log P_{ij} - \left(1 - \bar{P}_{ij}\right) \log\left(1 - P_{ij}\right).$$

4 EXPERIMENTAL SETUP

4.1 Dataset

We conducted experiments on the AOL search log dataset [19] and a commercial search log from a real search engine (written as "Commercial dataset" in the following). The detailed statistics of the datasets are shown in Table 1:

AOL Dataset is a dataset that contains a large amount of real user search history. It includes three months of query and click data from March 1, 2006 to May 31, 2006. Since there is only user click

Table 1: Basic statistics of the datasets.

Dataset	AOL	Commercial
# Days	91	58
# Users	110,439	33,204
# Queries	736,454	267,479
# Sessions	279,930	97,858
Average query length	2.87	3.25
Average #click per query	1.11	1.19

data in the original data, we follow [1] to use the BM25 algorithm to recall the top candidate documents. We also split logs into sessions, keeping the data preprocessing consistent with the previous work [42]. In order to make all the samples have sufficient user history, we divided the dataset by time. Specifically, we use the data of the first five weeks as user history, and the data of the next eight weeks as experimental data. Then, we divide the experimental data into the training set, validation set and test set at a ratio of 6:1:1.

Commercial Dataset contains two-month-long search logs from 2013. The commercial search engine did not use personalization technology at the time, so it has the potential to improve search quality through personalization. We use the search records of the first six weeks as historical data, and the last two weeks as experimental data. We follow [34] and regard 30 minutes of inactivity as a signal to divide the session. Since this dataset contains dwell time, we consider the click that stays for 30 seconds or the last click in the session as a satisfactory click.

4.2 Baselines

In order to better understand the performance of our model, we select some baselines for comparison. In terms of baseline types, we choose some ad-hoc models and previous personalized search methods:

Ori. [21] For the AOL dataset, since the original dataset only provides click data. We followed [1] to reconstruct the search results based on BM25 algorithm. For the commercial dataset, we directly use the original data because the initial displayed document list information has been given.

KNRM [37] It is a kernel-based neural ranking model that uses kernels to obtain soft-TF signals. The interactive matching features between query and document are extracted for ranking.

Conv-KNRM [8] It is the improved version of KNRM. Conv-KNRM adds n-gram convolution and model levels on the original basis, which allows it to capture more subtle semantic entities.

BERT [20] This model transforms from the pre-trained BERT model, which allows itself to predict the similarity score between the query and the document. The parameters of the entire model will be dynamically updated during the training.

HRNN [10] This model designs a hierarchical RNN and attention structure, which is to capture user interest in the long-term and short-term search history. Dynamically updated user interests help search engines to achieve personalization.

PSGAN [17]This model introduces the generative adversarial network into personalized search. The generator continuously learns to generate higher-quality negative examples during training, and the discriminator continuously strengthens to distinguish the subtle differences between the satisfying click document and other documents. Finally, this model uses the trained discriminator to re-rank candidate documents.

HTPS [42] This model uses transformer to disambiguate the current query in conjunction with contextual semantics. Meanwhile, in order to capture the user's query habit, it also designs a personalized language model to better predict the user's query intention.

PEPS [38] This model trains a personalized word embedding for each user, and improves the overall search quality through better data representation.

PSDFN (**P**ersonalized **S**earch with **D**ual- **F**eedback **N**etwork) This is the whole model proposed in Section 3

4.3 Evaluation Metrics

For evaluating the performance of the model, we select three evaluation metrics: mean average precise (MAP), mean reciprocal rank (MRR) and precision@1 (P@1). Although the above evaluation metrics are widely recognized, in some cases they are still flawed in some ways. For example, although some documents are relevant, they are still ignored due to the original low ranking position [15], which causes such documents to be marked as irrelevant. We consider using P-improve [17] as the fourth evaluation indicator, which can better evaluate the preference for reliability correlation. It is a remarkable fact that since the unclicked documents in the AOL dataset were subsequently recalled through BM25, it is not suitable to apply P-improve on the AOL dataset. Therefore, we only use the P-improve for evaluation on the commercial dataset.

5 RESULTS AND ANALYSIS

5.1 Overall Performance Comparison

The performance results of different models on the two datasets are shown in Table 2. It can be observed that:

(1) Our method vs. baselines of personalized search. **Our method outperforms all previous personalization models on both datasets**. Compared with the best baseline model PEPS and HTPS, our model has a significant improvement in all evaluation metrics with paired t-test at p <0.05 level. Specifically, compared to the best baseline PEPS on the AOL dataset, we have increased the MAP by 1.15%, and outperforms HTPS by 0.6% on the commercial dataset. The significant performance improvement on the two datasets shows that extracting dual-feedback information can improve the search quality.

(2) Personalized search vs. Ad-hoc search. Experimental results show that the personalized search model based on deep learning is better than all Ad-hoc search baseline models. The Ad-hoc search model ranks the candidate documents according to the relevance of the query and the document, while the personalized search also considers the user's preferences. The performance advantage of personalized search shows that the retrieval model of one-size-fits-all is difficult to meet the retrieval needs of different users. In addition, we observe that personalized search models make the most significant progress on P@1, which may be because the user's re-finding behavior is appropriately modeled.

Table 2: The results of all models on two datasets. The percentage is based on the SOTA baseline. '†' indicates the model outperforms all baselines significantly with paired t-test at p < 0.05 level. Best results are denoted in bold.

Model	AOL dataset						Commercial dataset							
	MAP		MRR		P@1		MAP		MRR		P@1		P-improve	
Ad-hoc search baselines														
Ori.	.2504	-64.9%	.2596	-64.2%	.1534	-75.6%	.7399	-10.2%	.7506	-10.0%	.6162	-15.6%	-	-
KNRM	.4291	-39.8%	.4391	-39.5%	.2704	-56.9%	.4916	-40.3%	.5001	-40.1%	.2849	-61.0%	.0655	-75.3%
Conv-KNRM	.4738	-33.5%	.4849	-33.2%	.3266	-48.6%	.5872	-28.7%	.5977	-28.4%	.4188	-42.7%	.1422	-46.5%
BERT	.5033	-29.4%	.5135	-29.3%	.3552	-43.4%	.6232	-24.4%	.6326	-24.2%	.4475	-38.7%	.1778	-33.1%
Personalized search baselines														
HRNN	.5423	-23.9%	.5545	-23.6%	.4854	-22.7%	.8065	-2.1%	.8191	-1.8%	.7127	-2.4%	.2404	-9.5%
PSGAN	.5480	-23.1%	.5601	-22.8%	.4892	-22.1%	.8135	-1.3%	.8234	-1.3%	.7174	-1.8%	.2489	-6.3%
HTPS	.7091	-0.5%	.7251	-0.1%	.6268	-0.1%	.8224	-	.8324	-	.7286	-	.2552	-
PEPS	.7127	-	.7258	-	.6279	-	.8221	-0.1%	.8321	-0.1%	.7251	-0.4%	.2545	-0.3%
Our method														
PSDFN	**.7242**†	+1.6%	**.7358**†	+1.4%	**.6403**†	+2.0%	**.8273**†	+0.6%	**.8374**†	+0.6%	**.7326**†	+0.5%	**.2688**†	+5.3%

(3) Single feedback vs. Dual feedback. Both PSGAN and HRNN regard the recurrent neural network as one of the important components, but HRNN only considers the positive feedback information of clicks, while PSGAN utilizes negative behavior feedback on this basis. Similarly, both HTPS and PSDFN adopt the transformer module, but PSDFN emphasizes the role of negative behavior feedback. The results show that **the models that take the negative feedback into account in the two comparison groups achieve better performance improvement**.

In summary, the experimental results prove that **personalized search with dual feedback network** is conducive to accurately capturing the current search intention and enhancing the personalization of search results. To analyze the model in more detail, we conducted the following supplementary experiment: ablation studies, effect of long-term and real-time feedback, performance of different query sets.

5.2 Ablation Analysis

To prove the effectiveness of the modules in our method, we conducted ablation studies on some important components in the model. The experimental settings on AOL dataset are as follows:

PSDFN w/o. SI. We remove the Sequence Interaction component in the long-term module.

PSDFN w/o. PFIS. We disgard the positive feedback interaction within each session in the long-term history dual-feedback module.

PSDFN w/o. NF. We remove the negative feedback and only keep the positive feedback.

The results of ablation experiments are shown in Table 3. First of all, the performance of all ablation experiments is not as good as the overall model, which proves that each component in the model makes a positive contribution. Specifically, among the three experimental settings, the performance degradation of the experiment without negative feedback is the most obvious. This strongly proves that the combination of negative feedback and positive feedback can effectively improve search quality. In addition, the PFIS component also improves performance to a certain extent, indicating that this component captures the potential relevance of feedback within the session. The Sequence Interaction component also plays a role in improving performance, which is consistent with the conclusion of the previous study. That is, the user's click decision is related to the order in which the list documents are displayed.

Table 3: Performance of ablation studies on different modules of our model (on AOL dataset)

Model	MAP		MRR		P@1	
w/o. SI	.7179	-0.9%	.7299	-0.9%	.6321	-1.3%
w/o. PFIS.	.7121	-1.7%	.7245	-1.6%	.6267	-2.2%
w/o. NF.	.7047	-2.8%	.7172	-2.6%	.6181	-3.6%
PSDFN	.7242	-	.7358	-	.6403	-

5.3 Long-term and Real-time Feedback

In the ablation study, we studied the influence of several main components on the model. For exploring the different contributions of long-term feedback and real-time feedback, we separately reserve one of the two modules for the experiment. In the experimental setting of long-term feedback, we removed the real-time feedback module and only used the dual-feedback output of long-term historical feedback. In the experimental setting of real-time feedback, we only use long-term interests as user interests for feature extraction, and finally only utilize the dual-feedback from the real-time feedback module.

The performance results of the experiment are shown in Table 4: We can observe that removing any module hurts the performance of the model, but the removal of long-term feedback has a greater impact on the model than real-time feedback. We believe this is because the long-term feedback contains a richer search history, and captures more accurate user preferences to get a higher benchmark performance. Real-time feedback focuses on modeling the impact of real-time interaction on query intention, and is better at mining user behavior patterns from the current conversation history. Therefore, these two modules are not a substitute for each other, but complement each other in their respective areas of expertise. Experimental results also show that adding real-time interaction

Table 4: Performance of Long-term Feedback and Real-time Feedback (on Commercial dataset)

Model	MAP		MRR		P@1	
PSDFN-L	.8263	-0.1%	.8375	-0.1%	.7321	-0.1%
PSDFN-S	.8240	-0.4%	.8348	-0.3%	.7313	-0.2%
PSDFN	.8273	-	.8374	-	.7326	-

on the basis of long-term feedback can improve the overall quality of search.

5.4 Performance of Different Query sets

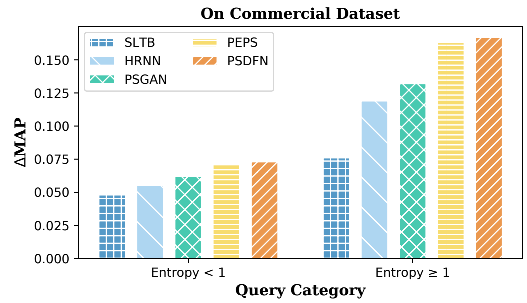
To explore the specific scenarios where our model has made progress, we divide the test set into different subsets based on conditions. By comparing the performance of the model under different query test sets, we can infer specific scenarios where the model has made progress. The details are as follows.

5.4.1 Performance on ambiguous and unambiguous queries. Different users may have inconsistent intentions for the same query. We can use click entropy to describe the degree of inconsistency. Generally speaking, the higher the degree of inconsistency, the higher the click entropy and the greater the potential for personalization. Previous work has pointed out that sometimes personalizing queries with low click entropy can hurt search quality. Therefore, we divide the query test set into two subsets according to the click entropy: the query click entropy in one set is less than 1.0, and the other is greater than or equal to 1.0. We choose SLTB, HRNN, PSGAN, PEPS and our model PSDFN to test on these two subsets, and the performance result is plotted.

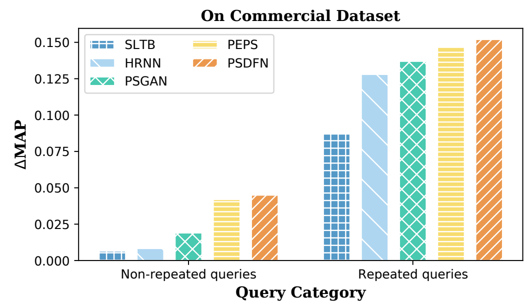
The performance result is shown in Figure 2 (a). The delta MAP of the ordinate refers to the improvement of the MAP relative to the original sort. In general, all personalization methods improve the performance both on ambiguous queries (click entropy ≥ 1) and unambiguous queries (click entropy < 1) compared to the original ranking. This shows that proper personalization is not only effective for ambiguous queries, but can also improve the performance for unambiguous queries. Comparing all personalization methods, our model performs significantly better than other methods on both sets of data. These results prove that our model significantly improve the personalized quality of search results.

5.4.2 Performance on repeated and non-repeated queries. Log statistical analysis shows that a major feature of user searches is to initiate repeated queries. Previous studies have shown that special structures can be designed to enhance repetitive search behavior, thereby improving the quality of personalization of search results. At the same time, non-repeated queries cannot directly use the same search information in the past, which has greater difficulty in retrieval. Therefore, we can divide the test set according to whether the query is repeated.

The performance results are shown in Figure 2 (b): On the repeated search subset, all the personalization methods significantly improve the search quality, and our model surpasses all baseline models. In the non-repeated search subset, the improvement of all models is significantly lower than that of the repeated query subset. This shows that non-repetitive queries are more difficult, because the model must capture content information in the history, but not



(a) Ambiguous / Unambiguous Queries



(b) Repeated / Non-repeated Queries

Figure 2: The performance on different query sets.

just the results of historical queries. The degree of improvement of our model on non-repeated queries also outperforms all baseline models, which is attributed to more comprehensive user profile construction.

6 CONCLUSION

In this paper, we propose a dual-feedback network framework for personalized search to enhance the understanding of search intention. First, we design a hierarchical feedback extraction model, which can utilize the dual-feedback under similar queries in the long-term history to modify the representation of the current query. To further consider the impact of user interaction patterns in the current session, we construct a fine-grained feature extraction model to capture the user's intention change pattern. After obtaining the dual-feedback of the two sub-models, we use a multi-layer neural network to fuse them into unified dual feedback representation to enhance the current query representation. Experimental results confirm that our framework is effective for enhancing personalized search.

ACKNOWLEDGEMENT

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, Beijing Outstanding Young Scientist Program NO.BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. I also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-task learning for document ranking and query suggestion. In *International Conference on Learning Representations*.

[2] Paul N Bennett, Krysta Svore, and Susan T Dumais. 2010. Classification-enhanced ranking. In *Proceedings of the 19th international conference on World wide web*. 111–120.

[3] 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 *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 185–194.

[4] Fei Cai, Shangsong Liang, and Maarten De Rijke. 2014. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the SIGIR'2014*. ACM, 835–838.

[5] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 1849–1852.

[6] Kyunghyun Cho, Bart van Merriënboer Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk, and Yoshua Bengio. [n.d.]. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. ([n. d.]).

[7] Kevyn Collins-Thompson, Paul N Bennett, Ryen W White, Sebastian De La Chica, and David Sontag. 2011. Personalizing web search results by reading level. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. 403–412.

[8] 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*. 126–134.

[9] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*. 581–590.

[10] 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 *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 347–356.

[11] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research* 3, Aug (2002), 115–143.

[12] Shufeng Hao, Chongyang Shi, Zhendong Niu, and Longbing Cao. 2019. Modeling positive and negative feedback for improving document retrieval. *Expert Systems with Applications* 120 (2019), 253–261.

[13] Morgan Harvey, Fabio Crestani, and Mark J Carman. 2013. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2309–2314.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, Vol. 51. Acm New York, NY, USA, 4–11.

[16] Lin Li, Zhenglu Yang, Botao Wang, and Masaru Kitsuregawa. 2007. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *Advances in Data and Web Management*. Springer, 228–240.

[17] 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 *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 555–564.

[18] Zhengyi Ma, Zhicheng Dou, Guanyue Bian, and Ji-Rong Wen. 2020. PSTIE: Time Information Enhanced Personalized Search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1075–1084.

[19] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*. 1–es.

[20] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531* (2019).

[21] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[22] Ahu Sieg, Bamshad Mobasher, and Robin Burke. 2007. Web search personalization with ontological user profiles. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 525–534.

[23] Jaime Teevan, Susan T Dumais, and Daniel J Liebling. 2008. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 163–170.

[24] Jaime Teevan, Daniel J Liebling, and Gayathri Ravichandran Geetha. 2011. Understanding and predicting personal navigation. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 85–94.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[26] Maksims Volkovs. 2015. Context models for web search personalization. *arXiv preprint arXiv:1502.00527* (2015).

[27] Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search personalization with embeddings. In *European Conference on Information Retrieval*. Springer, 598–604.

[28] Thanh Vu, Alistair Willis, Son N Tran, and Dawei Song. 2015. Temporal latent topic user profiles for search personalisation. In *European Conference on Information Retrieval*. Springer, 605–616.

[29] Thanh Tien Vu, Dawei Song, Alistair Willis, Son Ngoc Tran, and Jingfei Li. 2014. Improving search personalisation with dynamic group formation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 951–954.

[30] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W White, and Wei Chu. 2013. Personalized ranking model adaptation for web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 323–332.

[31] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. 2008. A study of methods for negative relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 219–226.

[32] Ryen W White, Paul N Bennett, and Susan T Dumais. 2010. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 1009–1018.

[33] Ryen W White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web*. 1411–1420.

[34] Ryen W White and Steven M Drucker. 2007. Investigating behavioral variability in web search. In *Proceedings of the 16th international conference on World Wide Web*. 21–30.

[35] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. FeedRec: News Feed Recommendation with Various User Feedbacks. *arXiv preprint arXiv:2102.04903* (2021).

[36] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation.. In *IJCAI*. 2519–2525.

[37] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*. 55–64.

[38] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2020. Employing Personal Word Embeddings for Personalized Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1359–1368.

[39] Jing Yao, Zhicheng Dou, Jun Xu, and Ji-Rong Wen. 2020. RLPer: A Reinforcement Learning Model for Personalized Search. In *Proceedings of The Web Conference 2020*. 2298–2308.

[40] Wen-jing Zhang and Jun-yi Wang. 2012. The study of methods for language model based positive and negative relevance feedback in information retrieval. In *2012 Fourth International Symposium on Information Science and Engineering*. IEEE, 39–43.

[41] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *SIGIR*. ACM, 92–101.

[42] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding History with Context-aware Representation Learning for Personalized Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1111–1120.

[43] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 789–797.

[44] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2021. Enhancing Potential Re-finding in Personalized Search with Hierarchical Memory Networks. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[45] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2021. PSSL: Self-supervised Learning for Personalized Search with Contrastive Sampling. In *CIKM*. ACM, 2749–2758.