Enhancing Sequence Representation for Personalized Search

Shijun Wang
School of Information
Renmin University of China
wangshijun@ruc.edu.cn

Han Zhang
China Unicom Research Institute
zhangh1550@chinaunicom.cn

Zhe Yuan
Academy of Regional and Global Governance
Beijing Foreign Studies University
yuanzhe@bfsu.edu.cn

Abstract

The critical process of personalized search is to reorder candidate documents of the current query based on the user's historical behavior sequence. There are many types of information contained in user historical information sequence, such as queries, documents, and clicks. Most existing personalized search approaches concatenate these types of information to get an overall user representation, but they ignore the associations among them. We believe the associations of different information mentioned above are significant to personalized search. Based on a hierarchical transformer as base architecture, we design three auxiliary tasks to capture the associations of different information in user behavior sequence. Under the guidance of mutual information, we adjust the training loss, enabling our PSMIM model to better enhance the information representation in personalized search. Experimental results demonstrate that our proposed method outperforms some personalized search methods.

1 Introduction

Search engines have emerged as pivotal instruments for users to access information. However, search engines often yield identical results in response to identical queries, regardless of different information needs from different users, e.g., some users use the word "MAC" to refer to "MAC computer", in contrast, some other users use it to denote "MAC lipstick". To tackle this problem, personalized search has emerged as a highly effective strategy. It models a user's preferences by user's historical behavior, and returns customized ranking results. Personalized search has been proved to provide a better re-ranking document list and improve user satisfaction (Dou et al., 2007; Bennett et al., 2012; White et al., 2013; Cai et al., 2014; Ge et al., 2018; Ma et al., 2020; Zhou et al., 2020b).

Prevalent models integrate semantic context from user histories with current queries and documents. Deep learning models like hierarchical recurrent network with query-aware attention and graph-augmented approach have advanced the field but typically neglect the interconnectedness of historical informational units (Ge et al., 2018; Lu et al., 2020).

We believe that capturing contextual relationships between historical queries/documents and their behavioral contexts is vital for better personalization. Hence, we introduce an innovative methodology rooted in Mutual Information Maximization (MIM, elaborated upon in Appendix A). As a form of self-supervised learning, MIM exploits the inherent correlations present in user historical sequences to blend different types of data into a cohesive representation. This learning paradigm (Devlin et al., 2019; Zhou et al., 2020a) enables models to learn from the underlying structure of raw data, creating self-derived training signals and initializing model parameters through specifically designed optimization objectives.

We present the PSMIM (Personalized Search Mutual Information Maximization) model, incorporating auxiliary tasks to enhance user representations for personalized search. By exploiting self-supervised information from user historical sequences, we design three auxiliary tasks to capture the following correlations: (1) the correlation between historical query and historical sequence information, (2) the correlation between historical document and historical sequence information, and (3) the correlation between the current query and historical sequence information. These objectives are unified under the framework of maximizing mutual information, enabling the model to uncover associations among different information types in user sequences and adapt to new data or association patterns. By effectively fusing various user information types through auxiliary tasks, we obtain a richer user historical context representation, which is then integrated with candidate document representations.

We employ Transformers to capture interactions among the historical behavior sequence, current query, and candidate documents to compute personalized ranking scores. Our model is trained with pair-wise ranking loss. Through experiments, we validate the effectiveness of our approach, demonstrating that our model, PSMIM, outperforms several previous personalized search models.

Our contributions include proposing a personalized search model PSMIM which systematically addresses inter-associations among user sequence information through auxiliary objectives, and empirically validating the practical effectiveness and robustness of PSMIM.

2 Related Work

2.1 Personalized Web Search

The essence of personalized search lies in deciphering users' preferences and traits based on their historical interactions. Hence, scholars have developed several techniques to extract user-specific details from historical sequences. Initially, research efforts such as (Dou et al., 2007; Teevan et al., 2011) emphasized click behavior in user history, proving it to be a straightforward yet insightful tool for predicting user inclinations. Moreover, many studies (Sieg et al., 2007; White et al., 2013; Harvey et al., 2013; Vu et al., 2015; Vu et al., 2017) endeavored to formulate user profiles by examining the thematic attributes of documents within the user's search history. However, they faced challenges, including high computational expenses and incomplete categorization.

Recently, deep learning has garnered broad acceptance across disciplines due to its exceptional representation of learning capabilities. It successfully uncovers nuanced user preferences in personalized search contexts, as documented in (Ge et al., 2018; Lu et al., 2019; Yao et al., 2020b; Lu et al., 2020; Yao et al., 2020a; Ma et al., 2020; Zhou et al., 2020b).

Distinctively, our methodology capitalizes on the interdependencies among historical queries, historical documents, the current query, and the overall user historical sequence. By doing so, we aim to refine their representations and thoroughly explore the dynamics between candidate documents and the user's historical context, thus contributing to advancements in personalized search.

2.2 Self-supervised Learning

Self-supervised learning techniques (Devlin et al., 2019; Hjelm et al., 2019) leverage auxiliary tasks to extract meaningful insights from unsupervised data. By generating supervised signals from intrinsic data relationships, these methods foster effective representations for subsequent tasks. Some self-supervised frameworks (Hjelm et al., 2019) utilize naturally associated features for visual feature learning. Pre-trained language models (PLMs) (Devlin et al., 2019) represent a prominent example of self-supervised applications in the NLP domain.

Mutual Information Maximization (MIM) (Hjelm et al., 2019; Kong et al., 2020) is a self-supervised learning technique. It operates by splitting input data into multiple perspectives and strives to maximize mutual information among the representations of these perspectives.

Unlike prior strategies, our work considers correlations among user search history and contextual information as a self-supervised cue. We present a model utilizing MIM to enhance sequence-level user information embeddings, thereby boosting personalized search task performance.

3 Problem Definition

First, let us define the personalized search problem. Consider a user's historical information sequence $H = \{P_1, P_2, \dots\}$, with each page P_i contains an issued query, a set of documents, and a clicked document with its timestamp. Acknowledging that issued queries and their corresponding clicked documents are more informational, so our study focuses primarily on modeling these two critical types of historical

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 986-998, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

data within the sequence. Thus, we can assume a user's history as $H = \{(q_1, d_1), \ldots, (q_n, d_n)\}$, with n representing the total count of queries in the user's history. Given a current query q_c and a set of candidate documents $D_c = \{d_{c_1}, d_{c_2}, \ldots, d_{c_m}\}$, the personalized search task is to score each document in D_c according to the current query q_c and the user's historical information sequence H.

4 Our Method

Our method maximizes mutual information across user historical data, including historical queries, historical documents, alongside the current query. Auxiliary tasks help to enhance the model's capacity to represent diverse historical information types. Leveraging word-level and sequence-level transformers, our method effectively encodes the contextual and sequential nuances within user historical data.

4.1 Base Model

Here we introduce the base model upon which our contributions are built. It serves as a solid foundation to present our method. Due to word polysemy and contextual variations, user historical informaiton reflects diverse user intentions and interests. So we design it as a hierarchical transformer architecture, illustrated in Figure 1, which is crucial address the complexities of user historical information.

User historical information includes queries, documents, and timestamps. We use a word-level transformer to fuse the context information of the words in a query or a document, and use a sequence-level transformer to fuse the context information of the queries and the documents, then get a context-aware representation for a user information sequence. This architecture considers context at both the word and sequence levels, allowing us to capture the nuances of user interactions with the system over time.

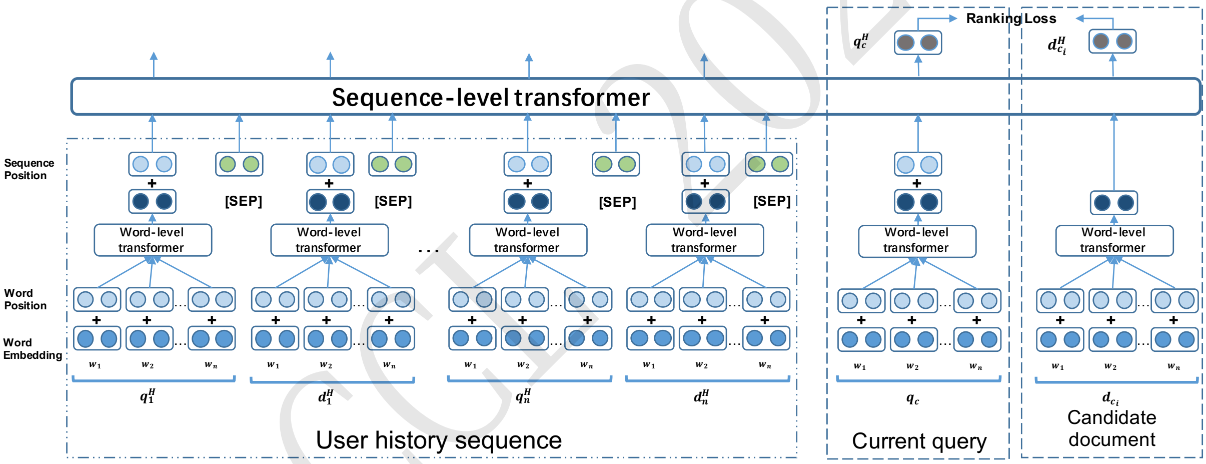

Figure 1: Base model of PSMIM (Personalized Search Mutual Information Maximization) framework.

The main component of the base model is a two-level context encoder, serving as context encoder for user history sequence.

The word-level context encoder is a transformer that takes in words from a query or a clicked document, and encodes them into a query embedding or a document embedding, preparing for the sequence-level context encoder. The goal of the word-level context encoder is to learn better representation for a query or a document, based on its whole text.

Taking a query q as an example (the word-level representation of document is the same as that of a query): suppose it consists of m words, expressed as $q = \{w_1, w_2, \ldots, w_m\}$. The word-level transformer takes in the word list $\{w_1, w_2, \ldots, w_m\}$, and then we get the embedding of query q.

The sequence-level context encoder is a transformer that takes in word-level embeddings of queries and documents from user history sequence, and encodes them into a user historical information sequence embedding. The goal of the sequence-level context encoder is to learn better representation for a user's historical information sequence, in other words, user history.

As we discussed earlier, the queries are usually very concise, sometimes just one word. In this case, there will still be ambiguity. Users usually put forward a series of queries for a single information requirement. In this process, historical queries and clicked documents provide rich context information to infer the user's intention. Therefore, we consider introducing the historical sequence context information to help represent the user's queries. Similarly, historically clicked documents represent the user's interest, and the combination of context query and document can better represent the user's interest in the documents.

After the word-level context encoder, a query or a document is integrated with the word-level context information. We take all queries and documents as context information to get the sequence-level representation. Suppose that the user historical information sequence is represented as the following list:

$$H = \{q_1, d_1, \ldots, q_t, d_t, \ldots, q_n, d_n, q_c\}, \tag{1}$$

the sequence-level transformer takes in the sequence list, then we get the embedding of the user historical information sequence H.

Current query representation: After the current query is sent into the Word-level Transformer as shown in Figure 1, each word in the current query is integrated with the word context information in the query. We conduct sum-pooling for all word representations of the query as the final representation. After the Sequence-level Transformer, the current query representation q_c^H incorporates information about the user's historical intentions and interests from the historical sequence. It should be noted that when the current query is fed into the Sequence-level Transformer, we only feed the historical information sequence into the model, and we do not feed the candidate document into the Transformer model.

Candidate document representation: Similar to the processing of the current query, after the candidate document is processed through the Word-level Transformer, we get a document representation that incorporates the word context, and after the sequence-level Transformer, the candidate document representation $d_{c_i}^H$ is represented with information that incorporates the user's historical intentions and interests. Similarly, when candidate documents are sent into the Sequence-level Transformer, we only consider historical information sequences in the model and do not use the current query in the Transformer model.

4.2 Auxiliary Learning Tasks

On the basis above, we design three auxiliary tasks with mutual information maximization using self-supervised information from user historical sequence to enhance the representation of user information. These three auxiliary tasks are respectively historical query prediction, historical document prediction and current query prediction. They model the associations of different information among the user historical sequence with mutual information maximization. These three auxiliary tasks are constructed as shown in Figure 2.

4.2.1 Historical Query Prediction

A historical query provides the user's fine-grained intention at a specific time. Our goal is establishing the correlation between a historical query and the user's whole historical sequence, to better represent the historical query. Existing personalized search models usually concatenate historical queries and documents, then regrade user historical information as a from-left-to-right sequence, which ignores the associations between historical queries and the whole sequence. It is also noted that during the training process, the model can observe the entire sequence of user historical information. Inspired by the model of BERT, we propose to use the bidirectional information in the historical sequence by a cloze task.

As an illustration, given a user historical sequence H and a historical query q_t, we mask the t-th query q_t. Then we take the rest sequence as its surrounding context, which is represented as:

$$C_{q_t} = \{q_1, d_1, \ldots, [mask], d_t, \ldots, q_n, d_n, q_c\}. \tag{2}$$

Considering the surrounding context C_{q_t} and masking term q_t, we treat them as distinct views for learning the data representation. According to the equations 19 in Appendix A, we minimize the Historical Query

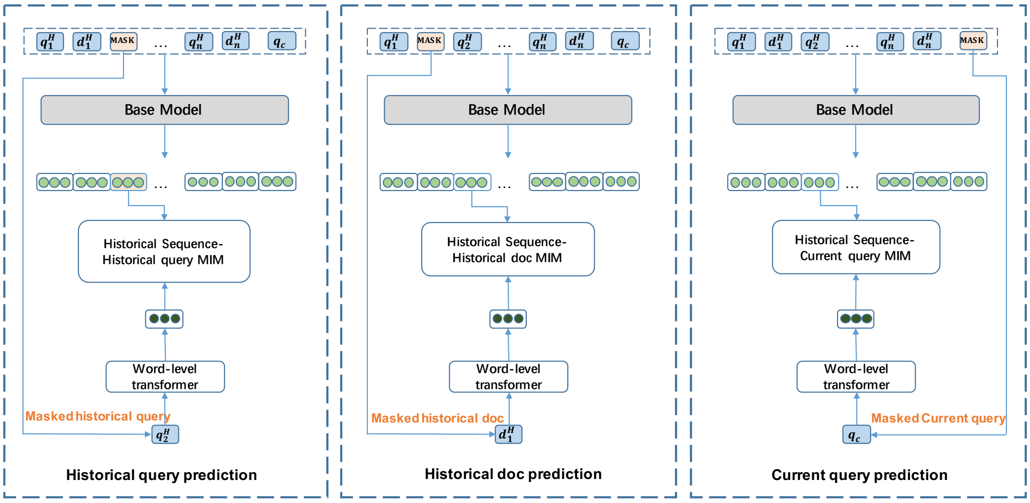

Figure 2: Three auxiliary tasks (Historical query prediction, Historical document prediction and Current query prediction) to model the associations of different information among the user historical sequence with Mutual Information Maximization to enhance the representation of user information.

Prediction (HQP) loss by:

$$L_{\text{HQP}}(C_{q_t}, q_t) = f(C_{q_t}, q_t) - \log\left(\sum_{\tilde{q}\in\mathcal{Q}\setminus\{q_t\}} \exp f(C_{q_t}, q_t)\right), \tag{3}$$

where $\tilde{q}$ denotes the negative sampling query and $f(\cdot)$ is calculated according to the following formula:

$$f(C_{q_t}, q_t) = \sigma(F_{q_t}^S \cdot W_{\text{HQP}} \cdot q_t^W), \tag{4}$$

where $W_{\text{HQP}} \in \mathbb{R}^*$ is the parameter matrix to be learned, $F_{q_t}^S$ is the learned sequence-level representation using the t-th query position of the user historical information sequence by sequence-level transformer, and q_t^W is the sequence-level representation of query q_t.

4.2.2 Historical Document Prediction

A clicked documents offers some fine-grained insights matching to the user's past interest in at a specific time. Our goal is establishing the correlation between a historical clicked document and the user's whole historical sequence, to better represent the clicked document.

Consistently with the previous subsection, given a user historical sequence H, we mask the t-th clicked document d_t, and take the rest sequence as its surrounding, and it is expressed as:

$$C_{d_t} = \{q_1, d_1, \ldots, q_t, [mask], \ldots, q_n, d_n, q_c\}. \tag{5}$$

We combine the context C_{d_t} and masking d_t as two different views. According to the equations in the 19 in Appendix A, we minimize the historical document prediction (HDP) loss by:

$$L_{\text{HDP}}(C_{d_t}, d_t) = f(C_{d_t}, d_t) - \log\left(\sum_{\tilde{d}\in\mathcal{D}\setminus\{d_t\}} \exp f(C_{d_t}, d_t)\right), \tag{6}$$

where $\tilde{d}$ denotes the negative sampling document and $f(\cdot)$ is calculated by:

$$f(C_{d_t}, d_t) = \sigma(F_{d_t}^S \cdot W_{\text{HDP}} \cdot d_t^W), \tag{7}$$

where $W_{\text{HDP}} \in \mathbb{R}^{d*d}$ is the parameter matrix to be learned, $F_{d_t}^S$ is the learned sequence-level representation using the t-th document position of the user historical information sequence by sequence-level transformer, and d_t^W is the word-level representation of document d_t.

4.2.3 Current Query Prediction

The current query reflects the user's current intent. Our goal is establishing the correlation between the current query and the user's whole historical sequence, to better represent the user's current intent. Similarly, we mask the current query q_c and take the rest of the information sequence as its surrounding context C_{q_c}, then we minimize the current query prediction (CQP) loss by:

$$L_{\mathrm{CQP}}(C_{q_c}, q_c) = f(C_{q_c}, q_c) - \log \left(\sum_{\tilde{q} \in \mathcal{Q}_{\mathcal{H}}} \exp f(C_{q_c}, q_c) \right), \tag{8}$$

where $\tilde{q}$ denotes the negative query sample and $f(\cdot)$ is calculated according to the following formula:

$$f(C_{q_c}, q_c) = \sigma(q_c^S \cdot W_{\mathrm{CQP}} \cdot q_c^W), \tag{9}$$

where $W_{\mathrm{CQP}} \in \mathbb{R}^{d*d}$ is the parameter matrix to be learned, $F_{q_c}^S$ is the learned sequence-level representation at the current query position in the user information sequence by sequence-level transformer, and q_c^W is the word-level representation of query q_c.

4.3 Candidate Document Score

In our method, given a current query q and a candidate document d, the score of the candidate document consists of two parts: the personalized relevance $p(d^H, q^H)$ and the ad-hoc relevance $p(d, q)$. As for the personalized relevance $p(d^H, q^H)$, we obtain the d^H of a candidate document integrating historical sequence information and the q^H of the current query through our model. After sum-pooling, our model acquire the integrated word-level context representation $d^{W_{sum}}$, upon which we conduct interaction with the current query q^H. Finally, we use one hidden layer MLP to automatically adjust the weights of various parts of these scores as below:

$$p(d^H, q^H) = \Phi(S(q^H, d^H), S(q^H, d^{W_{sum}})), \tag{10}$$

where, $S(\cdot)$ calculates similarity based on vector representation. In our work, we specifically employ cosine similarity.

The ad-hoc relevance $p(d, q)$ is computed through term matching between the current query and the candidate document, augmented by the context-aware word-level representations q^W and d^W. In addition, we extract the manual feature $F(q, d)$ of the clicks and topic for each document. These features are also weighted using one hidden layer MLP, as shown below:

$$p(d, q) = \Phi(I(q, d), I(q^W, d^W), \phi(F(q, d))), \tag{11}$$

where $I(\cdot)$ is based on the similarity of interactions. We adopt the KRNM (Xiong et al., 2017) method to implement the interaction. Specifically, given a query-document pair, we use the cosine similarity of each term in the query and document to construct an interaction matrix M. Then we use k kernels for matches. The final matching score $\mathrm{KNRM(M)}$ is obtained by aggregating them using one hidden layer MLP.

$$\mathrm{KNRM_k(M)} = \phi(\mathrm{K_1(M)}, \mathrm{K_2(M)}, \ldots, \mathrm{K_k(M)}), \tag{12}$$

$$K_k(M) = \sum_i \log \left(\sum_j \exp \left(\frac{(M_{ij} - \mu_k)^2}{2\sigma_k^2} \right) \right), \tag{13}$$

where μ_k is distributed between -1 and 1 depending on the number of kernels, and σ_k is set to 0.1.

Finally, we re-rank the candidate documents according to the final relevance score to obtain personalized search results that meet the users' needs.

4.4 Optimization Objective

In this section, we detail our model's optimization objective and the parameter configurations.

Ranking Loss. Many algorithms train the ranking task in pair-wise mode (Bennett et al., 2012; White et al., 2013; Ge et al., 2018; Lu et al., 2019; Yao et al., 2020b; Lu et al., 2020; Yao et al., 2020a; Ma et al., 2020; Zhou et al., 2020b). Each pair consists of a clicked document and an unclicked document, and the optimization goal is to maximize the gap in their scores. In particular, given a clicked document d_i and an unclicked document d_j, d_i is more relevant than d_j, and the prediction probability is $p(d_i|q,H) - p(d_j|q,H)$. The Ranking loss is the weighted cross-entropy between the ground-truth label $\overline{p}_{ij}$ and the predicted probability p_{ij}:

$$L_{\text{rank}} = -|\lambda_{ij}| \left(\overline{p}_{ij} \log p_{ij} + \overline{p}_{ji} \log p_{ji}\right), \tag{14}$$

where the weighting factor λ_{ij} represents the impact on ranking quality when documents d_i and d_j swap positions.

Auxiliary Task Losses. As mentioned above, we designed three self-supervised auxiliary tasks, each associated with a loss function: L_{HQP}, L_{HDP}, and L_{CQP}. We minimize the loss functions of the three auxiliary tasks together to update the parameters of our model. It should be noted that our model parameters are shared for the auxiliary and ranking tasks. The total loss of auxiliary tasks is as follows:

$$L_{\text{aux}} = z_1 L_{\text{HQP}} + z_2 L_{\text{HDP}} + z_3 L_{\text{CQP}}, \tag{15}$$

where z_1, z_2 and z_3 are the hyperparameters that control the weights.

Finally, we minimize the combination of these two losses, adjusted by a hyper-parameter α:

$$L_{\text{total}} = L_{\text{rank}} + \alpha L_{\text{aux}}. \tag{16}$$

5 Experiments

5.1 Dataset and Evaluation Metrics

Our study conducts experiments on the AOL dataset (Pass et al., 2006). The statistical details of the AOL dataset are summarized in Table 1.

Table 1: Statistics of the dataset

Type	AOL dataset
#Days	91
#Users	110,439
#Queries	736,454
Average query length	2.87
Average #click per query	1.11

The AOL dataset includes three months of user queries and corresponding clicked documents, fitting for personalized search purposes. Each user record includes an anonymous user ID, session ID, query, document, and click labels. We partition each user's data temporally into historical and experimental datasets. Historical data, encompassing the initial five weeks, serves as personalized historical information for each user's corresponding experimental data. The experimental data, covering the final eight weeks, is subsequently divided into training, validation, and testing sets following a 6:1:1 split ratio.

Given that the AOL dataset provides only clicked documents, we adhere to the methodology outlined in (Ahmad et al., 2018) and utilize the BM25 algorithm to identify the top-scoring candidate documents. Similarly, adopting the practices in (Ahmad et al., 2019; Huang et al., 2018), we extract 50 candidate documents per query in the testing set and five candidates per query in the training and validation sets. To ensure compliance with the fundamental principles of personalized search, we eliminate users whose historical data or training set is void. Furthermore, we exclusively use document titles for relevance calculations. In our study, we classify clicked documents in the AOL dataset as relevant, while the rest irrelevant. To assess the performance of various models, we employ three widely used metrics that gauge ranking quality: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision@1 (P@1). More implementation details are listed in Appendix B.

5.2 Baselines

We display the original ranking performance from the dataset and take several traditional and personalized search models as baseline models.

BM25 (Robertson and Zaragoza, 2009). As mentioned above, we use the BM25 algorithm to get one fundamental ranking.

KNRM (Xiong et al., 2017). This model conducts a kernel-pooling technique on a word-level similarity matrix to extract soft match features in multiple granularities. These features are then exploited by a pairwise Learning-to-Rank (LTR) algorithm to derive the final ranking score.

Conv-KNRM (Dai et al., 2018). In this model, a convolution layer is added based on KNRM to construct the n-gram soft matches, and the context information of words is added.

Bert (Devlin et al., 2019). We input the query and document simultaneously into a pre-trained BERT model and extract the final "[CLS]" token's output as the matching feature to compute the matching score.

P-Click (Dou et al., 2007). This model utilizes the click frequency and initial position of candidate documents in the user's historical sequence as personalized features for document re-ranking It has a good performance on user's refinding behavior.

SLTB (Bennett et al., 2012). In this model, click features, topic features, time features, and location features are taken as the features of personalized search. It utilizes a learning-to-rank approach to achieve document re-ranking.

HRNN (Ge et al., 2018). The model uses the hierarchical recurrent neural network and attention mechanism to model the historical information related to the current query and constructs the user profile based on the current query for the personalized search.

PSTIE (Ma et al., 2020). This model considers using time interval information in history to enhance the performance of the personalized search.

HTPS (Zhou et al., 2020b). This model divides the user history into long-term and short-term histories to disambiguate the current query and designs a personalized language model task to improve the model's effectiveness. Since the word embedding of most personalized search methods is fixed, we keep the word embedding static without fine-tuning for a fair comparison.

5.3 Experimental Results

We categorize our baseline models into two main groups: traditional search methods and personalized search methods. As shown in Table 2, our proposed personalized search model PSMIM surpasses traditional search models and other personalized search methodologies, outperforming several counterparts.

Table 2: Overall performance of all models on AOL dataset. "†" indicates the model outperforms all baselines significantly with paired t-test at p <0.05 level. Best results are shown in bold.

Task	Model	MAP		MRR		P@1	
Traditional Search	BM25	.250	-62.80%	.258	-62.40%	.148	-75.05%
	KNRM	.429	-36.16%	.439	-36.01%	.271	-54.31%
	Conv-KNRM	.474	-29.46%	.485	-29.31%	.327	-44.86%
	BERT	.483	-28.13%	.493	-28.14%	.335	-43.51%
Personalized Search	P-Click	.422	-37.20%	.431	-37.17%	.378	-36.26%
	SLTB	.511	-23.96%	.524	-23.62%	.469	-20.92%
	HRNN	.544	-19.05%	.557	-18.80%	.484	-18.39%
	PSTIE	.564	-16.07%	.577	-15.89%	.503	-15.18%
	HTPS	.672	-	.686	-	.593	-
	PSMIM	**.689**†	+2.53%	**.703**†	+2.48%	**.610**†	+2.87%

Subsequently, these personalized search methods can be divided into two categories: those based on user profiles (P-Click, SLTB, HRNN and PSTIE), and context-aware approaches (HTPS and PSMIM). By comparing context-aware personalized methods with those based on user profiles, we find that HTPS and PSMIM are superior to previous personalized search models. This finding emphasizes the importance of historical context.

Further, we contrast our model PSMIM against context-aware model HTPS. As evidenced in Table 2, our model excels over preceding personalized search models. Compared with the benchmark HTPS, our model exhibits statistically significant enhancements across all evaluation metrics, as confirmed by paired t-test at a $p < 0.05$ significance level. Significantly, on the AOL dataset, PSMIM records a notable improvement over HTPS with a 2.53% increase in MAP, a 2.48% boost in MRR, and a 2.87% rise in P@1. These results prove that our method is effective and robust.

In conclusion, the experimental outcomes affirm that accounting for the interconnections within user sequence information and employing context-aware candidate document representations effectively enhances the performance of personalized search. To further dissect the functionality of our model components, we conduct an ablation study and some further experiments in the upcoming subsections.

5.4 Discussion

5.4.1 Ablation Study

Our PSMIM model includes several essential components: the current query prediction task (CQP), the historical query prediction task (HQP), the historical document prediction task (HDP), and candidate document representation as historical context information (CDC). To prove the effectiveness of these main components in the PSMIM model, we perform several ablation experiments. We present the experimental results in Table 3 and have some discussions. Specifically, in setting PSMIM without CQP, we remove the current query prediction task of the PSMIM model and then train and test the model. In setting PSMIM without HQP&HDP, since the historical query and historical document are inseparable sources for deriving context information, we concurrently remove the historical query prediction and historical document prediction tasks from the original process. In setting PSMIM without CDC, we replace the candidate document representation integrating historical context information with word embedding representation of the original candidate document.

Table 3: Results of ablation experiments.

Model	MAP	MRR	P@1
PSMIM	**0.6887**	**0.7029**	**0.6096**
w/o. CQP	0.6789	0.6932	0.5976
w/o. HQP&HDP	0.6713	0.6853	0.5928
w/o. CDC	0.6800	0.6939	0.5999

In Table 3, it is evident that the performance of our model diminishes under the three implemented ablation strategies, falling short of the complete model's performance. Notably, the removal of historical query and historical document prediction tasks leads to the most substantial decrease in MAP, confirming the necessity and significance of capturing associations within users' historical interaction sequences. Discontinuing the current query prediction task also impairs the results, pointing to the importance of strengthening the representation of the current query for improved performance. Moreover, excluding historical context into candidate document embeddings leads to reduced model performance, highlighting the benefits of exploiting personalized candidate document representations.

5.4.2 Effect of Short-term and Long-term History

To validate the robustness of our model, akin to HTPS (Zhou et al., 2020b), we partition user history into short-term and long-term segments, and scrutinize the performance of our model under both short-term and combined short-term and long-term histories scenarios.

HTPS-S. We select HTPS, a model surpassing most prior personalized search approaches, as our baseline. We only use short-term history during training and testing the model.

PSMIM-S. We only use short-term user history during training and testing our model.

The results are shown in Table 4. It could be seen from the experimental results that our model PSMIM with all user history significantly outperforms the model HTPS (Zhou et al., 2020b), and the model PSMIM-S with short-term history still outperforms the model HTPS in terms of MAP and MRR, which can prove the robustness of our model.

Table 4: Performance of models with short and long term historys. † indicates the model outperforms baselines significantly with paired t-test at p-value < 0.05.

Model	MAP	MRR	P@1
HTPS-S	0.6280	0.6410	0.5210
HTPS	0.6720	0.6860	0.5930
PSMIM-S	0.6737†	0.6877†	0.5921
PSMIM	**0.6887**†	**0.7029**†	**0.6096**†

However, PSMIM-S exhibits marginally lower performance than HTPS (Zhou et al., 2020b) on P@1 metric. A possible reason is that the model PSMIM-S with the short-term history could not model the overall historical interests of a user.

5.4.3 Performance of Ambiguous and Non-ambiguous Query Sets

Typically, a user's queries fall into two categories: navigational and informational. The navigational queries usually have more precise query intent, and the informational queries are usually more ambiguous. Studies [8] have shown that it is more necessary to personalize the search results for informational queries, which typically exhibit higher click entropy. To verify the robustness of our model, we follow the setting in [29], computing click entropy for queries, using a threshold of 1.0 to separate queries into two subsets, and subsequently contrasting the results with the baseline model HTPS.

From the results in Table 5, our model outperforms the baseline model HTPS, especially on the queries with larger click-entropy. It indicates that our model can learn better user information representation when facing ambiguous queries.

Table 5: Performance of Ambiguous and Non-Ambiguous Query Sets.

Query sets	on Ambiguous Query Set	on Non-Ambiguous Query Set
HTPS	0.3322	0.4240
PSMIM	0.3540	0.4421

5.4.4 Performance of Repeated and Non-repeated Query Sets

A user's queries could be classified into repeated and non-repeated queries. It is easy to infer the user's intent based on the same queries in the user history for repeated queries. Non-repeated queries, however, offer limited information based on their historical clicked documents. In this situation, a better query representation may bring some performance benefits.

As the results are shown in Table 6, our model outperforms the baseline model HTPS on both repeated queries and non-repeated queries. This indicates that our model can better infer the intent when facing a new query with a better information representation.

Table 6: Performance of Repeated and Non-Repeated Query sets.

Model	on Repeated Query Sets	on Non-Repeated Query Sets
HTPS	0.6048	0.2576
PSMIM	0.6310	0.2908

6 Conclusion

In this paper, we propose a personalized search model PSMIM under the guidance of mutual information, and design three auxiliary tasks to obtain the associations among different types of data from user history. Experiment results verify the effectiveness and robustness of our personalized search model PSMIM.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper. The study was supported by the China Postdoctoral Fellowship Program of CPSF(GZC20230287) and the Fundamental Research Funds for the Central Universities(2024QY004).

References

Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-task learning for document ranking and query suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 385–394, New York, NY, USA. Association for Computing Machinery.

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*, SIGIR '12, page 185–194, New York, NY, USA. Association for Computing Machinery.

Fei Cai, Shangsong Liang, and Maarten de Rijke. 2014. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 835–838, New York, NY, USA. Association for Computing Machinery.

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*, WSDM '18, page 126–134, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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*, WWW '07, page 581–590, New York, NY, USA. Association for Computing Machinery.

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*, CIKM '18, page 347–356, New York, NY, USA. Association for Computing Machinery.

Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13(null):307–361, February.

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*, CIKM '13, page 2309–2314, New York, NY, USA. Association for Computing Machinery.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. 2018. Improving entity recommendation with search log and multi-task learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4107–4114. International Joint Conferences on Artificial Intelligence Organization, 7.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Proceedings of the 23rd China National Conference on Computational Linguistics, pages 986-998, Taiyuan, China, July 25 - 28, 2024.
Volume 1: Main Conference Papers

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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*, SIGIR'19, page 555–564, New York, NY, USA. Association for Computing Machinery.

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 and Development in Information Retrieval*, SIGIR '20, page 709–718, New York, NY, USA. Association for Computing Machinery.

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*, CIKM '20, page 1075–1084, New York, NY, USA. Association for Computing Machinery.

Paninski and Liam. 2014. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253.

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, page 1–es, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, apr.

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*, CIKM '07, page 525–534, New York, NY, USA. Association for Computing Machinery.

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*, WSDM '11, page 85–94, New York, NY, USA. Association for Computing Machinery.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Thanh Vu, Alistair Willis, Son N. Tran, and Dawei Song. 2015. Temporal latent topic user profiles for search personalisation. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr, editors, *Advances in Information Retrieval*, pages 605–616, Cham. Springer International Publishing.

Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search personalization with embeddings. In Joemon M Jose, Claudia Hauff, Ismail Sengor Altıngovde, Dawei Song, Dyaa Albakour, Stuart Watt, and John Tait, editors, *Advances in Information Retrieval*, pages 598–604, Cham. Springer International Publishing.

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*, WWW '13, page 1411–1420, New York, NY, USA. Association for Computing Machinery.

C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. *ACM SIGIR FORUM*, 51(cd):55–64.

Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2020a. Employing personal word embeddings for personalized search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1359–1368, New York, NY, USA. Association for Computing Machinery.

Jing Yao, Zhicheng Dou, Jun Xu, and Ji-Rong Wen. 2020b. Rlper: A reinforcement learning model for personalized search. In *Proceedings of The Web Conference 2020*, WWW '20, page 2298–2308, New York, NY, USA. Association for Computing Machinery.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020a. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1893–1902, New York, NY, USA. Association for Computing Machinery.

Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020b. 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*, SIGIR '20, page 1111–1120, New York, NY, USA. Association for Computing Machinery.

Appendix A Mutual Information Maximization

Mutual information maximization is a pivotal strategy for integrating diverse forms of historical information. Rooted in information theory, mutual information (MI) is a valuable tool for quantifying the dependence between random variables. Its mathematical definition is expressed as:

$$I(A, B) = P(A) - P(A|B) = P(B) - P(B|A). \tag{17}$$

Suppose A and B are different views of the input data, such as a word and its context in NLP tasks or a document and its historical context sequence in personalized search. Let f be a function receiving $A = a$ and $B = b$ as inputs. The primary aim of maximizing MI is to tune the parameters of f to maximize the mutual information $I(A, B)$, thereby extracting the most discriminative and salient attributes of the samples.

The essence of effective feature extraction involves distinguishing a sample from the entire dataset by capturing its distinctive information. By maximizing mutual information, one can isolate and harness such unique characteristics. However, when f constitutes neural networks or other encoders, directly optimizing MI is usually tricky (Paninski and Liam, 2014). Thus, a common workaround is to find a tractable lower bound for $I(A, B)$ that closely approximates the target function. A specific lower bound proved to be effective in practice is InfoNCE (Logeswaran and Lee, 2018; van den Oord et al., 2018), which is based on noise contrast estimation (Gutmann and Hyvärinen, 2012). InfoNCE is defined as follows:

$$\text{InfoNCE} = \mathbb{E}_p(A, B)\left(f_\theta(a, b) - \mathbb{E}_{q(\tilde{\mathcal{B}})}\left(log \sum_{\tilde{b} \in \tilde{\mathcal{B}}} \exp f_\theta(a, \tilde{b})\right)\right) + log|\tilde{\mathcal{B}}|, \tag{18}$$

where a and b are different views of the input data, and $f_\theta \in \mathbb{R}$ is a function whose parameter is θ (for example, dot product result expressed by word and context or cos distance). $\tilde{\mathcal{B}}$ is a set of samples taken from the distribution $q(\tilde{\mathcal{B}})$. The B set contains a positive sample b and $|\tilde{\mathcal{B}}| - 1$ negative samples. Learning representation based on this goal is also called contrastive learning.

We can see that the InfoNCE is analogous to the cross-entropy form the formula below when $\tilde{\mathcal{B}}$ can take all possible values of B (i.e., $\tilde{\mathcal{B}} = \mathcal{B}$) and they are uniformly distributed, maximizing InfoNCE is analogous to maximize the cross-entropy loss:

$$\mathbb{E}_p(A, B) = \left(f_\theta(a, b) - \log \sum_{\tilde{b} \in \mathcal{B}} \exp f_\theta(a, \tilde{b})\right). \tag{19}$$

Appendix B Implementation Details

The parameters of our model PSMIM are set as follows: The word embedding size is 100. The hidden size of the transformer layer in our base model is 512. The number of heads in multi-head attention is 8. The size of the MLP hidden layer is 256. In the experiment, we set the hyperparameters z_1, z_2 and z_3 in Formula 15 of three auxiliary tasks to 1.0. We use the Adam optimizer to minimize the final loss L_{total}, and the learning rate of our optimizer is $1e^{-3}$. In the experiment, we set α in Formula 16 to 1.0. In addition, the number of matched cores for the KRNM model is 11.