PSLOG: Pretraining with Search Logs for Document Ranking

Zhan Su
Zhicheng Dou
Yujia Zhou
suzhan@ruc.edu.cn
dou@ruc.edu.cn
zhouyujia@ruc.edu.cn
Renmin University of China
Beijing, China

Ziyuan Zhao
joshuazhao@tencent.com
Search Algorithm Group, WeChat
Tencent
Guangzhou, China

Ji-Rong Wen
jrwen@ruc.edu.cn
Engineering Research Center of
Next-Generation Intelligent Search
and Recommendation, Ministry of
Education, China
Renmin University of China
Beijing, China

ABSTRACT

Recently, pretrained models have achieved remarkable performance not only in natural language processing but also in information retrieval (IR). Previous studies show that IR-oriented pretraining tasks can achieve better performance than only finetuning pretrained language models in IR datasets. Besides, the massive search log data obtained from mainstream search engines can be used in IR pretraining, for it contains users' implicit judgment of document relevance under a concrete query. However, existing methods mainly use direct query-document click signals to pretrain models. The potential supervision signals from search logs are far from being well explored. In this paper, we propose to comprehensively leverage four query-document relevance relations, including co-interaction and multi-hop relations, to pretrain ranking models in IR. Specifically, we focus on the user's click behavior and construct an Interaction Graph to represent the global relevance relations between queries and documents from all search logs. With the graph, we can consider the co-interaction and multi-hop q-d relationships through their neighbor nodes. Based on the relations extracted from the interaction graph, we propose four strategies to generate contrastive positive and negative q-d pairs and use these data to pretrain ranking models. Experimental results on both industrial and academic datasets demonstrate the effectiveness of our method.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Pretrained Language Models, Search Log, Interaction Graph

ACM Reference Format:
Zhan Su, Zhicheng Dou, Yujia Zhou, Ziyuan Zhao, and Ji-Rong Wen. 2023. PSLOG: Pretraining with Search Logs for Document Ranking. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599477

KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599477

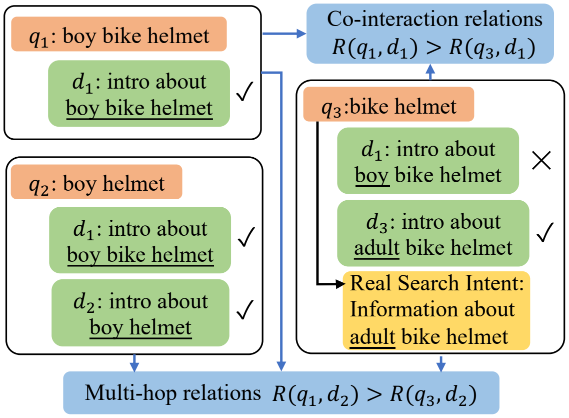
q_1: boy bike helmet
d_1: intro about boy bike helmet
✓
Co-interaction relations $R(q_1, d_1) > R(q_3, d_1)$
q_3:bike helmet
d_1: intro about boy bike helmet
×
d_3: intro about adult bike helmet
✓
Real Search Intent: Information about adult bike helmet
q_2: boy helmet
d_1: intro about boy bike helmet
✓
d_2: intro about boy helmet
✓
Multi-hop relations $R(q_1, d_2) > R(q_3, d_2)$

Figure 1: The example of mining potential query-document relevance relations from multiple search sessions. ✓ denotes more user clicks while × stands for fewer or no user clicks.

1 INTRODUCTION

Pretrained language models (PLM) like BERT [7] have demonstrated the powerful capability of understanding natural languages. Different from natural language processing (NLP), the relevance of the query and documents is an essential factor for the ranking models to measure in the information retrieval (IR) tasks, which is not explicitly considered in most language modeling pretraining tasks. Although finetuning the pretrained language models in the downstream IR tasks achieves excellent performance, recent studies [3, 4, 10, 20–22] find that pretraining ranking models with IR-tailored tasks will further improve the ranking performance of pretrained models in the IR tasks.

Search log data captures users' behaviors, such as issuing queries and browsing through documents. These logs can be utilized to enhance the quality of search results since they contain implicit judgments by users regarding the relevance of documents to specific queries. Moreover, search logs are widely exploited in IR tasks [1, 2, 8]. Considering the vast amount of search log data that can be obtained from commercial search engines, it becomes feasible to leverage this data for pretraining a ranking model. However, most existing approaches [18, 40] simply utilize clicks on documents within a single search session to pretrain ranking models. Consequently, they overlook the complex relationships and numerous potential supervised signals present in the search logs during the pretraining phase.

To fully leverage the weak supervised query-document relevance signals in the search log for pretraining ranking models, we propose to consider the q-d relations not only within the same search session but also from a global view of the search log. The benefits of considering global q-d relevance relations are shown in Figure 1. Considering that most users issuing query q_3 click document d_3 and ignore document d_1, it is more possible that the real search intent of query q_3 is to seek the information about adult bike helmet rather than boy bike helmet. In another word, we can discover that q_3 is more relevant to d_3 than d_1, namely $R(q_3, d_3) > R(q_3, d_1)$. Furthermore, if we jointly consider q_1 and q_3 that have common interaction on d_1, we can find that d_1 is more relevant to q_1 than q_3. Therefore, we can derive the co-interaction relations that $R(q_1, d_1) > R(q_3, d_1)$. Moreover, if we consider another search session like q_2, we can notice that both d_1 and d_2 are users' preference results. Based on this observation, it is reasonable to infer documents d_1 and d_2 are similar. Comprehensively leveraging the inferences above, we can obtain the multi-hop relations that $R(q_1, d_2) > R(q_3, d_2)$. Compared with only considering q-d relevance within a single session, exploiting q-d relations from all sessions is a benefit to obtain more insights about the real query-document relevance.

In this paper, we propose leveraging four kinds of query-document relations, including co-interaction and multi-hop relations, in the search log to pretrain ranking models. There are two main advantages of our method: (1) The human click signals in the search log can be used to provide human instructions about q-d relevance, which is a supplement to the text similarity for the model to measure the relevance of queries and documents. (2) The co-interaction and multi-hop relevance relations can provide more insights about the real query-document relevance, which is hard to obtain by only considering the relations within each separate search session.

To comprehensively model the relations in the search log, we build an **Interaction Graph** to represent the global relevance relations of queries and documents from all search sessions. On the interaction graph, queries and documents are nodes, while their relations are edges between them. There are two types of relations: clicked and unclicked. Note that we aggregate all the user clicks of the same queries and the interaction graph is built based on the aggregated search logs. The interaction relations of q and d denote whether most users preferred document d under the query q.

Concretely, our **p**retraining with **s**earch **log**s method **PSLOG** consists of four pretraining tasks. These pretraining tasks are designed to cultivate the ranking model's relevance sense capability by leveraging the query-document relations extracted from the interaction graph. **(1) Click Document Prediction (CDP)** task. The CDP task deals with the relations in the single search session and it is intended for the model to distinguish clicked (a.k.a. positive) document d^+ and unclicked (a.k.a. negative) document d^- under the same query q. **(2) Relevant Query Comparison (RQC)** task. This task deals with the co-interaction relations and it is designed for the model to choose which query of the two queries (*e.g.*q^+ and q^-) is more relevant to the given document d. **(3) Multi-hop Document Prediction (MDP)** task. The MDP task deals with the multi-hop relations extracted from the interaction graph. Provided with two q-d pairs (q, d_m^+) and (q, d_m^-), the model is expected to choose which document can better answer the question of q. Different from CDP task, d_m^+ and d_m^- are positive and negative documents sampled from q's multi-hop neighbors, respectively. **(4) Multi-hop Query Comparison (MQC)** task. This task also deals with multi-hop relations. A visual example is the multi-hop relations pair shown in Figure 1. Different from the MDP task, the MQC task prepares (q_m^+, d) and (q_m^-, d) pairs for the model to distinguish which multi-hop query search for the content contained in the document d.

Our main contributions can be summarized in these aspects:

(1) We propose to comprehensively consider different kinds of q-d relevance relations, including co-interaction relations and multi-hop relations, from the search logs to pretrain a ranking model.

(2) We introduce an Interaction Graph, offering a global perspective on query-document relations across all search sessions. We also utilize the interaction graph to figure out potential relations between queries and documents through their neighboring nodes.

(3) We propose four pretraining tasks based on relations extracted from the interaction graph for the document ranking model and obtain excellent performance in the experiments.

2 RELATED WORK

2.1 Pretrained Language Models

Pretrained language models (PLM) [7, 16, 19] have shown impressive performance on text modeling and many downstream tasks (*e.g.*Text Classification [35], Machine Translation [9]). As the representative one, BERT [7] is a pretrained on a large-scale corpus with masked language model (MLM) task and next sentence prediction (NSP) task based on Transformer encoders [31]. Both tasks were proposed to leverage self-supervised signals to pretrain models. Apart from the MLM and NSP tasks, there are also several pretraining tasks designed for PLM. For instance, token deletion, sentence permutation, and document rotation [16]. Besides, researchers also attempted to exploit external resources (*e.g.*knowledge base) to pretrain language models [30, 38].

2.2 Related IR Methods

2.2.1 IR-oriented Pretraining Methods.

Apart from semantic features, ranking models in IR are expected to extract the relevance features of queries and documents, which is ignored in the NLP pretraining tasks. Researchers found there is still improvement room for pretrained models in IR tasks. Then several IR-oriented pretraining tasks are proposed to pretrain ranking models. For example, Chang et al. [3] proposed several methods like Body First Selection to select possible queries for the given documents. Besides, Ma et al. [20] leveraged statistic methods to generate the representative words from the document as the pseudo-queries for relevance pretraining. Then the BERT-enhanced variant method B-PROP [21] was proposed to further improve the quality of generated queries of PROP. Chen et al. [4] proposed several axioms for pretraining IR models. These methods show the benefits of designing special pretraining tasks for information retrieval.

Other methods leveraged external sources to pretrain ranking models. For example, Seonwoo et al. [25] leveraged entity hyperlinks to find answers for pretraining in the QA task. Ma et al. [22] exploited hyperlinks in the document to generate related queries. And Guo et al. [10] pretraining ranking models with HTML structure information. Compared with the methods above, we also design

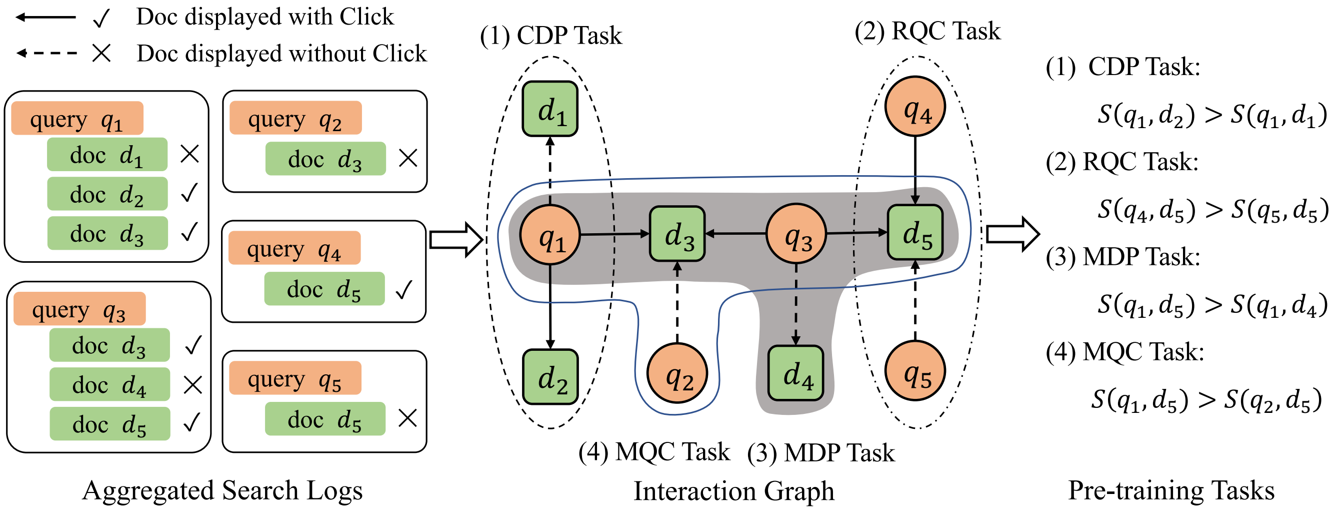

Figure 2: The procedure of extracting relations with the Interaction Graph. The query and document nodes used for each task are circled or in shadow. The positive neighbors are connected with solid arrows and the negative ones are with dash arrows.

IR-tailored pretraining tasks for ranking models. However, we obtain the training query-document pairs from real search logs, which is much closer to the practical situations.

2.2.2 Log-based Methods in IR.. Search log data has been widely used in the information retrieval area and it turns out to be useful to improve the model's ranking performance [1, 2, 5, 8, 13]. Pretraining ranking models with search log data have been used in the industry. For example, Liu et al. [18] used the clicked document as positive results while exposed documents without clicks were considered as negative results under the same query. Then the generated samples were used to pretrain ranking models. Besides, Zou et al. [40] focused on reducing noise in the search logs and leveraging human-labeled data to train a classifier to get revised document relevance labels for the given query. Then the revised labels are used to pretrain models. Different from these methods, we not only consider the innter-session click relations but also co-interaction and multi-hop relations, which can help ranking models acquire a better knowledge of relevance.

2.2.3 Graph-based Methods in IR. Graph structure is widely used in many IR tasks [23, 27, 37], and it turns out to be useful to handle multidimensional information. For instance, PageRank [23] and HITS [15] are representative methods that model node importance from their relationship with graph structure. Jiang et al. [12] exploited the click number of the query-document bipartite graph to estimate the relevance. In recent years, graph neural networks [14, 33] have also made a breakthrough in many IR tasks [17, 26–28, 32, 36]. Compared with these methods, we focus on mining the weak relevance signal from the interaction graph and apply multi-hop relevance propagation for the pretraining task.

3 OUR METHOD PSLOG

Given the requirement to generate high-quality rankings of relevant candidate documents for a given query, ranking models are expected to possess the ability to discern subtle differences between similar query-document pairs. To address this, we leverage hard negative query-document pairs, sampled from the related search log, to train the ranking model in the pretraining procedure.

To comprehensively leverage various types of query-document relationships, we construct an Interaction Graph based on large-scale search logs. Moreover, ranking models often encounter new queries in real search scenarios. Consequently, we employ multi-hop relevance propagation on the interaction graph to extract novel but relevant query-document pairs for pretraining.

This section introduces four pretraining tasks specifically tailored for training our ranking model, namely PSLOG, using the search log and an interaction graph. The tasks include the click document prediction (CDP) task and relevant query comparison (RQC) task, which focus on one-hop query-document relationships. Additionally, we propose the multi-hop document prediction (MDP) task and multi-hop query comparison (MQC) task, intended for multi-hop query-document relevance comparisons.

3.1 Interaction Graph

The overall procedure of our approach PSLOG is shown in Figure 2. Different from considering search sessions separately, we build a global interaction graph from all search sessions to comprehensively leverage the complicated relations of queries and documents. Given that users usually click on results that align with their search intents, click signals serve as a valuable measure of document relevance. Consequently, we can designate documents with a higher number of user clicks as positive relevant documents for a given query, while those with fewer clicks are considered negative documents. Moreover, we can present the relevance signals of the queries and documents on the graph.

Formally, we obtain an Interaction Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from aggregated search logs, where $\mathcal{V}$ represents the vertex set (or node set) on the interaction graph and $\mathcal{E}$ denotes the edge set between nodes. More concretely, the nodes set $\mathcal{V} = \{\mathcal{Q}, \mathcal{D}\}$ consists of query nodes and document nodes. In Figure 2, query nodes set $\mathcal{Q} = \{q_1, q_2, q_3, q_4, q_5\}$ and document nodes set $\mathcal{D} = \{d_1, d_2, d_3, d_4, d_5\}$. As shown in Figure 2, query nodes are only connected to their candidate documents nodes with two kinds of arrows. Specifically, query nodes are connected to the relevant documents with solid arrows (*e.g.*$q_1 \longrightarrow d_2$), while document nodes with negative signals are connected to their query nodes with dashed arrows (*e.g.*$q_1 \dashrightarrow d_1$). For the description convenience, we define a positive neighbor set $\mathcal{P}_v$ and a negative neighbor set $\mathcal{N}_v$ for each vertex v on the interaction graph. For query nodes on the graph, the positive neighbor set is the document set with positive click signals, while the negative neighbor set is the document set with negative click signals. For example, the positive neighbor set of the query q_1 is $\mathcal{P}_{q_1} = \{d_2, d_3\}$, while the negative neighbor set $\mathcal{N}_{q_1} = \{d_1\}$. By analogy, document d_3 has positive neighbor $\mathcal{P}_{d_3} = \{q_1, q_3\}$ and negative neighbor $\mathcal{N}_{d_3} = \{q_2\}$. It is worth noting that all the document nodes (documents with positive and negative signals) on the interaction graph are displayed in the search log. In another word, both positive and negative documents are examined by many different users. And the documents displayed on the search result page without click (*e.g.*d_1) for most users are hard negative samples compared with randomly sampled documents, which makes it difficult for the model to distinguish.

With the interaction graph, we can obtain both local and global relations of all the search sessions. Leveraging two kinds of relevance signals on the graph and exploring the relations of queries and documents by their neighbor nodes on the graph, we derive four pretraining tasks tailed for ranking models.

3.2 Click Document Prediction (CDP) Task

Considering that ranking models confront the problem of ranking candidate documents obtained in the first retrieval phase, they need to discern the relevant documents from irrelevant documents. Hence, we pretrain our model with Click Document Prediction (CDP) task. The key idea of the CDP task is to predict which document is more likely to be clicked by the users for the given query. Although user clicks data has noise and bias, it is a straightforward pretraining task and our method is compatible with other complicated noise reduction methods (*e.g.*[40]). To reduce costs and improve the ease of use of our model, we aggregate the click signals from different users for every query and filter out the search sessions with a low examined rate.

In the click document prediction task, we generate training samples from the query perspective. For each query node $q \in \mathcal{Q}$ on the interaction graph, we can obtain positive and negative documents from its neighbor set $\mathcal{P}_q$ and $\mathcal{N}_q$. For instance, the query node q_1 in Figure 2 has three document neighbors d_1, d_2, and d_3. These documents are all displayed on the search result page of query q_1. However, the document d_2 and d_3 are more preferred by the user according to the click rate. We hope the ranking model could discover the difference between the documents d_2, d_3 and d_1. Therefore, the output scores of the (q_1, d_2) pair and (q_1, d_3) pair should larger than (q_1, d_1) pair, namely $S(q_1, d_2) > S(q_1, d_1)$ and $S(q_1, d_3) > S(q_1, d_1)$. For the model architecture, we adopt the Transformer [31] encoder and the concatenation of query and document content as input. The relevance score of the input query-document pair is calculated by the [CLS] output and an MLP function. Formally, the score of query q and document d is generated as follows:

$$S(q, d) = \sigma\left(\mathrm{MLP}\left(\mathbf{E}(q, d)\right)\right), \tag{1}$$

$$\mathbf{E}(q, d) = \underset{[\mathrm{CLS}]}{\mathrm{Encoder}}([\mathrm{CLS}]; q; [\mathrm{SEP}]; d; [\mathrm{SEP}]), \tag{2}$$

where $\mathbf{E}(q, d)$ is the [CLS] embedding representation of (q, d) pair generated by the encoder, σ is the activation function and $[;]$ is the concatenation operation. In the pretraining task, we adopt the pairwise loss (*e.g.*hinge loss) for measuring the model's distinguish capability of positive pair (q, d^+) and negative pair (q, d^-).

$$\mathcal{L}_{\mathrm{CDP}} = \max\left(0, 1 - S(q, d^+) + S(q, d^-)\right), \tag{3}$$

where the score disparity of positive and negative pairs $S(q, d^+) - S(q, d^-)$ is larger, the loss is lower. Hence, the output score of positive pair $S(q, d^+)$ is expected to be higher than $S(q, d^-)$.

3.3 Relevant Query Comparison (RQC) Task

Different from the CDP task, the relevant query comparison (RQC) task is designed with a document-centric perspective to mine co-interaction relations. Considering that some relations that cannot be discovered in a single search session, we exploit the interaction graph to obtain a global view of all search sessions and their relations. For example, the document d_5 in Figure 2 exists in the aggregated search logs of query q_4 and query q_5, separately, while the query q_4 and the query q_5 have no direct relation of each other in the log data. On the interaction graph, we can easily observe that document d_5 has a positive neighbor q_4 and a negative neighbor q_5. Comparing the two related queries, document d_5 is more preferred in the search result of query q_4 than query q_5, which indicates that document d_5 may answer the query q_4 but not query q_5.

In the RQC task, we consider the situations that two different queries q and q' share the same candidate document d. There are three scenarios in total. (1) The document d is positive to both query q and q'. (2) The document d is positive to one query (*e.g.*q) and negative to another one (*e.g.*q'). (3) The document d is negative to both queries. Since we focus on the query-document relevance signals, we are interested in the second scenario for it provides the information to distinguish the relevance of two similar query-document pairs. To cultivate the model's capability of understanding the information needs from the query, we propose the pretraining task of relevant query comparison. In the RQC task, the score of the positive query-document pair generated by the model should be higher than the negative one, namely $S(q, d) > S(q', d)$.

More specifically, we can derive co-interaction relations on the interaction graph to distinguish which query has a closer relationship to the given document. For the document $d \in \mathcal{D}$ that has positive neighbor query $q^+ \in \mathcal{P}_d$ and negative neighbor query $q^- \in \mathcal{N}_d$, we can generate the pretraining pairs (q^+, d) and (q^-, d) for relevant query comparison task. The loss function could be calculated as follows:

$$\mathcal{L}_{\mathrm{RQC}} = \max\left(0, 1 - S(q^+, d) + S(q^-, d)\right), \tag{4}$$

where $S(q^+, d)$ and $S(q^-, d)$ can be obtained following the Equation (1). As illustrated above, the RQC task is designed for the model to discover the potential relationship between different search sessions on the interaction graph. By leveraging the contrastive pair $S(q^+, d)$ and $S(q^-, d)$, the model is driven to focus on the difference between positive query q^+ and negative query q^-.

3.4 Multi-hop Document Prediction (MDP) Task

The CDP task and RQC task deal with direct relations extracted from the interaction graph, while the MDP task and MQC task are tailed for the multi-hop query-document relations modeling. Considering that the majority of users prefer to examine a few documents, especially ranked at the top of the search results, compared with the long list of search results. Therefore, some relevant documents of the query q are not browsed by the users, and their positive signals will not be directly recorded in the single search log. The MDP task is designed to leverage multi-hop relations to measure the relevance relations of the new query-document pairs.

The key idea of the MDP task is to exploit documents from q's relevant queries to generate augmentative pretraining samples. Since the queries that share the same clicked document may have the same search intent, we sample both positive and negative documents from the queries with the same clicked document of query q. As shown on the interaction graph in Figure 2, query q_1 and q_3 are more similar for they share the same positive document d_3. Furthermore, query q_3 has a positive document d_5 and a negative document d_4 that have not been displayed or browsed in the session of query q_1. Given that the query q_3 is more relevant to document d_5 than document d_4, the q_3's similar query q_1 may have the same preference. Hence, we can derive the positive pair (q_1, d_5) and negative pair (q_1, d_4) by comprehensively considering the relations of related query sessions.

Formally, the approach of finding a relevant query for the given query q can be described as a kind of relevance propagation of the interaction graph. For the generated positive pair (q_1, d_5) in Figure 2, there is a path connected by solid arrows between them ($q_1 \rightarrow d_3 \leftarrow q_3 \rightarrow d_5$), while the path of negative pair (q_1, d_4) has dashed arrow ($q_1 \rightarrow d_3 \leftarrow q_3 \dashrightarrow d_4$). In short, we explore the multi-hop relations on the interaction graph with the assumption that the query-document relevance signals can be propagated along the solid-arrow paths without dashed arrows.

The procedure of the MDP task can be described as Algorithm 1. We explore the potential multi-hop relations of the given query q with its positive document d. If the document d is also clicked by another different query q^+, then we can leverage the positive document d_m^+ and negative document d_m^- that have not appeared in the neighbor set of query q to generate multi-hop training samples. With the multi-hop positive and negative samples (q, d_m^+) and (q, d_m^-), the loss function of the MDP task can be derived as:

$$\mathcal{L}_{\text{MDP}} = \max\left(0, 1 - S(q, d_m^+) + S(q, d_m^-)\right), \tag{5}$$

where (q, d_m^+) and (q, d_m^-) are not directly appear in the origin search result of query q, but there are still differences between these two pairs. The ranking models are expected to learn the relevance difference from these augmentative pairs in this task.

Algorithm 1 Pretraining Samples Generation of MDP task

```
Procedure MDP Samples Generation
Input: interaction graph $\mathcal{G}$, query $q$ and its positive neighbor set $\mathcal{P}_q$.
Output: MDP training pairs $\mathcal{T}_q$ generated from query $q$.
$\mathcal{T}_q \leftarrow \emptyset$
for document $d \in \mathcal{P}_q$ do
    $\mathcal{P}'_d \leftarrow \mathcal{P}_d \setminus \{q\}$ /* exclude the input query $q$ */
    for query $q^+ \in \mathcal{P}'_d$ do
        $\mathcal{P}'_{q^+} \leftarrow \mathcal{P}_{q^+} \setminus \{d\}$ /* exclude the one-hop document $d$ */
        if $|\mathcal{P}'_{q^+}| >= 1$ and $|\mathcal{N}_{q^+}| >= 1$ then
            $d_m^+ \leftarrow$ RandomSample($\mathcal{P}'_{q^+}$)
            $d_m^- \leftarrow$ RandomSample($\mathcal{N}_{q^+}$)
            $\mathcal{T}_q \leftarrow \mathcal{T}_q \cup \{(q, d_m^+, d_m^-)\}$
        end if
    end for
end for
return $\mathcal{T}_q$
```

3.5 Multi-hop Query Comparison (MQC) Task

Compared with the multi-hop document prediction (MDP) task, the multi-hop query comparison (MQC) task is designed from the perspective of documents. Considering that different users may issue similar yet distinct queries with the same search intent, document ranking models are expected to recognize queries that share the same search intent. However, due to several problems (*e.g.*document not being displayed), some relevant query-document pairs are absent in the search log. Hence, we propose the MQC pretraining task, which leverages the interaction graph to figure out the related and new q-d pairs with multi-hop relations.

The goal of the MQC task is to find similar queries that are not directly related to the given document d in the search log. Then similar queries are used for generating pretraining samples to enhance the robustness of the ranking models. To accomplish this, we leverage the document d^+ that share the same positive query with the document d. For example, the document d_3 and d_5 in Figure 2 are related for they share the same positive query q_3. Then we leverage the positive query q_1 and negative query q_2 of document d_3 to generate multi-hop pretraining samples. Specifically, (q_1, d_5) is a positive pair for the query and document between q_1 and d_5 are all positive ($q_1 \rightarrow d_3 \leftarrow q_3 \rightarrow d_5$), while (q_2, d_5) is a negative pair for query q_2 is a negative query to document d_5's related document d_3 ($q_2 \dashrightarrow d_3$). It is worth noting that (q_1, d_5) pair is positive in both examples of the MDP task and the MQC task. In the MQC task, positive and negative query-document pairs share the same document d_5, while in the MDP task, positive and negative pairs share the same query q_1 in Figure 2. The MQC task is intended for the model to distinguish positive and negative queries with multi-hop relations prediction for the given document d.

For the given document d, we try to find the positive document d^+ that shares the same positive query q. Then we randomly sample the positive query q_m^+ and negative query q_m^- from d^+'s neighbor set $\mathcal{P}'_{d^+}$ and $\mathcal{N}_{d^+}$, where $\mathcal{P}'_{d^+}$ is the positive neighbor set without the query q. Then the loss function of the MQC task can be calculated as:

$$\mathcal{L}_{\text{MQC}} = \max\left(0, 1 - S(q_m^+, d) + S(q_m^-, d)\right), \tag{6}$$

where (q_m^+, d) and (q_m^-, d) are multi-hop samples generated based on relations propagation on the interaction graph. These samples are prepared for the model to distinguish the relevant queries from irrelevant queries that do not appear in the session of document d.

3.6 Pre-taining and Fine-tuning

Apart from the four ranking tasks elaborated above, we also pretrain our model with masked language modeling (MLM) task. The key idea of the MLM task is to randomly mask some tokens in the text sequence. The pretrained model is encouraged to fill in the original token according to the remanent unmasked tokens. More specifically, we follow the setting of the BERT [7]. The whole sequence is used to randomly sample 15% masked tokens. We replace the masked token with the special "[MASK]" token for 80% of the time, a random token for 10% of the time, and the original token for 10% of the time. The MLM loss function can be defined as follows:

$$\mathcal{L}_{\text{MLM}} = -\sum_{k=1}^{M} \log p(x_k | S_d \setminus M_d), \tag{7}$$

where $M_d = \{x_1, \cdots, x_M\}$ is the tokens of document d that have been masked, $|M_d| = M$, S_d is the original token sequence of document d, and $p(x_k|S_d \backslash M_d)$ is the probability output by the pretrained model given the remanent sequence $S_d \setminus M_d$.

Our model PSLOG is pretrained with multiple tasks including ranking tasks and language modeling task. The final optimization loss function $\mathcal{L}$ is the sum of all task losses:

$$\mathcal{L} = \mathcal{L}_{\text{CDP}} + \mathcal{L}_{\text{RQC}} + \mathcal{L}_{\text{MDP}} + \mathcal{L}_{\text{MQC}} + \mathcal{L}_{\text{MLM}}. \tag{8}$$

The ranking loss consists of four losses from the ranking tasks. The CDP task is designed for the model to distinguish clicked documents in the single search session, while pretrained model needs to compare the queries from related search sessions in the RQC task. Moreover, the MDP task and MQC task are multi-hop relation prediction tasks intended to enhance the model's capability of dealing with unseen query-document pairs in the search log. Together with the MLM task, our model is pretrained to obtain both ranking capability and language modeling capability.

In the fine-tuning phase, we fine-tune our model with human-labeled relevance signals in the web search datasets, including a commercial document ranking dataset and two public web search datasets. We leverage the pairwise loss (*e.g.*hinge loss) to optimize our model, the loss is similar to the Equation (3). In the finetuning phase, the model is expected to get familiar with the new corpus and obtain a more stable document ranking capability.

4 EXPERIMENTS

4.1 Dataset and Evaluation

4.1.1 Pretraining Dataset. We pretrain our model at a large-scale search log of the commercial search engine. The search log data is collected from a three-month search log of the search engine. The total amount of the search session data is about three billion. Moreover, the ranking samples of the four ranking tasks are generated from the interaction graph built from the real search log data. More specifically, the aggregated search log data is obtained from the multiple users' behavior.

4.1.2 Fine-tuning Dataset. To evaluate the performance of our model, we fine-tune the pretrained models on three datasets.

Commercial Dataset. This dataset is a human-labeled document ranking dataset, containing 2,564 queries and 17,545 documents. The queries and their labeled documents are collected from the WeChat search engine. This dataset is intended for evaluating the model's ranking capability. The relevance label is a five-class measure that is scored from 0 to 4. The documents with label zero are irrelevant to the given query, while the ones with label four are more relevant to the query. Besides, the queries in this dataset have at least two candidate documents with different labels. The labeled document number of each query ranges from 2 to 9, and the average document number is 2.83. During the fine-tuning phase, the dataset was randomly partitioned into training, validation, and testing sets, comprising 1,864, 200, and 500 queries respectively. All the models are fine-tuned on the training set and selected according to the results of the validation set. The final evaluation and results were obtained from the testing set.

Tiangong-ST [6] is a public Chinese web search dataset. The dataset is collected from an 18-day search log of a large commercial search engine, with 2,000 human-labeled search sessions. To examine the effects of pretraining models, we leverage the human-labeled part of the dataset for fine-tuning. For the labeled sessions dealing with 610 queries, we split these sessions into training, validation, and testing sets with no shared queries. The query number of these sets are 380, 30, and 200, respectively. The average candidate document number is 9.841 for these sessions.

Sogou-QCL [39] is a Chinese document ranking dataset in IR, which contains 2,000 queries with human-assessed relevance labels. It consists of about 54 thousand queries and 9 million documents collected from the web search. It provides the relevance labels generated by the click models. We exploit the human-labeled part of this dataset for fine-tuning. Specifically, we leverage 900 queries for training and 100 queries for validation. The rest 1,000 queries are used for testing. The average candidate document number of each query is 24.95.

4.1.3 Evaluation Metrics. We adopt the widely used evaluation metrics nDCG, ERR, and MAP in document ranking. In the Sogou-QCL dataset, we use nDCG@10, ERR@10, and MAP for evaluation. For the document's relevance to be measured with multiple levels, we treat the documents with labels greater than 2 as relevant, otherwise, they are irrelevant while calculating MAP. Considering that queries from the other two fine-tuning datasets have fewer candidate documents (the average document numbers of Tiangong-ST and Commercial datasets are 9.841 and 2.83, respectively), we adjust the calculating positions of their metrics, namely calculating @5 and @1 in the Tiangong-ST dataset and Commercial dataset, respectively.

4.2 Baseline Methods

We compare our model with various kinds of methods, including traditional IR methods, pretrained language models, and pretrained ranking models.

(1) **Traditional IR Methods. BM25** [24] is a classical IR method that measures a document's relevance from the perspective of probability. It is widely used to retrieve documents in the first phase for

its efficiency. **KNRM** [34] is a neural ranking model that leverages kernel pooling to extract matching features for the query and document text. **DSSM** [11] is a representation-based learning-to-rank approach, which uses a DNN to extract semantic features of the input text for matching.

(2) **Pretrained Language Models. BERT** [7] is a representative language model pretrained with masked language modeling (MLM) task and next sentence prediction (NSP) task. BERT adopts the multi-layer bidirectional Transformer [31] encoder to model text semantic relations. Similar to BERT, **Transformer$_{\mathbf{MLM}}$** is a Transformer-based model pretrained with MLM task. pretrained with the MLM task, the Transformer encoder will be equipped with the basic language modeling capability. **ERNIE** [30] is a pretrained model that leverages knowledge to mask phases and named entities in the pretraining task, which demonstrates its capability in Chinese language modeling. **ERNIE-3.0** [29] is a pretrained model designed for the language understanding and generation task. Compared with ERNIE-1.0, ERNIE-3.0 is pretrained with more parameters.

(3) **Pretrained Ranking Models. PROP** [20] is a pretrained model tailored for ad-hoc retrieval. The key idea of PROP is to select representative words as the pseudo query from the document content using statistical methods. Following the approach of PROP, **B-PROP** [21] leverages token attentions extracted from the encoder layers of pretrained language model BERT to calculate token weights. Then the query terms are generated based on the distribution of the token weights for the given document.

4.3 Implementation Details

The implementation details of our experiments will be elaborated from three aspects: model architecture, pretraining settings, and fine-tuning settings.

4.3.1 Model Architecture. The encoder architecture we used in the experiments is the same as the base BERT [7]. The Transformer layer number is 12 with a hidden size of 768, and the feed-forward layer size is 3072. Given that the pretraining corpus is in Chinese, we adopt the basic Chinese BERT implemented by the huggingface as our baseline in the experiments. [1] The tokenizer is the same as the Chinese BERT with a vocabulary size of 21128.

4.3.2 Pretraining Settings. In the pretraining phase, we leverage the search log data collected from the commercial search engine to generate pretraining samples of four tasks elaborated in Section 3. For the large data amount of the search log, we only use part of the data to pretrain our model PSLOG. The pretraining data numbers of the CDP task, the RQC task, the MDP task, and the MQC task are 6M, 6M, 3M, and 3M samples, respectively. The total training samples of the model are 18 million samples. For a fair comparison, the baseline model Transformer$_{\mathrm{MLM}}$, PROP, and B-PROP are also pretrained with the same data amount based on the commercial corpus. For the PROP and B-PROP models, we leverage the pretraining code released by their authors. [2] Consistent with the previous studies [20, 21], we pretrain our model with a learning rate of 2e-5 and set the warm-up procedure for the first 10% steps. Besides, the batch size of PSLOG is set at 128 (same as PROP). The total training epochs are 10. Following the previous studies [4, 20, 21], the base Chinese BERT is used to initialize our model and baseline models. The pretraining task is conducted on 4 Nvidia A100-40GB GPUs.

4.3.3 Fine-tuning Settings. To comprehensively evaluate the capability of our pretrained model, we fine-tune PSLOG and other pretrained models on three different datasets Commercial, Tiangong-ST, and Sogou-QCL. Considering that BERT, ERNIE, and ERNIE-3.0 models are pretrained on a large-scale corpus different from ours, we fine-tune them based on the released checkpoints without pretraining on our corpus. For ERNIE and ERNIE-3.0, we adopt their pytorch version implements from huggingface, they can be found through the links below. [3] [4] As for traditional IR methods, we keep the same parameters set as the original paper. We use a three-layer DNN to extract semantic features with a hidden size of 128 for DSSM. The kernel number is set at 11 as the default setting of KNRM. For a fair comparison, all the baseline models are trained or fine-tuned on the same human-assessed datasets. In the fine-tuning phase, the training batch sizes are set at 16, 8, and 32 for Commercial, Tiangong-ST, and Sogou-QCL, respectively. We use the AdamW optimizer to adjust models with a learning rate of 5e-5. The final ranking results are generated by their best model according to the validation set. The overall performance of these baseline models is reported on the testing set of the datasets.

4.4 Experimental Results

The overall performance of our PSLOG model and other baseline models are shown in Table 1. PSLOG achieves better results compared with the baselines in terms of the metrics nDCG, ERR, and MAP. The counting positions of the metrics are adjusted to accommodate the different document numbers of three datasets.

(1) **PSLOG acquires the best results in the three fine-tuning datasets compared with all baseline methods in Table 1, including traditional IR methods, pretrained language models, and pretrained ranking models**. In the commercial dataset, PSLOG obtains 3.19% absolute improvement over the strongest baseline model B-PROP in terms of nDCG@1, which demonstrates the effectiveness of pretraining with search logs. Different from representative words generated by the task of the B-PROP, the queries used for pretraining PSLOG come from real search engines. Besides, given that the search log data records the user's interactions on the displayed documents, leveraging the search log for pretraining can provide the model with a human preference for different search results. This is a possible reason that PSLOG can outperform the other pretrained ranking models.

(2) In general, **the pretrained ranking models can achieve better performance than the pretrained language model in the three document ranking datasets**. Although ERNIE-3.0 acquire competitive results compared with PROP and B-PROP in Tiangong-ST dataset, it is worth noting that ERNIE-3.0 is pretrained in a larger corpus with more model parameters. This also implies that pretraining models with IR-oriented tasks can make up for some disadvantages of model size and training data amount. Moreover, we notice that the pretrained model Transformer$_{\mathrm{MLM}}$ outperforms BERT by a large margin in the commercial dataset, but

[1] https://huggingface.co/bert-base-chinese
[2] https://github.com/Albert-Ma/PROP
[3] https://huggingface.co/nghuyong/ernie-1.0-base-zh
[4] https://huggingface.co/nghuyong/ernie-3.0-base-zh

Table 1: Overall performances of all methods on three fine-tuning datasets. "†" denotes the significant improvement obtained by PSLOG from the same setting in t-test with $p < 0.05$ level. The best results are in bold and the second best results are underlined.

Method Type	Method Name	Commercial Dataset			Tiangong-ST			Sogou-QCL		
		nDCG@1	ERR	MAP	nDCG@5	ERR@5	MAP	nDCG@10	ERR@10	MAP
Traditional IR Methods	BM25	$0.3583^{\dagger}$	$0.2496^{\dagger}$	$0.3169^{\dagger}$	$0.6176^{\dagger}$	$0.3404^{\dagger}$	$0.8014^{\dagger}$	$0.4504^{\dagger}$	$0.4619^{\dagger}$	$0.6285^{\dagger}$
	KNRM	$0.3619^{\dagger}$	$0.2686^{\dagger}$	$0.3322^{\dagger}$	$0.7074^{\dagger}$	$0.3973^{\dagger}$	$0.8520^{\dagger}$	$0.5288^{\dagger}$	$0.4871^{\dagger}$	$0.6563^{\dagger}$
	DSSM	$0.3671^{\dagger}$	$0.2732^{\dagger}$	$0.3386^{\dagger}$	$0.7103^{\dagger}$	$0.4074^{\dagger}$	$0.8756^{\dagger}$	$0.5133^{\dagger}$	$0.4814^{\dagger}$	$0.6488^{\dagger}$
Pretrained Language Models	$\text{Transformer}_{\text{MLM}}$	$0.4948^{\dagger}$	$0.3103^{\dagger}$	$0.4358^{\dagger}$	$0.7192^{\dagger}$	$0.4272^{\dagger}$	$0.8811^{\dagger}$	$0.5377^{\dagger}$	$0.4926^{\dagger}$	$0.6811^{\dagger}$
	BERT	$0.4551^{\dagger}$	$0.2979^{\dagger}$	$0.4031^{\dagger}$	$0.7234^{\dagger}$	$0.4503^{\dagger}$	$0.8718^{\dagger}$	$0.5423^{\dagger}$	$0.5027^{\dagger}$	$0.6831^{\dagger}$
	ERNIE	$0.4615^{\dagger}$	$0.3044^{\dagger}$	$0.4142^{\dagger}$	$0.7371^{\dagger}$	$0.4554^{\dagger}$	0.8897	$0.5302^{\dagger}$	$0.4971^{\dagger}$	$0.6637^{\dagger}$
	ERNIE − 3.0	$0.4752^{\dagger}$	$0.3060^{\dagger}$	$0.4171^{\dagger}$	$\underline{0.7514}$	$\underline{0.4584}$	0.8890	$0.5261^{\dagger}$	$0.4937^{\dagger}$	$0.6617^{\dagger}$
Pretrained Ranking Models	PROP	$0.5393^{\dagger}$	0.3225	$0.4585^{\dagger}$	0.7491	0.4580	$\underline{0.9049}$	$0.5730^{\dagger}$	$0.5271^{\dagger}$	$0.7087^{\dagger}$
	B-PROP	$\underline{0.5421}^{\dagger}$	$\underline{0.3243}$	$\underline{0.4611}$	$0.7417^{\dagger}$	$0.4509^{\dagger}$	$0.8834^{\dagger}$	$\underline{0.5788}^{\dagger}$	$\underline{0.5306}^{\dagger}$	$\underline{0.7131}^{\dagger}$
	PSLOG (ours)	**0.5740**	**0.3305**	**0.4757**	**0.7648**	**0.4717**	**0.9084**	**0.6009**	**0.5496**	**0.7401**

gets similar results in the other two datasets. For we pretrained the $\text{Transformer}_{\text{MLM}}$ on the commercial corpus, it is reasonable that $\text{Transformer}_{\text{MLM}}$ is more familiar with the documents in commercial search scenarios. Besides, the pretrained ranking model PROP, B-PROP, and PSLOG lead the $\text{Transformer}_{\text{MLM}}$ in all datasets, validating the effects of pretraining tasks in document ranking.

(3) **The pretrained models can outperform the traditional IR methods**. Both pretrained language models and pretrained ranking models demonstrate outstanding document ranking capability on three datasets. Given that the pretrained models are developed in the large-scale corpus to obtain the basic language modeling capability, it has natural advantages compared with traditional IR models. The well-designed neural ranking models (*e.g.*KNRM) also achieve good results in the Sogou-QCL dataset. However, the pretrained models outperform them by a large margin in the other two datasets, which indicates that pretraining presents a promising approach for enhancing the document ranking performance of models. The experimental results validates our original motivation to design special pretraining tasks for retrieval. These results solidify the significance and effectiveness of pretraining as a means to improve document ranking performance.

In short, our pretrained ranking model PSLOG achieves superior results compared to baseline methods across multiple datasets. Leveraging search log data for pretraining contributes to its effectiveness. Pretrained ranking models, in general, outperform pretrained language models, showcasing the benefits of pretraining with IR-oriented tasks.

4.5 Ablation Study

To further investigate the effects of each pretraining task designed in PSLOG, we conduct an ablation experiment by removing the four pretraining tasks from the PSLOG, respectively. The ablation results in the commercial dataset are shown in Table 2. As shown in Table 2, all the pretrained ranking models obtain excellent ranking results for the whole ranking sequence of all methods reaching 0.9 in terms of nDCG, which validates the effectiveness of leveraging search logs to pretraining ranking models for web search. Besides, the metric nDCG@1 of all methods is more than half of the nDCG, which implies the majority of the relevant documents are ranked ahead of the irrelevant documents. Moreover, the removals of the CDP, RQC, MDP, and MQC tasks from PSLOG all lead to the decline of ranking performance in terms of all metrics. Compared with CDP and RQC tasks, the model PSLOG without MDP or MQC task gets lower metrics in Table 2. The possible reason is that the multi-hop relations prediction tasks can help the model discover the potential relations of the existing queries and documents. In the MQC task, the model is pretrained with positive and negative multi-hops queries for the given document d. The model is expected to focus on the different queries and figure out which one is more relevant to the document d. This may account for the PSLOG without the MQC task getting the lowest results.

Table 2: Performance of PSLOG with different tasks.

	nDCG@1	nDCG	ERR	MAP
PSLOG	**0.5740**	**0.9104**	**0.3305**	**0.4757**
w/o CDP	0.5717	0.9094	0.3295	0.4742
w/o RQC	0.5730	0.9097	0.3303	0.4756
w/o MDP	0.5670	0.9072	0.3288	0.4731
w/o MQC	0.5566	0.9001	0.3273	0.4708

4.6 Zero-shot and Few-shot Performances

Given that the pretrained models are usually fine-tuned in the different downstream datasets. We wonder how is the performance of our pretrained models. Therefore, we further investigate the zero-shot and few-shot document ranking performances of the pretrained models across three fine-tuned datasets. On the three datasets, training samples are generated from human-labeled data. Considering the different labeled document numbers of the three datasets, we adjust the query numbers to control the total number of the few-shot training samples.

The experimental results are shown in Figure 3. In general, the ranking results of the pretrained ranking models (PSLOG, B-PROP, and PROP) are higher than the pretrained language model BERT in all three datasets, which demonstrates that the knowledge learned

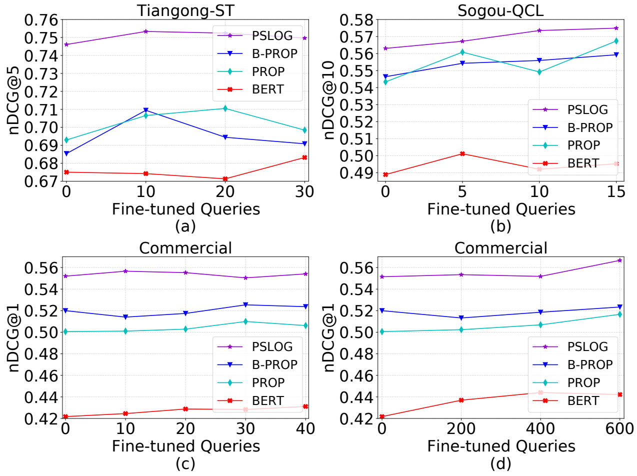

Figure 3: The zero-shot and few-shot ranking performances.

by the pretrained ranking models is effective in the different document ranking datasets. Notably, PSLOG achieved higher zero-shot ranking performance compared to PROP and B-PROP, indicating that our pretraining tasks enable the model to better comprehend the relevance signals in document ranking.

The zero-shot and few-shot ranking performance of the four pretrained models are shown in Figure 3 (a) and (b), respectively. Even in unfamiliar datasets like Tiangong-ST and Sogou-QCL, our model PSLOG can immediately improve document ranking quality only through a few queries, which demonstrates the adaptation capability of our model to different datasets. Furthermore, the higher zero-shot performance of PSLOG underscores the effectiveness of our pretraining methods in enhancing the model's understanding of relevance signals. As shown in Figure 3 (c), the pretrained models get a relatively small improvement with a few training samples. Considering that PROP, B-PROP, and PSLOG are pretrained on the large-scale commercial corpus, they are more familiar with the documents in the commercial dataset and already have good ranking performance. It is reasonable that only a small amount of training data is limited to improve the total ranking capability of a large model. Then we add more fine-tuned queries of the commercial dataset for the pretrained models and get the results as depicted in Figure 3 (d). The nDCG@1 metric of PSLOG increases as more fine-tuned samples are performed. Furthermore, we notice that PSLOG can stably get better ranking performance than PROP and B-PROP with different fine-tuned queries, which implies PSLOG can make good use of the fine-tuning data in different datasets.

4.7 User Clicks Prediction

Since our model is pretrained with the relevance signals mined from the search log data, we wonder whether PSLOG can distinguish the user's clicked document from the unclicked documents. Therefore, we randomly sample three thousand queries from the unseen search log. Within each query, a positive document and a negative document will be extracted based on their click data. The pretrained models are required to distinguish which one will obtain more clicks. Since the testing pairs are generated from the unseen search log data, the model's prediction results are their zero-shot performance on click document prediction.

Table 3: User clicks prediction results of pretrained models.

Type		PSLOG	B-PROP	PROP	BERT
Zero-shot Results	PNR	**2.3822**	1.4149	1.3346	0.9543
	ACC	**0.7043**	0.5859	0.5717	0.4883
Improvement	PNR	+**1.4279**	+0.4606	+0.3803	-
	ACC	+**0.2159**	+0.0976	+0.0834	-

The overall prediction results are shown in Table 3. We use PNR and accuracy (ACC) to evaluate the prediction performance. The PNR is obtained by dividing the right prediction number by the wrong prediction number. Compared with the baseline method BERT, PSLOG acquires more than 0.21 improvement in terms of ACC and a 1.42 increase in terms of PNR. This shows that our pretraining tasks can greatly improve the model's capability to predict user clicks. Moreover, the PSLOG's prediction accuracy of new search log data is about 0.7, which indicates our model has acquired a good knowledge of user click preferences. Together with the good ranking performance of PSLOG shown above, we believe it is beneficial of leveraging search log data to pretrain IR models.

5 CONCLUSIONS

In this paper, we propose to leverage the weak supervised signals of users in the search log data to pretrain document ranking models. To comprehensively consider the complicated relations of queries and documents in the search log, we build a global interaction graph to figure out the potential query-document relevance relations of each separate search session. Then we propose four pretraining tasks based on the relations extracted from the interaction graph. The pretraining tasks we have designed take advantage of both the direct click signals and the co-interaction relationship across different search sessions. Moreover, we utilize the multi-hop query-document relations to generate additional pretraining samples. Credit to the global perspective of the interaction graph, relations can be propagated through neighbors. Then we sample the q-d pairs with multi-hop relevance relations to cultivate the model's multi-hop relation inference capability. The experimental results on three datasets demonstrate the excellent ranking capability of our model. In the future, we plan to leverage more methods to reduce the noise in the search log and mine more reliable relations to instruct the pretraining of document ranking models.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China No. 2022ZD0120103, National Natural Science Foundation of China No. 62272467 and No. 61832017, Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098, Public Computing Cloud, Renmin University of China, and the Outstanding Innovative Talents Cultivation Funded Programs 2021 of Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods.

REFERENCES

[1] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving Web Search Ranking by Incorporating User Behavior Information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) *(SIGIR '06)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/1148170.1148177

[2] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. 2006. Learning User Interaction Models for Predicting Web Search Result Preferences. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) *(SIGIR '06)*. Association for Computing Machinery, New York, NY, USA, 3–10. https://doi.org/10.1145/1148170.1148175

[3] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rkg-mA4FDr

[4] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1524–1534. https://doi.org/10.1145/3477495.3531943

[5] Jia Chen, Jiaxin Mao, Yiqun Liu, Ziyi Ye, Weizhi Ma, Chao Wang, Min Zhang, and Shaoping Ma. 2021. A Hybrid Framework for Session Context Modeling. *ACM Trans. Inf. Syst.* 39, 3, Article 30 (may 2021), 35 pages. https://doi.org/10.1145/3448127

[6] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 2485–2488. https://doi.org/10.1145/3357384.3358158

[7] 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[8] Georges Dupret and Ciya Liao. 2010. A Model to Estimate Intrinsic Document Relevance from the Clickthrough Logs of a Web Search Engine. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining* (New York, New York, USA) *(WSDM '10)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/1718487.1718510

[9] Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4052–4059. https://doi.org/10.18653/v1/N19-1409

[10] Yu Guo, Zhengyi Ma, Jiaxin Mao, Hongjin Qian, Xinyu Zhang, Hao Jiang, Zhao Cao, and Zhicheng Dou. 2022. Webformer: Pre-training with Web Pages for Information Retrieval. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1502–1512. https://doi.org/10.1145/3477495.3532086

[11] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 2333–2338. https://doi.org/10.1145/2505515.2505665

[12] Shan Jiang, Yuening Hu, Changsung Kang, Tim Daly, Dawei Yin, Yi Chang, and Chengxiang Zhai. 2016. Learning Query and Document Relevance from a Web-Scale Click Graph. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) *(SIGIR '16)*. Association for Computing Machinery, New York, NY, USA, 185–194. https://doi.org/10.1145/2911451.2911531

[13] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Edmonton, Alberta, Canada) *(KDD '02)*. Association for Computing Machinery, New York, NY, USA, 133–142. https://doi.org/10.1145/775047.775067

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. of ICLR*.

[15] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* (1999).

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[17] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1259–1268. https://doi.org/10.1145/3404835.3462895

[18] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 3365–3375. https://doi.org/10.1145/3447548.3467149

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[20] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 283–291. https://doi.org/10.1145/3437963.3441777

[21] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: Bootstrapped Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1318–1327. https://doi.org/10.1145/3404835.3462869

[22] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 1212–1221. https://doi.org/10.1145/3459637.3482286

[23] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report.

[24] Stephen E. Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, W. Bruce Croft and C. J. van Rijsbergen (Eds.). ACM/Springer, 232–241. https://doi.org/10.1007/978-1-4471-2099-5_24

[25] Yeon Seonwoo, Sang-Woo Lee, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. 2021. Weakly Supervised Pre-Training for Multi-Hop Retriever. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 694–704. https://doi.org/10.18653/v1/2021.findings-acl.62

[26] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A Graph-to-Sequence Model for AMR-to-Text Generation. In *Proc. of ACL*.

[27] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. Modeling Intent Graph for Search Result Diversification. In *Proc. of SIGIR*.

[28] Zhan Su, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2022. Knowledge Enhanced Search Result Diversification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1687–1695. https://doi.org/10.1145/3534678.3539459

[29] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *CoRR* abs/2107.02137 (2021). arXiv:2107.02137 https://arxiv.org/abs/2107.02137

[30] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced

Representation through Knowledge Integration. *CoRR* abs/1904.09223 (2019). arXiv:1904.09223 http://arxiv.org/abs/1904.09223

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

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017). arXiv:1710.10903 http://arxiv.org/abs/1710.10903

[33] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[34] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 55–64. https://doi.org/10.1145/3077136.3080809

[35] Yijin Xiong, Yukun Feng, Hao Wu, Hidetaka Kamigaito, and Manabu Okumura. 2021. Fusing Label Embedding into BERT: An Efficient Improvement for Text Classification. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 1743–1750. https://doi.org/10.18653/v1/2021.findings-acl.152

[36] Victoria Zayats and Mari Ostendorf. 2018. Conversation Modeling on Reddit Using a Graph-Structured LSTM. *Trans. Assoc. Comput. Linguistics* (2018).

[37] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait (Eds.). ACM, 504–511. https://doi.org/10.1145/1076034.1076120

[38] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proc. of ACL*.

[39] Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-QCL: A New Dataset with Click Relevance Label. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1117–1120. https://doi.org/10.1145/3209978.3210092

[40] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model based Ranking in Baidu Search. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 4014–4022. https://doi.org/10.1145/3447548.3467147