PRADA: Pre-Train Ranking Models With Diverse Relevance Signals Mined From Search Logs

Shuting Wang, Zhicheng Dou, *Member, IEEE*, Kexiang Wang, Dehong Ma, Jun Fan, Daiting Shi, Zhicong Cheng, Simiu Gu, Dawei Yin, *Senior Member, IEEE*, and Ji-Rong Wen, *Senior Member, IEEE*

***Abstract*—Existing studies have proven that pre-trained ranking models outperform pre-trained language models when it comes to ranking tasks. To pre-train such models, researchers have utilized large-scale search logs and clicks as weak-supervised signals of query-document relevance. However, search logs are incomplete and sparse. Different users with the same intent tend to use various forms of queries. It is hard for recorded clicks to sufficiently cover diverse relevance patterns between queries and documents. Moreover, the diverse intentions of a large user base lead to long-tail distributions of search intents. Deriving sufficient relevance signals from sparse clicks of these long-tail intents poses another challenge. Therefore, there is significant potential for exploring richer relevance signals beyond direct clicks to pre-train high-quality ranking models. To tackle this problem, we develop two exploratory data augmentation strategies that consider the diversity of query forms from local and global perspectives, hence mining potential and diverse relevance signals from search logs. A generative augmentation strategy is also devised to create supplementary positive samples, to enhance the ranking ability for long-tail query intents. We leverage a multi-level pairwise ranking objective and a contrastive learning approach to enable our model to capture fine-grained relevance patterns and be robust for noisy training samples. Experimental results on a large-scale public dataset and a commercial dataset confirm that our model, namely PRADA, can yield better ranking effectiveness over existing pre-trained ranking models.**

***Index Terms*—Ranking model, data augmentation, diversity.**

I. Introduction

RECENTLY, pre-training technique has demonstrated remarkable capabilities not only in natural language processing (NLP) [1], [2], [3], [4], [5], [6], but also in information retrieval (IR) [7], [8], [9]. Previous studies [8], [9], [10], [11], [12], [13] confirm that pre-trained models tailored for IR surpass general pre-trained language models in downstream ranking tasks. These models are pre-trained with large-scale query-document relevance labels and hence capture relevance-matching signals more effectively than general pre-trained models.

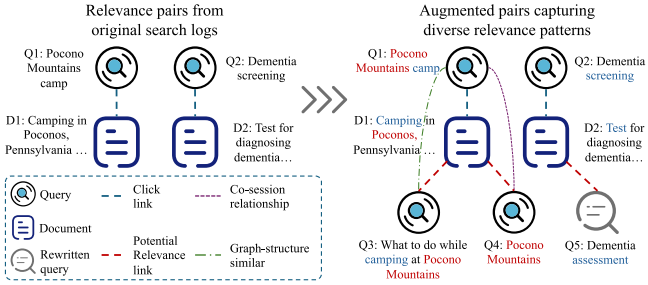

Fig. 1. Diversity of query form and query intention.

Nevertheless, acquiring sufficient human-annotated relevant query-document pairs is challenging due to expensive time and economic costs. Thus, existing studies explore alternative strategies to generate pseudo query-document pairs from the documents [8], [10] or the web structures [11], [12], [13], and pre-train the models with these weak-supervised relevance labels. Meanwhile, recent studies [14], [15] also leverage large-scale search logs to construct pre-training data since search logs cover numerous real queries, web documents, and click signals to reflect relevance to some extent.

Training a high-quality pre-trained ranking model necessitates sufficiently diverse training samples to ensure the stable capture of relevance features across varied search scenarios. Despite a wealth of click signals in search logs, the available relevance-matching patterns are still incomplete and sparse, which limits the performance of search-log-based pre-trained models. The main reasons are two-fold. One of them is the query form diversity. Many queries may contain similar intents but are phrased differently. Consequently, the click signals of these queries often scatter across different document sets, leading to incomplete coverage of diverse matching patterns between relevant queries and documents. As shown in Fig. 1, where Q_1, Q_3, and Q_4 all aim to search for information about the Pocono Mountains (camping). However, their query forms differ from each other. As a result, D_1 may not be displayed to Q_3 and Q_4, and there may be no corresponding click record in the search log. The other is query intention diversity. Various works [16], [17] have revealed that the popularity of web pages usually

Received 7 June 2024; revised 27 October 2024; accepted 25 November 2024. Date of publication 19 December 2024; date of current version 25 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272467, in part by Public Computing Cloud, Renmin University of China, and in part by the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE. Recommended for acceptance by G. Cong. *(Corresponding author: Zhicheng Dou.)*

Shuting Wang, Zhicong Cheng, and Ji-Rong Wen are with the Gaoling School of Artificial Intelligence, Remmin University of China, Beijing 100872, China (e-mail: wangshuting@ruc.edu.cn; chengzhicong01@baidu.com; jrwen@ruc.edu.cn).

Zhicheng Dou, Kexiang Wang, Dehong Ma, Jun Fan, Daiting Shi, Simiu Gu, and Dawei Yin are with Baidu Inc, Beijing 100193, China (e-mail: dou@ruc.edu.cn; wangkexiang@baidu.com; madehong@baidu.com; fanjun@baidu.com; shidaiting01@baidu.com; gusimiu@baidu.com; yindawei@acm.org).

Digital Object Identifier 10.1109/TKDE.2024.3515800

follows a power-law distribution. This implies that tremendous queries and documents are associated with a wide variety of tail intents. Typically, the click signals of these query intentions are extremely sparse, which makes it difficult for pre-trained models to maintain robustness across varied tail search intentions.

To address this issue, we introduce a pre-training method, PRADA, which **P**re-trains **RA**nking models with **D**iverse relev**A**nce signals mined from search logs by data augmentation techniques. Such an approach helps our pre-trained model stably capture accurate relevance features across diverse query forms and tail intentions, leading to improved generalization and overall performance. Concretely, to tackle the challenge of *query form diversity*, we propose exploratory augmentation strategies to uncover potential relevance signals within existing queries and documents. They involve exploring similar-intent queries and expanding positive training pairs by considering clicked documents of a query as augmented positive documents for other similar queries. We devise two exploratory strategies from different perspectives. From a local view, users typically issue a series of queries from multiple aspects to fulfill complex search intents, forming search sessions [18], [19], [20], [21]. Thus, we identify queries that frequently appear within the same session as similar-intent queries. From a global view, different users with the same search intents may issue diverse queries across different sessions. These queries often exhibit similar graph structures in the click-graph, e.g., overlapping click documents. Therefore, we introduce the global click-graph and examine similar queries based on graph similarities. By considering similar-intent queries from local and global perspectives, we can uncover potential relevance signals and augment the training data accordingly. As shown in Fig. 1, Q_1 frequently appears in the same session as Q_4, and has similar graph-structure to Q_3. Thus, they are both potentially relevant to Q_1's clicked document, D_1, and are favorable for providing more diverse relevance patterns. As for *query intention diversity*, since tail intents rarely appear in search logs, making it difficult to explore similar queries or documents, a rewriter-based generative augmentation strategy is devised to expand related pre-training samples.[1] We use a query rewrite model to take long-tail query-document click pairs as inputs and generate queries with the same search intent. An example is provided in Fig. 1, where Q_5 is the rewritten query of the Q_2. Pairing generated queries and original clicked documents allows for the augmentation of training data related to varied tail intents.

Additionally, considering the quality difference between positive documents, including clicked and augmented ones, we further assign multi-grade pseudo-relevance labels for these documents. The labels are determined by click frequencies and query similarities. With a multi-level pair-wise ranking loss function, we can prioritize more reliable relevant pairs while mitigating the impact of noisy samples on our model's performance. Furthermore, since generated queries have no candidate sets with click or non-click signals, pair-wise ranking loss functions are inapplicable to them. Thus, we design a contrastive learning optimization approach. It aligns the embeddings of original pairs and augmented pairs, and distinguishes the embeddings of positive and negative pairs. This method enables the model to capture accurate and consistent relevance features for same-intent input pairs.

We experiment with large-scale public and commercial search logs. Experimental results show that PRADA significantly outperforms existing pre-trained ranking models.

In summary, our main contributions are three-fold:

1) We develop exploratory data augmentation to dig sufficient and diverse relevance patterns within search logs, enhancing the robustness of pre-trained models across diverse query forms.
2) We devise generative data augmentation to create pre-training samples for varied tail intentions, improving the performance of pre-trained models on long-tail distributed search intents.
3) We employ a multi-level ranking objective and a contrastive learning approach to ensure the model captures reliable relevance features for various kinds of par-training samples.

II. Related Work

A. Pre-Trained Language Models

Recently, pre-trained language models (PLMs) have performed outstandingly in various NLP areas. These models usually use vast collections of web documents to acquire language modeling skills through unsupervised learning tasks. One popular PLM, BERT [2], employs Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks to pre-train a transformer encoder-based language model. It has provided superior language encoding ability for various downstream tasks [19], [20], [22], [23], [24], [25], [26], [27]. Some studies [3], [28], [29] also devised alternative pre-training tasks at both the word-level, e.g., Permutation Language Modeling [28] and Token Deletion [3], and the sentence-level, such as Sentence Order Prediction [29] and Sentence Permutation [3]. More recently, large language models (LLMs) [6], [30] have performed impressively in the NLP field, using huge model parameters to preserve extensive world knowledge. However, tremendous model parameters also impose expensive time and economic costs on the deep application of LLMs to other tasks. Thus, we focus on the comparison between PLMS, while the study of LLMs will be discussed in future work.

B. Pre-Trained Ranking Models

The goal of language models is to understand and generate texts while ranking models aim to accurately learn relevance patterns and rank search results. Such a difference limits the effectiveness of directly fine-tuning pre-trained language models for ranking tasks. Consequently, it is necessary to pre-train

[1]Note that the reason we denote the prior two strategies as "exploratory strategies" is that they focus on discovering similar-intent queries from collected large-scale search logs, which provide real user intents, to build augmentation training examples for high-frequency clicked query-document pairs. However, for long-tail query-document pairs, it is hard to apply these exploratory strategies due to limited similar intents among search logs. Thus, we apply the generative strategy to expand the corresponding training data.

ranking models specifically for IR tasks. To achieve this, some studies [7], [31] proposed extracting relevant sentence pairs from Wikipedia pages. For example, the Body First Selection task [7] views a random sentence from the first section of a Wikipedia page as the query and the remaining contents as the document to build relevant query document pairs. To improve the quality of these pairs, researchers [8], [9], [10] suggested building pre-training pairs by generating queries directly from documents using language models. Ma et al. [8], [10] identified the positive and negative queries according to their generative probabilities, and Chen et al. [9] further incorporated IR axioms to regularize the priority among generated queries. Beyond the document content, some methods, e.g., HARP [11] LOUVRE [12], and Webformer [13], also utilize the hyperlink or HTML structures to construct pre-training data and tasks. S^2phere [32] attempts to leverage open-source learning-to-rank (LTR) datasets to build data sources for pre-training high-quality ranking models. Additionally, a recent study [33] also focuses on pre-training legal search models to solve the ranking task for in-domain scenarios. Considering that search logs can naturally provide massive real search queries, web documents, and click information, recent studies [14], [15], [34] have introduced large-scale search logs to build pre-training samples using click relationships. Liu et al. [34] and Zou et al. [14] employed search logs to pre-train two-tower and cross-attention ranking models separately, where the latter further fine-tuned a denoise model to refine click signals. PSLOG [15] samples multi-hop relationships from click-graph to mine out broader query-document pairs.

However, existing search log-based pre-training rank models typically create relevant pairs from single [14], [34] or multi-hop [15] click relationships, overlooking the diversity of intent expressions and long-tail search intents. Therefore, in this paper, we adopt data augmentation strategies to discover various potential relevance patterns beyond click relationships. This approach aims to improve the generalization of pre-trained ranking models across diverse intent expressions and various long-tail intents, thereby enhancing their overall performance.

C. Query Augmentation Methods

The development of LLMs has significantly facilitated the query augmentation technique, which lightly expands training examples for information retrieval models. This approach increases both the volume and diversity of data more efficiently than human annotation and has received increasing attention in recent years. Considering the difficulty of collecting real user queries, some studies [35], [36], [37], [38], [39] leverage zero- or few-shot demonstrations to prompt LLMs to directly generate relevant queries based on given documents. It effectively augments the training query-document pairs for retrieval models. Singh et al. [40] view the generation probability of queries by LLMs, given a document, as their relevance labels, thereby simulating the relevance annotator to expand training examples. Different from the above studies that directly generate pseudo queries or relevance labels according to documents, our method emphasizes the creation of diverse patterns for the same query intents, using a real relevant query-document pair. Such an approach further ensures the reliability and authenticity of our generated queries while discovering diverse query formats, leading to facilitated ranking quality of our pre-trained ranking models.

III. Method

Existing studies have demonstrated the superiority of pre-trained ranking models over directly fine-tuning pre-trained language models for ranking tasks, particularly when using pre-training samples derived from real search logs. However, solely using click relationships is insufficient to cover the diverse and long-tail relevance patterns hidden in search logs. To address this challenge, our method adopts advanced data augmentations to uncover rich relevance signals from search logs beyond click relationships, thereby enhancing the robustness and effectiveness of pre-trained ranking models.

A. Overview

We present an overview of PRADA in Fig. 2. The middle part demonstrates the augmentation strategies. First, from the local view, we discover similar-intent queries by identifying query pairs that frequently occur in the same session. Such a strategy is called *session-based augmentation* (SEA). From the global view, queries with similar graph structures also share search intent, albeit expressed in different ways. Then, the clicked documents of these similar queries could be utilized to augment each other. We call this approach *graph-based augmentation* (GEA). After data expansion, we assign multi-grade pseudo-relevance labels to positive documents under all queries. With the multi-level pair-wise ranking loss, we can emphasize documents that are more likely to be relevant to queries and weaken the impact of noisy ones. Additionally, massive documents are tail documents that are rarely searched by users, and their click signals are very sparse. In this case, we employ a query rewriter to generate new queries with similar intents and expand training samples for these long-tail documents and queries. This method is called *rewriter-based generation augmentation*, i.e., REGA. We then use contrastive learning to capture robust and accurate relevance features from these samples. Our model is trained by aggregating the above multi-level ranking loss, the contrastive learning loss, and the original MLM loss.

B. Preliminaries

Before introducing our method, we first provide important symbol definitions. The search logs are composed of massive queries issued over some time and documents displayed to users. Considering the same query may be repeatedly issued at different timestamps, accumulating click signals at different timestamps is beneficial to recognizing reliable or noisy relevance signals. Therefore, we first aggregate the search logs based on queries. Specifically, for a recorded query q, we merge all its exposed documents to formulate its candidate document set, which is denoted as $\mathcal{D}(q)$. Each document in the candidate set has the associated click frequency, i.e., $cf(q, d) \in [0, +\infty), \forall d \in \mathcal{D}(q)$. We regard clicked documents as positive documents and unclicked

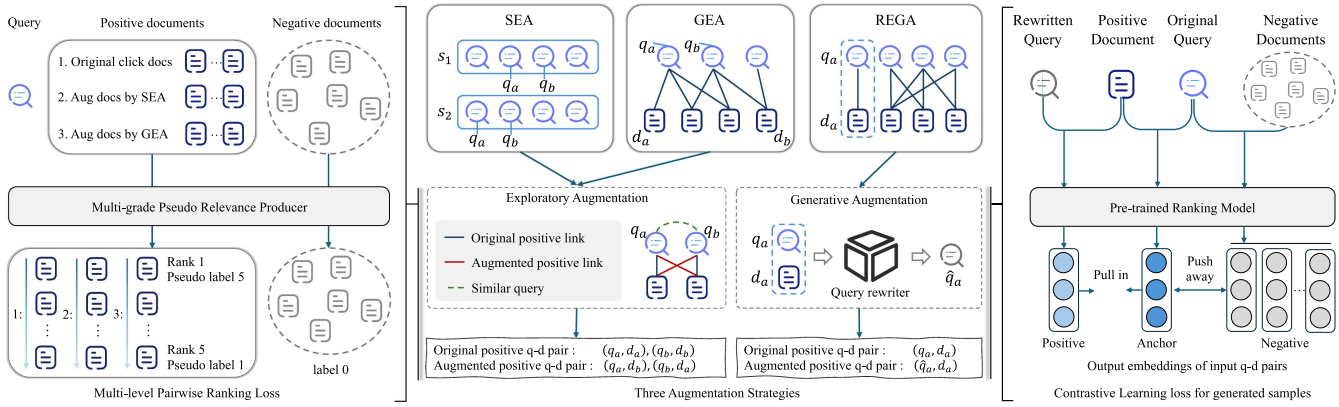

Fig. 2. The architecture of PRADA. The left part shows the process of assigning multi-grade pseudo-relevance labels, which are used to compute the multi-level pair-wise ranking loss.

documents as negative ones, which are referred to by d^+ and d^- respectively, to construct the basic pre-training samples.

C. Data Augmentation Strategies

Considering the diverse query forms of similar search intents and massive tail query intents, there are various uncovered relevance patterns in extensive search logs beyond one- or multi-hop clicks. To tackle this issue, we design the following three augmentation strategies: (1) session-based exploratory augmentation (SEA), (2) graph-based exploratory augmentation (GEA), and (3) rewriter-based generative augmentation (REGA) to sufficiently investigate possible relevance signals within search logs.

1) Session-Based Exploratory Augmentation: With the growth of users' information needs, their search intents are becoming increasingly complex and challenging to satisfy with a single query. Therefore, users prefer to yield a series of queries that cover the various aspects of their complex intents, which constitute search sessions [19], [20], [41]. As a result, we can observe that queries frequently co-appearing in sessions usually share similar search intents. This observation implies that among these similar queries, the documents clicked under one query may also be relevant to another one to some extent, thereby yielding more sufficient and diverse positive pre-training pairs. Specifically, we first count the co-session frequencies of all query pairs and normalize them as similarities between query pairs, i.e.,

$$w(q_j|q_i) = \frac{sf(q_i, q_j)}{\sum_{q_k \in \mathcal{S}(q_i)} sf(q_i, q_k)}. \tag{1}$$

Here $w(q_j|q_i)$ denotes the similarity of the query q_j with the query q_i, $sf(q_i, q_j)$ represents the co-session frequency of two queries, and $\mathcal{S}(q_i)$ is the set of queries that have appeared in the same session with q_i. Then, we regard the clicked document, d_j^+ of q_j that has not been clicked under q_i as the pseudo-relevant document of q_i. The pseudo-relevance degree can be measured as follows:

$$rd^{\text{SEA}}\left(d_j^+, q_i\right) = w(q_j|q_i) \cdot cf\left(q_j, d_j^+\right), \tag{2}$$

where $cf(\cdot,\cdot)$ is the click frequency, and the superscript "SEA" indicates that the pseudo-relevance degree is computed by the SEA strategy.

To ensure the quality of augmented positive documents, we filter the top-k augmented documents for the query q_i according to the above pseudo-relevance degree. We denote the query q_i's augmented documents based on the SEA approach with $\mathcal{D}^{\text{SEA}}(q_i)$. These documents will be further assigned multi-grade pseudo-relevance labels, which will be introduced in Section II-I-D.

2) Graph-Based Exploratory Augmentation: Apart from the local perspective of sessions, different users may issue different queries when searching for the same information. These similar queries typically come from different sessions. Consequently, to excavate such similar queries, we propose leveraging the global view of the click graph that is constructed from the whole search logs. We believe that two queries with similar search intents often exhibit similar graph structures, e.g., sharing similar co-clicked documents. According to this clue, we optimize a graph neural network based on the click graph using the GraphSage algorithm [42]. Subsequently, we identify similar queries based on the embedding similarities of their corresponding nodes in the graph.

$$w\left(q_j|q_i\right) = \text{GEmb}(q_i)^T \cdot \text{GEmb}(q_j), \tag{3}$$

where $\text{GEmb}(\cdot)$ denotes the graph embedding of the node. Since graph nodes are queries and documents, we initialize graph node embeddings by their text embeddings from a BERT model. This operation can integrate the semantic matching and graph structure matching capabilities simultaneously.

Similar to (2), we calculate the pseudo-relevance degree between the q_j's clicked documents, d_j^+ and q_i, i.e., $rd^{\text{GEA}}(d_j^+, q_i)$. By selecting the top-k documents based on pseudo-relevance degrees, we acquire the q_i's augmented positive document set of the GEA method, denoted with $\mathcal{D}^{\text{GEA}}(q_i)$

3) Rewriter-Based Generative Augmentation: The above ways simulate situations where the same search intent leads to different queries. These search intents are typically popular and produce various related queries from many users. However, it is hard to explore this kind of similar queries for various tail

intents because they are niche and rarely appear in search logs. To tackle this issue, we adopt a query rewrite model to generate supplementary queries with the same intents for long-tail query-document pairs.[2] This can enrich the training data related to long-tail intents, and finally improve the generalization of our ranking model on them.

• *Query generation for long-tail query-document pairs:* Given the entire click-graph, we simply select click query-document pairs (q, d) where both the degree of query and document, and the click frequency are lower than a threshold k as long-tail intents. k can be empirically decided considering the time range and data scale of the log. We then leverage a query rewrite model, $\mathrm{QRW}(\cdot)$, which is fine-tuned from a T5 model to generate additional queries for (q, d). The target of the rewriter is to generate an equivalent query $\hat{q}$ that shares the same intent as the original query q and is relevant to the clicked document d. The generated query $\hat{q}$ and the clicked document d form a new positive training sample.

• *Training of query rewriter:* Consequently, frequently co-clicked query pairs and their same clicked documents can be naturally used to build training samples for optimizing the query rewriter. Specifically, we create a piece of data (q_a, q_b, d) if q_a and q_b contain the same clicked document d in the log. The co-click sample is reserved only if the click frequencies of both queries on d are higher than 3 to reduce noise. Furthermore, since the navigational queries [43], [44] usually target specific websites without complex search intents, these queries are useless for us to capture users' diverse intent representations. Following previous studies [43], [44], we simply regard the queries whose click entropy is lower than 1 as navigational queries and remove them. We then use both "$prompt(q_a, d) \rightarrow q_b$" and "$prompt(q_b, d) \rightarrow q_a$" as training sequences to fine-tune the T5 model. The prompt content is shown in Section IV-C. In the generation stage, we feed "$prompt(q, d) \rightarrow$" to the model and get its output sequence as the rewrite query $\hat{q}$.

D. Multi-Grade Relevance Label Assignment

Previously, we introduced augmentation strategies to explore rich relevance signals beyond click-through. Furthermore, the reliability of these positive documents is also different, which can be reflected by click frequencies. Under the same query, documents with fewer click times compared to other frequently clicked documents should be paid less attention when pre-training ranking models to avoid the interference of noisy clicks. Additionally, the scale of click frequency under different queries may also be different. Some queries are popular, hence receiving more click signals, while niche queries usually receive fewer clicks. Thus, we set pseudo-relevance labels (5-scale from 1 to 5) for these clicked documents to distinguish their importance under the same query.

Specifically, for a query q, we first order its clicked documents based on click frequencies and acquire their rank position $pos(d|q)$, which starts at 0.[3] Then, the higher-ranked documents (smaller rank position) are assigned higher pseudo-relevance labels, $r^{\mathrm{C}}(d)$, by:

$$r^{\mathrm{C}}(d) = \max\left(5 - pos(d|q), 1\right), \quad \forall d \in \mathcal{D}^{\mathrm{C}}(q), \tag{4}$$

where $\mathcal{D}^{\mathrm{C}}(q)$ denotes the clicked document set of q.

In addition to the original clicked documents, i.e., $\mathcal{D}^{\mathrm{C}}$, there are other two types of augmented positive documents from our SEA and GEA strategies, i.e., $\mathcal{D}^{\mathrm{SEA}}$ and $\mathcal{D}^{\mathrm{GEA}}$. Their pseudo-relevance degrees calculated by (2) can also be utilized to assign relevance labels. After similar rank-then-assign steps, i.e., (4), we acquire the pseudo-relevance labels for these augmented documents, i.e., $r^{\mathrm{SEA}}(d), \forall d \in \mathcal{D}^{\mathrm{SEA}}$ and $r^{\mathrm{GEA}}(d), \forall d \in \mathcal{D}^{\mathrm{GEA}}$. Furthermore, it is difficult to ensure the priority among these three types of documents since they simulate different situations. Therefore, we separately assign relevance labels inner the same document type and do not compare the document relevance among different types. These three types of positive documents share the same negative document set. To ensure our model's ability to recognize simple negative documents, we further introduce some in-batch negatives for each query. As a result, negative documents for a query consist of unclicked documents that are not in augmented sets and in-batch negative documents. We assign their relevance labels to zero:

$$\begin{aligned} &r^{x}(d) = 0, \quad \forall d \in \mathcal{D}^{\mathrm{N}}(q), \quad \forall x \in [\mathrm{C}, \mathrm{SEA}, \mathrm{GEA}], \\ &\mathcal{D}^{\mathrm{N}}(q) = \left(\mathcal{D}^{\mathrm{UC}}(q) \cup \mathcal{D}^{\mathrm{IN}}(q)\right) \setminus \left(\mathcal{D}^{\mathrm{SEA}}(q) \cup \mathcal{D}^{\mathrm{GEA}}(q)\right), \end{aligned} \tag{5}$$

where $\mathcal{D}^{\mathrm{UC}}(q)$ denotes all unclicked documents that have been displayed under the query q. It is collected by the search engine automatically. $\mathcal{D}^{\mathrm{IN}}(q)$ represents the in-batch negatives of the query q. We use $\mathcal{D}^{\mathrm{N}}(q)$ to represent all negative documents of q.

E. Pre-Training

With pseudo-relevance labels for different types of positive documents, we can employ a multi-level pair-wise ranking loss function to capture fine-grained relevance knowledge from our training samples. Given a query-document pair, (q, d) that needs to predict relevance score, we formulate the input, I, following [22], [24], as follows.

$$\mathrm{I} = [[\mathrm{CLS}]; q; [\mathrm{SEP}]; d; [\mathrm{SEP}]]. \tag{6}$$

Subsequently, we view the output embedding of [CLS], $\mathrm{Emb}_{[\mathrm{CLS}]}$, as the relevance features. The predicted score of this pair, $s(d)$ is produced by applying an MLP layer, $f(\cdot)$ on the [CLS]'s embedding:

$$s(d) = f\left(\mathrm{Emb}_{[\mathrm{CLS}]}\right). \tag{7}$$

• *Multi-level pair-wise ranking loss:* Given a query q and its candidates of different relevance grades and types, we compute the multi-level pair-wise ranking loss as below:

$$\begin{aligned} &\mathcal{L}_{rk}(q) = \mathcal{L}^{\mathrm{C}}_{rk}(q) + \mathcal{L}^{\mathrm{SEA}}_{rk}(q) + \mathcal{L}^{\mathrm{GEA}}_{rk}(q), \\ &\mathcal{L}^{x}_{rk}(q) = \sum_{r^{x}(d_1) > r^{x}(d_2)} \left(r^{x}(d_1) - r^{x}(d_2)\right)\left(s(d_2) - s(d_1) + \gamma\right). \end{aligned} \tag{8}$$

[2]Although this generation strategy can also be used for popular intents, we prioritize it for long-tail intents to control the entire computational costs.

[3]Two documents with the same click frequencies have the same position.

$\mathcal{L}_{rk}(q)$ is the final ranking loss, $\mathcal{L}_{rk}^{x}$ denotes the ranking loss for the documents of type x, and the hyperparameter γ is the margin of hinge loss. This function enables the model to produce higher scores for more reliable positive documents than noisy positive documents, hence alleviating the interference of noisy clicks.

• *Contrastive learning for generated samples:* The queries generated from the query rewriter do not have candidate sets and are not suitable for using pair-wise ranking loss functions. To capture accurate and fine-grained relevance from generated training samples, we employ a contrastive learning method. Given a long-tail query-document pair (q, d^+) and their generated query $\hat{q}$, since the two input samples, (q, d^+) and $(\hat{q}, d^+)$, contain the same search intent, we view them as a positive pair, hoping to maximize the similarity between their relevance features. The negative samples are built by pairing the original query with its negative documents. Consequently, the contrastive learning loss is calculated as below (E() is an abbreviation of Emb()):

$$\mathcal{L}_{cl} = -\frac{\phi\left(\mathrm{E}_{[\mathrm{CLS}]}(q, d^+), \mathrm{E}_{[\mathrm{CLS}]}(\hat{q}, d^+)\right)}{\sum_{d^- \in \mathcal{D}^{\mathrm{N}}(q)} \phi\left(\mathrm{E}_{[\mathrm{CLS}]}(q, d^+), \mathrm{E}_{[\mathrm{CLS}]}(q, d^-)\right)}. \quad (9)$$

$\phi(\cdot)$ denotes the similarity function, here we use dot similarity to implement it.

• *The final loss:* To ensure the semantic modeling capability of our pre-trained model, we also use the masked language model (MLM) objective to optimize it [8], [15]. Consequently, the final loss function of our pre-training process can be formulated as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{rk} + \lambda_2 \mathcal{L}_{cl} + \lambda_3 \mathcal{L}_{MLM}, \quad (10)$$

where λ_i denotes the weights of different loss functions.

F. Ranking Model Fine-Tuning

Given the fine-tuning data, suppose $r(d)$ is the human-annotated relevance label of the document d under the query q. We fine-tune our model via the following pairwise loss:

$$\mathcal{L}_{ft}(q) = \sum_{r(d_1) > r(d_2)} \left(r(d_1) - r(d_2)\right)\left(s(d_2) - s(d_1) + \gamma\right). \quad (11)$$

IV. Experiment

A. Dataset and Evaluation Metrics

1) Datasets. Public dataset: The Baidu-ULTR dataset [45] is a publicly available pre-training dataset that is collected from the large-scale Baidu search engine. It is composed of a pre-training dataset and a fine-tuning dataset. 1) The pre-training dataset covers 383,429,526 issued queries sampled from the search logs of Baidu in April 2022. Each query has the documents displayed to the user as its candidate documents. 98.9% of issued queries contain no more than 10 displayed documents. Text features of documents consist of titles and abstracts. For the protection of user privacy, all text contents, including query texts, titles, and abstracts, are anonymized and represented by token IDs. The dataset provides no session information or issued timestamps, and it only has an aggregated reformulated query list for each unique query. Therefore, we directly view queries and their reformulated queries as co-session query pairs. 2) The fine-tuning dataset is also collected from search logs, which is similar to the pre-training dataset. It includes 7,008 queries, where 90% of queries have more than 50 candidate documents. Each query and document pair is annotated with a 5-scale (0-4) relevance label by expert annotators. To facilitate the analysis of long-tail queries, the dataset splits these queries into ten buckets based on search frequencies. The 0 bucket means the highest frequency and the 10 bucket means the lowest frequency. Considering the limited amount of the Baidu-ULTR fine-tuning dataset, we adopt 5-fold cross-validation rather than splitting training, validation, and test datasets when experimenting with it.

Commercial Settings: To validate the usability of our method in a real product setting, we further experiment with data from a commercial search engine. The pre-training dataset was sampled from the search logs in February 2023 and it contains 626,763,255 queries. Similarly, we record the documents displayed to users as the queries' candidate documents and their click information. Thus, each piece of the search log contains the issued query text, its candidate documents with titles and abstracts, the click-through of candidate documents, and the session ID of this search record. We leverage a large commercial ranking dataset containing more than 88 K queries with 5-scale expert-annotated relevance labels as our fine-tuning dataset. Specifically, we split the training, validation, and test sets in an 8:1:1 ratio, and filter out duplicate queries between them to avoid data leakage.

The statistical information of pre-training and fine-tuning datasets of both datasets is presented in Tables III and IV.

2) Evaluation Metrics: We leverage pair-wise and list-wise metrics to assess ranking performance. The Positive-Negative Ratio (PNR) [14] is a commonly used pair-wise ranking metric. It measures the ratio of consistent document pairs (where the predicted scores align with the relevance labels) to inconsistent pairs. For list-wise evaluation, since all our fine-tuning datasets record 5-scale relevance labels, we select the NDCG@K metric, which considers both ranking position and relevance scale. Concretely, we set $K = 1, 3, 5, 10$ to evaluate different top results. All NDCG values are calculated by TREC's official evaluation tool (trec_eval) [46].

B. Baselines

To validate the effectiveness of our proposed model, we select three types of baselines, including traditional IR models, pre-trained language models, and pre-trained ranking models.

1) *Traditional IR models:* **BM25** [47] is the most widely used unsupervised IR model. **K-NRM** [48] is a widely-used ad-hoc ranking model that uses Gaussian kernels to learn relevance features from word-level matching maps between queries and documents.

2) *Pre-trained language models:* **BERT** [2], a widely used PLM that is pre-trained via MLM and NSP tasks. **ERNIE** [49] leverages knowledge graph to pre-train model and provides promising performance in various NLP tasks. **ERNIE 3.0** [50]

is the latest public version of ERNIE that introduces more parameters with a continual multi-paradigm unified pre-training framework to achieve better performance. The base versions of these PLMs have comparable model structures and sizes with our pre-trained model, hence facilitating a fair comparison.

3) *Pre-trained ranking models:* **PROP** [8] and **BPROP** [10] adopt statistical and pre-trained language models to generate relevant queries from documents, hence pre-training ranking models.[4] We further chose two recent search-log-based pre-trained ranking models as our baselines. **Pyramid-ERNIE** [14] (PRE for short) uses a denoise model to recognize the relevance levels of click signals and pre-train ranking models based on search logs. **PSLOG** [15] samples first- and second-order query-document pairs from the click graph to pre-train ranking models.

To ensure the fairness of our comparison, we set the same model structures and parameter amounts for all pre-trained models following the BERT-base model. Furthermore, since data augmentation steps can be viewed as the pre-processing of training data construction, the computational cost of these steps will not impact the training and inference efficiency of our applied ranking model, which is comparable with all baselines due to the same model structures.

C. Implementation Details

1) Data Augmentation: We first demonstrate detailed operations for our three data augmentation strategies. 1) For the SEA strategy, we filter out the query pairs with co-session frequencies lower than 2 and select up to ten augmented positive documents for each query. 2) For the GEA strategy, considering the huge amount of graph nodes, we select PGLbox [51], a GPU-based hyper-scale graph model training engine, to optimize the GraphSage model. Specifically, the learning rate is 0.05, the batch size is 32, the max step is 400,000, the number of sampled neighbors is 5, and the optimizer is Adam [52]. Embeddings of graph nodes are initialized from the BERT-small model where the embedding size is 128. When augmenting pre-training samples, we consider the query pairs with graph similarities surpassing 0.95 as valid similar queries and augment at most ten positive documents for each query. 3) For the REGA strategy, we randomly select 100 M co-click query-document samples and split them into training, validation, and test sets by 6:2:2 ratio. The content of the prompt is "*Issued query: {#query}; the title of its clicked document: {#title}; the abstract of the clicked document:{#abstract}, rewrite the query to express a similar search intent that matches this document:*". Then, we fine-tune a T5-base model by 1e-3 learning rate, 24 batch size, and 5 epochs. The optimizer is Adam [52]. Note that for the public dataset without available pre-trained T5 models, the T5 model parameters are then initialized. Since the count of long-tail queries is massive, to trade-off the effectiveness and efficiency, we set $k_1 = k_2 = k_3 = 1$ to select long-tail q-d click pairs that are the most niche and generate one rewritten query for each pair. Finally, the amounts of original clicked q-d pairs and augmented q-d pairs from SEA, GEA, and REGA strategies are 500M, 10M, 30M, and 60M respectively. For the commercial dataset, the amounts are 800M, 20M, 24M, and 100M respectively.

2) Pre-Training Stage: Following previous studies [2], [8], [10], [15], we base transformer-encoder structures to build our pre-trained ranking model. Concretely, the number of transformer layers and multi-head is 12, the embedding size is 786, and the feed-forward hidden size is 3072. For a fair comparison, our model and baselines belonging to pre-trained ranking models are pre-trained from scratch based on their own pre-training ranking samples rather than continuing training based on some pre-trained language models. For models that need to be pre-trained, we use the same pre-training settings. Specifically, the maximum learning rate is 1e-4 with 1000 warmup steps and linearly decay during optimization, the max step is 200,000, the batch size is 1600, and the optimizer is Adam [52]. The margin, γ, of the pair-wise ranking loss function, $\mathcal{L}_{rk}$, is set as 0.1. Our preliminary studies uncover that the loss scales of our three pre-training optimization objections are different. To balance their contribution for the final loss function, we set their weights as $\lambda_1 = 1, \lambda_2 = 0.1, \lambda_3 = 1$, which can ensure the scales of these three loss functions are on the same magnitude order. Note that because the public dataset is anonymized and has no available pre-trained language models, we pre-train the BERT model based on the document corpus from scratch when experimenting on the public dataset. Since we cannot identify punctuations, for the NSP task, we randomly select start and endpoints to sample sentences. However, ERNIE models are pre-trained from knowledge-driven tasks, which is hard to complement in the anonymized dataset. Thus, there are no corresponding experiment results in the public dataset.

3) Fine-Tuning Stage: After pre-training, we further fine-tune pre-trained models and baselines on the two datasets to compare their ranking performance. We also adopt the same optimization settings for all models to ensure fair comparison. For the public dataset, we leverage 5-fold cross-validation to acquire reliable results due to its limited fine-tuning data amount. We set the learning rate as 1e-5, which linearly decays during fine-tuning, the max step as 8,000, and the batch size as 256. We also utilize the Adam algorithm to optimize models. For the commercial dataset, the learning rate is set as 1e-5 with linear decay, the max step is 8,000, the batch size is 1024 and the optimizer is Adam. The margin, γ, is set as 0.3. For the K-NRM model, we set the kernel number as 11 following existing studies [15], and set the learning rate as 1e-3 since it is not the pre-trained model.

• *Analysis of time consumption:* The computational cost of our methods mainly comes from two sources, i.e., the construction of pre-training instances and the pre-training of the ranking model. To reduce the time consumption for each data augmentation strategy, we utilize some high-efficiency toolkits to implement them. As we have shown before, for SEA, we employ the SQL server to organize complex session relationships in our large-scale search logs and use SQL statements to quickly retrieve co-session queries, thereby building augmented relevant q-d pairs. For GEA, we adopt PGLbox [51] to implement the GraphSage

[4]Some methods [9], [11], [13] also generate queries from documents beyond text contents, while they are inapplicable to the anonymized public dataset. Given our focus on pre-training ranking models via search logs, we select these two representative baselines.

TABLE I
OVERALL COMPARISON OF ALL MODELS ON THE PUBLIC DATASET

Models		Public Dataset				
		NDCG@1	NDCG@3	NDCG@5	NDCG@10	PNR
Traditional IR models	BM25	0.4666	0.4828	0.4942	0.5186	2.107
	K-NRM	0.4760	0.4910	0.5023	0.5252	2.116
Pre-trained language models	BERT	0.5328	0.5435	0.5534	0.5745	2.826
Pre-trained ranking models	PROP	0.5350	0.5462	0.5559	0.5780	2.851
	BPROP	0.5341	0.5438	0.5544	0.5751	2.925
	PRE	0.5515	0.5637	0.5731	0.5940	3.173
	PSLOG	0.5547	0.5640	0.5735	0.5940	3.132
	PRADA	**0.5884**‡	**0.5951**‡	**0.6034**‡	**0.6229**‡	**3.647**‡

We keep the four significant numbers to show the results. The symbol ‡ indicates that our model outperforms all baselines significantly in a paired t-test at the $p < 0.01$ level (with Bonferroni correction). The best results are highlighted in bold.

TABLE II
OVERALL COMPARISON OF ALL MODELS ON THE COMMERCIAL DATASET

Models		Commercial Dataset				
		NDCG@1	NDCG@3	NDCG@5	NDCG@10	PNR
Traditional IR models	BM25	0.5941	0.6237	0.6560	0.7244	1.476
	K-NRM	0.5982	0.6287	0.6580	0.7253	1.526
Pre-trained language models	BERT	0.6578	0.6805	0.7048	0.7603	2.012
	ERNIE	0.6741	0.6944	0.7174	0.7701	2.174
	ERNIE-3.0	0.6744	0.6951	0.7191	0.7715	2.194
Pre-trained ranking models	PROP	0.6741	0.6897	0.7145	0.7681	2.210
	BPROP	0.6711	0.6929	0.7157	0.7687	2.218
	PRE	0.6941	0.7136	0.7335	0.7831	2.497
	PSLOG	0.7010	0.7199	0.7406	0.7887	2.629
	PRADA	**0.7150**‡	**0.7294**‡	**0.7500**‡	**0.7948**‡	**2.743**‡

TABLE III
THE STATISTICAL INFORMATION OF PRE-TRAINING DATASETS

Item	Public	Commercial
# Query	383,429,526	626,763,255
# Document	1,287,710,306	3,232,513,622
# Candidate per query	13.05	17.82
# Click Candidate per query	2.35	2.53

algorithm on massive graph nodes, efficiently improving the optimization speed. The query rewrite model of the REGA method is initialized by a T5-based [53] model, which requires few computational costs for training and inference. In practice, data augmentation strategies' time consumption is less than 24 hours. Therefore, the time cost is led by model pre-training, whose time consumption is directly proportional to the pre-training q-d pairs. According to our implementation, the augmented pre-training pairs are around 20% of the original pairs. Thus, the time consumption is also improved by around 20%.

D. Overall Results

We provide the overall results of baselines on the two datasets in Tables I and II. From the comparison between our model and baselines, we derive the following analysis.

1) *Compared with all baselines, our model PRADA demonstrates the best ranking performance:* We observe that our model outperforms the existing models significantly with the t-test in $p < 0.01$ level, especially for the pre-trained ranking models. This result proves the assumption that there are various relevance patterns underlying search logs, hence, simply regarding click relationships as relevance signals is not enough to mine sufficiently diverse relevance patterns. Our method leveraging three types of data augmentation strategies from different perspectives is able to discover or create more general and diverse relevance patterns underlying search logs. Thus, our pre-trained models can robustly learn accurate relevance signals across varied search scenarios.

2) *Compared with pre-trained language models, pre-trained ranking models provide higher-quality ranking results:* It is obvious that pre-trained ranking models, e.g., PSLOG and PRADA, outperforms pre-trained language models, e.g., Bert and ERNIE, in general. It is consistent with our expectations since the data format of pre-trained language models is usually pure texts rather than the ranking data format, i.e., query and document pairs. Moreover, their pre-training objectives are not specific to IR tasks. Such different data distributions and learning tasks pose challenges to generalizing PLMs to ranking tasks. This phenomenon confirms the importance of pre-training models specific to ranking tasks.

3) We notice that the performance improvement on the commercial dataset is limited compared to the public dataset. This could be attributed to the fact that the commercial fine-tuning dataset contains more documents with high-relevance labels than the public fine-tuning dataset. This phenomenon can be

TABLE IV
THE STATISTICAL INFORMATION OF FINE-TUNING DATASETS

Item	Public Dataset	Commercial Dataset		
		Train	Valid	Test
# Query	7,008	73,411	7,901	7,270
# Document	381,599	1,631,002	79,173	84,215
# Rel(Q-D)=0	219,305	635,637	32,637	30,088
# Rel(Q-D)=1	36,622	150,632	5,302	8,834
# Rel(Q-D)=2	112,759	117,294	1,663	6,427
# Rel(Q-D)=3	28,172	461,204	27,544	19,021
# Rel(Q-D)=4	714	439,687	13,403	20,352

Rel(Q-D)=*l* denotes the number of query document pairs with relevance label = *l*.

TABLE V
ABLATION RESULTS OF PRADA ON THE PUBLIC DATASET

Models	Public Dataset				
	N@1	N@3	N@5	N@10	PNR
PRADA	**0.5884**	**0.5951**	**0.6034**	**0.6229**	**3.647**
w/o SEA	0.5806	0.5906	0.5996	0.6201	3.609
w/o GEA	0.5693	0.5842	0.5929	0.6153	3.575
w/o REGA	0.5818	0.5903	0.5986	0.6184	3.572
w/o MLoss	0.5732	0.5845	0.5936	0.6138	3.504
w/o IB	0.5787	0.5912	0.5991	0.6193	3.537

"N" is short for "NDCG".

TABLE VI
ABLATION RESULTS ON THE COMMERCIAL DATASET

Models	Commercial Dataset				
	N@1	N@3	N@5	N@10	PNR
PRADA	**0.7150**	**0.7294**	**0.7500**	**0.7948**	**2.743**
w/o SEA	0.7067	0.7238	0.7445	0.7910	2.642
w/o GEA	0.7101	0.7263	0.7462	0.7929	2.677
w/o REGA	0.7089	0.7229	0.7447	0.7910	2.618
w/o MLoss	0.7090	0.7251	0.7446	0.7915	2.660
w/o IB	0.7085	0.7234	0.7449	0.7917	2.694

found in Table IV. Such a data distribution requires ranking models to differentiate the relevance grades among documents with high-relevance labels, which is more difficult than distinguishing simple negative samples. *Nevertheless, our model still surpasses all baselines significantly in the commercial dataset,* which further verifies the ranking and generalization abilities of our proposed method.

E. Ablation Study

To test the effects of our modules respectively, we further conduct the following ablation studies and provide corresponding analyses. The experimental results are shown in Tables V and VI.

1) *Ablation of augmentation strategies:* To verify the importance of our data augmentation strategies, we conduct corresponding ablation studies that remove one of them separately, leading to three model variants, namely, w/o SEA, w/o GEA, and w/o REGA. We observe that dropping each augmentation strategy will hurt the model performance on both datasets. These results further prove that the expressions and distribution of user intents are diverse across the web. Therefore, click signals in search logs only cover limited relevance patterns. Our augmentation strategies specifically devised for these problems can reveal more diverse relevance signals, hence improving the ranking performance and generalization of our model.

2) *Ablation of multi-level ranking loss:* Considering the different reliabilities of clicked and augmented documents, we assign multi-grade pseudo-relevance labels for positive documents and leverage the multi-level hinge loss function to optimize our ranking model. To verify its effectiveness, we use two-level pseudo-relevance labels, i.e., 0 and 1, to compute the hinge loss, leading to a variant, w/o MLoss. It is noticeable that the ranking quality declines significantly. This result proves that positive signals may also contain noise. Consequently, it is important to discriminate the different relevance levels of positive samples that can enable the ranking model to focus on highly reliable relevance signals.

3) *Ablation of in-batch negatives:* Except for unclicked documents, we also introduce documents under other queries in the same batch as simple negatives. To test the role of this operation, we devise a variant, w/o in-batch that abandons IB negative samples. According to the result shown in Tables V and VI, we find that without in-batch negative samples, the variant performs worse ranking quality than PRADA. The main reason is that queries' candidates of our datasets are the documents displayed to users, these documents all contain relevance to the issued queries to some extent, referring to hard negative documents. However, there may also exist some easy negative samples for ranking models in real ranking scenarios. Thus, identifying simple negative documents is also important for models to provide high-quality ranking results.

F. Effectiveness on Long-Tail Queries

To assess the performance of our proposed method on long-tail queries, we conduct corresponding experiments on the public dataset, which classifies fine-tuning queries into ten buckets based on their search frequencies. According to the original paper [45], buckets 0, 1, and 2 include hot queries, buckets 3, 4, 5, and 6 contain queries with medium frequencies, and buckets 7, 8, and 9 include tail queries. We demonstrate the performance of our model and several strong baselines on these three subsets in Fig. 3 and analyze their effectiveness across different popularity levels of queries.

To save space, we select NDCG@1 and @3 to demonstrate the comparison results, the remaining metrics show similar trends. Based on the results, we find that PSLOG and PRADA that leverage relevance signals beyond one-hop clicks provide better performance on unpopular queries, including medium and tail queries. It proves that relying solely on one-hop click signals may further limit the generalization and performance of pre-trained ranking models. Furthermore, PRADA delivers the best results on all subsets, especially for tail queries. We believe the main reason is that our augmentations strategies

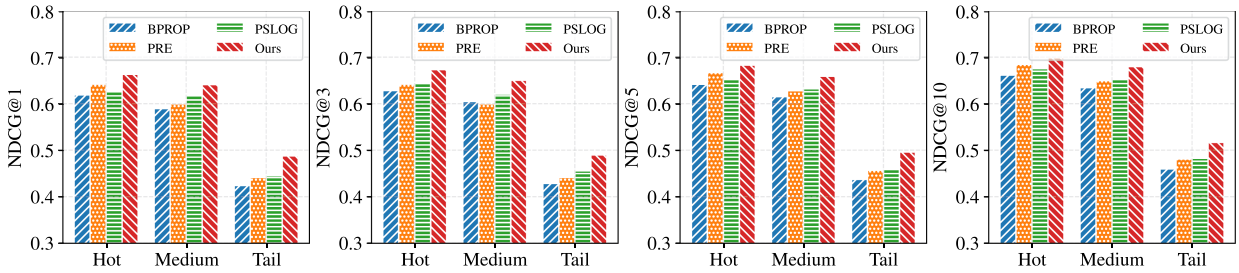

Fig. 3. The experimental results of test subsets with different search frequencies.

TABLE VII
SOME AUGMENTATION SAMPLES

Strategy	Original Query	Similar Query	Augmented Positive Document
SEA	us news World University Rankings by subject	us news World University Rankings	The 2023 US News World University Rankings are out...
GEA	How many minutes of egg custard?	How many minutes is best to steam the egg custard	Teach you the correct way to steam egg custard. Remember these skills, tender smooth...
Strategy	**Original Query**	**Original Positive Document**	**Rewritten Query**
REGA	Dementia simple screening score	# Alzheimer's # Teaches you how to assess FAQ; ## Alzheimer's ##;...	How is Alzheimer's assessed
REGA	Why cannot the configuration information be queried on the server's official website	... What is the configuration information of the server? What is the ISP server not configured? Soben's blog...	Configuration information cannot be queried on the server

For better understanding, we have translated the content into English.

discover more diverse and long-tail relevance patterns than pure multi-hop click signals, which benefits the generalization of our model on various search intents.

G. Case Study

1) Augmented Samples: To prove the quality of our expanded data, we sample some cases from three augmentation strategies and present them in Table VII. The first case shows two frequently co-session queries. It suggests that when users search for the world university rankings, they may be interested in both general rankings and subject rankings. Thus, the corresponding documents are all relevant to these two queries. The second case illustrates two expressions of the same search intent, e.g., the cooking time of chicken custard. The positive documents of one query are potentially relevant to another, but this relationship cannot be revealed by click signals alone. The last two samples present two long-tail search intents. By analyzing the issued query and its clicked document, our rewrite model can generate another relevant query to search for the same document, which can expand the training samples of long-tail intents. For example, when the query is "Dementia simple screening score" and the user clicks on the document related to Alzheimer's, the query rewriter analyzes that the user's search intent is the Alzheimer's assessment. To improve the query diversity, the query rewriter turns a statement into a question, and replaces common names with proper nouns to imitate different question habits of different users, hence improving the robustness of our ranking model on modeling user intents.

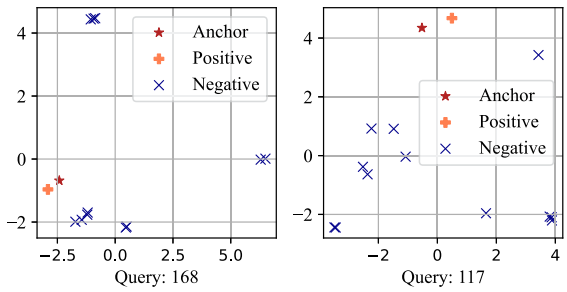

Fig. 4. Visualized embeddings of long-tail input pairs.

2) Visualization of Embeddings of Input Pairs: For the long-tail query-document clicked pairs (e.g., (q, d)), we leverage the query rewrite to generate similar queries (e.g., $\hat{q}$) that present the same search intents with original queries, and also be relevant to clicked documents. We use a contrastive learning method to align the embeddings of the augmented positive pair, $(\hat{q}, d)$, with the original positive pair, (q, d), a.k.a. anchor, while keeping their embeddings away from the embeddings of negative samples. To prove such ability of our model, we random sample two queries from the public dataset, i.e., query 168 and query 117, and visualize the embeddings of the original positive pair, the generated positive pair, and the negative pairs by red stars, orange pluses, and blue forks respectively. The visualization is complemented by PCA [54] algorithm, which is shown in Fig. 4. It is noticeable that the position of the embedding of the augmented pair (positive sample) is close to the embedding of the original pair (anchor sample), which is consistent with our

expectation. Since the augmented pair presents the same search intents as the original pair, pulling in their embeddings is able to ensure our model captures similar relevance features from them, hence grasping reliable relevance patterns.

V. Limitation Discussions

Our extensive experiments have proven the effectiveness of our proposed pre-training method for enhancing the ranking models. However, there are still some limitations that can be further promoted in the future.

First, due to limited available data resources, our main experiments are conducted on the single language (Chinese) search logs without considering multilingual scenarios. In the future, we plan to explore the effectiveness of our method across different language search logs further to verify its robustness. Second, in this paper, we still limit the model size to about the same as BERT to confirm the validity of our strategies. In the future, we expect to explore how to introduce larger models with sufficient world knowledge to benefit pre-trained ranking models.

VI. Conclusion

In this paper, we analyze that using click relationships alone to pre-train ranking models may limit the performance of pre-trained ranking models across diverse query forms and long-tail distributed search intents. According to existing problems, we propose a pre-trained method for ad-hoc ranking, namely PRADA. It leverages three augmentation strategies to diversify and expand the available relevance signals of our pre-trained ranking model. In addition, we assign multi-grade pseudo-relevance signals for positive documents and adopt the multi-level hinge loss function to distinguish the different reliability of relevance signals. Contrastive learning is further utilized to learn accurate relevance features from generated training samples. We conduct our experiments on both public and commercial large-scale search logs. The experimental results verify the effectiveness and generalization of our model, especially for long-tail intents.

References

[1] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Representations*, 2020.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[3] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Conf. Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2020, pp. 7871–7880.

[4] A. Radford and K. Narasimhan, *Improving Language Understanding By Generative Pre-training*, 2018.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language Models are Unsupervised Multitask Learners*, 2019.

[6] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[7] W.-C. Chang, F. X. Yu, Y.-W. Chang, Y. Yang, and S. Kumar, "Pre-training tasks for embedding-based large-scale retrieval," in *Proc. Int. Conf. Learn. Representations*, 2020.

[8] X. Ma, J. Guo, R. Zhang, Y. Fan, X. Ji, and X. Cheng, "PROP: Pre-training with representative words prediction for ad-hoc retrieval," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2021, pp. 283–291.

[9] J. Chen et al., "Axiomatically regularized pre-training for ad hoc search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1524–1534.

[10] X. Ma, J. Guo, R. Zhang, Y. Fan, Y. Li, and X. Cheng, "B-PROP: Bootstrapped pre-training with representative words prediction for ad-hoc retrieval," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 1318–1327.

[11] Z. Ma et al., "Pre-training for ad-hoc retrieval: Hyperlink is also you need," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1212–1221.

[12] Y. Seonwoo, S. Lee, J. Kim, J. Ha, and A. Oh, "Weakly supervised pre-training for multi-hop retriever," in *Proc. Findings Conf. Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2021, pp. 694–704.

[13] Y. Guo et al., "Webformer: Pre-training with web pages for information retrieval," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1502–1512.

[14] L. Zou et al., "Pre-trained language model based ranking in Baidu search," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 4014–4022.

[15] Z. Su, Z. Dou, Y. Zhou, Z. Zhao, and J. Wen, "PSLOG: Pretraining with search logs for document ranking," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, ACM, 2023, pp. 2072–2082.

[16] S. Goel, A. Z. Broder, E. Gabrilovich, and B. Pang, "Anatomy of the long tail: Ordinary people with extraordinary tastes," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2010, pp. 201–210.

[17] G. Forman, E. Kirshenbaum, and S. Rajaram, "A novel traffic analysis for identifying search fields in the long tail of web sites," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 361–370.

[18] W. U. Ahmad, K. Chang, and H. Wang, "Context attentive document ranking and query suggestion," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 385–394.

[19] Y. Zhu et al., "Contrastive learning of user behavior sequence for context-aware document ranking," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2780–2791.

[20] S. Wang, Z. Dou, and Y. Zhu, "Heterogeneous graph-based context-aware document ranking," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2023, pp. 724–732.

[21] S. Wang, Z. Dou, J. Liu, Q. Zhu, and J. Wen, "Personalized and diversified: Ranking search results in an integrated way," *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 81:1–81:25, 2024.

[22] Z. Dai and J. Callan, "Deeper text understanding for IR with contextual neural language modeling," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 985–988.

[23] L. Gao, Z. Dai, and J. Callan, "Rethink training of BERT rerankers in multi-stage retrieval pipeline," in *Proc. Adv. Inf. Retrieval - 43rd Eur. Conf.IR Res.*, Springer, 2021, pp. 280–286.

[24] R. F. Nogueira, W. Yang, K. Cho, and J. Lin, *Multi-Stage Document Ranking with BERT*, 2019, *arXiv: 1910.14424*.

[25] R. F. Nogueira and K. Cho, *Passage Re-Ranking with BERT*, 2019, *arXiv: 1901.04085*.

[26] J. Zhan, J. Mao, Y. Liu, M. Zhang, and S. Ma, "An analysis of BERT in document ranking," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1941–1944.

[27] C. Qu, C. Xiong, Y. Zhang, C. Rosset, W. B. Croft, and P. N. Bennett, "Contextual re-ranking with behavior aware transformers," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1589–1592.

[28] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5754–5764.

[29] W. Wang et al., "StructBERT: Incorporating language structures into pre-training for deep language understanding," in *Proc. Int. Conf. Learn. Representations*, 2020.

[30] H. Touvron et al., *LLaMA: Open and Efficient Foundation Language Models*, 2023, *arXiv:2302.13971*.

[31] K. Lee, M. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," in *Proc. Conf. Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2019, pp. 6086–6096.

[32] Y. Li et al., "S2phere: Semi-supervised pre-training for web search over heterogeneous learning to rank data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2023, pp. 4437–4448.

[33] H. Li et al., "SAILER: Structure-aware pre-trained language model for legal case retrieval," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 1035–1044.

[34] Y. Liu et al., "Pre-trained language model for web-scale retrieval in baidu search," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 3365–3375.
[35] L. H. Bonifacio, H. Abonizio, M. Fadaee, and R. F. Nogueira, *InPars: Data Augmentation for Information Retrieval Using Large Language Models*, 2022, *arXiv:2202.05144*.
[36] V. Jeronymo et al., *InPars-v2: Large Language Models as eFficient Dataset Generators for Information Retrieval*, 2023, *arXiv:2301.01820*.
[37] Z. Dai et al., "Promptagator: Few-shot dense retrieval from 8 examples," in *Proc. Int. Conf. Learn. Representations*, 2023.
[38] C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins, "Synthetic QA corpora generation with roundtrip consistency," in *Proc.Conf. Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2019, pp. 6168–6173.
[39] J. Saad-Falcon et al., "UDAPDR: Unsupervised domain adaptation via LLM prompting and distillation of rerankers," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2023, pp. 11265–11279.
[40] D. S. Sachan, M. Lewis, D. Yogatama, L. Zettlemoyer, J. Pineau, and M. Zaheer, "Questions are all you need to train a dense passage retriever," *Trans. Assoc. Comput. Linguistics*, vol. 11, pp. 600–616, 2023.
[41] H. Chen, Z. Dou, Y. Zhu, Z. Cao, X. Cheng, and J. Wen, "Enhancing user behavior sequence modeling by generative tasks for session search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 180–190.
[42] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[43] Z. Dou, R. Song, and J. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proc. Int. Conf. World Wide Web*, 2007, pp. 581–590.
[44] S. Wang, Z. Dou, J. Yao, Y. Zhou, and J. Wen, "Incorporating explicit subtopics in personalized search," in *Proc. Int. Conf. World Wide Web*, 2023, pp. 3364–3374.
[45] L. Zou, H. M. andXiaokai Chu, J. Tang, W. Ye, S. Wang, and D. Yin, "A large scale search dataset for unbiased learning to rank," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, *arXiv:2207.03051*.
[46] C. V. Gysel and M. de Rijke, "Pytrec_eval: An extremely fast python interface to trec_eval," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 873–876.
[47] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
[48] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 55–64.
[49] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. Conf. Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2019, pp. 1441–1451.
[50] Y. Sun et al., *ERNIE 3.0: Large-Scale Knowledge Enhanced Pre-Training for Language Understanding and Generation*, 2021, *arXiv:2107.02137*.
[51] X. Jiao et al., "PGLBox: Multi-GPU graph learning framework for web-scale recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2023, pp. 4262–4272.
[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[53] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.
[54] K. P. F. R. S., "LIII: On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

Shuting Wang received the BE degree in computer science and technology from Shandong University, in 2021. She is currently working toward the PhD degree with the Gaoling School of Artificial Intelligence, Renmin University of China. Her research interests include retrieval-augmented generation, large language models for IR, session-based document ranking, and personalized search.

Zhicheng Dou (Member, IEEE) received the BS and PhD degrees in computer science and technology from Nankai University, in 2003 and 2008, respectively. He is a professor with the Renmin University of China. His current research interests are information retrieval, natural language processing, and big data analysis. He received the Best Paper Runner-Up Award from SIGIR 2013, and the Best Paper Award from AIRS 2012. He served as the program co-chair of the short paper track for SIGIR 2019.

Kexiang Wang received the PhD degree in computer engineering and theory from Peking University, in 2021. He is a former software engineer with BAIDU Search Science team and currently serves as a research scientist with ALIBABA DAMO Academy. His current research interest spans virtual and embodied language-based agents, large language models, and general machine learning.

Dehong Ma received the PhD degree from the Insitute of Computational Linguistics (ICL), Peking University, in 2020. He is a staff algorithm engineer with Baidu Inc. Currently, he is broadly interested in natural language processing, information retrieval etc.

Jun Fan received the MSc degree from the Department of Computer Science, Beijing Institute of Technology, in 2014. He is now a senior algorithm engineer of Baidu Search.

Daiting Shi the current director architect of Baidu's Search Strategy Department. Since joining Baidu in 2014, his research areas include information retrieval, learning to rank, natural language processing, and large language models, and he has published multiple papers in these fields.

Zhicong Cheng received the master's degree from Peking University, in 2011. He is currently working with the Search Department of Baidu company as a principal architect. His current research interests are information retrieval and natural language.

Simiu Gu the current chief architect of Baidu's Search Strategy. Since joining Baidu in 2009, he has successively been in charge of Baidu's Knowledge Graph, Baidu's search recommendations, Baidu's information flow, and the overall search algorithm. Through the joint efforts of him and his team, Baidu has continuously improved and innovated in user information retrieval.

Dawei Yin (Senior Member, IEEE) received the BS degree from Shandong University, in 2006, and the MS and PhD degrees from Lehigh University, in 2010 and 2013, respectively. He is a senior director of engineering with Baidu inc.. He is managing the search science team with Baidu, leading Baidu's science efforts of web search, question answering, video search, image search, news search, app search, etc. He published more than 100 research papers in premium conferences and journals, and received 8 Best Paper Awards (or runner-ups), including KDD, WSDM, ICDM Best Paper Awards.

Ji-Rong Wen (Senior Member, IEEE) received the BS and MS degrees from the Renmin University of China, and the PhD degree from the Chinese Academy of Science, in 1999. He is a professor with the Renmin University of China. He was a senior researcher and research manager with Microsoft Research from 2000 to 2014. His main research interests include web data management, information retrieval (especially web IR), and data mining.