# Learning from the Wisdom of Crowds: Exploiting Similar Sessions for Session Search

**Yuhang Ye[1], Zhonghua Li[1], Zhicheng Dou[2], Yutao Zhu[3], Changwang Zhang[1], Shangquan Wu[1], Zhao Cao[1,*]**

[1] Huawei Poisson Lab, China
[2] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[3] University of Montreal, Quebec, Canada
{yeyuhang,lizhonghua3,zhangchangwang1,wushangquan,caozhao1}@huawei.com
dou@ruc.edu.cn, yutaozhu94@gmail.com

## Abstract

Search engines are essential internet services, enabling users to efficiently find the information they need. Session search employs users' session logs of queries to solve complex retrieval tasks, in which users search multiple times until interested documents are found. Most existing session search models focus on the contextual information within the current search, ignoring the evidence from historical search sessions. Considering the fact that many ongoing retrieval tasks should have already been carried out by other users with a similar intent, we argue that historical sessions with similar intents can help improve the accuracy of the current search task. We propose a novel Similar Session-enhanced Ranking (SSR) model to improve the session search performance using historical sessions with similar intents. Specifically, the candidate historical sessions are matched by query-level and session-level semantic similarity, and then query-level neighbor behaviors are aggregated by a Query-guided GNN (QGNN) while session-level neighbor behaviors are aggregated using the attention mechanism. Finally, we integrate the refined and aggregated historical neighbor information into the current search session. Experimental results on AOL and Tiangong-ST datasets show that our SSR model significantly outperforms the state-of-the-art models.

## Introduction

The goal of a search engine is to understand users' intentions and fulfill their information needs. Simple search tasks, such as navigation to a popular website, can be easily achieved by issuing one query and a click. However, for more complex search tasks, such as comparing mobile phones, users generally need multiple search interactions.

Session search emerges as an effective approach to solve complex search tasks, which not only focuses on the matching relationship between candidate documents with the current query but also considers users' previous search behaviors in a session. The search session context of the current query could possibly provide useful information to reduce query ambiguity and help understand the actual search intent of the current query. Figure 1 shows an example of web search session with queries and clicked documents. With the awareness of users' previous queries (i.e., $q_1$ Huawei mobile) and clicks, the search engine can better understand the search intent of the following query "Nova". In this case, the user is more likely to look for the series of Huawei Nova phones instead of the most-watched prime-time science series on American television with the same name "Nova".

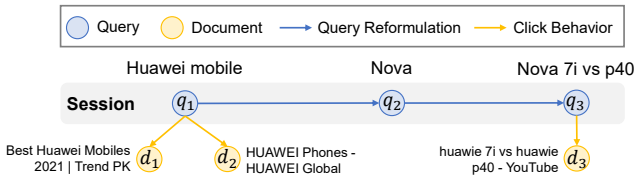


Figure 1: An example of session search. Three queries and the corresponding clicked documents are shown.

Existing session search works incorporate the search context information and achieve quite promising improvement on ranking performance (Sordoni et al. 2015; Ahmad, Chang, and Wang 2019; Qu et al. 2020; Chen et al. 2020). However, these works mainly utilize the history behavior only within the ongoing search session. When the session is very short, or the target query locates at the beginning part of a search task, the search context information could be very limited and even unavailable. As shown in (Chen et al. 2019; Qu et al. 2020), most sessions contain less than three queries, which further confirms this data sparsity limitation of these existing approaches.

To address the data sparsity problem, some researchers attempted to introduce long-term interests in users' previous searches (Guo et al. 2019; Cheng et al. 2021) or leverage social relationships (Zhou et al. 2021) to aggregate users' profiles for ranking. However, these methods are mainly for personalized ranking or recommendation, and the user-level history information may be unavailable in private search engines due to the privacy policy. Moreover, the above methods suffer from the cold start problem when no history behavior data is available for new users or new queries.

In this paper, we go beyond the ongoing search session of a particular user, by exploring historical behavior of all sessions to find queries or sessions with similar search intents to the current query and session. We explicitly use these

*Corresponding author.

matched queries and sessions to help improve the ranking performance of the current search session.

Specifically, we propose a novel Similar Session-enhanced Ranking (SSR) model: firstly, we perform both query-level and session-level neighbor session retrieval processes among all historical search logs, in order to find historical search sessions relevant to the current session. Secondly, we extract features from the retrieved neighbor queries and sessions using carefully designed graph and attention mechanisms, then aggregate them with the features of the current query and session for the ranking process.

Information from similar sessions has been explored in some Session-based Recommendation Systems (SRS) (Zheng et al. 2020; Wu and Gou 2021) to improve the recommendation performance. However, there are two main reasons why the methods in SRS studies cannot be applied to the session search directly. On the one hand, the user query is explicitly given in session search but not in SRS. On the other hand, document relevance is one of the main optimization criteria for session search but not in SRS. The search query is the key signal to represent users' search intent. How a similar session can be retrieved and used with the constraint of the current query, has not been explored in previous works.

Intuitively, by leveraging the "crowd wisdom" that how other users with similar search intent issue their queries and ultimately get satisfied, we can effectively solve the data sparsity problem. Since our method does not restrict to particular users, it can be applied to any search engine without privacy issues.

The main contributions of our work are summarized as follows:

- We propose a novel Similar Session-enhanced Ranking (SSR) model that explicitly employs relevant historical search queries and sessions to improve the performance of an ongoing session search. To the best of our knowledge, SSR is the first deep ranking model to consider both query and session level similar historical behaviors.
- We propose a novel Query-guided Graph Neural Network (QGNN) to aggregate the information in a query-neighbor graph. QGNN enables nodes with intent similar to the current search to gain higher weights in the final representation, increasing the effectiveness of query-level neighbor modeling.
- We design an asynchronous index encoding and updating mechanism to efficiently retrieve similar queries and sessions. This mechanism enables the time-consuming language model encoders to be used in industrial search systems.

## Related Work

**Session Search Models** When the search tasks are complex or ambiguous, users generally need to interact with the search engine multiple times. The queries, reformulated queries, and the corresponding clicked/skipped documents within the session are generally considered related and explored by many researchers. Several works have demonstrated the effectiveness of such session-level contextual information over a single query (Bennett et al. 2012; Carterette et al. 2016). Early studies often relied on manually extracted features or handcrafted rules (Shen, Tan, and Zhai 2005; Cao et al. 2008, 2009; Zhang et al. 2016), which limited their application in various search tasks.

With the rapid development of deep neural networks, the model of session behavior encoding has been significantly improved. Some researchers proposed a hierarchical neural structure with recurrent neural networks (RNNs) to model historical queries and predict the next query (Sordoni et al. 2015). Thereafter, researchers discovered that jointly learning document ranking and query suggestion can enhance both of them (Ahmad, Chang, and Wang 2019). Recently, pre-trained language models, such as BERT (Devlin et al. 2019), have been applied to capture search intent from user behavior sequences and conduct document ranking. To enhance the ability of aggregating session history, a hierarchical behavior-aware Transformer model was developed on top of BERT (Qu et al. 2020). Furthermore, it is shown that contrastive learning is also beneficial for optimizing the BERT encoder in context-aware document ranking (Zhu et al. 2021; Chen et al. 2022; Zhu et al. 2022; Wang, Dou, and Zhu 2023). Different from these methods, our proposed method investigates the power of similar sessions in search logs, which is effective for improving the performance of document ranking.

**Historical Session Based Ranking Model** According to the analysis of real-world web search logs (Qu et al. 2020; Chen et al. 2019), the average length of a session is about 2.5, therefore, the information within the ongoing sessions is generally very limited. Some researchers in the personalized search domain took the user's historical behavior both within the ongoing session and previous sessions into the ranking model. Ge et al. (Ge et al. 2018) regarded the current session and historical session as users' short-term and long-term interests respectively. To explore users' re-finding behavior, RPMN (Zhou, Dou, and Wen 2020) took advantage of memory networks for personalized search. Furthermore, LostNet (Cheng et al. 2021) developed a hierarchical session-based attention mechanism with a personalized multi-hop memory network, in order to learn the long-term and short-term users' interests.

Instead of using a single user's session information, many researchers harness collective intelligence (e.g. click-through bipartite graph and similar users' logs) from the search logs of different users to improve the ranking performance. Zhang et al. (Zhang, Wang, and Zhang 2019) proposed a graph embedding-based ranking model for product search (GEPS), which was the first to introduce click-graph embedding for enriching the representation of queries and items. Moreover, Li et al. (Li et al. 2020) represented the query and the document with the use of a session-flow as well as a click-through bipartite graph. Some group-based web search methods (Dou, Song, and Wen 2007; Teevan, Morris, and Bush 2009; Vu et al. 2014) made use of the query logs of groups of similar users, and also showed significant improvement. However, none of them investigate the potential of both similarity queries and sessions together to the current search. In our work, we employ and model

both query-level and session-level neighbor information to enrich the representation of the target query, and significantly alleviate the sparsity issue of search sessions.

# Methodology

## Problem Statement

We provide some definitions of important concepts and notations as follows: We denote the user's search behavior of a session as a sequence of $L$ queries $Q = \langle q_1, \cdots, q_L \rangle$. For each query $q_i \in Q$, the search engine returns top-$N$ relevant documents $D_i = \langle d_{i,1}, d_{i,2}, ..., d_{i,N} \rangle$. The binary label $y_{i,j}$ indicates whether the document $d_{i,j}$ is clicked or not. Each query $q_i$ is represented by the original text issued by the user, while each candidate document $d_{i,j}$ is represented by its text content. All queries are ordered according to their issued time. Note that we only consider queries and clicked documents in this work, since the unclicked documents are shown to be useless (Ahmad, Chang, and Wang 2019; Qu et al. 2020). Therefore, a search session can be denoted as: $S = \langle q_1, d_1, q_2, d_2, \cdots, q_L, d_L \rangle$.[1]

The context-aware document ranking task is to rank the candidate document set $D_i$ of query $q_i$ based on its search context $\langle q_1, d_1, \cdots, q_{i-1}, d_{i-1} \rangle$ in order to rank the clicked document as high as possible. In this work, in addition to search context, we propose to select similar sessions as supplementary information to rank the candidate documents.

## Overview

The overall structure of our SSR model is shown in Figure 2. SSR has four main components:

(1) **BERT encoders:** SSR uses BERT (Devlin et al. 2019) as basic encoders to represent a sequence (*i.e.*, a query, a document, or a session) as a vector.

(2) **ANN-based indices:** We build two indices for both query and session representations, respectively, using the Approximate Nearest Neighbor (ANN) algorithm. They are used for fast similar query/session selection.

(3) **Query-/session-level neighbor modeling:** The query-level neighbor modeling has two steps: first, queries similar to the current query are retrieved through the ANN index; then a query-neighbor graph is constructed, based on which a query-guided graph neural network (QGNN) is employed to learn an aggregated representation for these similar queries. For session-level neighbor modeling, we retrieve similar sessions based on the session-level ANN index and devise an attention fusion method to learn the representations of these neighbor sessions.

(4) **Ranking:** This component takes the representation of neighbor queries, neighbor sessions, current query, current session, and the candidate document as input and computes a final ranking score.

The details of these components are described as follows.

[1] If there are multiple clicked documents, they are all considered and arranged in chronological order.

## Encoders

We use a pre-trained BERT (Devlin et al. 2019) as the encoder to represent queries, documents, and sessions as vectors. Specifically, for queries and documents, we use the mean pooling of all token representations at the last layer as:

$$\tilde{\mathbf{q}} = \text{Mean}(\text{BERT}(q)), \quad \tilde{\mathbf{d}} = \text{Mean}(\text{BERT}(d)), \quad (1)$$

where $q = [w_q^1, \cdots, w_q^n]$ and $d = [w_d^1, \cdots, w_d^n]$ are the token sequence of the query and document. They are all padded to the length of $n$. $\tilde{\mathbf{q}}, \tilde{\mathbf{d}} \in \mathbb{R}^{768}$ are the query and document representation, respectively. For the sessions, we concatenate all queries and documents as a token sequence. Then, we feed it into the BERT encoder and also use the mean pooling to get the session representations as:

$$\begin{aligned} S &= \texttt{[CLS]}\, q_1\, \texttt{[EOS]}\, d_1 \cdots q_L\, \texttt{[EOS]}\, d_L\, \texttt{[EOS]}\, \texttt{[SEP]}, \\ S_t &= \texttt{[CLS]}\, q_1\, \texttt{[EOS]}\, d_1 \cdots q_t\, \texttt{[EOS]}\, \texttt{[SEP]}, \\ \tilde{\mathbf{S}} &= \text{Mean}(\text{BERT}(S)), \quad \tilde{\mathbf{S}}_t = \text{Mean}(\text{BERT}(S)) \end{aligned} \quad (2)$$

where $S$ is the complete session and $S_t$ is the ongoing session. $q_i$ and $d_i$ denote the query and document token sequence, respectively. The `[EOS]` tokens are added to indicate the end of a query/document. The `[CLS]` and `[SEP]` are two special tokens defined by the vanilla BERT to mark the start and end of a sequence.

## ANN-based Indices and Updating Mechanism

To support fast neighbor search, we build two indices to store the encoded history queries and sessions obtained in Equation (1) and (2). During the SSR training, the parameters of the BERT encoders are kept updating. Theoretically, the two indices should also be updated so as to retrieve similar queries/sessions more accurately. However, keeping updating indices at each training step is impractical, because both computing representations and refreshing indexes are very time-consuming.

To tackle this problem, we employ an asynchronous updating mechanism for the indices, which is shown in Figure 3. Concretely, we set an updating period $T$, and for the training step $t \bmod T = 0$, we recompute all queries' and sessions' representation by the BERT encoders and update the indices. For other training steps, we use the latest indices to retrieve the most relevant queries and sessions.

## Query-level Neighbor Modeling

According to our observation that similar queries are often issued by various users in different sessions for similar information need, we propose a graph-based method to expand the current query with other sessions that contains similar queries to help the model better understand the current search intent.

The graph construction process is shown in the upper part of Figure 2. First, we perform an ANN search using the current query $q_t$ to retrieve top-$n_q$ similar queries. Their representations are denoted as $[\tilde{\mathbf{q}}_1, \cdots, \tilde{\mathbf{q}}_{n_q}]$. Henceforth, we call these queries *neighbor queries*. Then, we recall the corresponding sessions that contain the neighbor queries. Next,

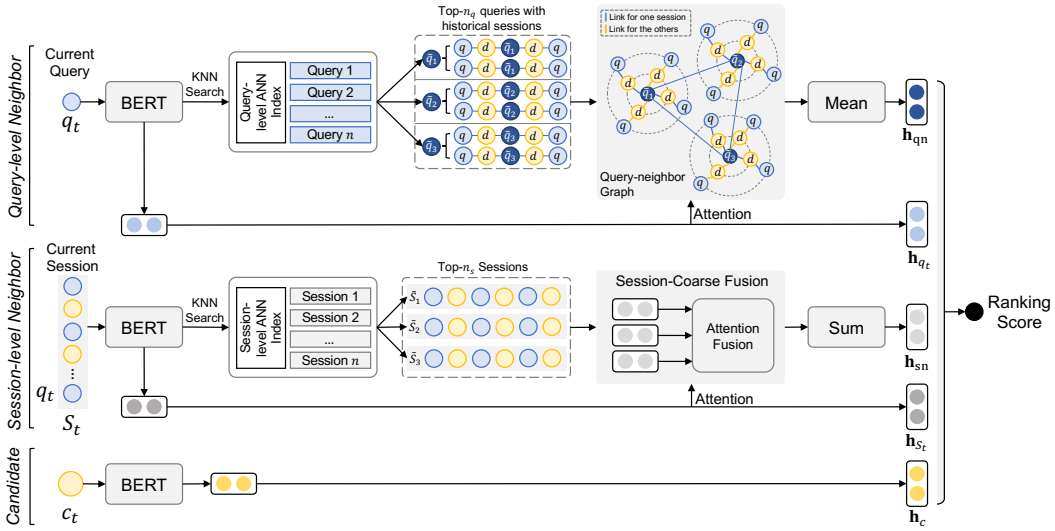


Figure 2: The architecture of the similar session-enhanced ranking model. Similar sessions retrieved by the query- and session-level finally enrich the representations of the session search

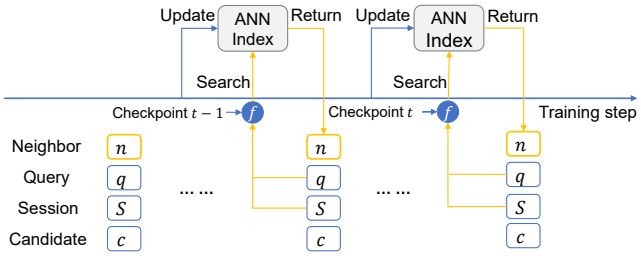


Figure 3: The training procedure of the MNRN

following previous studies (Li et al. 2020), for each retrieved session, we build a session-flow graph with a depth of $K$, which is centered at the neighbor query. Finally, we connect all neighbor queries (as they are similar) so that all session-flow graphs are connected. We call it *query-neighbor graph*. Note that all nodes in this graph are initialized by the vector representations stored in ANN indices.

After building the query-neighbor graph, we propose a query-guided graph neural network (QGNN) to compute the representation of the query-level neighbors. The basic idea of QGNN is using the current query to guide the message aggregation process so that the information most relevant to the current query can be updated into the node representations. Specifically, the input of the QGNN is a set of node vectors $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m\}$, where $m$ is the number of nodes. QGNN aggregates useful information from the neighbor nodes according to the attention with the current query $q_t$. Similar to the other GNNs (Kipf and Welling 2017; Velickovic et al. 2018; Wang et al. 2019), the representation propagation in QGNN involves two major operations: message aggregation and message transformation.

**Query-aware Message Aggregation**. For a node $v_i$ in the $k$-th layer with a set of neighbor nodes $\mathcal{N}(v_i)$, we define the message aggregation in node $v_i$ as:

$$\mathbf{h}^k_{\alpha_i} = \sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \alpha^k_{ij} \mathbf{h}^k_{v_j}, \tag{3}$$

where $\alpha^k_{ij}$ is the attention weight that determines how much information propagated from node $v_j$ to node $v_i$, which is calculated as:

$$\begin{aligned} e^k_{ij} &= \mathbf{a}^k \, \text{LeakyReLU}(\mathbf{W}^k[\mathbf{h}^k_{v_i}; \mathbf{h}^k_{v_j}; \mathbf{h}_{q_t}]), \\ &= \mathbf{a}^k \, \text{LeakyReLU}(\mathbf{W}^k_{v_i}\mathbf{h}^k_{v_i} + \mathbf{W}^k_{v_j}\mathbf{h}^k_{v_j} + \mathbf{W}^k_{q_t}\mathbf{h}^k_{q_t}), \\ \alpha^k_{ij} &= \text{softmax}(e^k_{ij}) = \frac{\exp(e^k_{ij})}{\sum_{v_p \in \mathcal{N}(v_i) \cup \{v_i\}} \exp(e^k_{ip})}. \end{aligned}$$

where $[;]$ is the concatenation operation; $\mathbf{W}^k \in \mathbb{R}$ and $\mathbf{a}^k$ are two parameters in the $k$-th layer; and LeakyReLU$(\cdot)$ is the activation function. By this means, $\alpha^k_{ij}$ can learn the relation between the current query and nodes $v_i$ and $v_j$, which are called current query guided attention.

Different from standard graph neural networks that only consider the contribution of $v_i$ and $v_j$, we introduce $q_t$ into attention weight computation. This guides the message aggregation to consider the relevance between the current query $q_t$ and each node in the graph. In other words, if a node is more similar to the current query, it will contribute more to the final representation. This query-aware message aggregation can help to refine more useful information for propagation and reduce the noise from irrelevant queries or documents, which increases the robustness of the query-level neighbor modeling.

**Message Transformation**. Thereafter, we update the node $v_i$'s representation by the aggregated message $\mathbf{h}^k_{\alpha_i}$ using a linear transformation layer, which can be defined as:

$$\mathbf{h}^{k+1}_{v_i} = \sigma(\mathbf{W}^k_t \cdot \mathbf{h}^k_{\alpha_i} + b^k), \tag{4}$$

where $\sigma$ is the Sigmoid function, and $\mathbf{W}_t^k, b^k$ are parameters. The number of layers $K$ determines the distance of the message passing on the graph. Through the message transformation, node representations can be refined layer by layer by aggregating information from their neighbors.

Finally, we take the mean pooling for the representation of $n_q$ neighbor queries as the final representation of the query-level neighbor modeling:

$$\mathbf{h}_{\text{qn}} = \text{Mean}(\tilde{\mathbf{h}}_{q_1}, \cdots, \tilde{\mathbf{h}}_{q_{n_q}}), \tag{5}$$

$\mathbf{h}_{q_i}$ is the updated representation after QGNN of the neighbor query $q_i$.

### Session-level Neighbor Modeling

Query-level neighbors are retrieved according to the current query. However, in practice, some queries are vague and ambiguous, which may bring noise neighbors and influence the understanding of the queries. To alleviate this problem, we propose to find session-level neighbors that are most similar to the current session (search context).

As shown in the middle part of Figure 2, the current search context $S_t$ is first encoded by the BERT encoder (as $s_t$ or $\mathbf{h}_{S_t}$) and used to retrieve several similar sessions through the session index. Similar to neighbor queries, we denote the top-$n_s$ sessions as *neighbor sessions*. Finally, an attention fusion method is devised to compute an integrated neighbor session representation based on all neighbor sessions' representations and the current session representation.

Specifically, considering the neighbor sessions' representations $\{\mathbf{s}_1, \cdots, \mathbf{s}_{n_s}\}$, the integrated vector of the neighbor sessions is calculated by the attention fusion as:

$$\mathbf{h}_{\text{sn}} = \sum_{i=1}^{n_s} \beta_i \mathbf{s}_i, \tag{6}$$

where the weights $\beta_i$ is given by:

$$\beta_i = \frac{\exp(\text{sim}(\mathbf{W} \cdot \mathbf{s}_i, \mathbf{W} \cdot \mathbf{s}_t))}{\sum_{j=1}^{n_s} \exp(\text{sim}(\mathbf{W} \cdot \mathbf{s}_j, \mathbf{W} \cdot \mathbf{s}_t))}, \tag{7}$$

where $\mathbf{W}$ is a parameter. $\mathbf{s}_t$ is the representation of the current search context. $\text{sim}(\cdot)$ is a function used to calculate the similarity of two vectors, and we use inner product in our experiments.

### Prediction and Ranking Loss

After the query and session-level neighbor modeling, we have five representations at different level: the query-level neighbor representation $\mathbf{h}_{\text{qn}}$, the session-level neighbor representation $\mathbf{h}_{\text{sn}}$, the current search context representation $\mathbf{h}_{S_t}$, the current query representation $\mathbf{h}_q$, and the candidate document representation $\mathbf{h}_c$. Following the design of sentence-BERT (Reimers and Gurevych 2019), we also use element-wise difference $|\mathbf{x} - \mathbf{y}|$ between the vector $\mathbf{x}$ and $\mathbf{y}$ as features. The element-wise difference measures the distance between the dimensions of the two representations, ensuring that similar pairs are closer and dissimilar pairs are further apart. Therefore, we also use the following features for document ranking:

$$\mathbf{I}_{qn} = |\mathbf{h}_{qn} - \mathbf{h}_c|, \qquad \mathbf{I}_{sn} = |\mathbf{h}_{sn} - \mathbf{h}_c|, \tag{8}$$

$$\mathbf{I}_{q_t} = |\mathbf{h}_{q_t} - \mathbf{h}_c|, \qquad \mathbf{I}_{S_t} = |\mathbf{h}_{S_t} - \mathbf{h}_c|. \tag{9}$$

Finally, all features are concatenated and fed into a linear transformation layer as:

$$p = \sigma\left(\mathbf{W} \cdot [\mathbf{h}_{qn}; \mathbf{h}_{sn}; \mathbf{h}_{q_t}; \mathbf{h}_{S_t}; \mathbf{h}_c; \mathbf{I}_{qn}; \mathbf{I}_{sn}; \mathbf{I}_{q_t}; \mathbf{I}_{S_t}]\right),$$

where $p$ is the predicted probability of a document being clicked.

The document ranking task then can be trained by minimizing the binary cross-entropy loss:

$$\mathcal{L}(\theta) = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)). \tag{10}$$

## Experiments

### Datasets and Evaluation Metrics

We conduct our experiments on two benchmark datasets: AOL search log data (Pass, Chowdhury, and Torgeson 2006) and TianGong-ST query log data (Chen et al. 2019). These two datasets are widely used in existing session search works (Qu et al. 2020; Zhu et al. 2021). It is also worth noting that most existing public datasets do not contain realistic search sessions and thus are not used in our session model test.

**AOL:** It is a large-scale real user search log. To make a fair comparison, we directly use the version provided by Ahmad, Chang, and Wang (2019) where the candidate documents are retrieved by BM25 and the datasets are split into background, training, validation, and test sets. Each query contains five candidate documents in training and validation sets. In the test set, there are 50 candidate documents for each query. The background set is used to generate candidate queries for query suggestion in (Ahmad, Chang, and Wang 2019), but in this paper, we use this set for similar session retrieval.

**TianGong-ST:** It is a public dataset collected from a Chinese commercial search engine. It contains a search log for 18 days and each query has 10 candidate documents. In both training and validation sets, click labels are given. In the test set, the last query of each session is manually annotated with relevance scores as labels, while other (previous) queries in the session only has click signals. To keep the consistency between the training and inference, we only use those previous queries and corresponding documents with click labels as test data. Furthermore, we use the training set for similar session retrieval.

Following previous works (Huang et al. 2018; Ahmad, Chang, and Wang 2019; Qu et al. 2020), we only use the document title as its content so as to speed up our model training.

**Evaluation Metrics:** Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) are used as our evaluation metrics. The TREC's evaluation tool (trec_eval) is used for the metric implementation.

| Dataset | Metric | ARC-I | ARC-II | K-NRM | Duet | M-NSRF | M-Match | CARS | HBA | COCA | SSR | Improv. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AOL | MAP | 0.3361 | 0.3834 | 0.4038 | 0.4008 | 0.4217 | 0.4459 | 0.4297 | 0.5281 | 0.5500 | **0.5560**$^{\dagger}$ | 1.09% |
| | MRR | 0.3475 | 0.3951 | 0.4133 | 0.4111 | 0.4326 | 0.4572 | 0.4408 | 0.5384 | 0.5601 | **0.5673**$^{\dagger}$ | 1.29% |
| | NDCG@1 | 0.1988 | 0.2428 | 0.2397 | 0.2492 | 0.2737 | 0.3020 | 0.2816 | 0.3773 | 0.4024 | **0.4157**$^{\dagger}$ | 3.31% |
| | NDCG@10 | 0.3953 | 0.4486 | 0.4761 | 0.4675 | 0.4886 | 0.5103 | 0.4971 | 0.5951 | 0.6160 | **0.6193** | 0.54% |
| Tiangong-ST | MAP | 0.6597 | 0.6729 | 0.6551 | 0.6745 | 0.6836 | 0.6778 | 0.6909 | 0.6957 | 0.7481 | **0.7515** | 0.45% |
| | MRR | 0.6826 | 0.6954 | 0.6748 | 0.7026 | 0.7065 | 0.6993 | 0.7134 | 0.7171 | 0.7696 | **0.7768**$^{\dagger}$ | 0.94% |
| | NDCG@1 | 0.5315 | 0.5458 | 0.5104 | 0.5738 | 0.5609 | 0.5499 | 0.5677 | 0.5726 | 0.6386 | **0.6515**$^{\dagger}$ | 2.02% |
| | NDCG@10 | 0.7509 | 0.7608 | 0.7469 | 0.7621 | 0.7691 | 0.7646 | 0.7746 | 0.7781 | 0.8180 | **0.8213** | 0.40% |

Table 1: Document Ranking Performance on all datasets. The best performance and the second best performance are in bold and underlined, respectively. The improvement of SSR over the best baseline is given in the last column. † indicates SSR achieves significant improvements over all existing methods in paired t-test with $p$-value $< 0.01$.

## Baselines

The proposed SSR is compared with two types of baseline methods: ad-hoc ranking models and context-aware ranking models.

**Ad-hoc ranking models.** This type of models does not consider the contextual information (*i.e.*, the precedent queries and clicked documents within a session) and only use the current query to rank the candidate documents. It can be categorized into representation-based and interaction-based models. **ARC-I** (Hu et al. 2014) and **ARC-II** (Hu et al. 2014), as representative models in each of the categories are compared here as baseline models. **K-NRM** (Xiong et al. 2017), as another interaction-based model, has shown competitive ranking performance and is also compared as one of the baseline models. As a combination of representation-based and interaction-based methods, **Duet** (Mitra, Diaz, and Craswell 2017) computes the local and distributed representations of query and document by CNN and MLP layers, and the final ranking scores are computed by integrating both representation-based and interaction-based features.

**Context-aware ranking models.** Previous studies have shown the superiority of context-aware ranking models compared with the ad-hoc ones. We consider them as our competitive baselines. **M-NSRF** (Ahmad, Chang, and Wang 2018) adopts the multi-task learning technique and models the current query, history query, and document representation respectively using a deep interactive architecture. Similar to M-NSRF, **M-Match-Tensor** (Ahmad, Chang, and Wang 2018) learns the contextual representations combined with the current query and session information for multi-task learning. **CARS** (Ahmad, Chang, and Wang 2019) and **HBA** (Qu et al. 2020), are both advanced context-aware document ranking models and are implemented as strong baselines. **COCA** (Zhu et al. 2021) designs a pre-training stage based on contrastive learning to improve a BERT's representation of user behavior sequences. Then, the BERT model is further fine-tuned for context-aware document ranking. This is the state-of-the-art method.

## Overall Results

Experimental results are shown in Table 1. We can observe that SSR performs the best among all existing models on both AOL and Tiangong-ST. This strongly demonstrates the effectiveness of our proposed method. We further have several observations as follows.

(1) Context-aware ranking models perform better than ad-hoc ranking models. In our experiments, we can find that the weak contextualized model M-NSRF can even outperform the strong ad-hoc ranking model K-NRM. This reflects the importance of contextual information in capturing search intent.

(2) Pre-trained language model-based approaches (such as HBA, COCA, and SSR) perform better than RNN-based methods (such as M-NSRF and M-Match). Note that the later methods jointly learn both document ranking and query suggestion tasks, which utilizes more supervision signals. Therefore, this result reflects the clear advantage of applying pre-rained language models in document ranking.

(3) HBA designs complex interaction layers on the top of BERT, and COCA develops a session-based contrastive learning method to pre-train the BERT for document ranking. Both of them are interaction-based models, which means the context and document are concatenated as a sequence for ranking score calculation. In contrast, our SSR model is representation-based, where the document representation can be pre-computed offline. The results demonstrate that SSR can achieve better performance while keeping high efficiency.

## Ablation Study

We conduct a series of experiments to investigate the influence of our proposed neighbor enhance strategies. Specifically, we first remove the query-level neighbors ("*w/o* query-level") and session-level neighbors ("*w/o* sess-level") from the full model, respectively. Then, we remove both of them ("*w/o* both"), in which only the information in the current session is used. These variants are tested on AOL, and the results are shown in Table 2. We can see:

(1) Removing any kind of neighbors leads to performance degradation. This clearly validates the effectiveness of our proposed neighbor-enhanced method. (2) Compared with query-level neighbors, removing session-level neighbors has more impact. We attribute this to the noise nature of single queries. As some queries are vague and ambiguous, they will affect the accuracy of the retrieved neighbor queries. In con-

| | MAP | MRR | NDCG@1 | NDCG@10 |
|---|---|---|---|---|
| SSR (Full) | **0.5560** | **0.5673** | **0.4157** | **0.6193** |
| *w/o* query-level | 0.5520 | 0.5635 | 0.4123 | 0.6155 |
| *w/o* sess-level | 0.5514 | 0.5626 | 0.4110 | 0.6142 |
| *w/o* both | 0.5321 | 0.5442 | 0.3894 | 0.5987 |

Table 2: Performance of SSR on the AOL dataset with different historical neighbors retrieval strategies.

trast, session-level neighbors consider the previous behaviors in a session, which can alleviate some ambiguity of the query, thus providing more relevant and valuable neighbor sessions to improve the ranking. (3) Equipping with either kind of neighbor can improve the performance significantly. This indicates the great potential of similar sessions in the search log for context-aware document ranking.

## Sessions with Different Lengths

To investigate the robustness of SSR, we further test the performance with different session lengths on the AOL dataset. We split all testing sessions into three bins according to their session lengths: (1) Short sessions: 1-2 queries - 77.13% of test set; (2) Medium sessions: 3-4 queries - 18.19% of test set; (3) Long sessions: $\geq 5$ queries - 4.69% of test set.

We compare SSR with several competitive baselines, *i.e.*, Duet, HBA, and COCA. The MAP and NDCG@1 results are shown in Figure 4. **First**, we find SSR achieves the highest results than baseline models on all three bins of sessions, which indicates the superiority of SSR. **Second**, the context-aware ranking models, such as HBA, COCA, and SSR, outperform the ad-hoc ranking model Duet. These observations demonstrate once again that the contextual information in sessions is important for document ranking. **Third**, we can observe SSR performs relatively worse when ranking for the long sessions. Other baseline methods' results also have similar trends. Long sessions may contain more exploratory queries and clicks which increase the difficulty of document ranking. **Finally**, we calculate the improvement of SSR compared with the COCA. We find that SSR improves about 4.50%, 2.60%, 5.0% on short, medium, and long sessions respectively in terms of NDCG@1. It can be found that the improvement in short and long sessions is greater than that in medium sessions. This result implies that our method is extremely effective in two situations: 1) sparse data scenario: when session context is limited, *e.g.*, at the beginning of a session; 2) exploratory search scenario: when users are not sure about their goals.

## Performance with Different Hyperparameters

Some hyperparameters, such as the number of query-/session-level neighbors ($n_q$ and $n_s$) and the maximum node number $m$ in query-neighbor graph, may influence SSR's performance. We conduct several experiments on AOL dataset to test such influence. For convenience, we empirically set $n_q = n_s$, and the results are given in Tabel 3.

For the number of neighbors, we can see SSR performs better as the number of neighbors increasing at the beginning

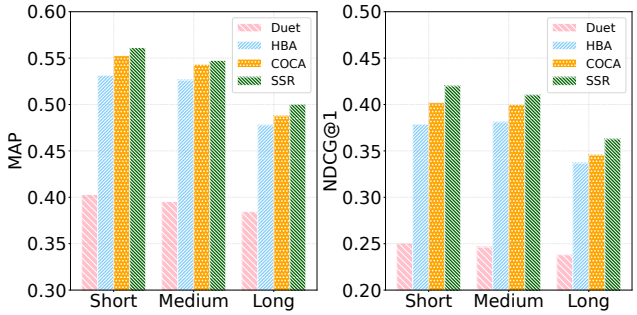


Figure 4: Performance on different lengths of sessions.

| Model | # Neighbors | MAP | NDCG@1 | NDCG@10 |
|---|---|---|---|---|
| SSR | 1 | 0.5553 | 0.4150 | 0.6186 |
| SSR | 3 | 0.5558 | 0.4158 | 0.6189 |
| SSR | 5 | 0.5560 | 0.4157 | 0.6193 |
| SSR | 7 | 0.5552 | 0.4148 | 0.6185 |
| Model | Max Nodes | MAP | NDCG@1 | NDCG@10 |
| SSR | 12 | 0.5537 | 0.4139 | 0.6164 |
| SSR | 24 | 0.5543 | 0.4148 | 0.6171 |
| SSR | 36 | 0.5560 | 0.4157 | 0.6193 |
| SSR | 48 | 0.5535 | 0.4146 | 0.6164 |

Table 3: Performance of SSR on AOL dataset with different hyperparameters.

(from 1 to 5 neighbors). This indicates that more neighbors can bring more sufficient information and improve the performance. However, when we keep increasing the number of neighbors (from 5 to 7 neighbors), the performance starts decreasing. The potential reason is that more neighbors introduce some noise. Consequently, five neighbors are the best setting for our SSR. Similar results can be observed when we increase the maximum number of nodes in the neighbor graph. The performance peak at $m = 36$. All these results demonstrate that a reasonable number of neighbors is critical for achieving the best performance.

## Conclusion

In this paper, we propose a novel framework namely Similar Session-enhanced Ranking (SSR) model, which employs both query-level and session-level similar sessions to guide the current search. Specifically, we design a novel Query-guided Graph Neural Network (QGNN) and an attention fusion mechanism to aggregate the information in query-neighbor graphs and relevant historical sessions. Experimental results on public datasets demonstrate the effectiveness of the proposed model and confirm the potential of relevant historical session information in improving session search performance.

## References

Ahmad, W. U.; Chang, K.; and Wang, H. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *6th International Conference on Learning Representations,*

*ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ahmad, W. U.; Chang, K.; and Wang, H. 2019. Context Attentive Document Ranking and Query Suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 385–394. ACM.

Bennett, P. N.; White, R. W.; Chu, W.; Dumais, S. T.; Bailey, P.; Borisyuk, F.; and Cui, X. 2012. Modeling the impact of short- and long-term behavior on search personalization. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, 185–194. ACM.

Cao, H.; Jiang, D.; Pei, J.; Chen, E.; and Li, H. 2009. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, 191–200. ACM.

Cao, H.; Jiang, D.; Pei, J.; He, Q.; Liao, Z.; Chen, E.; and Li, H. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, 875–883. ACM.

Carterette, B.; Clough, P. D.; Hall, M. M.; Kanoulas, E.; and Sanderson, M. 2016. Evaluating Retrieval over Sessions: The TREC Session Track 2011-2014. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, 685–688. ACM.

Chen, H.; Dou, Z.; Zhu, Y.; Cao, Z.; Cheng, X.; and Wen, J. 2022. Enhancing User Behavior Sequence Modeling by Generative Tasks for Session Search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, 180–190. ACM.

Chen, J.; Mao, J.; Liu, Y.; Zhang, M.; and Ma, S. 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*, 2485–2488. ACM.

Chen, J.; Mao, J.; Liu, Y.; Zhang, M.; and Ma, S. 2020. A Context-Aware Click Model for Web Search. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, 88–96. ACM.

Cheng, Q.; Ren, Z.; Lin, Y.; Ren, P.; Chen, Z.; Liu, X.; and de Rijke, M. 2021. Long Short-Term Session Search: Joint Personalized Reranking and Next Query Prediction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 239–248. ACM / IW3C2.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Association for Computational Linguistics.

Dou, Z.; Song, R.; and Wen, J. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, 581–590. ACM.

Ge, S.; Dou, Z.; Jiang, Z.; Nie, J.; and Wen, J. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 347–356. ACM.

Guo, Y.; Cheng, Z.; Nie, L.; Wang, Y.; Ma, J.; and Kankanhalli, M. S. 2019. Attentive Long Short-Term Preference Modeling for Personalized Product Search. *ACM Trans. Inf. Syst.*, 37(2): 19:1–19:27.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2042–2050.

Huang, J.; Zhang, W.; Sun, Y.; Wang, H.; and Liu, T. 2018. Improving Entity Recommendation with Search Log and Multi-Task Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 4107–4114. ijcai.org.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Li, X.; de Rijke, M.; Liu, Y.; Mao, J.; Ma, W.; Zhang, M.; and Ma, S. 2020. Learning Better Representations for Neural Information Retrieval with Graph Information. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 795–804. ACM.

Mitra, B.; Diaz, F.; and Craswell, N. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, 1291–1299. ACM.

Pass, G.; Chowdhury, A.; and Torgeson, C. 2006. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1, 2006*, volume 152 of *ACM International Conference Proceeding Series*, 1. ACM.

Qu, C.; Xiong, C.; Zhang, Y.; Rosset, C.; Croft, W. B.; and Bennett, P. N. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *Proceedings of the 43rd International ACM SIGIR conference on research and development*

*in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 1589–1592. ACM.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 3980–3990. Association for Computational Linguistics.

Shen, X.; Tan, B.; and Zhai, C. 2005. Context-sensitive information retrieval using implicit feedback. 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*, 43–50. ACM.

Sordoni, A.; Bengio, Y.; Vahabi, H.; Lioma, C.; Simonsen, J. G.; and Nie, J. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, 553–562. ACM.

Teevan, J.; Morris, M. R.; and Bush, S. 2009. Discovering and using groups to improve personalized search. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, 15–24. ACM.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 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.

Vu, T. T.; Song, D.; Willis, A.; Tran, S. N.; and Li, J. 2014. Improving search personalisation with dynamic group formation. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, 951–954. ACM.

Wang, S.; Dou, Z.; and Zhu, Y. 2023. Heterogeneous Graph-based Context-aware Document Ranking. In *WSDM '23: The Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, Singapore, February 27-March 3, 2023*. ACM.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 165–174. ACM.

Wu, Y.; and Gou, J. 2021. Leveraging neighborhood session information with dual attentive neural network for session-based recommendation. *Neurocomputing*, 439: 234–242.

Xiong, C.; Dai, Z.; Callan, J.; Liu, Z.; and Power, R. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, 55–64. ACM.

Zhang, Y.; Wang, D.; and Zhang, Y. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2390–2400. ACM.

Zhang, Z.; Wang, J.; Wu, T.; Ren, P.; Chen, Z.; and Si, L. 2016. Supervised Local Contexts Aggregation for Effective Session Search. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, volume 9626 of *Lecture Notes in Computer Science*, 58–71. Springer.

Zheng, Y.; Liu, S.; Li, Z.; and Wu, S. 2020. DGTN: Dual-channel Graph Transition Network for Session-based Recommendation. In *20th International Conference on Data Mining Workshops, ICDM Workshops 2020, Sorrento, Italy, November 17-20, 2020*, 236–242. IEEE.

Zhou, Y.; Dou, Z.; Wei, B.; Xie, R.; and Wen, J. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, 92–101. ACM.

Zhou, Y.; Dou, Z.; and Wen, J. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, 789–797. ACM.

Zhu, Y.; Nie, J.; Dou, Z.; Ma, Z.; Zhang, X.; Du, P.; Zuo, X.; and Jiang, H. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, 2780–2791. ACM.

Zhu, Y.; Nie, J.; Su, Y.; Chen, H.; Zhang, X.; and Dou, Z. 2022. From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, 2784–2794. ACM.