Heterogeneous Graph-based Context-aware Document Ranking

Shuting Wang
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
wangshuting@ruc.edu.cn

Zhicheng Dou
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
dou@ruc.edu.cn

Yutao Zhu
University of Montreal, Montreal Quebec, Canada
yutaozhu94@gmail.com

ABSTRACT

Users' complex information needs usually require consecutive queries, which results in sessions with a series of interactions. Exploiting such contextual interactions has been proven to be favorable for result ranking. However, existing studies mainly model the contextual information independently and sequentially. They neglect the diverse information hidden in different relations and structured information of session elements as well as the valuable signals from other relevant sessions. In this paper, we propose HEXA, a heterogeneous graph-based context-aware document ranking framework. It exploits heterogeneous graphs to organize the contextual information and beneficial search logs for modeling user intents and ranking results. Specifically, we construct two heterogeneous graphs, *i.e.*, a session graph and a query graph. The session graph is built from the current session queries and documents. Meanwhile, we sample the current query's k-layer neighbors from search logs to construct the query graph. Then, we employ heterogeneous graph neural networks and specialized readout functions on the two graphs to capture the user intents from local and global aspects. Finally, the document ranking scores are measured by how well the documents are matched with the two user intents. Results on two large-scale datasets confirm the effectiveness of our model.

CCS CONCEPTS

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

KEYWORDS

Context-aware Document Ranking; Heterogeneous Graph; Related Sessions

ACM Reference Format:
Shuting Wang, Zhicheng Dou, and Yutao Zhu. 2023. Heterogeneous Graph-based Context-aware Document Ranking. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570390

1 INTRODUCTION

With more complex information needs, users' search behaviors have evolved from one-shot queries to multiple consecutive queries. Such a series of queries and the corresponding user activities (*e.g.*, clicked documents) are often regarded as a *search session* [2]. Exploiting the contextual information in a session (such as user historical behavior or interaction) is known to be effective for understanding the user's search intent and ranking the search results [2, 52, 56, 58].

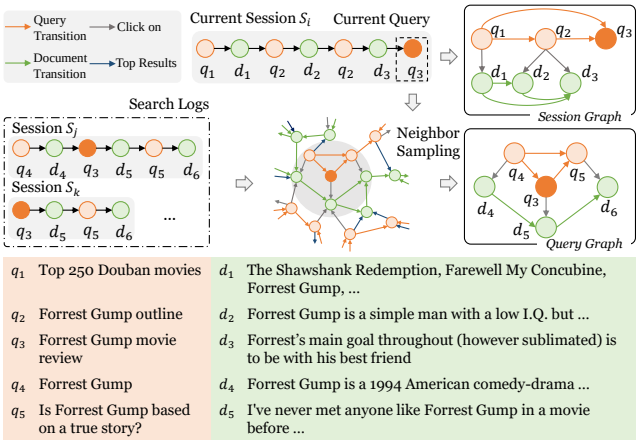


Figure 1: An example processed by our method. For the current session S_i, our method represents queries, documents, and their relationship by a heterogeneous session graph. Besides, queries/documents in other sessions (such as S_j and S_k) related to the current query q_3 are sampled to construct a heterogeneous query graph. We measure the context-aware relevance of a document based on these two graphs.

Various methods have been proposed for utilizing contextual information in a search session. Early studies designed some heuristic methods to capture the influence of contextual information on document ranking [42]. Later, many studies employed neural networks (such as RNNs) to learn user intents from sequential session behaviors [1, 9, 44]. With the development of large-scale pre-trained language models, recent studies have adopted these advanced models to identify users' search intent from their behaviors and have achieved significant improvement [41, 56].

The majority of existing methods formulate context-aware document ranking as a sequential modeling problem, that is, they treat each search session as a **sequence**. Although they have achieved promising performance, in this paper, **we argue that it would be better to represent a session in a heterogeneous graph**. This is because there are various relations between queries and documents in a session, and they convey information in different ways. We think these different relations should be modeled discriminately, and they have been overlooked by existing sequential methods. We illustrate this with some examples. As shown in Figure 1, considering the current session $\{q_1, d_1, \cdots, d_3, q_3\}$, the transition between

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM '23, February 27-March 3, 2023, Singapore, Singapore
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570390

q_1 and q_2 reflects the way the user's search intent changes from top-rated movies to a specific movie (*Forrest Gump*), while the click action of d_2 under q_2 reflects how the user's query is satisfied by d_2. It is evident that mixing together such query transition and click relations will disturb the model in distinguishing different user intentions behind them. Furthermore, the structures of different relation combinations can present abundant information. For example, combining query transition and click relations can reveal the path of how the user's information needs change and are satisfied by the clicked document. The combination of click relations (*e.g.*, $d_2 \leftarrow q_2 \rightarrow d_3$) reflects that both documents meet the query intent in different aspects. **Heterogeneous graphs can naturally represent various relations by multiple edge types and capture the structural information through graphs, hence they would be superior to sequences in modeling a session.**

Another shortcoming of existing methods is that they treat each search session as an **independent** sample. In other words, only the queries and documents in the current session are used for boosting the ranking of a subsequent query. In fact, **other related sessions in search logs can also provide useful knowledge for understanding the current query and improving the ranking**. As shown in Figure 1, the current query q_3 "Forrest Gump movie review" also appears in other sessions (*e.g.*, S_j and S_k). If we check the movie review document d_5 by the click relation, we can better capture the user's information need; and if we know that the query q_5 (whether the movie is based on a true story) is often issued after q_3, we can infer the user's other potential interests. Such supplementary and diverse information from other sessions in search logs is helpful and critical for understanding a query, but it is neglected by existing methods. In this paper, **we propose to represent the query-related information contained in related sessions by a heterogeneous graph**, similar to the way we propose to represent a single session.

Consequently, we propose a **HE**terogeneous graph-based model for conte**X**t-**A**ware document ranking, which is called HEXA. We build two heterogeneous graphs, *i.e.*, a **session graph** and a **query graph**, to capture the user intents from different perspectives. (1) For the **session graph**, we view the query and document as two types of nodes due to their different natures. Three directed edge types are considered to connect nodes, *i.e.*, query transition, document transition, and click-through. In detail, we link any two queries that occurred in the same session by a directed transition relation to capture the long dependency of the user intent transfer. Another similar relation is also introduced to connect clicked documents within a session to learn the user intent evolution in a different perspective. Each query is linked with its clicked documents to model the intent satisfaction relation. (2) For the **query graph**, we use the same graph schema as the session graph. Due to the sparsity of click-through of search logs, we further introduce a top result relation between queries and top returned documents. Based on the four directed edge types, we transfer search logs into a global graph. The query graph is built by sampling the current query's k-layer neighbors with our developed sampling method from the global graph. Therefore, the understanding of the current query can be enhanced by the supplementary information from other sessions. We illustrate the two graphs in Figure 1.

After graph construction, we apply heterogeneous graph neural networks to compute node representations. Different readout functions are devised for different graphs to learn user intents. Finally, we compute the similarity between candidate documents and the obtained user intent representation and rank the documents accordingly. We conduct experiments on two large-scale datasets, AOL and Tiangong-ST, and the results show that our model significantly outperforms state-of-the-art methods, demonstrating the effectiveness of modeling search sessions by heterogeneous graphs.

Our main contributions are three-fold:

(1) We exploit heterogeneous graphs to model different roles of queries and documents in a session and differentiate their relations.

(2) We build two heterogeneous graphs, which capture user intents within a search session and exploit supplementary information from other sessions, to enrich the intent representation.

(3) We propose a context-aware ranking framework based on these heterogeneous graphs. Experimental results on two real search logs verified both the effectiveness and efficiency of our method.

2 RELATED WORK

2.1 Context-aware Document Ranking

It has been verified that contextual information of a session search is conducive to user intent modeling. Early studies leveraged statistical features and heuristic algorithms to quantify the contextual information and characterize the user intent [42, 49, 52]. However, such methods heavily rely on human experience, thus limiting the application in various retrieval tasks. Thereafter, researchers started to build predictive models for learning user intent. For example, hidden Markov model and Reinforcement learning were introduced to model the evolution of user intent [3, 15, 16, 30].

With the development of deep learning, numerous neural network-based approaches have achieved great success. For instance, recurrent neural networks (RNNs) were used to represent user behavior sequences and yielded positive results in both context-aware document ranking and query suggestion tasks [9, 44]. Researchers further discovered that jointly learning document ranking and query suggestion can boost the performance of both tasks [1, 2]. Recently, large-scale pre-trained language models, *e.g.*, BERT [11], have exhibited satisfying performance in various NLP and IR tasks [5, 13, 14, 23, 31, 32, 55, 57]. In context-aware document ranking, BERT has also achieved significant improvement with the help of additional behavior structures or contrastive learning tasks [41, 56].

Though promising results have been obtained, the above studies use each piece of user behavior in search logs as a single sequence to train the model. Different from them, we propose using heterogeneous graphs to represent user behavior sequences where queries/documents and their relations are modeled differently.

2.2 Application of Graph Structure in IR

Graphs are widely used in data mining as they can naturally represent structured knowledge. Based on whether involving diverse node and relation types, graphs can be roughly categorized into homogeneous graphs and heterogeneous graphs. The homogeneous graph considers all relations and nodes to be of the same types and has numerous applications in the IR area. For instance, HITS [33] and PageRank [35] were proposed to measure the importance of the

documents based on their link relationships; click-through graphs are employed in many IR tasks [10, 22, 27, 51]. Recently, graph neural networks (GNNs) have shown considerable potential in modeling graph-structured data. Homogeneous GNNs [18, 24, 46] have also been applied to many IR tasks [7, 26, 28], and have exhibited their advantages of encoding the structural information. Some session-based recommendation algorithms employed GNNs to model user intent based on the graph structure of the session items [8, 36, 40, 43, 48].

In many practice scenarios, graphs typically consist of heterogeneous nodes and relations with different semantics; yet, the homogeneous GNNs are insufficient for exploring such potential information. Several studies [20, 47, 54] therefore developed heterogeneous graph neural networks (HGNNs) to address this problem. Recent studies [21, 25, 37] have exploited HGNNs on recommendations to simulate various relations between the items, attributes, and users. They focus primarily on the heterogeneity of item attributes rather than the different relationships between search behaviors. In contrast to these studies, in this work, we use heterogeneous graphs and HGNNs to learn the user intent by capturing the semantics of search behavior from both the contextual information within a session and the entire search log.

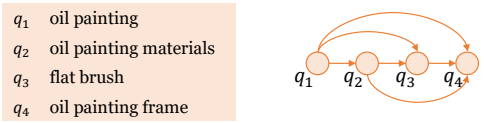
3 OUR PROPOSED METHOD

Context-aware document ranking aims at learning user intent from contextual behaviors and improving ranking quality. However, existing methods mainly focus on encoding the independent sequence of the current session, while neglecting the information hidden in the complex inside structure of the session and useful signals from other sessions. In this paper, we leverage heterogeneous graphs to capture the rich information contained in both the current session and similar sessions, to help better understand the current intent and improve result ranking in the end.

3.1 Problem Definition

The problem of context-aware document ranking has been extensively studied in existing works [2, 41, 56]. We briefly formulate the task as follows. We denote the user's search behavior of a session as a sequence of M queries $\mathcal{S}_r = \{q_1, \cdots, q_M\}$. Each query q_i has a list of candidate documents $\mathcal{D}_i = \{d_{i,1}, \cdots, d_{i,n}\}$ with binary click signals ($y_{i,j} = 1$ if clicked). Each query q_i is represented by the original text string submitted to the search engine, while each candidate document $d_{i,j}$ is represented by its text content. All queries are ordered according to their issued timestamps. For a specific query q_i, we denote all its previous queries $\{q_1, \cdots, q_{i-1}\}$ and the corresponding clicked documents as its search context.[1] With the above notations, the context-aware document ranking task is defined as: reranking the candidate document set $\mathcal{D}_i$ of query q_i based on its search context so as to rank the clicked document as high as possible, and each candidate document is scored by its relevance with respect to the current query and its context. In this paper, we propose taking more information from other sessions related to q_i in search logs as a supplement to the context.[2]

[1]The first query q_1 does not have a search context.
[2]We only use sessions from training set as search logs to avoid data leakage in inference



(a) A series of issued query (b) The soft transition

Figure 2: An example of soft transition.

3.2 Overview

The overall structure of our model is shown in Figure 3. First, we organize the user's contextual behaviors within the current session as a **session graph**. Meanwhile, given the current query and predefined edge types, we sample the query's k-layer neighbors from search logs by our developed sampling method to form the **query graph**. Then, we employ the Heterogeneous Graph Transformer (HGT) and different readout functions on both graphs to model the user intents from two perspectives, respectively. Finally, we compare each candidate document with the two intent representations. The resulting similarity scores are combined with the score calculated by the sequential model, to yield the final ranking score.

3.3 Heterogeneous Graphs Construction

A heterogeneous graph can model complex relations and structures, hence is more representative than a homogeneous graph when involving diverse node and relation types. Formally, the definition of a heterogeneous graph is $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R}\}$, where $\mathcal{V}$, $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{R}$ denote the sets of the nodes, edges, node types, and edge types respectively. The type mapping functions are presented as, $\tau(v) : \mathcal{V} \mapsto \mathcal{T}, \phi(e) : \mathcal{E} \mapsto \mathcal{R}$. Following [12], $\mathcal{S} = \{\mathcal{T}, \mathcal{R}\}$ is called the graph schema of a heterogeneous graph where $|\mathcal{T}| + |\mathcal{R}| > 2$.

In this section, we will introduce how to construct our two heterogeneous graphs, *i.e.*, the query graph G_q and the session graph G_s. They are derived from the similar graph schema.

3.3.1 Graph Schema. Our graph schema $\mathcal{S}$ includes two node types, *i.e.*, query and document. The edge types come from three aspects, *i.e.*, query-query, query-document, and document-document. Different edge types represent different relations between nodes.

Query-query. Users usually perform multiple interactions with the search engine to satisfy their vague and changeable information needs. Thus, the transition relation between queries is favorable for mining the user search intent. A straightforward method is to connect adjacent queries. However, merely considering such a transition is hard to capture long dependence between queries and makes the model sensitive to noisy queries. As shown in the example of Figure 2 (a), the query q_4 "Oil painting frame" is more relevant to q_1 "Oil painting" and q_2 "Oil painting materials" than q_3 "flat brush". Therefore, we define a soft transition relation, *i.e.*, $\vec{r}_{\text{stq}}$, for all query pairs in the same session – a previous query is connected to all future queries, which can be described as $(q_i, q_{i+k}, \vec{r}_{\text{stq}})$, as shown in Figure 2 (b). We believe this edge type is more general and robust than adjacent transition.

Query-document. For the large-scale search log, the click-through is a commonly used evidence of relevance between queries and documents [10, 27]. As a result, we consider the click relation $\vec{r}_{\text{c}}$ between queries and their clicked documents *i.e.*, $(q, d, \vec{r}_{\text{c}})$. Due to the sparsity of the click-through of the entire search log, we

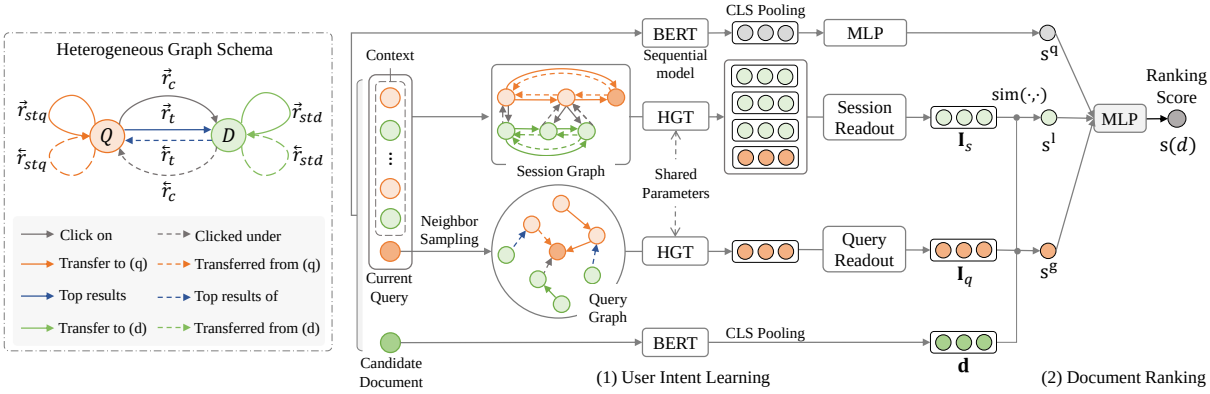


Figure 3: The architecture of our proposed model and graph schema.

further introduce a top result relation, $\vec{r}_{\text{t}}$, between queries and their top returned results for the query graph to provide more abundant relevance signals. It can be denoted as $(q, d, \vec{r}_{\text{t}})$.

Document-document. Queries and documents reflect the user's information need from two perspectives, where the former is usually a short and vague description, and the latter contains specific and sufficient information about the query. We deem that the clicked document transition can also reflect the user's intent evolution, which is complementary to query transition. Similar to the query pair, we introduce the soft transition relation $\vec{r}_{\text{std}}$ for clicked document pairs in the same session as $(d_t, d_{t+k}, \vec{r}_{\text{std}})$, where the subscript indicates the order of the clicked documents.

It is worth noting that all relations above are asymmetrical, thus the edges in the graph are directed. This is because the message passing in opposite directions has different meanings. Taking the click relation as an example, the information flows from documents to queries is more concrete and detailed, while the opposite direction brings more general and critical descriptions from queries to documents. Consequently, we further introduce the corresponding reverse edge types for these directed relations to model the message passing in different directions, *e.g.*, $(q, d, \overleftarrow{r}_{\text{c}})$. Formally, our heterogeneous graph schema can be defined as $\mathcal{S} = \{\mathcal{T} = \{Q, D\}, \mathcal{R} = \{\vec{r}_{\text{x}}, \overleftarrow{r}_{\text{x}} | \text{x} \in \{\text{stq, c, t, std}\}\}\}$, which is visualized in the left of Figure 3.

3.3.2 Session Graph. The session graph G_s is built on queries and documents within the current session, which provides a *local* view of the search intent. Considering that contextual behaviors reflect the current user's information needs, the top results of the session queries that are ignored by the user may imply that the current user is not interested in them. Thus the top results of current session queries are hard to provide valuable relevance signals. Therefore, we omit it and exploit the left relations to construct the session graph based on the session queries and documents.

3.3.3 Query Graph. The query graph G_q is designed to expand the current query with relevant queries or documents in search logs, which provides a *global* view of the query intent. Specifically, we view queries/documents containing identical content as a query/document node. Based on our defined edge types, we convert the search logs into a global graph. To consider the degree of association between connected nodes, the number of times an edge appears in search logs determines the edge weight. As the global graph is usually very large, it is impractical to process it online. Thus, we choose to offline store the global graph and sample the query's k-layer neighbors from it to build the query graph.

Weighted Neighbors. To sample a high-quality query graph, we distinguish the importance of neighbors for target nodes. Specifically, for edge $(v_i, v_j, \vec{r})$, where v_i is the source node, v_j is the target node and $\vec{r}$ is the edge type (when the edge type is $\overleftarrow{r}$, source and target nodes are switched), we denote the edge weight as $\eta(v_i, v_j, \vec{r})$. , representing how many times the connections between the nodes have appeared in the search log. For the target node v_j, the weight of its neighbor v_i on relation $\vec{r}$, $w(v_i|v_j, \vec{r})$, is the normalization of the edge weight:

$$w(v_i|v_j, \vec{r}) = \frac{\eta(v_i, v_j, \vec{r})}{\sum_{v_k \in \mathcal{N}_{\vec{r}}(v_j)} \eta(v_k, v_j, \vec{r})}. \tag{1}$$

$\mathcal{N}_{\vec{r}}(v_j)$ is the neighbor set of v_j connected by the edge type $\vec{r}$.

Graph Sampling Method. Given the issued query q, we sample its k-layer neighbors from the global graph as the query graph G_q. Unfortunately, the traditional sampling method based on for-loop introduces a large time overhead. Concretely, we set the number of sampled neighbors for the node at layer l is $m \times e$, where m is the number of edge types connected to the node, and e is the number of sampled neighbors in each edge type. For a batch nodes of size B, the time complexity of the k-layer neighbor sampling is $O(B \sum_{l=1}^{k} (me)^{l-1})$. To accelerate the sampling process, we devise a batch sampler with a new storage form of the global graph. Specifically, we allocate C neighbor slots for each node on each edge type. These slots are filled by neighbors based on their weight:

$$Occ(v_i|v_j, \vec{r}) = \left[C \times w(v_i|v_j, \vec{r})\right], \tag{2}$$

where $[\cdot]$ is the rounding operation. By this means, we obtain a tensor $\mathcal{M}$ with size $|\mathcal{R}| \times |\mathcal{V}_l| \times C$ as the global graph, where $\mathcal{V}_l$ is the node set of the global graph. When sampling, we take out a tensor with size $|\mathcal{R}| \times B \times C$ from $\mathcal{M}$ based on the target nodes in the batch, then uniformly sample e indexes in the last dimension to acquire e neighbors for all nodes in one time. The k-layer neighbors can be obtained by repeating the above process k times, and the overall time complexity is reduced to $O(k)$.

3.4 Modeling User Intent Based on Graphs

Given the session graph G_s and the query graph G_q, we apply heterogeneous graph neural networks (HGNNs) to them for learning node representations. Then, different readout functions are devised for the two graphs to capture user intents from different perspectives. For clarity, the obtained intent representations are called session intent and query intent, *i.e.*, $\mathbf{I}_s$ and $\mathbf{I}_q$, respectively.

The Heterogeneous Graph Transformer (HGT) [20] is a recently proposed method that can model different relations and capture flexible heterogeneous structural information. We adopt this advanced network to process our graphs. Such a choice is flexible, and HGT can be replaced by any other HGNNs.

3.4.1 Learning Session Intent. With the session graph composed of contextual queries, documents, and their link edges, we exploit HGT to learn the representation of queries and documents in the graph. Since the user intent transition is hidden in the internal structure of the session queries and documents, a single node is inadequate to represent the complete user intent in the session. Thus, we design a session readout layer to learn the representation of the entire graph and capture the user's session intent.

Session Readout. The node heterogeneity makes it hard to capture reasonable session intent by aggregating session queries and documents simultaneously. Moreover, because the previously clicked documents reflect the user's potential information needs concretely, after the process of HGT, their node representations are refined by the rich structural information of user intent transition and query information of the session. Therefore, we aggregate their node representations to compute the overall session intent:

$$\mathbf{I}_s = \sum_{i=1}^{c} \alpha_i \mathbf{d}_i; \quad \mathbf{d}_i = \mathrm{HGT}(G_s)[d_i], \tag{3}$$

where c is the number of clicked documents, α_i is the document weights, and $\mathbf{d}_i$ is the updated node representation of the document d_i. The weight α_i is yielded by the soft attention mechanism [37] based on the current query's node embedding, $\mathbf{q}$:

$$\alpha_i = \mathrm{Softmax}_i\left(\gamma_i\right); \quad \gamma_i = \mathbf{v}^T \sigma(\mathbf{W}^a \mathbf{q} + \mathbf{W}^b \mathbf{d}_i + \mathbf{b}^a), \tag{4}$$

where $\mathbf{v} \in \mathbb{R}^{h\times 1}$, $\mathbf{W}^a, \mathbf{W}^b \in \mathbb{R}^{h\times h}$, and $\mathbf{b}^a \in \mathbb{R}^{h\times 1}$ are parameters, and h is the dimension of the node representation.

3.4.2 Learning Query Intent. Considering that the query graph is centered on the current query and includes the query's k-layer neighbors, applying the HGT to the query graph can sufficiently aggregate high-order neighbor features and structural information to the current query's node representation. We believe such a representation can present the user's query intent accurately. Consequently, we directly readout the node representation of the query q as the query intent $\mathbf{I}_q$ representation, namely query readout:

$$\mathbf{I}_q = \mathrm{HGT}(G_q)[q], \tag{5}$$

3.4.3 Processing Steps of HGT. Inspired by the Transformer encoder [45] the HGT adopts the self-attention mechanism to aggregate neighbor features into target nodes. Particularly, the projection weights of HGT are specific to node and edge types for modeling their heterogeneity. We state its steps below.

Attention Mechanism. Take the triplet $(s, t, \vec{r})$ as an example, where s, t denote the source and target nodes, and $\vec{r}$ is the edge type. The attention value is calculated as:

$$\mathrm{Attn}(s, t, \vec{r}) = \mathrm{Softmax}_{s\in\mathcal{N}(t)}\left(\left(K(s)\mathbf{W}^A_{\vec{r}} Q(t)\right) \cdot \frac{\mu_{\vec{r}}}{\sqrt{h}}\right), \tag{6}$$

where $\mathcal{N}(t)$ is the set of node t's neighbors from all relations; $\mathbf{W}^A_{\vec{r}}$ is an edge type-specific matrix (different edge types have different parameters) for modeling the edge heterogeneity; $\mu_{\vec{r}}$ are trainable parameters to capture the prior importance of each edge type; and $K(s)$ and $Q(t)$ are Key and Query vectors mapped from source and target nodes. To encode node heterogeneity, $Q(t)$ and $K(s)$ are produced by node type-specific linear projections $\text{K-Linear}_{\tau(s)}$ and $\text{Q-Linear}_{\tau(t)}$ as:

$$K(s) = \text{K-Linear}_{\tau(s)}\left(H^l[s]\right), \quad Q(t) = \text{Q-Linear}_{\tau(t)}\left(H^l[t]\right).$$

$H^l[x]$ denotes the node x's representation after l layer HGT, and $H^0[x]$ is initialized by the pre-trained BERT embeddings.

Massage Passing. To model diverse semantics conveyed in different types of edges and nodes, the massage passing is also sensitive to the graph heterogeneity, which is defined as:

$$\mathrm{Message}(s, t, \vec{r}) = \text{V-Linear}_{\tau(s)}\left(H^l[s]\right)\mathbf{W}^M_{\vec{r}}. \tag{7}$$

Similar to the attention mechanism, $\text{V-Linear}_{\tau(s)}(\cdot)$ and $\mathbf{W}^M_{\vec{r}}$ are used to learn the heterogeneity of node and edge types.

Nodes Updating. Given attention values and message features, the information from source nodes is aggregated as follows:

$$\tilde{H}^{l+1}[t] = \sum\nolimits_{s\in\mathcal{N}(t)} \mathrm{Attn}(s, t, \vec{r}) \cdot \mathrm{Message}(s, t, \vec{r}). \tag{8}$$

It is used to update the representation of the target node by:

$$H^{l+1}[t] = \sigma\left(\text{A-Linear}_{\tau(t)}\left(\tilde{H}^{l+1}[t]\right)\right) + H^l[t], \tag{9}$$

where the linear projection $\text{A-Linear}_{\tau(t)}(\cdot)$ is applied to map the aggregated vector $\tilde{H}^{l+1}[t]$ back to the type-specific space of the target node. σ is an activation function.

The multi-head attention is employed to capture multi-granularity features from different heads. We conduct a k-layer HGT on the graphs and view the $H^k[x]$ as the node x's representation.

3.5 Candidate Document Ranking

With the query intent modeling user intent from global search logs and the session intent capturing user intent from the local session, the ranking score of the candidate document d is measured by how well it matches the user intents. Specifically, we first calculate two types of ranking scores, *i.e.*, $\mathrm{s}^{\mathrm{g}}(d)$ and $\mathrm{s}^{\mathrm{l}}(d)$, by dot-production as:

$$\mathrm{s}^{\mathrm{g}}(d) = \mathbf{I}_q^\top \mathbf{d}, \quad \mathrm{s}^{\mathrm{l}}(d) = \mathbf{I}_s^\top \mathbf{d}. \tag{10}$$

where $\mathbf{d}$ is the document vector produced by the BERT, which is the same model used to initialize the node vector. Since BERT has achieved great performance in sequential modeling [41, 56], we retain it to model the user's session behaviors. Following [56], all queries and the corresponding documents are concatenated alternatively as a long sequence X and fed into the BERT. The output of the `[CLS]` token is used to compute the matching score:

$$\mathrm{s}^{\mathrm{q}}(d) = f(\mathrm{BERT}(X)_{[\mathtt{CLS}]}), \tag{11}$$

Table 1: The statistics of two datasets. We abbreviate "Query", "Document", and "Session" to "Qry", "Doc", and "Sess".

Items	AOL			Tiangong-ST		
	Train	Valid	Test	Train	Valid	Test
# Sessions	219,748	34,090	29,369	143,155	2,000	2,000
# Queries	566,967	88,021	76,159	344,806	5,026	6,420
Avg. # Qry/Sess	2.58	2.58	2.59	2.41	2.51	3.21
Avg. # Doc/Qry	5	5	50	10	10	10
Avg. Qry Len	2.86	2.85	2.9	2.89	1.83	3.46
Avg. Doc Len	7.27	7.29	7.08	8.25	6.99	9.18
Avg. # Clicks/Qry	1.08	1.08	1.11	0.94	0.53	3.65

where $f(\cdot)$ is an MLP layer. Finally, We yield the ranking score by combining the three scores through an MLP layer $\psi(\cdot)$:

$$\mathrm{s}(d) = \psi([\mathrm{s}^{\mathrm{g}}(d); \mathrm{s}^{\mathrm{l}}(d); \mathrm{s}^{\mathrm{q}}(d)]), \tag{12}$$

where [;] is the concatenation operation.

3.5.1 Optimization. Consistent with previous studies [2, 41, 56], we apply a point-wise learning-to-rank algorithm to optimize our model. The loss function is formulated as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log z_i + (1 - y_i) \log (1 - z_i) \tag{13}$$

where N is the number of training samples, where $z_i = \mathrm{sigmoid}(\mathrm{s}(d_i))$.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

4.1.1 Datasets. Following previous studies [4, 56, 58], we conduct experiments on two large-scale search log datasets, *i.e.*, AOL [38] and Tiangong-ST [6].

We use the AOL dataset provided by [2]. It contains numerous search logs grouped as sessions. Specifically, five candidate documents are provided for each query in both training and validation sets. In the test set, 50 documents retrieved by BM25 are used as candidates for each query. The candidate construction process can be referred to at [1]. There is at least one clicked document under a query. For the Tiangong-ST dataset, the session data are extracted from an 18-day search log provided by a Chinese search engine, and each query has ten candidate documents. In training and validation sets, we use the click-through labels as relevant signals, while in the test set, the dataset provides an annotated relevance score (0-4) for the last query of each session. Following [2, 56], we use the document title as its content to reduce memory load and improve efficiency. The statistics of both datasets are shown in Table 1.

4.1.2 Evaluation Metrics. We employ three common metrics to evaluate the models' performance, *i.e.*, Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position k (NDCG@k, $k = \{1, 3, 5, 10\}$). As the relevance labels provided in the Tiangong-ST dataset are human-annotated, MAP and MRR may be inaccurate for evaluation. We follow the original authors' suggestion [6] and concentrate on NDCG metrics. All evaluation results are computed by the TREC's official evaluation tool (trec_eval) [17].

4.2 Baselines

In our experiment, we compare the performance of our model with two kinds of baselines, including (1) ad-hoc ranking methods; and (2) sequence-based ranking methods.

(1) **Ad-hoc ranking.** These methods focus on the matching between the issued query and candidate documents but do not use the information from the search context.

- `ACR-I` [19] is a representation-based method that applies convolutional neural networks (CNNs) to learn the representations of queries and documents.
- `ACR-II` [19] employs CNNs on the matching map between query and document terms to better capture their interactions.
- `KNRM` [53] is another interaction-based model that captures relevance signals from the matching map by Gaussian kernels.
- `Duet` [34] combines interaction-based and representation-based features to learn more reliable ranking scores.

(2) **Sequence-based ranking.** It utilizes the session behavior sequence to learn the user intent and rank the candidates.

- `M-NSRF` [1] jointly optimizes the next query prediction and context-aware document ranking tasks by a multi-task framework.
- `M-Match-Tensor` [1] (`M-Match` for brevity) improves M-NSRF by learning contextual representations for query and document terms. It yields the ranking score by word-level representations.
- `CARS` [2] incorporates implicit feedback from contextual information to enhance both query suggestion and document ranking.
- `HBA-Transformer` [41] (`HBA` for brevity) exploits BERT and a high-level Transformer structure to learn the contextual information by CLS-pooling.
- `COCA` [56] leverages the contrastive learning to pre-train the BERT and improves its robustness for representing user behavior sequences. It is the state-of-the-art (SOTA) method for context-aware document ranking.

4.3 Implement Details

We implement our model based on PyTorch [39], and the BERT checkpoint is provided by HuggingFace [50]. For the storage of the global graph, we set the count of neighbor slots C as 100. For the query graph, we set the order of sampled neighbor $k = 2$ and we sample two neighbors for each edge type. Therefore, there are at most eight neighbors for each target node. We conduct a two-layer HGT on the session and query graphs to alleviate the over-smoothing problem. The head number of HGT is six. We pre-train the BERT used to initialize the node embedding by positive query-document pairs and in-batch negative sampling to offer high-quality node features. The BERT used for sequential modeling is initialized with the original parameters from HuggingFace. We adopt AdamW optimizer [29] to train our model. The learning rate is set as 2e-5 with a linear decay. We train our method by five epochs and the batch size is set as 64. Our code was released on GitHub via https://github.com/ShootingWong/HEXA.

4.4 Overall Results

Experimental results are shown in Table 2. We can see that our `HEXA` outperforms all existing methods. It demonstrates the advantage of our approach. Additionally, we have the following observations.

Table 2: The overall results of our model and compared baselines. "‡" indicates the model outperforms all baselines significantly in paired t-test at $p < 0.01$ level (with Bonferroni correction). The best result is emphasized by bold.

Models	AOL Dataset						Tiangong-ST Dataset			
	MAP	MRR	NDCG@1	NDCG@3	NDCG@5	NDCG@10	NDCG@1	NDCG@3	NDCG@5	NDCG@10
ARC-I	0.3361	0.3475	0.1988	0.3108	0.3489	0.3953	0.7088	0.7087	0.7317	0.8691
ARC-II	0.3834	0.3951	0.2428	0.3564	0.4026	0.4486	0.7131	0.7237	0.7379	0.8732
KNRM	0.4038	0.4133	0.2397	0.3868	0.4322	0.4761	0.7560	0.7457	0.7716	0.8894
Duet	0.4008	0.4111	0.2492	0.3822	0.4246	0.4675	0.7577	0.7354	0.7548	0.8829
M-NSRF	0.4217	0.4326	0.2737	0.4025	0.4458	0.4886	0.7124	0.7308	0.7489	0.8795
M-Match	0.4459	0.4572	0.3020	0.4301	0.4697	0.5103	0.7311	0.7233	0.7427	0.8801
CARS	0.4297	0.4408	0.2816	0.4117	0.4542	0.4971	0.7385	0.7386	0.7512	0.8837
HBA	0.5281	0.5384	0.3773	0.5241	0.5624	0.5951	0.7612	0.7518	0.7639	0.8896
COCA	0.5500	0.5601	0.4024	0.5478	0.5849	0.6160	0.7769	0.7576	0.7703	0.8932
HEXA	**0.5625**‡	**0.5727**‡	**0.4142**‡	**0.5631**‡	**0.5974**‡	**0.6279**‡	**0.7791**	**0.7734**‡	**0.7945**‡	**0.9011**‡
Improv. over COCA	+2.27%	+2.25%	+2.93%	+2.79%	+2.14%	+1.93%	+0.28%	+2.09%	+3.14%	+0.88%

Table 3: The results of ablation studies.

Models	AOL		Tiangong-ST	
	MAP	NDCG@1	NDCG@3	NDCG@5
HEXA (Full)	0.5625	0.4142	0.7734	0.7945
w/o SGM	0.5535	0.4053	0.7668	0.7821
w/o QGM	0.5506	0.4019	0.7647	0.7916
w/o INE	0.5588	0.4115	0.7598	0.7747
w/o SRL	0.5595	0.4124	0.7617	0.7823
w/ GCN	0.5496	0.4003	0.7585	0.7699
w/ Max	0.5442	0.3967	0.7534	0.7606

Table 4: The results of different sampling strategies/methods.

Models	AOL		Tiangong-ST	
	Qry Latency (s)	MAP	Qry Latency (s)	NDCG@3
HEXA (Full)	0.0548	0.5625	0.0139	0.7734
+for-loop	0.0585	0.5621	0.0169	0.7729
+uni-spl	0.0581	0.5543	0.0164	0.7673
+top-spl	0.0547	0.5606	0.0139	0.7728
COCA	0.0531	0.5500	0.0120	0.7576

(1) **Compared with all baselines, our HEXA achieves the best results**, confirming that modeling search sessions and valuable search logs based on heterogeneous graphs is effective. Generally, context-aware ranking models perform better than ad-hoc ranking methods, indicating the benefit of contextual information for understanding user intent and improving ranking. Further, the significant improvement achieved by HBA and COCA (more than 15% in terms of all metrics on AOL) implies the superiority of the pre-trained language models (*e.g.*, BERT) in capturing sequential information.

(2) **Our graph-based HEXA model significantly outperforms the state-of-the-art method COCA** (in paired t-test at p-value < 0.01). All previous context-aware document ranking approaches modeled the user's session behavior as a sequence. In contrast, our HEXA leverages heterogeneous graphs to represent the session and expand the current query in local and global views, respectively. The improvement obtained by HEXA suggests that the heterogeneous graph can more accurately represent the session, which can help determine the user intent and enhance the document ranking.

(3) It is interesting that the improvement of HEXA on the AOL is greater than that on the Tiangong-ST. The potential reasons include: (i) On Tiangong-ST, the model is trained on click labels but tested with human-annotated relevance labels. This gap makes the task considerably more difficult, but we speculate that the problem can be alleviated if there are additional training relevance labels available. (ii) Candidate documents in Tiangong-ST are returned by the modern search engine, resulting in a higher overall relevance than candidates in AOL. It requires models to discern finer-grained differences between documents and increases the difficulty of the task. Nevertheless, **our HEXA performs best on both datasets, indicating its high generalizability and broad application.**

4.5 Ablation Study

We conduct several ablation studies to investigate the effects of various modules in HEXA. The results are shown in Table 3.

(1) **Effectiveness of the two graphs.** The two heterogeneous graphs play different roles in HEXA. To study their impact, we first eliminate the session graph and construct a variant, namely "*w/o* SGM". We find the performance drops significantly. This demonstrates the importance of modeling the heterogeneous structure of session elements and their internal relations. Next, we drop the query graph and denote the variant as "*w/o* QGM", where the model only uses the current session to rank documents. The performance degradation reveals that the other relevant sessions are extremely useful for identifying the intent of the current query.

(2) **Effectiveness of encoding two graphs independently.** Our HEXA builds two graphs from different information sources, *i.e.*, contextual information and relevant search logs. Alternatively, they can be combined into a hybrid graph. We denote this variant as "*w/o* INE". The result indicates that HEXA performs worse with the hybrid graph than separately modeling the two graphs. The possible explanation is that the semantics contained in the two graphs are derived from different perspectives. The query graph captures the relevant information from the global search logs, whereas the

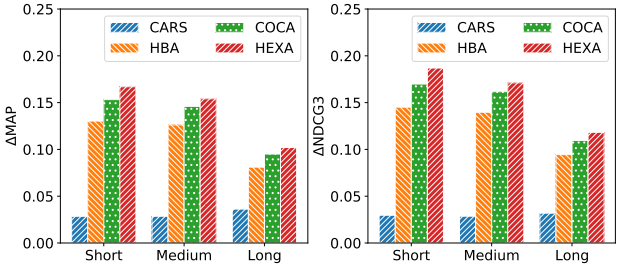


Figure 4: Results on different session lengths.

session graph displays the user's intent in the local view. Therefore, they are incompatible and may affect each other. Despite this, HEXA with the hybrid graph can still outperform COCA. It again demonstrates the limitations of existing sequential methods for representing heterogeneous objects and relations.

(3) **Effectiveness of session readout layer.** To learn the user's session intent, we devise a session readout layer to capture the embedding of the session graph. To test its effect, we replace it with the query readout layer and build a variant "*w/o* SRL". The results show that adopting query readout on the session graph is detrimental to the model's performance. We analyze the cause since the graph structure of session items represents complex user intents, which is hard to capture with a single node. Thus, the session readout layer performs better than the query readout layer.

(4) **Effectiveness of heterogeneous graphs.** To model the heterogeneity of graph structures, we employ HGT on the two heterogeneous graphs for representation. To validate its usefulness, we replace HGT with the widely used homogeneous GNN, GraphSAGE [18]. Specifically, we apply GraphSAGE with two aggregation functions, GCN and Max-pooling, denoted as "*w/* GCN" and "*w/* Max", respectively. Based on the results, all homogeneous GNNs perform worse than our model. These results confirm our assumption that different relations should be modeled separately, hence heterogeneous graphs can better capture user intents.

4.6 Efficiency and Effectiveness of Sampling

To test the efficiency of our proposed batch sampler, we compare it with the traditional for-loop-based sampling in terms of the average query latency. To verify the effectiveness of our weighted sampling, we further employ two alternative sampling strategies. One is uniformly sampling neighbors without weights (denoted as "uni-spl"), and the other is sampling the top-weighted neighbors fixedly (denoted as "top-spl"). We compare the results with COCA. Based on the results in Table 4, we have the following observations:

(1) Our batch sampler is faster than for-loop sampling, which implies its efficiency for inference on large-scale datasets. Since uniform sampling is also based on for-loop, they have similar processing costs. The efficiency of top sampling is similar to our batch sampler as we also store the top neighbors via a tensor. Besides, we also notice that the average query latency of all variants of HEXA is comparable to COCA. This is because the sequential modeling part of HEXA is the same as COCA (*i.e.*, a BERT model), and it consumes the majority of processing costs. As a comparison, our heterogeneous graph modules only require a little additional overhead, but bring significant improvement. This clearly demonstrates its superiority in context-aware document ranking. (2) For effectiveness, all HEXA variations outperform the SOTA model, confirming the power and robustness of our proposed model. In detail, uniform sampling performs worstly among all variants, and top sampling underperforms batch sampling. The reason is that the uniform sampling ignores the importance of each neighbor, which introduces considerable noise. On the other hand, top sampling uses fixed neighbors, which may lose diverse training signals and hurt model optimization.

4.7 Impact of Session Lengths

The session length controls the richness of contextual information and influences the performance of context-aware ranking models. We examine this effect by dividing the test sessions into short (length≤2), medium (length=3 or 4), and long sessions (length>4). The results of HEXA and some baselines are illustrated in Figure 4, where the y-axis denotes the improvement of the models over the strong ad-hoc ranking model Duet.

It is evident that HEXA performs better than all baselines, which implies the robustness of our model across various session lengths. Furthermore, we discover that the shorter the session length, the greater the improvement in HEXA. We attribute this to the use of the query graph, which expands the current query throughout the entire search log beyond the contextual information. These results verify the advantages of heterogeneous graphs in modeling session behaviors and the benefits of relevant search logs on ranking.

5 CONCLUSION

In this paper, we propose a heterogeneous graph-based model for context-aware document ranking, which leverages the current session and other sessions by heterogeneous graphs to capture accurate user intents. Specifically, we view queries and documents as two node types and devise four directed relations to build a trustworthy graph schema. Two heterogeneous graphs are derived from it by organizing the current session and other relevant sessions, respectively. We further develop a batch sampler to accelerate the sampling process. The heterogeneous graph transformers and two readout functions are applied to capture user intents from the two graphs. Finally, we measure the document scores by their similarity with the user intents. Experimental results prove the effectiveness and efficiency of our proposed method.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by the National Natural Science Foundation of China No. 62272467 and No. 61872370, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China NO. 22XNKJ34, Public Computing Cloud, Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. The work was partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE, and Beijing Key Laboratory of Big Data Management and Analysis Methods.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *ICLR (Poster)*.

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *SIGIR*.

[3] Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. 2009. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *WWW*.

[4] Haonan Chen, Zhicheng Dou, Qiannan Zhu, Xiaochen Zuo, and Ji-Rong Wen. 2022. Integrating Representation and Interaction for Context-Aware Document Ranking. (2022).

[5] Haonan Chen, Zhicheng Dou, Yutao Zhu, Zhao Cao, Xiaohua Cheng, and Ji-Rong Wen. 2022. Enhancing User Behavior Sequence Modeling by Generative Tasks for Session Search. In *CIKM*.

[6] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *CIKM*.

[7] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. A Context-Aware Click Model for Web Search. In *WSDM*.

[8] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling Information Loss of Graph Neural Networks for Session-based Recommendation. In *KDD*.

[9] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *SIGIR*.

[10] Nick Craswell and Martin Szummer. 2007. Random walks on the click graph. In *SIGIR*.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[12] Yuhui Ding, Quanming Yao, Huan Zhao, and Tong Zhang. 2021. DiffMG: Differentiable Meta Graph Search for Heterogeneous Graph Neural Networks. In *KDD*.

[13] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *EMNLP (1)*.

[14] John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *ACL/IJCNLP (1)*.

[15] Dongyi Guan and Hui Yang. 2014. Query Aggregation in Session Search. In *DUBMOD@CIKM*.

[16] Dongyi Guan, Sicong Zhang, and Hui Yang. 2013. Utilizing query change for session search. In *SIGIR*.

[17] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. In *SIGIR*.

[18] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

[19] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NeurIPS*.

[20] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *WWW*.

[21] Yugang Ji, MingYang Yin, Yuan Fang, Hongxia Yang, Xiangwei Wang, Tianrui Jia, and Chuan Shi. 2020. Temporal heterogeneous interaction graph embedding for next-item recommendation. In *ECML PKDD*. Springer.

[22] Shan Jiang, Yuening Hu, Changsung Kang, Tim Daly Jr., Dawei Yin, Yi Chang, and ChengXiang Zhai. 2016. Learning Query and Document Relevance from a Web-scale Click Graph. In *SIGIR*.

[23] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR*.

[24] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*.

[25] Chen Li, Linmei Hu, Chuan Shi, Guojie Song, and Yuanfu Lu. 2021. Sequence-aware heterogeneous graph neural collaborative filtering. In *SDM*. SIAM.

[26] Xiangsheng Li, Maarten de Rijke, Yiqun Liu, Jiaxin Mao, Weizhi Ma, Min Zhang, and Shaoping Ma. 2020. Learning Better Representations for Neural Information Retrieval with Graph Information. In *CIKM*.

[27] Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *SIGIR*.

[28] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *SIGIR*.

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR (Poster)*.

[30] Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win search: dual-agent stochastic game in session search. In *SIGIR*.

[31] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *WSDM*.

[32] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *CIKM*.

[33] Alberto O. Mendelzon. 2000. Review - Authoritative Sources in a Hyperlinked Environment. *ACM SIGMOD Digit. Rev.* 2 (2000).

[34] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *WWW*.

[35] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. (11 1998).

[36] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star Graph Neural Networks for Session-based Recommendation. In *CIKM*.

[37] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous Global Graph Neural Networks for Personalized Session-based Recommendation. In *WSDM*.

[38] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Infoscale (ACM International Conference Proceeding Series, Vol. 152)*.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8024–8035.

[40] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. In *CIKM*.

[41] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W. Bruce Croft, and Paul N. Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *SIGIR*.

[42] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Context-sensitive information retrieval using implicit feedback. In *SIGIR*.

[43] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-Based Social Recommendation via Dynamic Graph Attention Networks. In *WSDM*.

[44] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *CIKM*.

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

[46] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR (Poster)*.

[47] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*.

[48] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xianling Mao, and Minghui Qiu. 2020. Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In *SIGIR*.

[49] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. 2010. Predicting short-term interests using activity-based search context. In *CIKM*.

[50] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP (Demos)*. Association for Computational Linguistics, 38–45.

[51] Wei Wu, Hang Li, and Jun Xu. 2013. Learning query and document similarities from click-through bipartite graph with metadata. In *WSDM*.

[52] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. 2010. Context-aware ranking in web search. In *SIGIR*.

[53] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR*.

[54] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *KDD*.

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

[56] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *CIKM*.

[57] Yutao Zhu, Jian-Yun Nie, Yixuan Su, Haonan Chen, Xinyu Zhang, and Zhicheng Dou. 2022. From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking. In *CIKM*.

[58] Xiaochen Zuo, Zhicheng Dou, and Ji-Rong Wen. 2022. Improving Session Search by Modeling Multi-Granularity Historical Query Change. In *WSDM*.