Session Search with Pre-trained Graph Classification Model

Shengjie Ma
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
msj@ruc.edu.cn

Chong Chen
Huawei Cloud BU
Beijing, China
chenchong55@huawei.com

Jiaxin Mao*
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
maojiaxin@gmail.com

Qi Tian*
Huawei Cloud BU
Shenzhen, China
tian.qi1@huawei.com

Xuhui Jiang
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
jiangxuhui19g@ict.ac.cn

ABSTRACT

Session search is a widely adopted technique in search engines that seeks to leverage the complete interaction history of a search session to better understand the information needs of users and provide more relevant ranking results. The vast majority of existing methods model a search session as a sequence of queries and previously clicked documents. However, if we simply represent a search session as a sequence we will lose the topological information in the original search session. It is non-trivial to model the intra-session intractions and complicated structural patterns among the previously issued queries, clicked documents, as well as the terms or entities that appeared in them. To solve this problem, in this paper, we propose a novel Session Search with Graph Classification Model (SSGC), which regards session search as a graph classification task on a heterogeneous graph that represent the search history in each session. To improve the performance of the graph classification, we design a specific pre-training strategy for our proposed GNN-based classification model. Extensive experiments on two public session search datasets demonstrate the effectiveness of our model in the session search task.

CCS CONCEPTS

• **Information systems → Users and interactive retrieval**.

KEYWORDS

session search, heterogeneous information network, heterogeneous graph neural networks, graph classification

ACM Reference Format:
Shengjie Ma, Chong Chen, Jiaxin Mao, Qi Tian, and Xuhui Jiang. 2023. Session Search with Pre-trained Graph Classification Model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591766

1 INTRODUCTION

As users increasingly rely on search engines to access needed information, the search tasks tend to be more diverse and complex. In many cases, a single query is insufficient to meet a user's information needs, leading them to submit additional queries until they are satisfied or cease their search efforts. This process, known as session search [10, 15, 35], typically involves a series of queries, top candidates returned per query, and the user clicks within a short time frame. These interactions between the user and the search engine in a session are known to be valuable for multiple purposes [2, 4, 13], such as intent understanding [7], query suggestion [1], and personalized ranking [4, 13]. In this paper, we focus on how to take full advantage of the session history to optimize the ranking result for users' current query.

Previous research[1, 3, 32, 42] has sought to leverage session data to enhance document ranking. However, most existing methods model search sessions as sequences of queries and clicked documents, which may be sub-optimal in two ways.

First, modeling a search session as a sequence of interactions ignores the topological interactions among queries, documents, and the related entities and topics existing in the data. For example, as shown in Fig.1, a user typed in two queries and clicked several candidate documents per query. The text colored blue, orange, and purple represent the topics that possibly meet the user's interest. Obviously, objects covering the blue topic are relevant in the context. While the second query is more related to the first two clicked documents of the first query, which suggests that the user could be concerned about the energy market but not other markets during the Russia-Ukraine war. The last query and the corresponding clicks further confirmed this. It is worth noting that the blue topic appearing in the clicked documents under the last query was not explicitly mentioned in the last query, but could be inferred from the former query-document interaction. Besides, although the purple and the orange topic are not directly related, they are semantically related to the blue topic. Intermediaries like the blue topic could establish the semantic structural patterns or semantic topological relatedness existing among session data. As the figure

*Corresponding authors.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan.

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591766

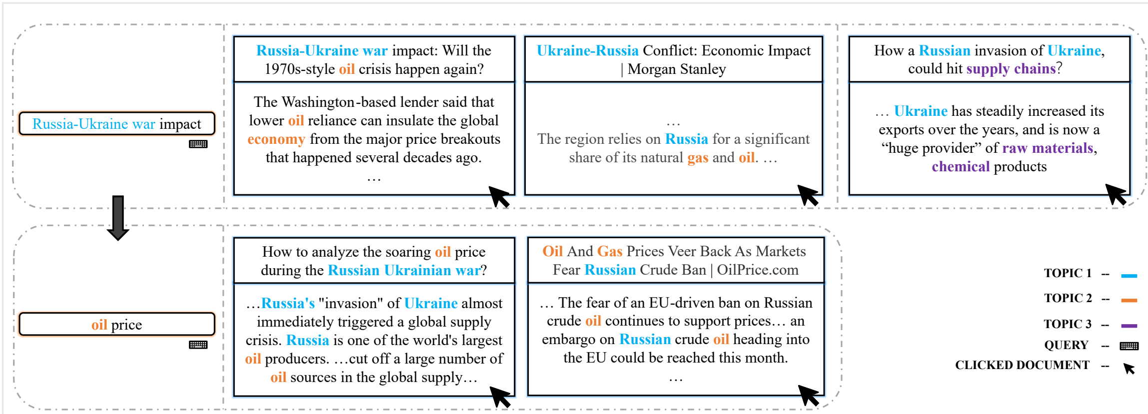

Figure 1: An example to illustrate the intra-session data associations and structural patterns in a session: Objects covering the blue topic share common context relation. The second query has more relatedness to the first two clicked documents of the first query, which suggests that the user could be concerned about the energy market but not other markets during the Russia-Ukraine war. The blue topic appearing in the clicked documents under the last query was not mentioned in the last query, which is inferred from the former query-document interaction. The purple and the orange topic are not context-related, but they are semantically linked by an intermediary — the blue topic, which shows the semantic structural patterns in session data.

shows, instead of a one-dimensional sequence, the natural form of a search log is a graph that contains comprehensive intra-session data associations as well as more complicated structural patterns, which helps reflect how user intents evolve and change.

Second, from a fine-grained perspective, different data types have different characteristics, which may have different influences on user behaviors. For instance, even the URL can provide auxiliary information for people to judge its relevance. Traditional methods fail to consider such heterogeneous features. Additionally, due to the restriction of the maximum input length, such as the 512-token limit for BERT [12], most sequence-based methods have to compromise some useful information, such as only retaining the first clicked document for each query, which can lead to an inevitable loss of valuable information.

Observing these two limitations of existing session search models, we argue that a better solution is to model a session as a graph (called *session graph*), where rich information can be modeled in a more comprehensive yet flexible way. However, leveraging the information of session graphs for document ranking is challenging. A carefully-designed framework is required to model various types of semantic information and estimate the context-dependent relevance of candidate documents.

For a more reasonable representation of session information and better encoding of the context in the search history, we propose a Session Search with Graph Classification Model (SSGC). With SSGC, we first represent a session as a heterogeneous graph with four node types (query, document, keyword, and the current query), and formulate the session search task as a graph classification task. Specifically, rather than respectively encoding the session history and candidate documents into representational vectors and computing their similarities, we include each candidate document of the current query with corresponding session history together in a session graph and build a graph classification model to predict whether the user will like or click this candidate document. As mentioned above, the session graph contains different types of nodes, we apply a heterogeneous graph pooling method [36], which can hierarchically condense the session information while considering the characteristics of different types of data, in building the graph classification model. Additionally, we develop a specific pre-training approach for our proposed session-based graph classification model. Through optimizing a local-global objective, the Deep Graph Infomax (DGI) [34] pre-training enables each node to view an arbitrarily large region of the graph, which further helps the graph classification model better leverage the global and structural information in session graphs.

Extensive experiments conducted on two public large-scale web search session datasets TianGong-ST [8] and AOL [24] demonstrate the effectiveness of our model in the session search task.

In summary, the main contributions of this paper are as follows:

- We propose a new session modeling method, which represents each candidate document and its corresponding search session as a heterogeneous graph to better utilize the comprehensive data association and complex structural patterns within the session for session search.
- We propose a novel re-ranking method for the session search task in a graph classification fashion, which to the best of our knowledge has not been proposed in previous literature.
- We design a contrastive pre-training strategy to improve the ranking performance of our proposed model SSGC.
- We conduct extensive experiments on two large-scale web search session datasets TianGong-ST [8] and AOL [24] to evaluate the proposed method. The results demonstrate the advantages of modeling session search as a graph classification task and the effectiveness of the graph pooling technology.

2 RELATED WORK

2.1 Neural Ranking with Modeling Search History

User search history is known to contain rich contextual information. Especially, it is well-known that the short-term history in a session contributes a lot to the inference of users' information needs and personalization[2, 4, 13]. Researchers have tried various attempts to exploit the user search history [1, 3, 32, 42]. Early studies tend to manually extract click-based and rule-based features from search history [3, 28, 32]. However, most of them are limited by data sparsity and manually-designed rules. With the prevalence of deep learning, researchers start to model users' search history with various solutions. Next query suggestion is an important and effective additional task in neural information retrieval, which aims to enhance the context feature modeling capacity by predicting the next query in a context-aware search scenario[21, 30]. Sordoni et al. [30] propose a hierarchical RNN encoder-decoder architecture that encodes history queries and supports query suggestions. Based on the earlier hierarchical RNN model, Chen et al. [11] apply an attention mechanism to better capture user preferences. Later, Ahmad et al. [1] propose CARS, a multi-task neural framework that can jointly learn document re-ranking and next query prediction to infer a user's hidden intent from both the history queries and clicked documents. As mentioned before, CARS is a representation-based neural model. Recently, as the incredible power of BERT [12] shines in the NLP field, many BERT-based methods are published. A common approach is to concatenate the document and query text together and feed them into the downstream tasks, where the '[CLS]' token embeds the representation of the sequence. Qu et al. [26] proposed Hierarchical Behavior Aware Transformers (HBA-Transformers), which use a BERT encoder to capture the word-level interactions between queries. Chen et al. [5] integrate the latent representation of a session and the word-level interaction between queries and documents. Zuo et al. [43] focused on extracting user intent by modeling the behavior of the user to re-formulate the query a session. Furthermore, researchers are trying to improve session search tasks from a wider perspective. Chen et al. [9] combines the click model with session search tasks. Zhu et al. [42] further improves the ability to represent history sequences by contrastive learning.

Nevertheless, in the majority of session search methods, such as RNN-based and BERT-based methods mentioned above, simply compressing a session into a linear sequence inevitably limits the model capacity. To tackle this problem, we propose Session Search with Graph Classification Model for a more reasonable representation of session information and better encoding of the context in the search history.

2.2 Graph Pooling

Pooling operations are wildly used in developing graph neural networks, which aim to hierarchically aggregate node information and extract spatial-locality information. In general, the pooling methods on graph data are classified into two categories: node sampling and node clustering. Node sampling methods[38] mainly rank all nodes by importance, then keep the top-ranked nodes and discard the others. However, as this selection process is based on a global ranking for all nodes, these methods fail to take into account the topology information within the graph. Node clustering methods mainly cluster nodes into super-nodes. For example, Diffpool [37] learns to summarize a graph by learning some assignment matrices that assign nodes to clusters (i.e. super-nodes) according to their similarity and connectivity. Based on these assignment matrices, Diffpool can hierarchically aggregate the graph information. One of the most significant advantages of this method is that the topological structures are taken into account from hierarchical views. Since the session graphs contain rich semantic structural features, node clustering pooling methods are more suitable for building our model. We will use the graph-pooling technique to build a heterogeneous graph classifier for the session search task.

2.3 Graph in Information Retrieval

Graphic structure is a very intuitive way to represent relationships. In the field of IR, it is common to use graphs to represent query-document relationships at the entire corpus level[14, 18, 19]. For example,Zhang et al. [40] build a corpus-level graph of documents by the co-click connections on click intention level, instead of document level. Zhang et al. [41] build document-level word relationships on graphs through the graph-of-word text format.

Although previous works provide a better understanding of how to model graph information effectively for IR, they cannot be applied to the session search task compatibly. There are two major unsatisfactory of these methods for the session search task: (1) Session is a short time interval user interaction process, sensitive to the serving time. However if constructing a corpus-level graph, when new data arrives, the model has to be retrained from scratch, which is time-consuming. Meanwhile, it consumes immense memory to learn on the corpus-level graph. (2) There are few designs specifically considering the characteristics of the session data, which hinders fully utilizing the rich information in the session.

Furthermore, although there are several graph methods[23, 29] for session-based recommendation system, they are also not suitable for session search due to a major gap: search is an active behavior based on queries, while the recommendation is a passive message pushing based on user profiles.

2.4 Self-surpervised Graph Neural Networks

Despite the significant achievements of graph neural networks in recent years, like most supervised or semi-supervised machine learning models, they require a substantial amount of labeled data to optimize learning objectives in order to acquire powerful expressive capabilities. Self-supervised learning methods can be broadly classified into two main categories: generation-based learning and contrast-based learning. Generative self-supervised learning methods learn the inherent characteristics of the data by making the model generate and reconstruct the input data. Contrastive self-supervised methods construct positive and negative samples from the input data, enabling the model to discriminate between positive and negative samples in the implicit representation space. These two approaches construct pre-training tasks from unlabeled input data in different ways, serving as supervisory signals. In recent years, pre-training methods have also been applied in session search, such as the contrastive learning task designed to pre-train the language model BERT in COCA [42] and a generative pre-training strategy proposed to enhance the expressive ability of the downstream model in [6]. However, the application of pre-training strategies in session search remains an area for further exploration. In this work, we design a specialized graph pre-training task that facilitates the extraction of relatedness between type-specific local features and the entire session context.

3 METHODOLOGY

Our main objective in this study is to enhance the utilization of session history for document ranking. To achieve this, besides the context of the previous queries and clicked documents, we aim to exploit the semantic topological relations and type-specific characteristics between them to improve the ranking performance in session search. In this section, we will briefly introduce the problem definition in Section 3.1 and explain how we construct the session graph to represent the search session in Section 3.2. We will then describe how we formulate the session search task as a graph classification problem and how we employ the graph-pooling technique to develop a GNN-based graph classifier for this problem in Section 3.3. Lastly, we will introduce how to pre-train the GNN to further improve the ranking performance.

3.1 Problem Definition

Prior to presenting our methodology, we will first establish the definitions and necessary notations pertaining to the task of session search. In the context of a search session, a user will engage in a continuous iteration of query submission and examination of the documents returned by the search system, until satisfaction or cessation of the search is attained. Assuming n submitted queries, the search behavior history $\mathcal{H}$ is denoted as:

$$\mathcal{H} = \{(q_1, C_1), (q_2, C_2), \ldots, (q_{n-1}, C_{n-1})\}, \quad (1)$$

where q_i is the i-th query, $C_i = \{d_{i,1}, \ldots, d_{i,m}\}$ is the list of m clicked documents of q_i and $1 \leq i < n$. The goal of session search is to re-rank the candidates in C_n for the current query q_n, where $C_n = \{d_{n,1}, \ldots, d_{n,k}\}$ represents the list of k candidates of q_n.

3.2 Session Graph

As demonstrated in Figure 1, the search history of a session possesses abundant intra-data associations and structural patterns, which can potentially aid in the inference of user intent and context-aware document ranking. To represent session data, we propose a graph-based session modeling approach referred to as Session Graph (SG), as depicted on the left side of Figure 2.

Graph Construction. As shown in the figure, a session graph consists of four types of nodes, namely the query nodes, document nodes, keyword nodes, and a special node for the current query. Given a search history $\mathcal{H}$, we represent each history query q_i ($1 \leq i < n$) and corresponding clicked documents $d_{i,j} \in C_i$ ($1 \leq i < n$) as query and document nodes, respectively. We then establish connections between each clicked document $d_{i,j}$ ($1 \leq j \leq m$) and the corresponding query q_i, as well as between every two queries that are adjacent in time. Additionally, we utilize a TF-IDF-based method[39] to extract all keywords present in the queries and documents in $\mathcal{H}$. Each unique keyword is represented as a keyword node and linked to the nodes where it appears. Furthermore, we observe that many documents are clicked on in different sessions under different queries and that similar or identical queries with the same search intentions also have common clicks in different sessions. This inter-session level of common click information can provide valuable auxiliary clues for the current search task. As such, we establish connections between each document in the current session with an external query (if available), as well as between each query and an external document that was clicked on by similar or identical queries (if available). We follow the equation suggested by Chen et al. [5] to find similar (supplemental) queries:

$$\sup(q_a \mid q_b) = \mathrm{spe}(q_a \mid q_b) + \mathrm{sim}(q_a \mid q_b), \quad (2)$$

where q_a and q_b are two queries, $\sup(q_a \mid q_b)$ is the supplemental rate of q_a, $\mathrm{spe}(q_a \mid q_b) = \frac{\mathrm{len}(q_a) - \mathrm{len}(q_b)}{\mathrm{len}(q_b)}$ when every word of q_a is in q_b otherwise $\mathrm{spe}(q_a \mid q_b) = 0$, $\mathrm{sim}(q_a \mid q_b)$ is the similarity that computed by the python class SequenceMatcher [1]. If there is no identical query, we attempt to find a supplemental query with the highest supplemental rate.

Our primary goal in this research is to construct a session graph for each candidate document, $d_x \in C_n$, of the current query, q_n, and to estimate the relevance between d_x and the session context using graph classification method. To emphasize the role of the current query and model its relation with the whole session context, we regard the current query q_n as a special type of node and link it with all the other query nodes in the history $\{q_1, \ldots, q_{n-1}\}$. For the candidate document, we do not connect it with the current query initially, as we cannot predict whether the user will click on the candidate document under the current query. Nevertheless, the candidate document node will be indirectly connected to the session graph if it shares some keywords with the session history $\mathcal{H}$, which benefits the representation of the candidate document to some extent. It is worth noting that h^{d_c} the embedding of the candidate document before the pooling layer will be retained for the MLP layer, similar to a residual network:

$$\hat{y} = MLP\left(h^{d_c}, h^q, h^d, h^k, h^{sq}\right), \quad (3)$$

where h^q, h^d, h^k, h^{sq} are summarized representation of query type, document type, keyword type and special query type respectively, and $\hat{y}$ is the ranking score of the current candidate document.

Node initialization. For each node in the session graph, we need to determine its initial node feature. For the encoder of queries and documents, we apply Transformer [33] to learn contextualized word embeddings and then apply attention pooling networks to learn the node features. The encoders of queries and documents are independent and trainable. For the keywords, we extract them from queries and documents and keep several high-ranking keywords by TF-IDF score. We use pre-trained word embeddings of keywords as their initial node features.

Formally, the session graph is denoted as $G^{(0)} = (A^{(0)}, X^{(0)})$, where $A^{(0)}$ is the adjacent matrix of the session graph and $X^{(0)}$ is the matrix for the initial features.

[1] https://docs.python.org/3/library/dilib.html

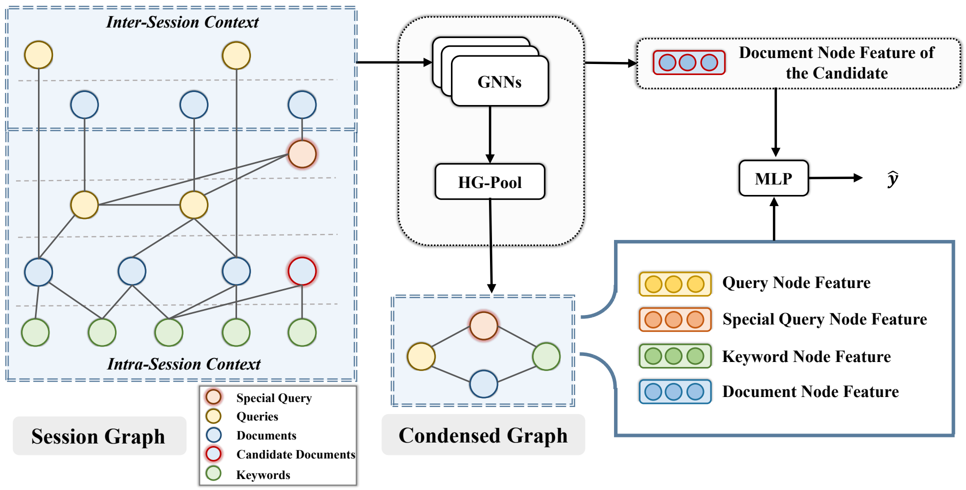

Figure 2: The architecture of Session Search with Graph Classification Model (SSGC).

3.3 Session Search as Graph Classification

To leverage rich intra-session data associations and data structural patterns in SG, We propose Session Search with Graph Classification Model (SSGC). The overall framework is shown in Fig.2. After initializing the representation of T types of nodes in the graph, we apply multiple Graph Neural Network (GNN) Layers to aggregate neighbor information. The output of the l-th GNN layer is $G^{(l)} = (A^{(l)}, X^{(l)})$, where $A^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)}}$, $X^{(l)} \in \mathbb{R}^{N^{(l)} \times \mathcal{F}}$, $N^{(l)} = \sum_{i=1}^{T} N_i^{(l)}$ is the summation of the number of each kind of nodes and $\mathcal{F}$ is the node feature dimension. We keep the node feature of the candidate document for later graph classification.

After several GNN layers, all nodes are encoded with their local structural patterns and neighbor semantic information. Due to the varying number of nodes across different session graphs, we utilize the graph pooling technique to obtain a fixed-length representation of the entire session graph. As there are four distinct types of nodes in the session graph, we apply a heterogeneous graph pooling method HG-Pool [36] to encode the characteristic of each node type.

Here we briefly introduce details of HG-pool [36]. Motivated by Diffpool [37], HG-Pool assigns a pooling assignment matrix for each type of node, which allows for encoding various type features in heterogeneous graphs. Specifically, given T types of nodes in session graphs, $A^{(l)}$ is divided into T^2 sub-matrices and $X^{(l)}$ is divided into T sub-matrices. $A_{i,j}^{(l+1)}$ denotes the adjacent sub-matrix of the i-th and the j-th node type. The formulation of HG-pool is as follows:

$$S_i^{(l)} = \text{PoolGNN}\left(A^{(l)}, X^{(l)}; \Theta_i^{(l)}\right), \tag{4}$$

$$T_i^{(l)} = \text{softmax}\left(W_i^{(l)} S_i^{(l)} + B_i^{(l)}\right), \tag{5}$$

$$A_{i,j}^{(l+1)} = P_i^{(l)\top} A^{(l)} P_j^{(l)}, \tag{6}$$

$$X_i^{(l+1)} = P_i^{(l)\top} X^{(l)}, \tag{7}$$

where $S_i^{(l)} \in \mathbb{R}^{N^{(l)} \times N_i^{(l+1)}}$ is the learned pooling matrix for the i-th node type, $\Theta_i^{(l)}$ is the parameter set of this pooling GNN, $W_i^{(l)}$ and $B_i^{(l)}$ are parameters for condensing $S_i^{(l)}$ to $T_i^{(l)}$. And $P_i^{(l)} \in \mathbb{R}^{N^{(l)} \times N_i^{(l+1)}}$ is the padding of $T_j^{(l)}$ for avoiding indexing operations, where only rows corresponding to the i-th kind of nodes are non-zero. Finally, after concatenation of $X_i^{(l+1)}$and $A_{i,j}^{(l+1)}$ according to the coordinate of the node type, we get the pooling output at $(l+1)$-th layer as follows:

$$\text{HG-Pool}\left(A^{(l)}, X^{(l)}; \Theta^{(l)}\right) = G^{(l+1)}, \tag{8}$$

$$G^{(l+1)} = (A^{(l+1)}, X^{(l+1)}), \tag{9}$$

where $A^{(l+1)} \in \mathbb{R}^{N^{(l+1)} \times N^{(l+1)}}$ is the pooled adjacent matrix and $X^{(l+1)} \in \mathbb{R}^{N^{(l+1)} \times \mathcal{F}}$ is the output node feature matrix. In this work, the HG-Pool layer clusters every type of node into one representation as shown in the 'Condensed Graph' in Fig.2, which is the globally summarized type-specified feature. Especially, as mentioned in 3.2, we use a special node to represent the current query so that its feature can be retained and avoid being integrated with other queries in the search history. Because the size of a SG is moderate (on average SG contains 28 nodes and 47 edges), we apply only one pooling layer. The final step is to train a classifier based on the previously learned graph features. The inputs of the MLP classifier are the features of the condensed graph as well as the representation of the candidate document node output by the aggregation GNN. The output of the classifier $\hat{y}$ is an estimate of the probability that the current candidate document in the session graph is relevant to the user's intent.

For model optimization, we use the cross-entropy loss as follows:

$$L_{CE} = -\frac{1}{N}\Sigma_{n=1}^{N}\left(y^{(n)} \log \hat{y}^{(n)} + \left(1 - y^{(n)}\right) \log \left(1 - \hat{y}^{(n)}\right)\right), \tag{10}$$

where N is the number of samples for training and $y^{(n)}$ is the real label of the current candidate in the training data.

3.4 Unsupervised Pre-training for SSGC

To enable each node to consider an arbitrarily large portion of the graph and discover structural similarities, we have devised a specialized pre-training strategy for the proposed session-based graph classification model, as illustrated in Figure 3.

Our approach involves obtaining graph representations through maximizing the mutual information between graph-level and local-level type-specific representations. Formally, we denote a batch of session graphs as $\mathbb{G} = \{G_1, \cdots, G_{batchsize}\}$, where $batchsize = |\mathbb{G}|$. Note that for unsupervised learning the candidate document note is excluded. Denote ϕ as the set of parameters of our graph neural networks and ϕ' as the set of parameters of the pooling layer. After all the graph neural network layers, representations of all nodes are aggregated with their local information, denoted as $\left\{h^i_\phi\right\}_{i=1}^N$, where N is the number of nodes of a graph. After the applying pooling operation, we have four type-specific representations: $\left\{h^q_{\phi'}, h^d_{\phi'}, h^k_{\phi'}, h^{sq}_{\phi'}\right\}$), which represent the graph-level query node type, document node type, keyword node type and special query node type representation respectively. Note that we slightly abuse the notation of h. The goal is to train our model by maximizing the mutual information (MI) between local-level and global-level type-specific representations through an estimator over the given graphs $\mathbb{G}$:

$$\hat{\phi}, \hat{\phi}', \hat{\psi} = \underset{\phi,\phi',\psi}{\arg\max} \sum_{G \in \mathbb{G}} \frac{1}{|G|} \sum_{u \in G} I_\psi \left(h^u_\phi ; h^{T_u}_{\phi'} \right), \tag{11}$$

where $T_u \in \{q, d, k, sq\}$ is the node type of u. The MI estimator I_ψ is modeled by a discriminator D_ψ with parameters ψ. Following the formulation of [22], here we apply the Jensen-Shannon MI estimator:

$$\begin{aligned} I_\psi \left(h^i_\phi(G); h^{T_i}_{\phi'}(G) \right) = \mathbb{E}_{\mathbb{P}} \left[-\operatorname{sp} \left(-D_\psi \left(h^i_\phi(x), h^{T_i}_{\phi'}(x) \right) \right) \right] \\ - \mathbb{E}_{\mathbb{P} \times \tilde{\mathbb{P}}} \left[\operatorname{sp} \left(D_\psi \left(h^i_\phi\left(x'\right), h^{T_i}_{\phi'}(x) \right) \right) \right], \end{aligned} \tag{12}$$

where $\operatorname{sp}(z) = \log(1 + e^z)$ is the softplus function, x is a positive sample from $\mathbb{P}$, the empirical probability distribution of the training set on the input space, and x' is the negative sample from $\tilde{\mathbb{P}} = \mathbb{P}$, the same distribution as the empirical probability distribution of the input space. In our process, negative samples are all the possible combinations of local node representation and the global representation of the specific type in all other graphs in a batch. The discriminator consists of several feed-forward neural networks, residual networks, an activation function, and a dot calculation unit.

Through optimizing a local-global type-specific objective that enables each local element to view an arbitrarily large distance of the graph, the aggregating GNNs and pooling GNNs are capable of recognizing multi-granularity structural similarities to enhance their overall ability of representation.

4 EXPERIMENTS

4.1 Dataset and Evaluation matrics

We conduct experiments on two public datasets: Tiangong-ST [8] and AOL [24] search data. Tiangong-ST [8] is a large-scale Chinese-centric web search session dataset. It consists of 147,155 refined

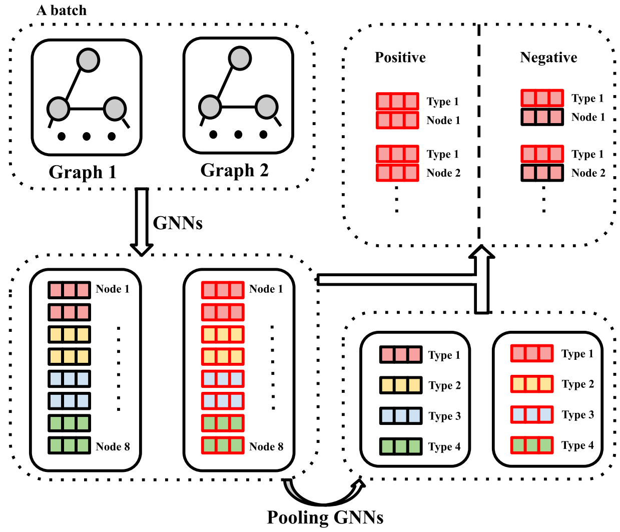

Figure 3: A simple demonstration of the pre-training strategy of SSGC. In this example, we assume the batch size is 2, and each graph has 8 nodes, containing 4 types of nodes, with 2 nodes for each type.

Table 1: The details of the datasets.

Dataset	Tiangong-ST	AOL
# Sessions	120,256	137,530
# Queries	256,513	450,796
# Unique Queries	18.125	220,924
Avg. # Query per Session	2.21	3.28
Avg. # Document per Query	10	10
Avg. # Click per Query	0.71	0.39

Web search sessions collected from an 18-day search log of Sogou, the second-largest search engine in China. A session in this dataset consists of several search interactions together with a list of clicked documents. Each interaction represents a search iteration where a user submits an independent query and receives the top 10 documents from the search engine. AOL search data [24] is a widely used dataset, which was collected from March 1st, 2006 to May 31st, 2006, with a span of 11 weeks. In our experiment, the queries within 30 minutes by the same user are regarded as a session. We reprocessed the AOL dataset following the protocol in [7]. We discard sessions that contain only one query or no click for the last query in a session. We also discard the sessions where there is no clicked document. There are 10 candidates per query and we use the title as the document content (much of the documents' content no longer exists). The statistics of these two datasets are shown in Table 1. For both datasets, we split them into the training, validation, and test sets with a ratio of 8:1:1. We use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position k (NDCG@k, $k = \{1, 3, 5, 10\}$) as evaluation matrics. All evaluation results are calculated by TREC's evaluation tool [31].

Table 2: The Experimental Results on Tiangong-ST. The best results are bold. The second-best results are underlined. Improv. indicates the statistical improvement of SSGC over the best baseline. The asterisks denote the significance level(*p < 0.05, **p < 0.01).

Models	MAP	MRR	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3528**	0.3528**	0.2212**	0.2930**	0.3498**	0.3961**
CARS	0.5856**	0.6016**	0.4388**	0.5536**	0.6145**	0.6920**
HBA-Transformer	0.6710**	0.6886**	0.5392**	0.5392**	0.7047**	0.7539**
RICR	0.6806*	0.6902*	0.5600*	0.6657*	0.7202*	0.7667*
COCA	0.7033*	0.7246*	0.6138*	0.7245*	0.7678*	0.7900*
ASE	0.7213*	0.7426*	0.6305*	0.7441	0.7866*	0.8070*
SSGC	**0.7279**	**0.7490**	**0.6371**	**0.7511**	**0.7898**	**0.8119**
Improv.	0.91%	0.86%	1.05%	0.93%	0.40%	0.61%

Table 3: The Experimental Results on AOL. The best results are bold. The second-best results are underlined. Improv. indicates the statistical improvement of SSGC over the best baseline. The asterisks denote the significance level(*p < 0.05, **p < 0.01).

Models	MAP	MRR	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.2145**	0.2308**	0.1431**	0.2517**	0.2950**	0.3773**
CARS	0.5409**	0.5422**	0.3149**	0.5660**	0.6152**	0.6398**
HBA-Transformer	0.5773**	0.5821**	0.3433**	0.5916**	0.6487**	0.6795**
RICR	0.5866**	0.5919**	0.3512**	0.6031**	0.6555**	0.6930**
COCA	0.5928*	0.5981*	0.3574*	0.6092*	0.6610*	0.6953*
ASE	0.6030*	0.6078*	0.3645*	0.6144	0.6673*	0.6989*
SSGC	**0.6092**	**0.6161**	**0.3702**	**0.6219**	**0.6746**	**0.7054**
Improv.	1.02%	1.35%	1.55%	1.28%	1.08%	0.87%

Table 4: SSGC Variant Methods Evaluated in Ablation Studies on TianGong-ST: (w/o I.) (w/o H.) (w/o H.&P.) indicate no inter-session context, no node type differentiation and no pre-training & node type differentiation respectively.

Models	MAP	MRR	NDCG@1	NDCG@3	NDCG@5	NDCG@10
SSGC	**0.7279**	**0.7490**	**0.6371**	**0.7511**	**0.7898**	**0.8119**
(w/o I.)	0.7245	0.7429	0.6316	0.7438	0.7859	0.8084
(w/o P.)	0.7232	0.7337	0.6230	0.7395	0.7834	0.8082
(w/o H.&P.)	0.6642	0.6712	0.5047	0.6340	0.6872	0.7418

4.2 Implementation Details

We use PyTorch [25] to implement our model with. We use Nvidia 3090 with 24G memory. The pre-trained Chinese word vectors are provided by Li et al. [17], which is trained in the Baidu Encyclopedia corpus. We use the Chinese tokenizer Jieba [2] in data pre-processing. We use GCN [16] as the GNN model and set the hidden dimension as 200. For the small number of nodes in a session, we apply one graph pooling layer. We regard the click as a relevance label in all the sets and use AdamW [20] optimizer to optimize the loss function defined in Eqn.8. The batch size is set as 32. The learning rate is set as 3e-5 and linearly decayed during the training. We train the model for three epochs. All other hyper-parameters are tuned based on the performance on the validation set.

[2]https://github.com/fxsjy/jieba

Efficiency Issue. Session search is an online service that needs to take into account computation efficiency. Therefore, in order to complete the graph construction and encoding within a relatively reasonable amount of time, we have made the following settings: inter-session context is generated offline beforehand and saved as a dictionary. The range of variation in session length is relatively small. To facilitate real-time construction of graphs, we have fixed the size of the session graph and the number of nodes of each type, padding where necessary and keeping the later clicks in case of exceeding. The average running time for processing a single candidate document is 1.37e-2 seconds, and re-ranking for a single query with about 80 candidate documents costs about 1.12 seconds (including graph construction and computing ranking scores), which can be improved through parallelizing the computation process for each candidate document. .

4.3 Baseline

To examine the performance of SSGC, we choose baselines in three categories, including a traditional Ad-hoc model (BM25 [27]), an RNN-based model (CARS [1]), a Bert-based sequence model HBA-Transformer [26] and two recently proposed context-aware ranking models based on pre-train strategies (COCA [42] and ASE[6]). Here we briefly introduce them as follow:

BM25 [27] is an effective and widely used classical probabilistic retrieval model.

CARS [1] is a multitask model, which learns query suggestions and document ranking simultaneously. It models the click documents in the search history with an RNN. An attention mechanism is applied to compute representations for each query and document. The final ranking score is computed based on the representation of historical queries, clicked documents, current queries, and candidate documents.

HBA-Transformer [26] concatenates historical queries, clicked documents, and unclick documents into a long sequence and applies BERT [12] to encode them. Then, a higher-level transformer structure with behavior embedding and relative position embedding is employed to further enhance the representation. Finally, the representation of the first token ("[CLS]") is used to compute the ranking score.

RICR [5] uses the overall history representation to enhance the information of queries and documents on word level.

COCA [42] is one of the state-of-the-art methods in the session search task. It is a two-step training model. First, it uses three data augmentation strategies to create similar user history sequences and applies a contrastive learning objective to a BERT encoder to pull close the representation of similar sequences and push apart different ones. Next, it finetunes the trained BERT encoder in the session search task.

ASE[6] applies a pre-trained BART with three generative tasks, including information of future queries, future clicked documents and a supplemental query, to enhance the BART-based session encoder.

4.4 Experimental Results and Analysis

Table 2 and Table 3 list the overall experimental results of baselines and our model on the two mentioned datasets. According to the results, we have some observations as follows:

First, the proposed SSGC model outperforms all baseline models, which demonstrates the overall effectiveness of our model. To be specific, ASE shows the best performance among all baseline methods. Compared to the best baseline model ASE, SSGC achieves consistent improvements on all matrics. This further shows the search graphs contain structural and topological information. Meanwhile, with our design of session graphs and the representative power of the graph-based modeling method on session graphs, rich interactions and structural information of multi-type nodes in a session can be better extracted and encoded. Additionally, the most improvement in NDCG@1 indicates that our model is good at recognizing the most relevant documents.

Secondly, the superiority of ASE and SSGC over two BERT-based models highlights the impact of unsupervised learning methods on enhancing the expressiveness and performance of the model on downstream tasks.

Interestingly, our SSGC method demonstrates a more pronounced improvement in the AOL dataset. This can be attributed to the high levels of noise present in the AOL dataset. Our approach not only leverages multi-granularity and multi-perspective features, but also utilizes a DGI-based pre-training methodology that strengthens the correlation between session-level local features and global features, thus providing enhanced robustness to noise and increased ability to identify anomalous data, which implies that our approach has better robustness.

In general, the BERT-based models outperform the RNN-based model CARS and the traditional method BM25. This observation reflects the advantage of BERT in encoding sequence data.

Furthermore, the main difference between the two BERT-based models is the contrastive learning part. The better performance of COCA against HBA-Transformer shows the effectiveness of contrastive learning.

4.5 Evaluation of Session Length

To further investigate the model performance with different lengths of session data, we equally extract sessions of different lengths ($session\_length = 2, 3, 4$ and $5 \leq session\_length \leq 10$) to construct four fixed-length testing sets. Again, note that because we are more concerned with the accuracy of the top rankings, we do not include NDCG@10 in the chart for a more concise comparison. As depicted in Figure 4, the proposed SSGC model demonstrates superior performance compared to the other two models across all session lengths. Additionally, the results indicate that session length significantly influences the model performance. Specifically, for sessions of length 3, the performance of all models is enhanced, indicating that longer sessions contain more useful information for inferring user intent. Notably, the improvement of SSGC is particularly remarkable, highlighting its strong capability in extracting features from session data. However, the performance of all models decreases when handling sessions of length 4 and from 5 to 10. The possible causes of this can be traced to two factors: firstly, the limited representation of long sessions in the training data (10,119 out of 120,256) and secondly, the increased noise in longer sessions, which often indicates a lack of satisfactory and relevant documents to meet the user's information needs. Despite these challenges, SSGC demonstrates improved performance compared to baseline methods, reflecting its strong effectiveness and resilience.

4.6 Ablation Studies

Our focus will be on the effects of variations in graph construction, graph encoding, and training methodology on the overall performance of the model, to gain a deeper understanding of the contributions of each component of our model to its overall performance and to identify areas for potential improvement.

4.6.1 ***Contribution of Heterogeneity, Pre-training and Inter-session Context.*** As shown in Table 4, it can be observed that the performance of the models has been degraded to varying extents when certain components or strategies are not utilized. Especially, in order to investigate the impact of incorporating heterogeneity in search logs on the session search task, a variant of SSGC was

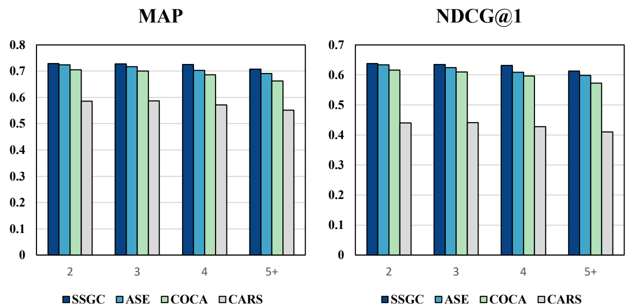

Figure 4: Visualization: Coarse comparison of intent modeling capability of four models (SSGC, ASE, COCA and CARS) on multi-length testing session data. The numbers (2, 3, 4, 5+) **indicate the fixed-length testing sets that the models are tested in, representing** $session\_length = 2, 3, 4$ **and** $5 \leq session\_length \leq 10$ **respectively.**

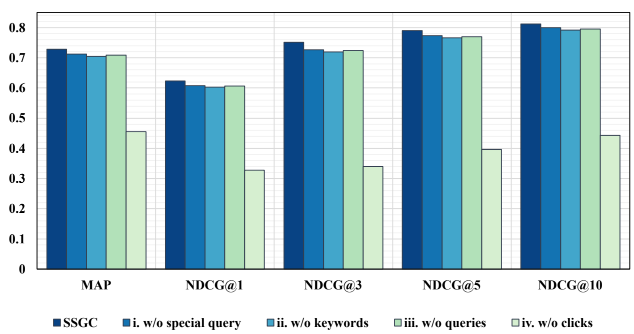

Figure 5: Visualization: Comparison of contributions of multi-type Nodes. We perform the following operations respectively: i. remove the current query node. Therefore the last query is set as the same as the history query node, not a unique query type (current query). ii. remove all of the keyword nodes. iii. remove all of the query nodes (except for the current query). iv. remove clicked document nodes and the corresponding keyword nodes.

constructed by building session graphs as homogeneous graphs and replacing the heterogeneous pooling method (HG-Pool) with diffpool [37], which is a method for homogeneous graphs. This variant of SSGC is referred to as SSGC(w/o H.&P.). Results show that compared to SSGC(w/o P.), there is a significant decrease in performance. However, when compared to other methods, SSGC(w/o H.&P.) still outperforms CARS, but does not surpass BERT-based models. From these observations, it can be inferred that incorporating type-specified features in a session leads to a notable enhancement in SSGC performance. Additionally, even when not utilizing the feature of heterogeneity and pre-train, SSGC(w/o H.&P.) still surpasses CARS in performance, highlighting the superiority of the graph-based session modeling approach over traditional recurrent neural network-based sequence modeling methods.

4.6.2 ***Contribution of Multi-type Nodes.*** To evaluate the contribution of each type of node in session graphs to the performance of the proposed model, we conduct three ablation experiments. These experiments involve replacing the current query node type with query node type, removing all keyword nodes, removing all query nodes (except for the current query), and removing clicked document nodes and their corresponding keywords. The results of these experiments are illustrated in Figure 5.

Our findings indicate a decrease in the performance of SSGC by 2.5%, 1.8%, 1.3% and 1.1% for NDCG@1, @3, @5 and @10 respectively when the current query type setting is removed and the last query is treated equally with the historical queries. Notably, there is a greater impact on the accuracy of the top-5 rankings. This observation highlights the effectiveness of extracting the type-specific features of the current query in improving re-ranking accuracy in the session search task. Additionally, our results indicate that the features of keyword nodes, query nodes, and clicked documents (together with their corresponding keyword nodes) make varying degrees of improvement in model performance. The results of *iii. w/o queries* and *iv. w/o clicks* demonstrate that the deletion of clicks impairs performance significantly. In contrast, the deletion of query nodes causes relatively little influence, which can be attributed to the rich information in clicked documents partially compensating for the loss of clues in queries. These findings indicate that clicked documents and their corresponding keywords carry crucial information for better user intent inference. Furthermore, the results also suggest that the keyword nodes in clicked documents play a important role similar to the history query nodes.

5 CONCLUSION AND FUTURE WORKS

In this paper, we propose a novel model, SSGC, for session search. Previous methods have largely overlooked the topological nature of session data and represented sessions as linear sequences. In contrast, we propose a more effective method of representing a search session as a graph and design a new graph classification model for the session search task. Additionally, we develop an unsupervised pre-training strategy to further enhance the graph representation. Our proposed SSGC method consistently and significantly outperforms state-of-the-art methods in extensive experiments, demonstrating the effectiveness of our model and the superior ability of graph-based solutions for the session search task.

ACKNOWLEDGMENTS

This research was supported by the Natural Science Foundation of China (61902209), and Beijing Outstanding Young Scientist Program (NO. BJJWZYJH012019100020098) and Intelligent Social Governance Platform, Major Innovation Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China".

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 385–394.

[2] Paul N Bennett, Ryen W White, Wei Chu, Susan T Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short-and long-term

behavior on search personalization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval.* 185–194.
[3] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management.* 1849–1852.
[4] Ben Carterette, Paul Clough, Mark Hall, Evangelos Kanoulas, and Mark Sanderson. 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.* 685–688.
[5] Haonan Chen, Zhicheng Dou*, Qiannan Zhu, Xiaochen Zuo, and Ji-Rong Wen. 2022. Integrating Representation and Interaction for Context-aware Document Ranking. *ACM Transactions on Information Systems (TOIS)* (2022).
[6] 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 *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 180–190.
[7] Jia Chen, Jiaxin Mao, Yiqun Liu, Ziyi Ye, Weizhi Ma, Chao Wang, Min Zhang, and Shaoping Ma. 2021. A Hybrid Framework for Session Context Modeling. *ACM Trans. Inf. Syst.* 39, 3, Article 30 (may 2021), 35 pages. https://doi.org/10.1145/3448127
[8] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A new dataset with large-scale refined real-world web search sessions. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 2485–2488.
[9] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. A context-aware click model for web search. In *Proceedings of the 13th International Conference on Web Search and Data Mining.* 88–96.
[10] Jia Chen, Weihao Wu, Jiaxin Mao, Beining Wang, Fan Zhang, and Yiqun Liu. 2022. Overview of the NTCIR-16 Session Search (SS) Task. *Proceedings of NTCIR-16. to appear* (2022).
[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based hierarchical neural query suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 1093–1096.
[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[13] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing search results using hierarchical RNN with query-aware attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 347–356.
[14] 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 *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* 185–194.
[15] Rosie Jones and Kristina Lisa Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management.* 699–708.
[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[17] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical Reasoning on Chinese Morphological and Semantic Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Melbourne, Australia). Association for Computational Linguistics, 138–143. http://aclweb.org/anthology/P18-2023
[18] 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 *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 795–804.
[19] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1259–1268.
[20] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7
[21] Bhaskar Mitra. 2015. Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval.* 3–12.
[22] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems* 29 (2016).
[23] 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 *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining.* 775–783.
[24] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems.* 1–es.
[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[26] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W Bruce Croft, and Paul Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1589–1592.
[27] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94.* Springer, 232–241.
[28] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.* 43–50.
[29] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining.* 555–563.
[30] 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 *proceedings of the 24th ACM international on conference on information and knowledge management.* 553–562.
[31] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 873–876.
[32] Christophe Van Gysel, Evangelos Kanoulas, and Maarten de Rijke. 2016. Lexical query modeling in session search. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval.* 69–72.
[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[34] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. *ICLR (Poster)* 2, 3 (2019), 4.
[35] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W White, and Wei Chu. 2013. Learning to extract cross-session search tasks. In *Proceedings of the 22nd international conference on World Wide Web.* 1353–1364.
[36] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-Graph: User Modeling with Heterogeneous Graph Pooling for News Recommendation. In *IJCAI.*
[37] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems* 31 (2018).
[38] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
[39] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of TF* IDF, LSI and multi-words for text classification. *Expert systems with applications* 38, 3 (2011), 2758–2765.
[40] Yusi Zhang, Chuanjie Liu, Angen Luo, Hui Xue, Xuan Shan, Yuxiang Luo, Yiqian Xia, Yuanchi Yan, and Haidong Wang. 2021. MIRA: Leveraging Multi-Intention Co-click Information in Web-scale Document Retrieval using Deep Neural Networks. In *Proceedings of the Web Conference 2021.* 227–238.
[41] Yufeng Zhang, Jinghao Zhang, Zeyu Cui, Shu Wu, and Liang Wang. 2021. A graph-based relevance matching model for ad-hoc retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4688–4696.
[42] 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 *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 2780–2791.
[43] Xiaochen Zuo, Zhicheng Dou, and Ji-Rong Wen. 2022. Improving Session Search by Modeling Multi-Granularity Historical Query Change. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining.* 1534–1542.