Effective Personalized Search With Heterogeneous Graph Based Hawkes Process

Xiang Wu, Hongchao Qin, Rong-Hua Li, Yuchen Meng, Huanzhong Duan, Yanxiong Lu, Yujing Gao, Fusheng Jin, and Guoren Wang

***Abstract*—Personalized search aims at re-ranking search results with reference to users' background information. The state-of-the-art personalized search methods often consider both the short-term search interests from current session behaviors and the long-term search interests from previous session behaviors. However, sessions in real-world search scenarios are usually very short, and a large number of sessions contain only one query, which makes it difficult to model short-term search interests. Intuitively, apart from current session behaviors, some recent historical session behaviors could also contribute to the current search interests, and the influence of these behaviors typically decays over time. Based on this intuition, we propose a novel heterogeneous graph based Hawkes process to improve the effectiveness of personalized search. Specifically, we first construct a heterogeneous graph to model multiple relations between users, queries, and documents. Then, we propose a heterogeneous graph neural network based algorithm to encode the representations of users' historical search behaviors. After that, we develop a multivariate Hawkes process to capture the influence of historical search behaviors on the current search intent. Our approach can dynamically model the influence of historical behaviors in a continuous time space. Thus, both the current session behaviors and the historical session behaviors can be utilized to characterize a more accurate current search intent. We evaluate our method using three real-life datasets, and the results show that our approach significantly outperforms the state-of-the-art methods in terms of several widely-used precision metrics.**

***Index Terms*—Hawkes process, heterogeneous graph, personalized search.**

I. Introduction

SEARCH engine is an important tool for information retrieval. Apparently, for a search engine, returning the same set of results to different users is not sensible, because individuals could have diverse search intents even though they issue the same query. As a result, tailoring the search result based on the user's personal search interest is an effective approach to improving users' search experience.

Manuscript received 11 February 2023; revised 5 December 2023; accepted 6 May 2024. Date of publication 13 May 2024; date of current version 14 March 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3301301 and in part by the NSFC under Grant U2241211 and Grant 62072034. Recommended for acceptance by Y. Yang. *(Corresponding Author: Rong-Hua Li.)*

Xiang Wu, Hongchao Qin, Rong-Hua Li, Yuchen Meng, Yujing Gao, Fusheng Jin, and Guoren Wang are with the Department of Computer Science, Beijing Institute of Technology, Beijing 100811, China (e-mail: 3120211060@bit.edu.cn; qhc.neu@gmail.com; lironghuabit@126.com; meng_yc@163.com; jfs21cn@bit.edu.cn; paulgyj@bit.edu.cn; wanggrbit@gmail.com).

Huanzhong Duan and Yanxiong Lu are with Wechat Search Application Deparment, Tencent, Beijing 100080, China (e-mail: boosterduan@tencent.com; alanlu@tencent.com).

Digital Object Identifier 10.1109/TBDATA.2024.3399606

In the literature, there exist a large number of methods that are proposed to extract users' search interests. Traditional methods [1] mainly aim to extract explicit features in search behaviors. Recent deep learning-based methods [2], [3], [4], [5], [6], [7], [8], [9], [10], however, aim to learn users' search interests via deep neural network models. The candidate documents in the returned list are re-ranked based on the relevance to the user's profile. Due to the powerful expressive ability of the deep neural network models, these deep learning-based methods have been recognized as the state-of-the-art solutions for personalized search.

In real search scenarios, a user's current search intent is often closely related to their historical search behaviors. Therefore, utilizing the temporal features in search behavior is important for personalized search. Existing methods [3], [4], [8], [11] have utilized previous session behaviors and current session behaviors for learning long-term and short-term search interests, respectively. However, in real scenarios users are not always likely to issue a series of queries in a session. As we observed in a recently-collected mobile search dataset from WeChat, more than half of the sessions contain only one query, which hampers existing methods from accurately modeling short-term search interests. To address the above problem, we argue that short-term search interests should not be limited to the current session. For example, a user may repeatedly search the result of games during the Beijing Winter Olympics in a number of sessions, and after the event closes, their interest in the Olympics gradually decays. In this case, although their behaviors spread in various sessions, they are more likely to reflect the user's short-term search interest in a period of time. In fact, this kind of phenomenon is common when a user has information need for some breaking news or hot topics. Their interest is excited by the events instantaneously, but decays over time. In this paper, different from all existing methods, we do not explicitly partition the user's search history into current and previous session behaviors by specific criteria. Instead, we make use of the Hawkes process [12] to model the influence of previous behaviors in a continuous time space.

The Hawkes process [12], which is a kind of temporal point process that can capture the time dependencies in discrete event sequences, is especially excellent at modeling the influence of previously happened events on the occurrence of a future event. Some popular methods for learning time series data, such as recurrent neural networks and Transformer [13], only consider the order of the event sequence, but ignore the interval between two events. Unlike these methods, hawkes process utilize timestamps to model temporal influences, thus can reap the benefits

from both orders and intervals. The vanilla Hawkes process oversimplifies the complex dynamics in point process [14], which hampers it from accurately modeling real-world events. Recently, the so-called neural Hawkes process [14], [15], [16] endows Hawkes process with more powerful expressive ability. However, the neural Hawkes process is a general framework for modeling event sequences [14], [15], [16], which is nontrivial to adopt it to handle specific domains [17], [18], [19] and also has not yet been used to model users search behaviors. To fill this gap, in this paper, we propose a neural Hawkes process to model users' evolving search intents so as to improve the performance of personalized search.

Although the Hawkes process can well model the influence of historical behaviors, how to effectively represent users' heterogeneous search behaviors is still a nontrivial problem, For example, "issuing a query" and "clicking a document" are two major behaviors, representing two types of relations between user and query as well as between user and document. We argue that query reflects the user's search intent, while "click" reflects that the user finds relevant information. Differentiating these two behaviors can provide a more comprehensive view on user's search interest. Furthermore, documents clicked under the same query and queries with similar returned documents often exhibit similar content features [7], which can be modeled as graphs to enhance their representations. Prior studies [1], [8] have also demonstrated that similar users are beneficial for modeling search interests, particularly when search history is limited. Formulating the relations between users with graphs could better alleviate the cold-start problem. Therefore, to more effectively capture users' search behaviors, we propose a heterogeneous search graph to model those complicated and multiple types of relations. Based on this heterogeneous search graph, we present a heterogeneous graph neural network based method for learning query-specific representations of the search behaviors. Armed with those representations, we develop a multivariate Hawkes process to model users' dynamic search intents, which will be used for improving personalized search.

The main contributions of this paper are summarized as follows. (1) We propose a heterogeneous search graph model to encode the multiple types of relations in search behaviors and a query-specific heterogeneous graph neural network algorithm to learn the representations of users' search behaviors. (2) We present a multivariate Hawkes process to model the evolving process of a user's search intent based on their historical search behaviors. To our knowledge, this is the first time that the Hawkes process is used to study the dynamic influence of long-term and recent behaviors in personalized search. (3) We conduct extensive experiments on three real-world datasets, which validate the effectiveness and superiority of our method.

II. Related Works

A. Personalized Search

Traditional methods: Personalized search has been widely studied in the literature. Traditional methods, such as [1], [20], focus on extracting click features from history logs. For example, SLTB [11] extracts multiple statistic features from search history and studies the benefits of these features to personalized search. The Open Directory Project (ODP) [21], [22] contains the classification of large numbers of websites, which can be utilized as the website features for personalized web search, but such a program is difficult to maintain with the increasing number of websites, thus automatically extracting features from documents is a better choice. Many studies [23], [24], [25] use the Latent Dirichlet Allocation (LDA) algorithm [26] to extract topic features in the documents and queries, which is then utilized for predicting the relevance between documents and queries.

Deep learning based methods: Recently, deep learning-based approaches have been widely applied to personalized search. Song et al. [27] studied the application of the Ranknet [28] on personalized search, which is a neural network trained for matching query and url pairs. More recent methods mostly adopt sequence models, such as RNNs [29] and Transformers [13], to model users' historical behavior. For example, Ge et al. [3] proposed a hierarchical RNN to generalize user profile from historical data. Moreover, a query-aware attention model is proposed to dynamically construct user profiles based on current queries. Zhou et al. [4] proposed to learn context-aware representation of current query to improve ranking quality. Their technique comprises two hierarchical transformer models, namely the query disambiguation model and the personalized language model, aiming to disambiguate the query based on its words and historical behaviors. Beyond sequential models, Lu et al. [6] conducts personalized entity linking on queries to enhance search intent representations, and construct knowledge-enhanced user profiles using memory networks to store predicted search intents and linked entities from search histories.

Apart from explicitly establishing user profiles, some methods implicitly utilize user background or various signals from search behaviors. Yao et al. [5] acknowledged the ambiguity and varying interpretations of words in queries stems from the different backgrounds of users. Hence, it incorporates personal word embeddings trained from users' search history, to improve the modeling of query intent. Zhou et al. [30] studied the effects of re-finding behaviors in personalized search with Hierarchical Memory Networks. The method explores re-finding behavior from granularity and query intent. From granularity aspects, it considers re-finding at word, sentence, and session levels. From query intent aspects, it incorporates both query-based and document-based re-finding to accommodate different user query intents. Yao et al. [31] proposed active learning based methods to enhance the selection of labeled samples with historical evaluation results. In the work conducted by Lu et al. [2], adversarial training [32] is utilized for personalized search. Their approach involves the utilization of a generator designed to create a negative document to deceive the discriminator. Simultaneously, the discriminator is trained to differentiate between positive and negative documents. Unlike all these mentioned methods, we study personalized search from a novel view by coupling the heterogeneous graph and the Hawkes Process.

B. The Hawkes Process

The Hawkes Process [12] is a temporal point process (TPP) which assumes that historical events should have positive excitation effects on the occurrence of future events. This

excitation effect is modeled by the *conditional intensity function*. Although the vanilla Hawkes process has been applied to many fields, the major limitation is that its conditional intensity function is too simple to capture many complicated real-world time dependencies. To address this problem, some recent studies propose to enhance the Hawkes process with deep neural networks. For example, the neural Hawkes Process [15] was proposed to capture historical influence with LSTM. Self-attention Hawkes Process [16] and Transformer Hawkes Process [14] were proposed to investigate the temporal influence of previous events with Transformer. The Hawkes process has been widely applied to numerous domains. For example, in dynamic graph embedding, Hawkes Process is utilized to study the neighborhood formation process of nodes on dynamic graphs [17], marco and micro dynamics in temporal network [18], temporal heterogeneous dynamic graph embedding [19] and interaction between nodes on dynamic graph [33]. In recommendation systems, Hawkes process is used to model the time intervals between sessions [34] and the evolving purchase interest of users [35]. This paper conduct studies on the influence of historical search behaviors on current search intent with hawkes process.

III. Preliminaries

In this section, we first formulate our problem, followed by a definition of our heterogenous search graph and a brief review of the Hawkes process.

A. Problem Formulation

At first, a formulation of personalized serach is given. For a user u and the current timestamp t, his/her historical search behavior is defined as $H = \{S_1, S_2, \ldots S_{t-1}\}$. Here $S_{t'} = \{q_{t'}, D_{t'}\}$ denotes the historical search behavior at the previous timestamp t', which contains the query $q_{t'}$ and the document list $D_{t'} = \{d_1, d_2, \ldots\}$ returned by the search engine, including both the clicked and the non-clicked documents.

Given a user $u's$ historical search behavior H, a current query q and a candidate document list D returned by the search engine at t, our goal is to predict a relevance score p for each document d in the return list D, which measures the personal preference of u to d. The final search results exhibited to the user are reranked so that the document with a higher score is given a higher priority.

B. The Hawkes Process

The Hawkes Process is a kind of temporal point process which models the self-excitation effect of historical events in a continuous time space. Self-excitation assumes that historical events excite the occurrence of events with the same type. This excitation effect can be modeled by the conditional intensity function. Assuming there are only one type of events, the conditional intensity function of Hawkes process is defined as:

$$\lambda(t) = \mu + \sum_{(v',t') \in H_t} \kappa\left(t - t'\right), \tag{1}$$

where $\mu \geq 0$ is a history independent parameter called base intensity. H_t is the history set before t, v' and t' represents each historical event and it's occurrence time. κ is the kernel function representing the time decay effect, which is usually an exponential function:

$$\kappa(t) = \exp(-\gamma t), \tag{2}$$

where γ is the parameter that controls the decaying speed of intensity.

The Multivariate Hawkes process: If the event sequence encompasses various types of events, Hawkes process can be extended to multivariate Hawkes process, which has the property of mutual-excitation. The mutual-excitation effect assumes that historical events could excite the occurrence of different types of events. The conditional intensity function of multivariate Hawkes process is defined as:

$$\lambda_x(t) = \mu_x + \sum_{(v',t') \in H_t} \alpha_{i,j} \kappa\left(t - t'\right) \tag{3}$$

where event x belongs to event type i and v' belongs to event type j, $\alpha_{i,j}$ is the excitation rate of event type j to event type i.

C. Definition

The Heterogeneous graph: Let $G = \{\mathcal{V}, \mathcal{E}, \phi, \tau\}$ be a heterogeneous graph, where $\mathcal{V}$ and $\mathcal{E}$ denotes the set of nodes and edges, respectively. $\phi(v) \rightarrow \mathcal{A}$ is a mapping function which maps nodes to node types and $\tau(e) \rightarrow \mathcal{R}$ maps edges to edge types.

A heterogeneous graph $\mathcal{G}$ is a heterogeneous search graph if there are three types of nodes in the graph, which are users (U), queries (Q), and the clicked documents (D). Among these three types of nodes are four types of relations, which are user-query (issue), user-document (click), query-document (match), and user-user (similar), respectively.

IV. The Proposed Method

The proposed heterogeneous graph based Hawkes process model, abbreviated as HGHP, is comprised of three key components. First, we construct a heterogeneous search graph to represent the historical search behaviors. Second, a heterogeneous graph neural network based model is applied to the constructed heterogeneous graph to learn the representation of search behaviors. Third, equipped with the representations of search behaviors, we propose a method based on multivariate Hawkes process to model the dynamic search intent based on long-term and recent search behaviors. The overall architecture of HGHP is illustrated in Fig. 1.

A. Heterogeneous Search Graph Construction

The heterogeneous search graph is constructed based on users' historical search behaviors. As we mentioned above, there are three types of nodes and four types of edges in the graph. For each user u, we first add the user u, the queries and the clicked documents in the search history to the graph. Next, we add edges between these nodes, where the user-query edge represents the user once issued such query in the search history, the user-document edge represents the user once clicked the document,

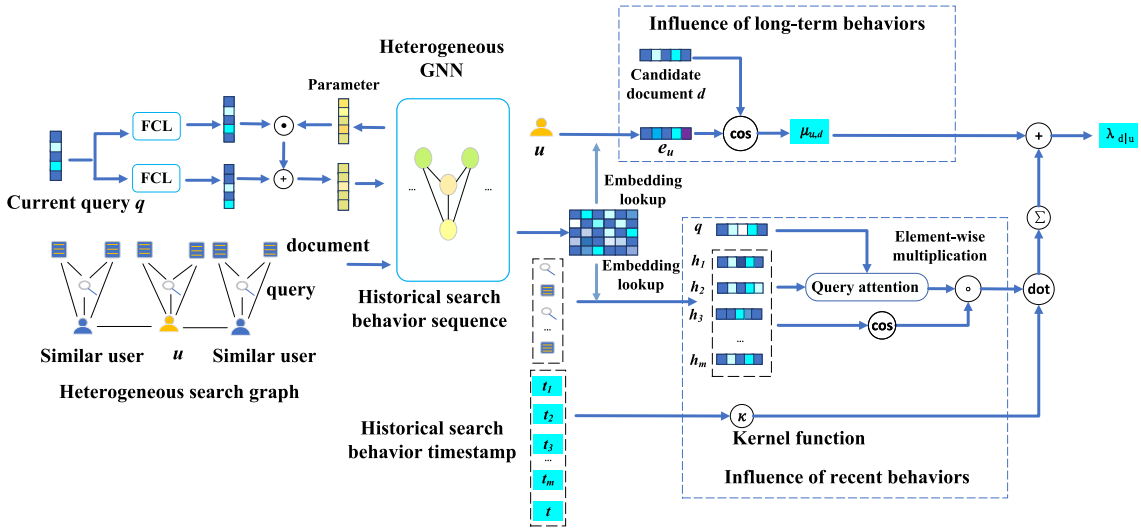


Fig. 1. The overall architecture of our HGHP model. First, a heterogeneous graph is constructed to represent historical search behaviors. The query-specific embedding of search behaviors is generated by a heterogeneos graph neural network, whose parameter is query-specified by affine transformation. Multivariate Hawkes process is then adopted to modeled the influence of long-term and recent historical behaviors of the user. $\lambda_{d|u}$ is the conditional intensity that represents the preference of u to a candidate document d.

and the query-document edge represents the document is in the returned list of the query. Note that the edges are unweighted, if a document is clicked multiple times, there will be multiple edges between two nodes.

We also add edges between two similar users into the heterogeneous search graph (i.e., user-user edges). For a user u in the heterogeneous search graph, we first find the top-k similar users of u in terms of a newly-defined meta-path based similarity metric [36]. And then, we add the edges between u and each of its top-k similar users into the heterogeneous search graph. We mainly consider joint click behaviors to measure the similarity. First, for each document in the graph, we select users who clicked the same document and add these users to the candidate similar users list. Then, we compute the similarity between two users in the candidate similar users list. Here the similarity is defined based on a concept of meta-path [36] in the heterogeneous graph. The meta-path is a predefined composite relation between two node types. In our heterogeneous search graph, the node type sequence user-document-user can be regarded as a meta path, which semantically represents a joint click behavior between two users. A node sequence p that follows one specific meta path P is a *meta-path instance*. Based on these two concepts, our similarity metric between two users is computed by:

$$sim(x,y) = \frac{2 \times \sum_{i=1}^{m_{xy}} \frac{1}{sum(p_{xy}^i)}}{\sum_{i=1}^{m_{xx}} \frac{1}{sum(p_{xx}^i)} + \sum_{i=1}^{m_{yy}} \frac{1}{sum(p_{yy}^i)}}, \quad (4)$$

where m_{xy} represents the sum of meta-path instances between x and y, and p_{xy}^i is the i^{th} meta-path instance between x and y. $sum(p_{xy}^i)$ is the total click count of the jointly clicked document (intermediate node) of p_{xy} by all users in the dataset. For each user, we rank the candidate users according to the proposed similarity metric and keep only the top-k similar users in the candidate similar user list.

Note that the difference between our method and the original meta-path method PathSim [36] is that we consider the total clicks of the intermediate nodes (documents). The reason why total clicks are considered is that there are some cases PathSim may be unreasonable in the search scenario. For example, the joint click of some common documents or websites such as "Google.com" should not be interpreted as having similar search interests while the joint click of some less popular documents should represent a higher probability of having similar search interests. As a result, we quantify the popularity of a document by the total click times of the document.

B. Search Behavior Representation Learning

To learn representations of the nodes in the heterogeneous search graph, an initial embedding is required. For queries and documents, Word2vec [37] is adopted to generate the initial embedding according to the content of the queries and the title of the documents. For users, we simply take the average of the embedding of the issued queries and the clicked documents in the search history as the initial embedding.

The key to predicting search interest is to govern the influence of search behaviors in the past. However, historical behaviors often contribute differently to search intent with respect to (w.r.t.) the current query, which motivates us to devise a query-specific learning method for search behavior representation learning. To be more specific, we want the representation of historical behaviors to be unique w.r.t. the current query q. The merit of such design is that not only the semantics of behaviors but also their relations to the current query can be considered, which better benefits search intent modeling. Our method is inspired by the Hypernetworks [38] which includes a primary network and a secondary network. The secondary network is used for generating parameters for the primary network.

Specifically, the primary network in our method is a heterogeneous graph neural network [39] for embedding the heterogeneous search graph. The secondary network is composed of fully connected neural networks which generate query-specific parameters for the primary network. The query-specific parameters contain two types of parameters, which are scaling parameter β and shifting parameter γ. Affine transformation [40] is performed on the parameters of the primary network based on these scaling and shifting parameters. Thus, the primary network can fit the current query.

The scaling and shifting parameters are computed by:

$$\beta_q^W = \mathrm{FCL}_\beta^W(\mathbf{e}_q), \tag{5}$$

$$\gamma_q^W = \mathrm{FCL}_\gamma^W(\mathbf{e}_q), \tag{6}$$

$$\beta_q^a = \mathrm{FCL}_\beta^a(\mathbf{e}_q), \tag{7}$$

$$\gamma_q^a = \mathrm{FCL}_\gamma^a(\mathbf{e}_q), \tag{8}$$

where $\mathbf{e}_q$ is the embedding of query q, and the superscript $\mathbf{W}$ and a of FCLs denote that they are generated for the parameter $\mathbf{W}$ and a of the primary network, respectively. $\mathbf{W}$ is the feature transformation parameter matrix, while a is the parameter for computing attention weights. To make the primary network query-specific, affine transformation is performed on the original parameter $\mathbf{W}$ and $\mathbf{a}$:

The representation of the behaviors is generated by the query-specific heterogeneous graph neural network, which is re-parameterized with $\mathbf{W}_q$ and $\mathbf{a}_q$. The attention weight between node i and j is computed by:

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakyReLU}\left(\mathbf{a}_q^T\left[\mathbf{z}_i||\mathbf{z}_j||\mathbf{z}_{e_{\tau(<i,j>)}}\right]\right)}{\sum_{k\in\mathrm{Neighbor}(i)}\exp(\mathrm{LeakyReLU}\left(\mathbf{a}_q^T\left[\mathbf{z}_i||\mathbf{z}_k||\mathbf{z}_{e_{\tau(<i,k>)}}\right]\right)}. \tag{9}$$

Here $\mathbf{z}_i = \mathbf{W}_q\mathbf{e}_i$ and $\mathbf{z}_{\mathbf{e}_{\tau(<i,j>)}} = \mathbf{W}_q\mathbf{e}_{\tau(<i,k>)}$, where $\mathbf{e}_{\tau(<i,k>)}$ is the edge type embedding of the edge type between node i and node j. $\mathbf{e}_i$ and $\mathbf{e}_j$ are the features of node i and node j. The superscript of layer l is omitted for simplicity. The node embedding of the next layer $l+1$ is computed by the aggregation of neighbors based on the attention weight computed above:

$$\mathbf{e}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathrm{Neighbor(i)}}\alpha_{ij}\mathbf{W}_q^l\mathbf{e}_j^l\right), \tag{10}$$

where $\mathbf{W}_q^l$ is the linear transformation of the feature and σ is the sigmoid activation function.

By the heterogeneous graph neural network, we can obtain the node embeddings of the heterogeneous search graph. We take the node embedding of a target user u, the node embedding set of their queries and their clicked documents, denoted as $\mathbf{e}_u$, E_{q_u} and E_{d_u}, respectively. We regard each historical query and click of u as a distinct behavior, whose representation is the element of E_{q_u} and E_{d_u}. Combining E_{q_u} and E_{d_u} forms a behavior set $\mathrm{S}_u^t = E_{q_u} \cup E_{d_u}$. We arrange these behaviors in chronological order, and append the timestamp w.r.t. each behavior to form a behavior sequence $\mathrm{H}_u^t = \{(\mathbf{e}_{h_i}, t_i)\}_{i=1}^m$, where $\mathbf{e}_{h_i} \in S_u^t$ and $m = |\mathrm{S}_{q_u}|$.

Based on the behavior sequence, we can model the influence of long-term and recent behaviors using the multivariate Hawkes process to predict the current search intent. Below, we detail our multivariate Hawkes process modeling method.

C. Modeling Search Behavior With Multivariate Hawkes Process

Based on the historical search behaviors, we propose to predict the current search intent with multivariate Hawkes process. The key point of our approach is to predict the preference of u to each candidate document d with the conditional intensity function.

Specifically, given a user u, $u's$ historical behavior sequence $\mathrm{H}_\mathrm{u}^\mathrm{t}$, and the current query q at timestamp t, the conditional intensity function for each candidate document d is defined as:

$$\lambda_{d|u} = \mu_{u,d} + \sum_{(h_i,t_i)\in H_u^t}\alpha_{d,h_i}\kappa(t-t_i). \tag{11}$$

The conditional intensity function consists of two terms, the first term $\mu_{\mathrm{u,d}}$ is the base intensity, which denotes the influence of long-term behavior. We define it as the cosine similarity between candidate document d and u's long-term search behavior $\mathbf{e}_u$:

$$\mu_{u,d} = \mathrm{cosine}(\mathbf{e}_u, \mathbf{e}_d). \tag{12}$$

$$\mathbf{W}_q = \beta_q^W \odot \mathbf{W} + \gamma_q^W, \tag{13}$$

$$\mathbf{a}_q = \beta_q^a \odot \mathbf{a} + \gamma_q^a, \tag{14}$$

where $\odot$ denotes the Hadamard product.

The second term is the summation of the effect of historical behaviors. α_{d,h_i} is the excite rate deciding to what extent historical behavior h_i excites the behavior of clicking candidate document d. We argue that α_{d,h_i} should not only be determined by the relevance between h_i and d, but also the relevance between h_i and current query q. For example, if a user issues a query "sweet apple", the excitation effect of a document relevant to fruit such as "eatapples.com" should be much stronger than that of "Apple.com". Therefore, we calculate α_{d,h_i} as follows:

$$\alpha_{d,h_i} = w_{q,h_i}\mathrm{cosine}(\mathbf{e}_d, \mathbf{e}_{h_i}), \tag{15}$$

where

$$w_{q,h_i} = \frac{\exp(\mathrm{cosine}(\mathbf{e}_q, \mathbf{e}_{h_i}))}{\sum_{(\mathbf{e}_{h_j},t_j)\in H_u^t}\exp(\mathrm{cosine}(\mathbf{e}_q, \mathbf{e}_{h_j}))}. \tag{16}$$

Since $\mathrm{cosine}(\mathbf{e}_d, \mathbf{e}_{h_i})$ can be positive or negative, historical behaviors may have excitation effects or inhibition effects on future behaviors.

The kernel function is an exponential function modeling the time decay effect of excitation:

$$\kappa(t) = \exp(-\theta t), \tag{17}$$

where θ can either be a predefined hyperparameter or a learnable parameter.

An intuitive illustration of how the influence of long-term and recent behaviors are modulated is shown in Fig. 2. The influence of long-term behavior represents the user's stable

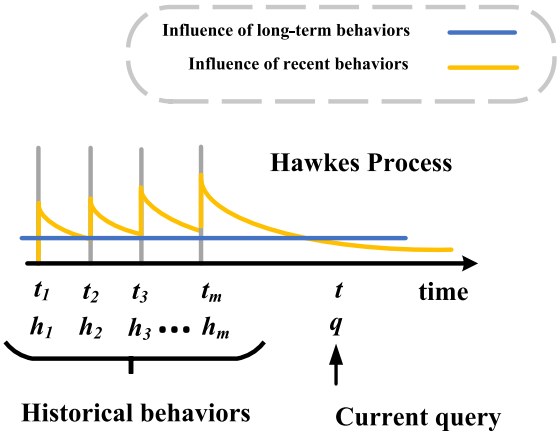


Fig. 2. An intuitive illustration of how dynamic influence of long-term and short-term behaviors are modulated.

interest, such as his profession, while the influence short-term behavior represents recent search interests. The preference of a user to a document is the aggregation of these two kinds of influence.

D. Search Result Re-Ranking

In the above section we modeled the dynamic search intent of user based on Hawkes process, here we introduce how to obtain the final re-ranked search results.

First, given the initial representation of the current query q_u, the condition intensity $\lambda_{d|u}$ is directly adopted as the preference of u to d based on the current search intent. Second, similar to existing works [4], we also collect ranking features according to [11], which are aggregated by an MLP to a similarity score p_r. The final similarity score is the aggregation of these three kinds of similarity scores. We concatenate these scores and adopt an MLP to compute the final score for each candidate document d:

$$p = \tanh\left(\mathrm{MLP}\left(p_q, \lambda_{d|u}, p_r\right)\right), \tag{18}$$

where $\tanh$ denotes the $tanh$-activation function.

We make use of a loss function as used in [28] to train our model. The training data is composed of pairs of documents, including a positive document (clicked) and a negative document (non-clicked). Δ is the value change of MAP (Mean Average Precision) when exchanging the positions of two documents. The loss function is defined as:

$$L = |\Delta|\left(-\hat{p}_{ij}\log(p_{ij}) - \hat{p}_{ji}\log(p_{ji})\right), \tag{19}$$

where $\hat{p}_{ij}$ is the true label that document i is more relevant than document j, and p_{ij} is the predicted probability that document i is more relevant than document j. Finally, we use the Adam optimizer to optimize the parameters of the model.

E. Complexity Analysis

Time complexity: The time complexity of heterogeneous graph neural network is $\mathcal{O}(lN^2)$, where l is the maximum number of layers, and N is the number of nodes in the heterogeneous search graph. The time complexity of the Hawkes process is $\mathcal{O}(h)$, where h is the maximum number of background behaviors of each user. Consequently, the overall time complexity of training HGHP is $\mathcal{O}(|\mathcal{E}|PlN^2)$, where $|\mathcal{E}|$ is the number of epochs, and P is the number of users. To circumvent the substantial computational cost that could arise due to a large N, we only select most recent h_s behaviors to construct the heterogeneous search graph. Hence, N can be regarded as a constant, and the time complexity is linear to the number of users.

Space complexity: The space complexity for storing the adjacent matrix of graph is $\mathcal{O}(N^2)$. Additionally, the space complexity for storing node and edge embeddings is $\mathcal{O}(NF_n + EF_e)$, where F_n and F_e are the maximum dimensionality of node embeddings and edge embeddings, respectively, during forward propagation. The space complexity for the learnable parameters in heterogeneous graph neural network is $\mathcal{O}(lF_e^2H)$. Here, H is the number of attention heads. The space complexity for the learnable parameters in affine transformation is $\mathcal{O}(F_e^2)$. Furthermore, the space complexity for the learnable parameters in the Hawkes process is $\mathcal{O}(h_s)$. Consequently, the overall space complexity of HGHP in one forward propagation of $\mathcal{O}(N^2 + NF_n + EF_e + lF_n^2H + F_n^2)$.

TABLE I
STATISTICS OF DATASET

Dataset	AOL	WeChat	Amazon
User count	24,227	101,852	192,403
Day count	91	45	3702
Query count	181,257	12,003,667	134,9851
Session count	90,345	8,162,282	-
Total clicks	188,007	24,479,377	1,689,188

V. EXPERIMENTS

In this section we conduct extensive experiments to evaluate the effectiveness of the proposed method, which is named as HGHP. We use "performance" to denote the ranking quality of search results. Our experiments are designed to answer the following five questions.

Q1: Could HGHP achieve better performance compared to the state-of-the-art (SOTA) personalized search methods?

Q2: What do major components of HGHP contribute to the performance?

Q3: How does the number of similar users on the heterogeneous search graph affect the performance?

Q4: Does heterogeneous graph modeling perform better than homogeneous graph modeling?

Q5: How important is each type of edge in the heterogeneous search graph.

A. Experimental Setup

Datasets: We use three real-world datasets in our experiments: AOL [41], [42], WeChat and Amazon [43]. The statistics of the datasets are shown in Table I. A brief introduction of the datasets is as follows.

The AOL dataset, introduced by [41], [42], is a widely recognized and publicly accessible dataset for personalized search studies. This dataset comprises user click data collected from March 1, 2006, to May 31, 2006. Each data entry in the AOL

dataset includes an anonymized user ID, a timestamp of the search, a query, a clicked URL, and a rank position. The dataset used in our study was constructed by [41] and [42], and its candidate documents are ranked by the BM25 algorithm [44]. The candidate document list ($D_{t'}$) contains 5 items in the training and validation sets and 50 in the testing set. The data from the initial five weeks (35 days) is employed as background behavior to identify similar users, while the remaining data is divided in a 6:1:1 ratio for training, validation, and testing, respectively.

The Wechat dataset is collected from Wechat, the largest social platform in China. This dataset was collected from the historical log of the search engine embedded in Wechat, encompassing the search behavior of 101,852 users from September 4, 2021, to October 19, 2021. The candidate document list encompass 20 items ranked by the search engine. The data from the first 30 days is used as background behavior to identify similar users on a heterogeneous search graph. The remaining 15 days' data is partitioned in a 6:1:1 ratio for training, validation, and testing.

The Amazon dataset [43] is a widely utilized dataset for personalized product search studies, comprising user reviews on purchased items. For our evaluation, we select the *Electronics* subset. In this context, we interpret purchase behavior as click behavior and construct a candidate document with a length of 10 for each query. The ranking of candidate documents follows the same methodology as in the AOL dataset. The data from the initial five years serves as the background behavior, while the remaining data is divided in a 6:1:1 ratio for training, validation, and testing. It worth noting that this dataset does not provide the partition of sessions.

Baselines: We select 7 methods for personalized search as our baselines, which include several SOTA methods. *P-Click.* [1] It utilizes the re-finding behavior of users. The clicks under the same query of the same person are recorded to re-rank the current query.

PSGAN: [2] It employs generative adversarial networks to train search models. A generator is adopted to generate the distribution of relevant documents and a discriminator is used to distinguish the relevant documents.

HRNN: [27] It uses hierarchical recurrent neural networks to model the search sequence. Attention is used to generate the dynamic query-aware representation of user profiles for reranking.

RPMN: [45] It uses the re-finding behaviors in personalized search. Three types of memory networks are devised to identify query, document, and session based re-finding behaviors, respectively.

PEPS: [5] It employs personal word embeddings on personalized reranking, which are trained from user's history, it also takes account of global word embeddings. The embeddings are used for the representation of the documents and queries in the personalized search model.

HTPS: [4] It uses two hierarchical models containing high-level and low-level transformers which are the query disambiguation model and the personalized language model. These two models are combined with a gate join for the representation of queries.

PSSL: [7] It is a self-supervised contrastive learning model. It extract four kinds of contrastive pairs from search histories, which are used to pre-train the encoders. Pre-trained models are then fine-tuned on the personalized reranking stage.

Parameters and Settings: The initial embedding of queries and document titles is generated using a pre-trained Word2vec model [37] for a fair comparison with previous personalized search methods [4], [5], [7]. The number of dimension of initial embedding, determined through experimentation with values in {50,100,200,300}, was set to be 100 to strike a balance between performance and computational trade-offs. Both the hidden and output dimensions of the model were set to 64. γ is a predefined parameter because it performs better than the learnable setting. We configure it to 0.001,0.001 and 0.1 on AOL, Wechat and Amazon respectively. Given the differing time spans across these datasets, normalization of timestamps was applied by setting the first timestamp in the training set to 0 and the last to 1 on three datasets. The learning rate was determined through a search in the set $\{0.0001, 0.001, 0.01\}$ and set to 0.001 as it yielded the optimal performance. The batch size of 64 was selected. To balance performance and computational trade-offs, the number of similar users was set to 10, 15, and 10 for the AOL, Wechat, and Amazon datasets, respectively. For each user, we select 20 most recent historical queries to construct their heterogeneous search graph. The number of heads of attention mechanism is set to 8 for the consistency with previous studies [8], [13], [46]. The model converges in about 5-10 epochs of training. All experiments are performed on a Nvidia Tesla V100 cluster.

B. Evaluation Metrics

For result evaluation, we select three commonly-used metrics MAP (mean average precision), MRR (mean reciprocal rank), and P@1 (precision@1) to measure the ranking quality of baselines and our method. We regard the clicked documents as the relevant and non-clicked documents as the irrelevant. AP (Average Precision) is defined as

$$\mathrm{AP} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Precision@i} \times \mathrm{hit(i)}}{\mathrm{Number\ of\ clicked\ documents}} \tag{20}$$

where N is the length of the returned list, $\mathrm{Precision@i}$ denotes the top-n precision, $\mathrm{hit(i)}$ is whether the document at position i is clicked, and MAP is the mean of AP of all the searches in the test data. RR(reciprocal rank) is computed by

$$\mathrm{RR} = \sum \frac{1}{\mathrm{rank_i}} \tag{21}$$

where $rank_i$ is the the position of the first clicked document in the returned list, and MRR takes the mean of all the searches in the test data. $P@1$ is defined as the proportion of documents ranked at position 1 be clicked.

C. Experimental Results

*Overall Performance (Q1):*The overall experimental results are shown in Table II. From Table II, we can obtain the following observations.

TABLE II
RANKING RESULTS OF HGHP AND BASELINES ON THREE DATASETS

Model	AOL			Wechat			Amazon		
	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
ori	.2502%–66.00%	.2596%–65.07%	.1534%–76.15%	.4045%–7.18%	.4470%–4.69%	.2674%–3.95%	-	-	-
P-Click	.4224%–42.59%	.4298%–42.18%	.3788%–41.27%	.4063%–6.77%	.4476%–4.56%	.2680%–3.73%	.4756%–40.06%	.4793%–39.75%	.3385%–45.67%
HRNN	.5423%–26.30%	.5545%–25.41%	.4854%–24.54%	.4207%–3.46%	.4601%–1.93%	.2732%–1.90%	.6533%–17.66%	.6561%–17.53%	.5075%–18.55%
PSGAN	.5480%–25.52%	.5601%–24.65%	.4892%–23.95%	.4245%–2.59%	.4622%–1.45%	.2746%– 1.36%	.7125%–10.20%	.7148%–10.15%	.5526%–11.31%
RPMN	.5926%–19.46%	.6049%–18.60%	.5322%–17.27%	.4288%–1.60%	.4623%– 1.45%	.2743%–1.47%	.7354%–7.32%	.7376%–7.29%	.5751%–7.70%
HTPS	.7091%–3.63%	.7251%–2.46%	.6268%–2.63%	.4293%–1.49%	.4643%–1.10%	.2749%–1.26%	.7816%–1.49%	.7823%–1.67%	.6155%–1.22%
PEPS	.7127%–3.14%	.7258%–2.37%	.6279%–2.45%	.4301%–1.30%	.4674%–0.31%	.2755%–1.04%	.7814%–1.52%	.7837%–1.49%	.6156%–1.20%
PSSL	.7358%–	.7434%–	.6433%–	.4358%–	.4690%–	.2784%–	.7935%–	.7956%–	.6231%–
HGHP	**.7436***%+1.06%	**.7525***%+1.22%	**.6713***%+4.35%	**.4456***%+2.48%	**.4742***%+1.10%	**.2849***%+2.33%	**.8167***%+2.96%	**.8182***%+2.94%	**.6527***%+4.75%

* indicates that our model significantly outperforms the best baseline method with two-tail t-test at t <0.05 level.

First, HGHP outperforms all other baselines on three datasets. For example, on AOL, HGHP outperforms the SOTA method PSSL by 1.06% on MAP, 1.22% on MRR, and 4.35% on P@1. On WeChat, our method outperforms the original ranking by 10.16% on MAP, and outperforms PSSL by 2.48% on MAP. On Amazon, HGHP outperforms SOTA method PSSL by 2.96% on MAP, 2.94% on MRR and 4.75% on P@1. These results confirm that our heterogeneous graph-based Hawkes process is an effective way to enhance the quality of personalized re-ranking.

Second, the improvement on AOL over the original ranking is more significant than that on WeChat. The reason could be that the original ranking result of AOL is generated by BM25, while the original ranking result of WeChat is generated by a highly-optimized ranking algorithm deployed in WeChat . The experimental results also indicate that on AOL dataset, the improvement on P@1 is more significant compared with that on other metrics. The reason could be that on AOL dataset users are more likely to click the historically-clicked documents, which are more likely to be ranked at position one by our method. On Wechat and Amazon dataset, users are more likely to explore new documents, thus ranking the clicked documents at the first position could not result in similar performance improvement. Besides, there are several other subtle differences between web search and mobile search such as click preference [47], which make the search behavior on WeChat slightly different from that on AOL. Despite these differences, our methods can outperform existing personalized search methods on three datasets, validating the effectiveness of our method.

Third, the improvement over the SOTA methods is more pronounced on Amazon dataset compared to the other two datasets. The reason for this phenomenon may be the absence of sessions in the Amazon dataset. Because the SOTA methods relies on modeling short-term interests through session behaviors [4], the absence of sessions can lead to suboptimal results. Compared to existing methods, HGHP directly models short-term interests with the Hawkes Process, and therefore more robust to the absence of sessions, thus achieving more significant improvement on Amazon dataset.

Ablation Experiments (Q2): To evaluate the contributions of each main component of our model, we conduct ablation experiments on our method, we use "w/o" to denote removing respective components from HGHP.

w/o HSG denotes a variant that removes the heterogeneous search graph and directly uses the original word2vec representation of queries and documents as the search behavior representation. The multivariate Hawkes process component is reserved to model the evolving search intent.

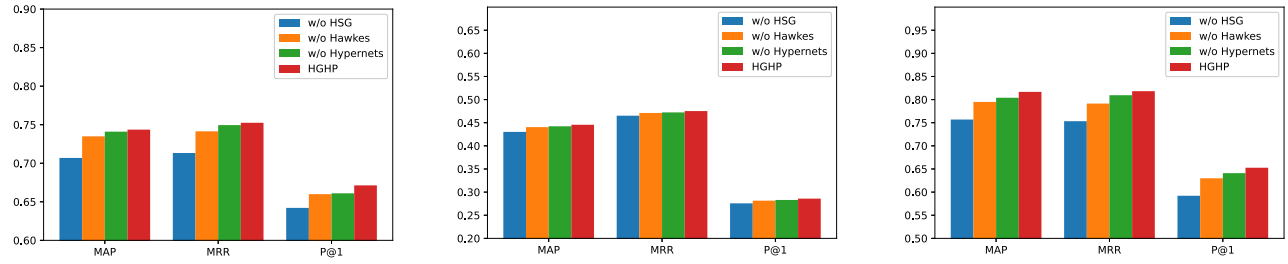
w/o Hawkes is a variant that keeps the heterogeneous search graph, but removes the multivariate Hawkes process. In this case, we only calculate the base rate as the search intent and remove the dynamic features modeled by the multivariate Hawkes process.

w/o Hypernets is a variant that removes query-specific affine transformation on the parameters of the heterogeneous graph neural network. Other components of the model remain unchanged.

The results of ablation studies are shown in Fig. 3. As can be seen, all variants lead to performance decline compared to the original HGHP. For example, on AOL , "w/o HSG" leads to 4.88% drop on MAP, "w/o Hawkes" leads to 1.11% drop on MAP, and "w/o Hypernets" leads to 0.64% drop on MAP. On WeChat , "w/o HSG" leads to 3.41% drop on MAP, "w/o Hawkes" leads to 0.88% drop on MAP, and "w/o Hypernets" leads to 1.01% drop on MAP. On Amazon, "w/o HSG" leads to 3.78% drop on MAP, "w/o Hawkes" leads to a 2.74% drop on MAP, and "w/o Hypernets" leads to 1.79% drop on MAP. The results indicate that each component is beneficial to the performance. The proposed heterogeneous graph based Hawkes process integrating three components achieves the best performance.

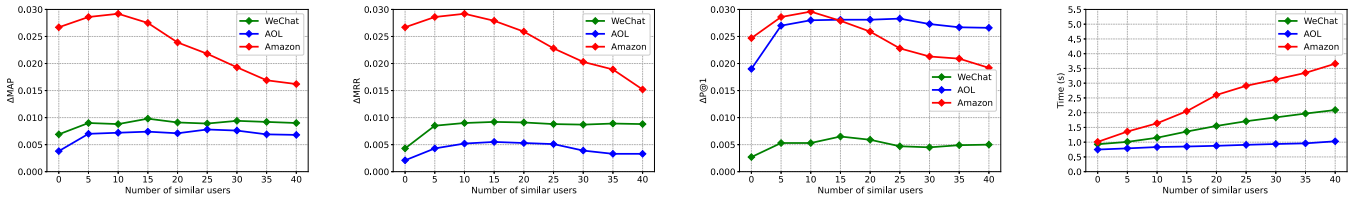
It can also be observed that that the "w/o HSG" variant resulted in a poorer performance on the AOL and Amazon datasets compared to Wechat. This result suggests that the contribution of the heterogeneous search graph is more significant on these two datasets. The potential reason for this phenomenon could be that the text of queries and documents in AOL and Amazon are manually established [41], [42], [43], which may introduce some noise and bias. The heterogeneous search graph leverages collaborative signals in search behaviors, thereby mitigating the impact caused by this noise and bias. On the other hand, the text of queries and documents on WeChat is directly crawled from the search engine, making it more accurate. These results confirm the advantage of the heterogeneous search graph in learning more accurate representations.

Impact of the Number of Similar Users (Q3): One of the merits of the heterogeneous search graph is that it can easily



(a) AOL (b) Wechat (c) Amazon

Fig. 3. Results of ablation experiments.



(a) MAP (b) MRR (c) P@1 (d) time cost

Fig. 4. The experimental result of different number of similar users three datasets. ΔMAP, ΔMRR, and ΔP@1 represents the increase on three metrics compared with the SOTA method PSSL, respectively. (d) shows the the time cost of processing a batch during the training phase.

utilize similar users. Here we study the impact of the number of similar friends on the performance. To this end, we construct heterogeneous search graphs with different numbers of similar users. The number of similar users increases from 0 to 40 with a step of 5. The results on two datasets are shown in Fig. 4. From Fig. 4, we have the following observations.

First, on all datasets, the performance increases when the number of friends increases from 0 to 5, indicating that adding similar users to the heterogeneous search graph can improve the performance. The improvement on AOL is more significant than on Wechat . The reason could be that the historical search behavior on AOL is more limited than that on Wechat. Adding similar users on AOL could better alleviate the data sparsity problem.

Second, the optimal number of similar users is 25,15 and 10 on AOL , Wechat and Amazon, respectively. The performance does not consistently improve as the number of users increases. The reasons could be two-folds. First, the order of adding similar friends is based on the similarity to the target user. When the number of similar users increases, the newly-added users are the least similar to the target user, which may contribute less compared to the users added earlier. This diminishing marginal effect could limit the effect of adding more similar users. Second, too many similar users may incur noises, which is harmful to the performance.

Third, the performance of HGHP on Amazon consistently declines when the number of similar users exceeds 15. This may be attributed to the fact that the products in the Amazon dataset are predominantly related to electronics, leading to a more concentrated topic compared two other two datasets. A high number of similar users could potentially weaken the discriminative ability of user interests, resulting in a more pronounced impact on performance compared to the other two datasets.

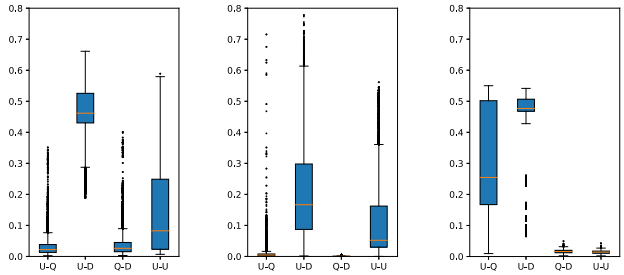
Discussions: Although incorporating similar users can improve the performance, it also incurs computation overhead. The time cost of processing a batch with respect to the number of similar users are shown in Fig. 4(d). It can be observed the time cost increases linearly with the number of similar users. It can also be observed from Fig. 4 that the improvement is marginal when the number of similar users exceeds 5. Therefore, considering the trade-offs between performance and computational overheads, selecting 5 similar users on AOL and Wechat, and 5-10 similar users on Amazon is sufficient to achieve comparable results to those obtained with optimal numbers.

Comparison to Homogeneous Graph Modeling (Q4): To demonstrate the advantage of the heterogeneous graph, we convert the heterogeneous search graphs to homogeneous graphs, which assumes the edges on the graph are of the same type. Then, we utilize two homogeneous graph neural network, GCN [48] and GAT [46] to learn the node embeddings of the graph. We denote such a method using GCN as w/ GCN, and using GAT as w/ GAT. Other components of the model remain unchanged. We compare the performance of two homogeneous graph based variants with our original HGHP. The results are reported in Table III.

As shown in Table III, two homogeneous graph based methods achieve inferior results compared to HGHP on all datasets. Specifically, GCN yields the worst performance, indicating neglecting the weights of nodes in the heterogeneous search graph leads to a significant loss in performance. GAT, by calculating node weights to aggregate messages from neighbors, achieved better performance than GCN. HGHP explicitly considers the

TABLE III
COMPARISON TO HOMOGENEOUS GRAPH MODELING

Model	AOL			Wechat			Amazon		
	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
w/ GCN	.7152% –3.82%	.7213%–4.14%	.6490%–3.32%	.4285%–3.84%	.4609% –2.80%	.2740%–3.83%	.7951% –2.64%	.7969%–2.60%	.6323%–3.13%
w/ GAT	.7345% –1.22%	.7409%–1.54%	.6597%–1.73%	.4407%–1.09%	.4696% –0.97%	.2814%–1.22%	.8058% –1.33%	.8087%–1.16%	.6435%–1.41%
HGHP	**.7436***	**.7525***	**.6713***	**.4456***	**.4742***	**.2849***	**.8167***	**.8182***	**.6527***



(a) AOL (b) Wechat (c) Amazon

Fig. 5. The distribution of attention weights on AOL dataset (left), WeChat (middle) and Amazon dataset (right).

types of edges and has achieved better performance than GAT, indicating that modeling the heterogeneity in search behaviors is important to enhancing the effectiveness of personalized search.

Importance of each relation (Q5): From Q4 we find that considering the heterogeneous relations is of benefit. In this subsection we further investigate how important is each type of relation. As we mentioned above, there are four types of relations on the graph, including user-query (U-Q), user-document (U-D), query-document (Q-D) and user-user (U-U).[1] Specifically, we visualize the attention weights of these four types of relations, and display their distribution in a box plot.

Fig. 5 shows the distribution of attention weights on three datasets. It can be observed that on all datasets, the edges of relation "user-document" has the highest attention weights among all relations, suggesting that clicking behavior plays the most important role. The attention weights of "user-document" is higher on AOL dataset than that on WeChat dataset, the reason could be re-finding behaviors is more frequent on AOL dataset and users are more likely to click previously clicked documents, which makes the model pay more attention to the click behavior. The attention weights of user-user relation is the second highest on AOL and Wechat datasets, suggesting that similar users are useful information for modeling search behaviors. The attention weights of the U-Q relation is relatively low on AOL and Wechat but higher on Amazon, the reason could be queries on AOL and Wechat are often ambiguous than that on Amazon. Hence, the contribution of U-Q relation is more pronounced on Amazon than on the other two datasets.

[1]There are actually eight types of relation on the graph since we regard reverse edges as different relations, however, we found that the weights of reverse edges are similar to the original edges, so we omit them on the figure for brevity.

VI. CONCLUSION

In this work, we propose a new personalized search method which uses a novel heterogeneous graph based Hawkes process to model user's search behaviors. Specifically, we propose a heterogeneous search graph to represent search behaviors. A query-specific heterogeneous graph neural network model is then developed to learn the embeddings of users' search behaviors. With those embeddings, the dynamic influence of users' long-term and recent historical behaviors is modeled by the multivariate Hawkes process. The final search result personalization is achieved by re-ranking the documents based on the dynamic search intent. Extensive experimental results on two real-world datasets demonstrate the effectiveness and superiority of the proposed method.

The potential limitations of this paper are too-folds. First, the heterogeneous graph neural network model may be computationally expensive when dealing with an excessively large heterogeneous search graph. Therefore, a promising avenue for future research involves developing sampling strategies to select informative historical behaviors, thereby mitigating computational costs. Second, there may be instances where the title of a document does not contain relevant keywords, potentially leading to inaccurate initial features. A possible solution for this issue could involve the use of collaborative filtering to identify similar documents, and generating additional features for the document based on the features of these similar documents.

VII. ETHICAL IMPLICATIONS

To develop personalize search methods, we have collected user's search log from search engine. To ensure user privacy and data security, all personally identifiable information and user identities in the data we used have been replaced with pseudonyms with data anonymization techniques.

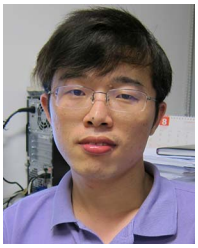
REFERENCES

[1] Z. Dou, R. Song, and J. Wen, "A large-scale evaluation and analysis of personalized search strategies," in *Proc. Int. Conf. World Wide Web*, 2007, pp. 581–590.
[2] S. Lu, Z. Dou, X. Jun, J.-Y. Nie, and J.-R. Wen, "PSGAN: A minimax game for personalized search with limited and noisy click data," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 555–564.
[3] S. Ge, Z. Dou, Z. Jiang, J.-Y. Nie, and J.-R. Wen, "Personalizing search results using hierarchical RNN with query-aware attention," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 347–356.
[4] Y. Zhou, Z. Dou, and J. Wen, "Encoding history with context-aware representation learning for personalized search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1111–1120.
[5] J. Yao, Z. Dou, and J.-R. Wen, "Employing personal word embeddings for personalized search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1359–1368.

[6] S. Lu, Z. Dou, C. Xiong, X. Wang, and J.-R. Wen, "Knowledge enhanced personalized search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 709–718.
[7] Y. Zhou, Z. Dou, Y. Zhu, and J.-R. Wen, "PSSL: Self-supervised learning for personalized search with contrastive sampling," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2749–2758.
[8] Y. Zhou, Z. Dou, B. Wei, R. Xie, and J.-R. Wen, "Group based personalized search by integrating search behaviour and friend network," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 92–101.
[9] J. Yao, Z. Dou, R. Xie, Y. Lu, Z. Wang, and J.-R. Wen, "User: A unified information search and recommendation model based on integrated behavior sequence," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2373–2382.
[10] Z. Ma, Z. Dou, G. Bian, and J.-R. Wen, "PSTIE: Time information enhanced personalized search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1075–1084.
[11] P. N. Bennett et al., "Modeling the impact of short-and long-term behavior on search personalization," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 185–194.
[12] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.
[13] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[14] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, "Transformer Hawkes process," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11692–11702.
[15] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6754–6764.
[16] Q. Zhang, A. Lipani, O. Kirnap, and E. Yilmaz, "Self-attentive hawkes process," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11183–11193.
[17] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, "Embedding temporal network via neighborhood formation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2857–2866.
[18] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro-and macro-dynamics," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 469–478.
[19] Y. Ji, T. Jia, Y. Fang, and C. Shi, "Dynamic heterogeneous graph embedding via heterogeneous hawkes process," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2021, pp. 388–403.
[20] J. Teevan, D. J. Liebling, and G. R. Geetha, "Understanding and predicting personal navigation," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2011, pp. 85–94.
[21] P. N. Bennett, K. Svore, and S. T. Dumais, "Classification-enhanced ranking," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 111–120.
[22] A. Sieg, B. Mobasher, and R. Burke, "Web search personalization with ontological user profiles," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2007, pp. 525–534.
[23] M. J. Carman, F. Crestani, M. Harvey, and M. Baillie, "Towards query log based personalization using topic models," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1849–1852.
[24] T. Vu, A. Willis, S. N. Tran, and D. Song, "Temporal latent topic user profiles for search personalisation," in *Proc. Eur. Conf. Inf. Retrieval*, 2015, pp. 605–616.
[25] M. Harvey, F. Crestani, and M. J. Carman, "Building user profiles from topic models for personalised search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2309–2314.
[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[27] Y. Song, H. Wang, and X. He, "Adapting deep RankNet for personalized search," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2014, pp. 83–92.
[28] C. Burges et al., "Learning to rank using gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 89–96.
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] Y. Zhou, Z. Dou, and J.-R. Wen, "Enhancing potential re-finding in personalized search with hierarchical memory networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3846–3857, Apr. 2023.
[31] J. Yao, Z. Dou, J.-Y. Nie, and J.-R. Wen, "Looking back on the past: Active learning with historical evaluation results," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4921–4932, Oct. 2022.
[32] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[33] Z. Wen and Y. Fang, "Trend: Temporal event and node dynamics for graph representation learning," in *Proc. Int. Conf. World Wide Web*, 2022, pp. 1159–1169.
[34] B. Vassøy, M. Ruocco, E. de Souza da Silva, and E. Aune, "Time is of the essence: A joint hierarchical rnn and point process model for time and item predictions," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2019, pp. 591–599.
[35] T. Bai et al., "CTRec: A long-short demands evolution model for continuous-time recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 675–684.
[36] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.
[37] T. Mikolov, I. Sutskever, K.G. S. ChenCorrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, C.L. BurgesM. BottouZ. Welling-Ghahramani, and K.Weinberger, Eds., Curran Associates, Inc., 2013, pp. 3111–3119.
[38] D. Ha, A. Dai, and Q. Le, "Hypernetworks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–11.
[39] Q. Lv et al., "Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1150–1160.
[40] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3942–3951.
[41] W. U. Ahmad, K.-W. Chang, and H. Wang, "Multi-task learning for document ranking and query suggestion," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, pp. 1–12.
[42] W. U. Ahmad, K.-W. Chang, and H. Wang, "Context attentive document ranking and query suggestion," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 385–394.
[43] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft, "Learning a hierarchical embedding model for personalized product search," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 645–654.
[44] S. Robertson and H. Zaragoza, *The Probabilistic Relevance Framework: BM25 and Beyond*. Boston, MA, USA: Now Publishers Inc, 2009.
[45] Y. Zhou, Z. Dou, and J.-R. Wen, "Enhancing re-finding behavior with external memories for personalized search," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2020, pp. 789–797.
[46] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.
[47] J. Mao, C. Luo, M. Zhang, and S. Ma, "Constructing click models for mobile search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 775–784.
[48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–10.

Xiang Wu received the BS degree in computer science and technology from the Beijing Institite of Technology, in 2021. He is currently working toward the MS degree with the Department of Computer Science, Beijing Institute of Technology. His research interests include graph neural networks, graph representation learning, and data mining.

Hongchao Qin received the BS degree in mathematics, the ME and PhD degrees in computer science from Northeastern University, China, in 2013, 2015, and 2020, respectively. He is currently a postdoc with the Beijing Institute of Technology, China. His current research interests include social network analysis and data-driven graph mining.



Rong-Hua Li received the PhD degree from the Chinese University of Hong Kong, in 2013. He is currently a professor with the Beijing Institute of Technology, Beijing, China. His research interests include graph data management and mining, social network analysis, graph computation systems, and graph-based machine learning.

Yuchen Meng received the BE degree in computer science and technology from the Beijing Institute of Technology, in 2022. He is currently working toward the ME degree in computer science and technology from the Beijing Institute of Technology. His research interests include Big Data management, graph generation, and topological data analysis.

Huanzhong Duan is currently an expert researcher with Wechat search application department, Tencent. His research interests include mechine learning and data mining.

Yanxiong Lu is currently an expert researcher with Wechat Search Application Department, Tencent. His research interests include information retrieval, natural language processing and mechine learning. He has published several papers on some conferences such as ACL and AAAI.

Yujing Gao received the PhD degree from the Beijing Institute of Technology, in 2014. He is currently a lecturer with the School of Computer Science, Beijing Institute of technology. His research interests include data mining, meching learning and software engineering.

Fusheng Jin is currently an associate professor with the school of computer science, Beijing Institute of technology. His research interests included graph-based machine learning block chain Big-Data analysis and software architecture.

Guoren Wang received the BSc, MSc, and PhD degrees from the Department of Computer Science, Northeastern University, China, in 1988, 1991, and 1996, respectively. Currently, he is a professor with the Department of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include XML data management, query processing and optimization, bioinformatics, high dimensional indexing, parallel database systems, and cloud data management. He has published more than 100 search papers.