Group based Personalized Search by Integrating Search Behaviour and Friend Network

Yujia Zhou[2,3], Zhicheng Dou[1*], Bingzheng Wei[3], Ruobing Xie[3], and Ji-Rong Wen[4,5]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]WeChat Search Application Department, Tencent, China
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods
[5]Key Laboratory of Data Engineering and Knowledge Engineering, MOE
{zhouyujia,*dou}@ruc.edu.cn

ABSTRACT

The key to personalized search is to build the user profile based on historical behaviour. To deal with the users who lack historical data, group based personalized models were proposed to incorporate the profiles of similar users when re-ranking the results. However, similar users are mostly found based on simple lexical or topical similarity in search behaviours. In this paper, we propose a neural network enhanced method to highlight similar users in semantic space. Furthermore, we argue that the behaviour-based similar users are still insufficient to understand a new query when user’s historical activities are limited. To tackle this issue, we introduce the friend network into personalized search to determine the closeness between users in another way. Since the friendship is often formed based on similar background or interest, there are plenty of personalized signals hidden in the friend network naturally. Specifically, we propose a friend network enhanced personalized search model, which groups the user into multiple friend circles based on search behaviours and friend relations respectively. These two types of friend circles are complementary to construct a more comprehensive group profile for refining the personalization. Experimental results show the significant improvement of our model over existing personalized search models.

CCS CONCEPTS

• **Information systems → Personalization**.

KEYWORDS

Personalized search; Friend network; Group formation

ACM Reference Format:
Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462918

SIGIR ’21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462918

1 INTRODUCTION

Search engine is a common tool for obtaining information, but returning the same results to different users is often not optimal. Personalized search aims to customize ranking lists to meet individual needs for each user. It has been proven to be effective in improving user search experience [5]. Previous studies [3, 5, 11, 21, 23, 30, 31] usually build user profiles based on historical behaviours, and consider the matching between the user profiles and documents when ranking the results. Recently, the emergence of deep learning allows models to capture user interests in semantic space [10, 14, 15, 40] and brings significant improvement in search quality.

Although the existing strategies of personalization are diversified, most of them point out that personalization is extremely dependent on user’s historical behaviour [8, 23]. When the user has only limited activities in history, the effect of personalized strategies is also constrained. To tackle this problem, some group based methods [22, 24, 30] were proposed to integrate query logs of similar users to expand user profiles. However, these studies mainly identify similar users based on lexical or topical similarity of queries and documents. This approach is too simple to inevitably introduce a lot of noisy users. In fact, even if users issue same queries, they may be driven by different intents. Besides, hot events will also encourage two users with different profiles to have similar search behaviours. In this paper, **we propose a better personalized method in highlighting similar users with the help of neural networks**. This enables us to calculate the similarity between users in semantic space and in a more reliable manner.

Previous approaches for finding similar users mainly refer to the user’s historical search or click behaviours, and have demonstrated some effectiveness in group based personalized search. However, due to the reliance on the historical behaviour, these methods are still insufficient when the users only have limited historical data. It is unreliable to understand the intent of a new query based on similar users found only with scant query log. Recently, the integration of social platforms and search engines has gradually become a hot field. Many APPs, such as WeChat and Toutiao, have both social attributes and information acquisition functions. These APPs generate two types of connected data: search activities of individual users, and the friend relationship among users. This enables the possibility of leveraging friend network to improve personalized web search, and in this paper, **we carry out a preliminary study on introducing the user’s friend network into personalized search** to solve the above-mentioned problem. Friend network has been applied on many user modeling tasks, especially on social recommendation

[9, 19, 32], and shows the effectiveness in personalization. In real life, the establishment of friendship is often based on the same background or interest. This will help capture the user's stable profile based on his friends even if his query log is empty. Endowed with the benefit of friend network, the reliability of group formation is enhanced to address the problem of data sparsity.

Intuitively, not all friends contribute equally to building the user profile. Both the search behaviour similarity and friend relation closeness can determine their contribution to the current user in personalization. The former reflects the similarities between users' information needs, while the latter reveals the strength of relationship between users in the real world. Their complementary advantages can help build a more comprehensive user profile. In other words, if a close friend also has similar search behaviours as the current user, he should be given more attention in personalization. Based on this consideration, **we propose integrating both factors, behaviour similarity and friend relationship, in measuring the similarity between users**.

Furthermore, in real life, a user usually belongs to multiple friend circles due to his diverse backgrounds or interests. Users in the same circle generally share similarities in one aspect, such as belonging to the same university or loving to play football. This observation inspires us to group the user into multiple friend circles to build the group profile in a fine-grained way. The user's historical search behaviour and the friend network provide us with two grouping angles: one is based on similar information needs, and the other is based on the relationship in real life. To ensure the diversity of friend circles, we design an algorithm to maximize the difference of the users included in different circles. Thus, each friend circle reflects one aspect of the user profile, and they together form a complete one for better personalization.

Inspired by the above observation, we propose a group based personalized search model FNPS, which integrates the search behaviour and friend network to model the group profile with neural networks. The construction of group profile can be summarized as the following four steps. **Firstly**, we leverage the friend network and the user's historical search behaviours to form friend circles from two angles. **Secondly**, the graph attention network is applied to aggregate the friends' profiles with different weights. **Thirdly**, the representations of two types of friend circles are fed into a cross attention layer to enhance each other. **Finally**, in response to the current query intent, we use the query-aware attention to highlight relevant friend circles and construct the group profile dynamically. The final personalized results are generated under the combined effect of the user's individual profile and the group profile.

Our main contributions can be summarized as follows. (1) To the best of our knowledge, this is the first time that similar users are used to build profiles for neural personalized web search; (2) In addition to the user similarity calculated based on users' search and click behaviour, we introduce the friend network into personalized search with neural networks to further tackle the problem of historical data sparsity. (3) Empowered by the combination of search behaviour similarity and friend relation closeness, the measurement of similarity between users is more reliable. (4) To build the group profile in a fine-grained way, we group the user into different friend circles from two angles. They will be considered with dynamic weights based on the current query in personalization.

2 RELATED WORK

2.1 Personalized Web Search

The goal of personalized search is to re-rank the results to match the individual needs for different users. The key to personalized search is how to build user profiles based on their search logs. Some early studies mainly construct user profiles by extracting features from query logs, such as click features and topic features. Dou et al. [8] and Teevan et al. [23] predicted the click probability of a document by counting the number of historical clicks. They pointed out that it was reliable to personalize search results with respect to user's re-finding behaviours. The extraction of topic features has undergone a transition from manual design to automatic learning [2, 20, 33]. Due to the huge human cost of the former, some studies [6, 11, 29, 31] focused on how to automatically learn the topic representation of the document with Latent Dirichlet Allocation (LDA). With the emergence of the learning to rank method, many studies [3, 27, 33] combined multiple personalized features for ranking with LambdaMART [35] algorithm.

Recently, deep learning has been applied in various research fields and has achieved success. Due to its powerful representation learning capabilities, it is good at extracting user potential interests in personalized search. Song et al. [21] designed a deep neural network to adapt a generic RankNet for personalized search. Ge et al. [10] focused on sequential information hidden in query log and learned the user interests with recurrent neural network. Zhou et al. [40] built memory networks to generalize the re-finding behaviour to semantic space. Lu et al. [16] introduced knowledge base to refine user profiles. Moreover, methods based on adversarial neural networks [15] and reinforcement learning [37] have also been proposed for enhancing data quality. In addition to these user profile-based methods, Zhou et al. [39] and Yao et al. [36] argued that the query representation is dynamically changing in different historical contexts. They used the history to learn the embedding of the current query. This paper attempts to introduce the friend network into personalized search with deep learning to enhance user profiles, especially for the users who lack historical data.

2.2 Group based Web Search

Group based search aims to improve search results with the help of similar users' query logs. The existing methods for extracting similar users can be divided into two categories, based on search behaviour or social relations. For the first method, Dou et al. [8] proposed the G-Click model to find top K users with similar search behaviour, and then ranked the results based on these users. Morris et al. [18] focused on enhancing the quality of traditional web search with collaborative behaviours. Teevan et al. [22, 24] tried different grouping ways and showed that grouping could identify what users would consider relevant to the query. However, these static grouping methods ignore the users' diverse interests with respect to different topics. Vu et al. [30] proposed to construct the group dynamically in response to the input query.

The social network based methods were proposed to model user preferences based on social relations. Bender et al. [1] designed a new approach to exploit social relations by combining semantic and social signals during the ranking. They put the users, tags, and documents into a friendship graph and applied PageRank computation on it. Similarly, Kashyap et al. [12] designed six social groups and formed a social aware search graph for ranking. Bjorklund et al. [4] and Carmel et al. [7] constructed the social network from some social applications, and took the social relations into account while ranking. Recently, some studies concentrated on the personalized search in microblog, which has an explicit Follow-Follower social network. Vosecky et al. [28] and Zhao et al. [38] leveraged social relations in Twitter to construct a better user preference for personalized re-ranking of tweets. Different from these social search methods in a specific system, we focus on web search which retrieves the documents from the whole Internet.

3 METHODS

Referring to the profiles of similar users to personalize the results can improve the ranking quality when the user lacks historical activities. As we stated in Section 1, finding similar users based on simple lexical and topical similarity of query logs is weak in measuring the closeness between users in semantic space. Moreover, when the user has limited interactions with search engine, similar users found based on it are still insufficient to ensure reliability of personalization for new queries. To handle these problems, we propose the model FNPS to highlight similar users in semantic space with neural networks. It integrates the friend network to further address the problem of historical data sparsity. Specifically, to construct the group profile in a fine-grained manner, the user is grouped into multiple friend circles with respect to his search behaviours and friend relations respectively. They are called behaviour-based friend circles and relation-based friend circles in the following. Under the complementary effect of them, similar users are highlighted and a group profile is constructed based on them.

To begin with, suppose that for the user u, his historical query log H_u can be divided into the long-term history H_u^l and the short-term history H_u^s. The former includes a series of queries and satisfied documents in previous sessions, $H_u^l = \{\{q_1, D_1\}, ..., \{q_n, D_n\}\}$, where n is the number of queries issued in previous sessions. The latter contains user's recent interactions in the current session, $H_u^s = \{\{q_{n+1}, D_{n+1}\}, ..., \{q_{t-1}, D_{t-1}\}\}$, where t is the current timestamp. The friend network of the user can be denoted as a graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes containing the current user and his friends, and $\mathcal{E}$ represents the friend relations between users. Given a new query q and candidate documents $D = \{d_1, d_2, ...\}$ returned by the search engine, we need to score each element in D to represent the probability of being clicked. The scoring process should refer to the current query q, the historical data H, and the friend network $\mathcal{G}$. Thus, we denote the score of the document d as $p(d|q, H, \mathcal{G})$, which consists of two parts:

$$p(d|q, H, \mathcal{G}) = \phi(p(d|q), p(d|P_u^I, P_u^G, q)), \tag{1}$$

where $p(d|q)$ is the adhoc relevance between the document and the query, and $p(d|P_u^I, P_u^G, q)$ represents the personalized relevance based on the user profile. Specifically, the P_u^I denotes u's individual profile, which is constructed with respect to the query q and history H. The P_u^G corresponds to u's group profile, which is generated by combining all the information q, H, and $\mathcal{G}$. The function $\phi(\cdot)$ is the multilayer perceptron (MLP) with $tanh(\cdot)$ as the activation function, which is used to combine the scores of these two parts with different weights.

The architecture of our model is shown in Figure 1. The input consists of the user's friend network, historical search behaviours and the current query. At first, in order to construct the individual profile, we use two transformers to model the long-term and short-term history respectively based on the current query. And then, for building the group profile, we combine the friend relations and search behaviours to form multiple friend circles. Under the interaction between two types of friend circles, more central circles are highlighted to personalize the results. We will introduce the details in the remaining parts of the section.

3.1 Modeling individual profile

The key to personalized search is how to model user interests based on user's historical search behaviour. Inspired by previous researches [3, 13, 39], we model the long-term and short-term historical search behaviour of users separately. The former describes more long-standing user characteristics, and the latter usually represents the user's recent interests or temporary information needs. Due to the powerful ability of Transformer [25] on long-term dependency, we attempt to apply it to model the individual profile with the following two steps.

Modeling user current query intent. A common situation in the search engine is that users often put forward a series of queries for a single information need in a session. Therefore, recent interactions in short-term history are helpful for clarifying the intent of current query, especially when the current query is ambiguous. Inspired by the previous work [40], we apply the short-term transformer to model the current query intent.

For each query and document, the vector representation is computed by summing up the words together based on word embedding, which is trained by the word2vec method [17]. Formally, for each interaction in short-term history, we sum up the query vector and the average satisfied document vector to represent the search intent, denoted as $h_i = q_i + \text{average}(D_i)$. We concatenate the representation of short-term history $H_u^s = \{h_{n+1}, ..., h_{t-1}\}$ with the current query q as the input of the short-term transformer:

$$q^s = \text{Transformer}^{last}([H_u^s, q] + \text{PE}([H_u^s, q])), \tag{2}$$

where $\text{Transformer}(\cdot)^{last}$ is implemented by the transformer encoder [25] and the output of the last position is taken. The function $\text{PE}(\cdot)$ is the position embedding function of transformer. The output q^s indicates the user's current query intent, which contributes to the construction of the user profile and matching with candidate documents in the following.

Building individual profile dynamically. Long-term search behaviour often reflects the user's background and stable interest. For example, the user who frequently submits queries related to "pytorch" is more likely to be a programmer. In order to model the user's long-term interests, we devise another transformer to model the long-term dependencies between historical behaviours. Similar to the short-term history, the representations of interactions in long-term history $H_u^l = \{h_1, ..., h_n\}$ constitute the input of the long-term transformer:

$$O_u^l = \text{Transformer}(H_u^l + \text{PE}(H_u^l)),$$

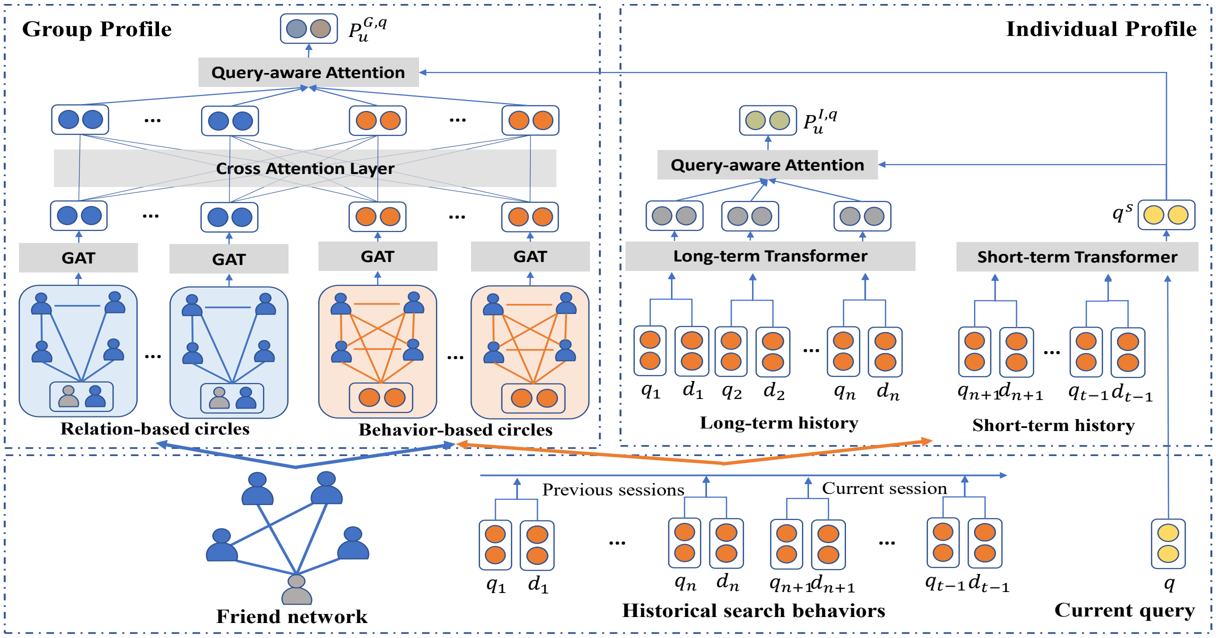

Figure 1: The architecture of FNPS. Given the user's historical search behaviours and current query, the query intent and the individual profile are modeled with transformer layers. With the help of friend network, relation-based and behaviour-based friend circles are formed to build the group profile. These two profiles are integrated to enhance search results personalization.

where $O_u^l = \{o_1, ...o_n\}$ is the set of outputs with respect to u's long-term interactions. Intuitively, not all long-term interactions are valuable for current query intent [10]. Based on this idea, we dynamically adjust the weight of each interaction o_i with attention mechanism based on the user's current query intent q^s. The dynamic individual profile $P_u^{I,q}$ is computed by weighted summation:

$$P_u^{I,q} = \sum_{i=1}^{n} \alpha_i o_i, \tag{3}$$

where α_i is the weight of o_i with respect to the current query intent. It is calculated by feeding o_i and q^s into MLP, and is normalized by the softmax function:

$$\alpha_i = \text{softmax}(\phi(o_i, q^s)), \text{softmax}(e_i) = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}.$$

Finally, we obtain the individual profile $P_u^{I,q}$ and the current intent q^s related to the current query. They will act on matching with candidate documents and compute the personalized relevance. However, when the user's historical data is limited, the individual profile is weak for personalization. To tackle this issue, a group profile is built in the following.

3.2 Friend Circle Formation

In real life, a user can often be grouped into different friend circles, such as colleagues, relatives, classmates, etc. Each friend circle can reflect an aspect of user characteristics. In this section, we attempt to form multiple friend circles of the user to capture the group profile in a fine-grained way. As we discussed in Section 1, finding similar users based only on search behaviour is not enough when the user lacks historical interactions. The friend relation can also provide us with a way to measure the closeness between users without any behaviours. Based on this consideration, we group the user into multiple friend circles based on relation and behaviour respectively. The former inclines to group users with similar backgrounds into a circle, while the latter focuses on similar information needs. The details of group formation from two angles are as follows.

Algorithm 1 Friend Circle Formation

```
Input: friend graph $\mathcal{G}$, candidate nodes $N$, maximum number of friend circle $k$
Output: friend circles $C = \{c_1, ...c_k\}$, core nodes $F = \{f_1, ...f_k\}$
  for $i$ in range($k$) do
    for all $n \in N$ do
      E ← GetNumberofEdges(n) from $\mathcal{G}$
    end for
    $f_i$ ← N[Argmax(E))]
    $c_i$ ← SubGraph($f_i$) from $\mathcal{G}$
    $\mathcal{G}$ ← DeleteEdges($c_i$) from $\mathcal{G}$
  end for
  return $C$, $F$
```

Relation-based friend circle. Since the establishment of friendship in real life is often based on shared experiences, the users in the same relation-based friend circle may reflect the same background. In general, there are some friends who are closer with u in each circle. Forming friend circles based on these close friends can more accurately reflect the user's group information. To find the close friends, we take the number of mutual friends as the indicator to measure the closeness between users.

Formally, we develop an algorithm of friend circle formation as described in Algorithm 1. Given the user's friend network $\mathcal{G}$ =

$\{\mathcal{V}, \mathcal{E}\}$ and the candidate nodes N, our goal is to find out the close friends from N and to form top k_{r} friend circles based on them. The candidate nodes here contain all friends of the current user u, $N = \{v_i | v_i \in \mathcal{V}, v_i \neq u\}$. Specifically, we first select the node with the most edges on $\mathcal{G}$ from candidate nodes N, which represents the friend who has the most mutual friends with user u. The subgraph related to this node on $\mathcal{G}$ is regarded as the first friend circle c_1. The node can be regarded as the core of this friend circle f_1, which has the same friend relations as the user u in c_1. To ensure the distinction between different circles, we remove the edges of this friend circle from G and repeat the above process to form other circles. Finally, we obtain k_{r} diverse relation-based friend circles $C^{\mathrm{r}} = \{c^{\mathrm{r}}_1, ...c^{\mathrm{r}}_{k_{\mathrm{r}}}\}$ and corresponding core friends $F^{\mathrm{r}} = \{f^{\mathrm{r}}_1, ...f^{\mathrm{r}}_{k_{\mathrm{r}}}\}$. They will act on modeling the group profile in the next section.

Behaviour-based friend circle. The formation of friend circle is not only based on the same background, but may also establish upon similar interests, such as a sport, a movie star, etc. Historical search behaviour can reflect user interests to a certain extent. In this part, we attempt to group users based on their historical search behaviours. To implement this idea, the edges of the friend network $\mathcal{G}$ are replaced based on the common search behaviour in this section. The users who issued the same query or were satisfied with the same documents are connected with an edge. Moreover, we put u's historical search behaviours (queries and satisfied documents) into the graph to form a heterogeneous graph. Each query or document is regarded as a node and the other users whose query logs contain it are connected with this node. Formally, this new heterogeneous graph is considered as the behaviour-based friend graph, and the nodes which represent u's search behaviours are set as candidate nodes. We apply the same algorithm as above on them to group the user into different friend circles based on search behaviours. Finally, we generate top k_{b} behaviour-based friend circles $C^{\mathrm{b}} = \{c^{\mathrm{b}}_1, ...c^{\mathrm{b}}_{k_{\mathrm{b}}}\}$ and the corresponding core behaviours. Note that since the core behaviour must appear in the query log of each friend in the circle, each behaviour-based friend circle is a fully connected graph.

These two types of friend circle focus on different angles of grouping. We believe they complement each other and can build a more comprehensive group profile. The following section will introduce how to integrate them to get the group profile.

3.3 Constructing Group Profile

Although we formed multiple friend circles according to the friend relation and search behaviour, how to aggregate these information is still a challenge. In this section, we propose three insights on modeling group profile. Firstly, we hold the idea that the users in the friend circle have different distances to the current user. Secondly, relation-based and behaviour-based friend circles are formed from two angles. They complement each other and the users who are essential in both dimensions should be given more attention. Thirdly, the contribution of each circle is different in facing different information needs. We will elaborate on the role of each opinion and the implementation details in the following.

GAT on friend circle. The friend circle we formed in last section can be regarded as a group of users with similar background or interest around the core node. Based on the first insight, we intend to explore the structure information of the circle to characterize the influence of each friend. Due to the powerful ability of Graph Attention Network (GAT) [26] in modeling the structure of the graph, we apply it on each friend circle to highlight similar users.

Specifically, we model the user profile of each friend based on his historical behaviours referring to Section 3.1. For the group user g in friend circles, we apply the long-term transformer with the same parameters and average the outputs to represent his individual profile P^I_g. Formally, for each friend circle c_i and the representation of core node f_i, the aggregation function of GAT is defined as:

$$c^f_i = \mathrm{LeakyReLU}(W \sum_{g \in c_i} \alpha_{ig} P^I_g), \alpha_{ig} = \mathrm{softmax}(\phi(f_i, P^I_g)).$$

where c^f_i is regarded as the feature of the circle c_i. The W is the model parameter which is learned during the training. The α_{ig} is the aggregation weight of the user g based on the core node and it is parameterized with MLP. The representations of relation-based friend circles $C^{\mathrm{r},f}$ and behaviour-based friend circles $C^{\mathrm{b},f}$ will interact in the next step to complement each other.

Cross attention on relation and behaviour. As we proposed above, the relation-based friend circles tend to dig out the background information, while the behaviour-based friend circles capture the interests of the user. Based on the second insight, we believe the friends who appear in both types of circles have more contribution to the user profile. In other words, if a relationship-based friend circle and a behaviour-based friend circle contain many common users, we should pay more attention to these users through the interaction of these two circles.

To achieve such an interaction between the two types of friend circles, we propose to use a masked transformer which only keeps the connections between different types of circles. We concatenate the representations of two types of circle $C^{\mathrm{r},f}$ and $C^{\mathrm{b},f}$, and feed them into the masked transformer. We have:

$$C^f = \mathrm{Transformer}^{masked}([C^{\mathrm{r},f}, C^{\mathrm{b},f}]),$$

where $\mathrm{Transformer}^{masked}(\cdot)$ means adding the mask matrix M to the weight matrix of the transformer encoder before softmax. The computation of the mask matrix M is based on the relations among different friend circles:

$$M_{ij} = \begin{cases} 0 & \text{if } c^f_i \text{ and } c^f_j \text{ are connected} \\ -inf & \text{if } c^f_i \text{ and } c^f_j \text{ are disconnected} \end{cases}$$

The output of cross-attention layer $C^f = \{c^f_1, ...c^f_{k_{\mathrm{r}}+k_{\mathrm{b}}}\}$ represents the enhanced friend circles combining the information of search behaviour and friend relation. It provides a foundation for building a dynamic group profile in the next step.

Query-aware attention on different circles. Intuitively, not all friend circles are helpful when the user issues a new query. For adjusting the weight of each friend circle, we apply an attention mechanism on different circles with respect to the user's current query intent. Formally, we take the output of the cross-attention layer C^f and the user's query intent q^s to learn the weight of each circle by MLP. Similar to the Eq. (3), we have:

$$P^{G,q}_u = \sum_{i=1}^{k_{\mathrm{r}}+k_{\mathrm{b}}} \alpha_i c^f_i, \alpha_i = \mathrm{softmax}(\phi([c^f_i, q^s])). \quad (4)$$

The final output $P_u^{G,q}$ is the group profile, which is constructed based on friend network, historical search behaviours, and the current query. It will play an essential role in the search results personalization, especially when the individual profile is weak in personalizing the results of current query.

3.4 Search Results Personalization

In this section, we compute the score of each candidate document based on the individual profile and the group profile. The computation of each part in Eq. (1) is introduced as follows.

For adhoc relevance $p(d|q)$, we take the vector similarity between the query and the document to represent their semantic relevance. Moreover, we also extract a series of relevance features $\mathcal{F}_{q,d}$ following [3], mainly including original ranking, click features, and topical features. We use MLP to aggregate these features to compute a relevant score. The adhoc relevance consists of two parts:

$$p(d,q) = \phi\left(\text{sim}(q,d), \phi(\mathcal{F}_{q,d})\right),$$

where the function $\text{sim}(\cdot)$ is computed by cosine similarity.

The personalized relevance $p(d|P_u^I, P_u^G, q)$ is related to the individual profile and the group profile we obtained above. Intuitively, the group profile is more useful when the individual profile cannot provide effective personalized information for the current query. Therefore, we devise a gate unit to combine two parts with regard to the current query q and individual profile $P_u^{I,q}$ from Eq. (3). The gate weight is computed by feeding them into MLP, denoted as $c = \phi([q, P_u^{I,q}])$. Finally, the personalized relevance combines the individual part and group part with this weight:

$$p(d|P_u^I, P_u^G, q) = c * p(d|P_u^I, q) + (1-c) * p(d|P_u^G, q).$$

For the individual part, the current query intent q^s from Eq. (2) and the individual profile $P_u^{I,q}$ are matched to the candidate documents. And the matching between the profile $P_u^{G,q}$ from Eq. (4) and the document d represents the score of group part. We have:

$$p(d|P_u^I, q) = \phi\left(\text{sim}(d, P_u^{I,q}), \text{sim}(d, q^s)\right),$$
$$p(d|P_u^G, q) = \text{sim}(d, P_u^{G,q}).$$

Finally, the results are personalized by re-ranking the document list according to the score of each candidate documents. We adopt the LambdaRank algorithm [35] to train our ranking model, which constructs the document pair with a satisfied document and an irrelevant document. Formally, the distance between the positive sample d_i and the negative sample d_j is computed by $|p(d_i|q,H,\mathcal{G}) - p(d_j|q,H,\mathcal{G})|$ with sigmoid normalization, denoted as p_{ij}. The loss function is defined as the weighted cross entropy between true distance and predicted distance:

$$\mathcal{L} = -|\Delta|\left(\overline{p}_{ij} log(p_{ij}) + \overline{p}_{ji} log(p_{ji})\right),$$

where the weight Δ represents the change of ranking quality after swapping the position of d_i and d_j. By minimizing the loss function with Adam optimizer, the model gradually reaches convergence.

Table 1: Basic statistics of the dataset.

Item	Statistic	Item	Statistic
#days	92	#seed users	43,770
#friends	2,195,625	#friend relations	126,860,976
#queries	14,930,839	#distinct queries	9,334,434
#clicks	15,843,926	#sessions	8,226,296

4 EXPERIMENTAL SETUP

4.1 Dataset and Evaluation Metrics

There is no public dataset that have both search logs and friend networks with shared users. To evaluate the effectiveness of the model, we collect the data from a large social platform, which embeds a search engine for users to acquire information. Since the social network of the platform is too large for experiments, we extract a small friend network from it for experimentation. To avoid too sparse friend network, we randomly sample some seed users and take their friends to form the friend network in experiments. We collect their search behaviours in the last three months of 2020. Each piece of data in the query logs contains *an anonymous user ID, a query string, query issued time, top 20 documents returned by search engine, clicked documents, and click dwell times*. Note that the returned results include not only the contents created in the platform, but also the general web pages of the entire Internet. The basic statistics are shown in Table 1.

Since personalization requires historical behaviours to build basic user profile, we regard the first 8 weeks data as the history. The last 5 weeks data is divided with 4:1:1 ratio for training, validation, and testing. To ensure that there is at least one session for each part, we remove the users whose active time is less than four sessions. The session is demarcated by 30 minutes of user inactivity [34].

To evaluate the ranking quality, we define the satisfied documents as the documents whose click dwell time is more than 30s [3, 31]. Moreover, we also identify a document as relevant if it is satisfied under the next two queries following [3, 10]. We choose mean average precision (MAP), mean reciprocal rank (MRR), precision@1 (P@1), average click position (Avg. Click), and normalized discounted cumulative gain@K (NDCG@K) to measure the results.

4.2 Baselines

The original ranking of the search engine is considered as a basic baseline. Additionally, we compare our model with the group based methods, social network based methods, and deep learning based methods for personalized search.

G-Click [8]. This is a method of grouping based on click behaviours, which scores documents by counting the same behaviours of other users, thereby re-ranking the list.

DGF [30]. It finds similar users based on topic features constructed using the LDA method, and then considers the interests of these similar users when personalizing search results.

PSSN [7]. It achieves personalization based on the user's social relations, which considers the familiarity-based network, similarity-based network, and overall network. We regard the friend network as the familiarity-based network in this baseline.

Table 2: Overall performance of FNPS and other baselines. The percentage reflects improvements over original ranking. "†" indicates the model outperforms all baselines significantly with paired t-test at $p < 0.05$ level. Best results are shown in bold.

Model	MAP	MRR	P@1	Ave. Rank	NDCG@3	NDCG@5	NDCG@10
Ori.	.6614 -	.6859 -	.5766 -	3.898 -	.6330 -	.6720 -	.7148 -
Group based methods							
G-Click	.6625 +0.17%	.6874 +0.22%	.5746 -0.35%	3.849 +1.26%	.6371 +0.65%	.6754 +0.51%	.7166 +2.52%
DGF	.6491 -2.02%	.6769 -1.31%	.5583 -3.17%	3.920 -0.56%	.6253 -1.22%	.6616 -1.55%	.7069 -1.11%
Social network based methods							
PSSN	.6642 +0.42%	.6883 +0.35%	.5732 -0.59%	3.812 +2.21%	.6389 +0.93%	.6778 +0.86%	.7193 +0.63%
SonetRank	.6659 +0.68%	.6898 +0.57%	.5731 -0.61%	3.764 +3.44%	.6409 +1.25%	.6796 +1.13%	.7220 +1.01%
Deep learning based methods							
HRNN	.6707 +1.41%	.6951 +1.34%	.5800 +0.59%	3.727 +4.39%	.6478 +2.34%	.6839 +1.77%	.7258 +1.54%
RPMN	.6724 +1.66%	.6962 +1.50%	.5795 +0.50%	3.642 +6.57%	.6475 +2.29%	.6859 +2.07%	.7296 +2.07%
PEPS	.6727 +1.71%	.6979 +1.75%	.5811 +0.78%	3.626 +6.98%	.6501 +2.70%	.6867 +2.19%	.7303 +2.17%
HTPS	.6749 +2.04%	.6991 +1.92%	.5821 +0.95%	3.615 +7.26%	.6513 +2.89%	.6894 +2.59%	.7326 +2.49%
Our designed model							
FNPS	**.6827**†+3.22%	**.7058**†+2.90%	**.5902**†+2.36%	**3.486**†+10.57%	**.6585**†+4.03%	**.6985**†+3.91%	**.7402**†+3.55%

SonetRank [12]. It builds a Social Aware Search Graph, which consists of groups, users, queries and clicked documents. It aggregates the relevance feedback of the similar users in the group. We treat the user's friends as a group of similar users.

HRNN [10]. This is a deep learning based method which leverages the sequential information hidden in query logs by RNN. Attention mechanism is applied to build dynamic user profiles.

RPMN [40]. It enhances re-finding behaviours in personalized search with memory networks. Query-based re-finding and document-based re-finding are considered for different search intent.

PEPS [36]. It trains the personal embedding matrix for each user based on historical behaviour. The global embedding and personal embedding are taken into account in personalization.

HTPS [39]. It abandons the construction of user profiles, and instead encodes the history as context to disambiguate the query with Transformer encoder, which is also used in our model.

The first two models group users based on their click information, while the next two methods leverage the social network to find similar users. The last four approaches mine user preferences hidden in historical behaviours with deep learning. Our method combines the advantages of these baselines and we named the model as FNPS (**F**riend **N**etwork enhanced **P**ersonalized **S**earch)[1].

The parameters of the model are selected through multiple experiments. We choose the word embedding size in $\{\mathbf{100}, 300\}$, the hidden state size of transformer and GAT in $\{128, 256, \mathbf{512}\}$, the number of MLP hidden units in $\{128, \mathbf{256}, 512\}$, the learning rates in $\{10^{-2}, \mathbf{10^{-3}}, 10^{-4}\}$. For hyperparameters k_{r} and k_{b}, we compute them according to the number of friends and search behaviours, $k_{\mathrm{r}} = \frac{\#\text{friends}}{20}$, $k_{\mathrm{b}} = \frac{\#\text{behaviours}}{5}$. Considering the performance and memory usage, we select the parameters in bold to train the model.

5 RESULTS AND ANALYSIS

5.1 Overall Performance and Analysis

The overall results of models are listed in Table 2. Some findings are summarized as follows:

(1) Our proposed model FNPS outperforms all baselines on all evaluation metrics, which shows the effectiveness of our friend network enhanced personalized search model on constructing a more reliable user profile. Compared with the best personalized baseline model HTPS, our model improves the ranking quality significantly with paired t-test at $p < 0.05$ level. Specifically, FNPS expands the improvement on MAP over original ranking from 2.04% to 3.22%. A more noteworthy evaluation metric is P@1, on which the improvement increases from 0.95% to 2.36%. These results confirm that the behaviour of similar users can provide effective personalized signals when re-ranking the results. And our model is able to accurately strengthen the influence of similar users by integrating search behaviour and friend network.

(2) Comparing different types of personalization methods, we find that group based methods have minimal improvement in results and even have side effects. The DGF model, which uses topical similarity to find similar users, causes a serious decline on the results. A possible reason is that in the case where the quality of the original ranking is high, users with similar topics are not enough to distinguish the candidate documents. Social network based methods achieve better results by introducing social relations to find similar users, which indicates the familiar users tend to have similar interests. Moreover, deep learning based methods show powerful ability to learn user profiles based on historical interactions automatically, and they outperform previous models significantly. Our model combines the advantages of these methods and further introduces the friend network into personalized search with deep learning, which is proven to be effective.

(3) The improvement on P@1 is obviously lower than other metrics, which may be caused by characteristics of mobile search. In this scenario, the user's re-finding behaviour may be weaker than searching on the computer, such as searching for "Gmail" every day to check emails. Users sometimes want to get latest content instead of what they have seen before. Therefore, it may not be the best choice to rank the viewed documents at the first position. The group based methods and social network based methods even get worse results on P@1. A possible reason is that these methods do

[1] The code of the model is available on https://github.com/smallporridge/FNPS

Table 3: Performance of ablation studies of the FNPS model. The percentage is calculated based on the whole model.

Model	MAP	MRR	P@1	Ave. Rank	NDCG@3	NDCG@5	NDCG@10
w/o. RGF	.6766 -0.89%	.7001 -0.81%	.5838 -1.25%	3.611 -1.08%	.6526 -0.90%	.6924 -0.87%	.7349 -0.72%
w/o. BGF	.6772 -0.80%	.7008 -0.72%	.5840 -1.22%	3.604 -1.04%	.6531 -0.82%	.6930 -0.78%	.7356 -0.62%
w/o. GAT	.6790 -0.54%	.7023 -0.50%	.5870 -0.37%	3.585 -0.54%	.6549 -0.55%	.6950 -0.49%	.7373 -0.39%
w/o. CA	.6801 -0.37%	.7034 -0.34%	.5876 -0.27%	3.570 -0.44%	.6562 -0.35%	.6964 -0.29%	.7383 -0.26%
w/o. QA	.6793 -0.49%	.7026 -0.45%	.5869 -0.39%	3.579 -0.69%	.6551 -0.52%	.6953 -0.45%	.7374 -0.38%
FNPS	.6827 -	.7058 -	.5902 -	3.486 -	.6585 -	.6985 -	.7402 -

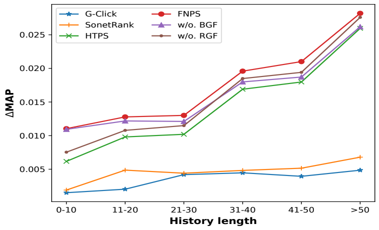

Figure 2: The study of different lengths of history.

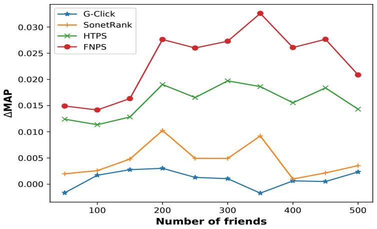

Figure 3: The study of different numbers of friends.

not take timeliness into account, which is a key feature in mobile search. All of deep learning based methods extract the sequential information of query logs and achieve better results. Our model has a significant advantage on P@1, which verifies the reliability of re-ranking the results according to the profiles of similar users.

In summary, **our model integrates search behaviour and friend network to improve the reliability of finding similar users, and a more accurate group profile can be constructed based on them with neural networks for personalized search.**

5.2 Ablation Experiments

The FNPS model mainly proposes a group formation method combining the friend relation and search behaviour, and then uses neural networks to model the group profile for personalized search. To verify the effectiveness of each component in this process, we conduct ablation experiments on the relation-based group formation, the behaviour-based group formation, the GAT, the cross attention layer, and the query-aware attention. Specifically, the implementation details are as follows.

FNPS w/o. RGF. We remove the relation-based group formation and only keep the behaviour-based friend circles.

FNPS w/o. BGF. The behaviour-based group formation is removed and only the relation-based friend circles are reserved.

FNPS w/o. GAT. We replace the GAT with averaging the vector of users in the friend circle to represent this circle.

FNPS w/o. CA. We discard the cross attention layer, which means there is no interaction between two types of friend circles.

FNPS w/o. QA. We abandon the query-aware attention when building the group profile. Instead, we simply average the representation of each circle after cross attention layer.

The ablation results are shown in Table 3. It can be observed that all ablation models perform worse than the whole framework. Specifically, removal of either relation-based or behaviour-based group formation causes considerable damage to the results, which shows the necessity of each type of friend circle on building the group profile. The contribution of the other three components are relatively smaller, but they still have a certain impact on the results. The drop caused by removing GAT shows that deep learning can discover common interests in a friend circle, thereby reducing the impact of irrelevant behaviours of others. Eliminating cross attention layer leads to about 0.37% decline on MAP, which reveals that two types of friend circles are complementary in building the group profile. The effectiveness of query-aware attention has been proved by the previous work [11]. Similarly, our results consistently show that discarding the query-aware attention will reduce the accuracy of the user profile and damage the results.

5.3 Comparison of Different History Lengths

Personalized search usually depends on the user's historical behaviour. The more the user interacts with the search engine, the more accurate the user profile is constructed by the model. In order to analyze the impact of different history lengths on the model more specifically, we divide the queries with different history lengths into different groups at 10 intervals. We choose G-Click, SonetRank, and HTPS models as the representatives of each type of baseline methods. To observe the role of the two kinds of friend circles, we also take w/o. RGF and w/o. RGF for comparison. Here we calculate the MAP improvement over original ranking to show the performance.

The results are shown in Figure 2. It can be seen that a longer history helps all personalization methods to improve the ranking quality in general. As the history length grows, the deep learning based methods show more powerful ability to extract the personalized information in the history. Comparing our model FNPS and the best baseline model HTPS, we find that when the user's historical activity is limited, our model improves more obviously. As the history length increases, the gap between them becomes smaller. This indicates that **when the user has enough search history, the user's query intent can be well determined according to his individual profile. The group profile is more useful when the individual profile is weak in personalizing the results**. The performance of the two ablation models demonstrates that the relation-based and behaviour-based friend circles are complementary. Removing RGF causes more drops when the

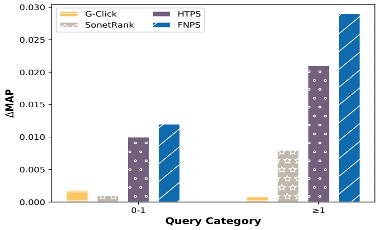

Figure 4: The study of queries with different click entropies.

history length is short, while discarding BGF damages the results obviously for queries with long history. Two grouping ways focus on different scenarios and the combination of them makes our model more robust.

5.4 Effect of the Number of Friends

Since our personalization model is based on a friend network, the number of friends will have an impact on the model effect. To explore this issue, we divide users into multiple groups according to the number of friends at intervals of 50, and test the performance of different models. For comparison, we select two friend network independent baselines G-Click and HTPS, and two relation-based models SonetRank and FNPS to conduct the experiment.

The results shown in Figure 3 reveal that our model FNPS outperforms all baselines in each interval. We also find that friend network related models fluctuate more when the number of friends changes, while the G-Click and HTPS models are relatively stable. Comparing our model FNPS with the best baseline model HTPS, the gap between them implies the contribution of introducing friend network into personalized search for better group formation. It can be seen that the contribution of the friend network is growing when the number of friends increases from 0 to 350. This demonstrates that **more friends can provide more effective reference information for the model to optimize the user profile**. However, when the number of friends is larger than 350, the gap between HTPS and FNPS is getting smaller. This indicates that too many friends will lead to more noise in modeling the group profile.

5.5 Effect of Different Click Entropy

In search engines, the query can be divided into navigational query and non-navigational query according to the purpose of the query. The intent of the former is clearer and has little diversity for different users, while the latter often has multiple meanings. We choose the click entropy [8] to measure the ambiguity of the query and divide the tested queries with cutoff of click entropy at 1.0. Queries with larger click entropy are ambiguous which require more personalization. To study the contribution of our model, we compare the improvement over original ranking of FNPS with the baselines models G-Click, SonetRank, and HTPS on the two query sets.

As shown in Figure 4, all personalized models outperform the original ranking on both query sets, while the improvement on non-navigational queries is more obvious. Specifically, deep learning based models show significant improvement over the traditional models. Compared to the best baseline model HTPS, our model improves the ranking quality on both query sets, especially on the query set with larger click entropy. This indicates that by integrating search behaviour and friend network, a more accurate group profile can be constructed to clarify the query intent.

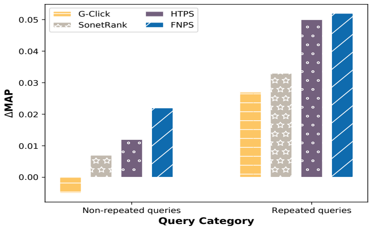

Figure 5: The study of repeated and non-repeated queries.

5.6 Repeated Queries vs. Non-repeated Queries

In this experiment, we categorize the test query sets into repeated and non-repeated queries. A repeated query is a query that the user has submitted before, which is easy to infer the query intent based on the historical click-through data. But for the non-repeated queries, it is hard to understand the user's intent directly based on existing click-based features. According to our statistics, about 75% of queries are issued for the first time. The understanding of these queries will greatly contribute to the improvement of the results. To compare the performance of different models on these two query sets, we choose the same baselines as the previous experiment.

From Figure 5, we observe that all personalized models perform better on repeated queries. However, the G-Click even performs worse than original ranking on non-repeated queries, which indicates the difficulty of inferring users' click behaviour on these queries. To tackle this issue, the SonetRank introduces social network and improves the results. The deep learning based model HTPS further enhances the performance by context-aware representation learning. Our model FNPS outperforms all baselines on both query sets, especially on non-repeated queries. This confirms that the group profile constructed by our model can accurately understand user query intent even if the query is a new one.

6 CONCLUSION

In this work, we propose a group based personalized search model integrating search behaviour and friend network to refine the user profile. We combine the user's search behaviour and the friend network to form multiple friend circles and highlight similar users with neural networks. Under the interaction of relation-based and behaviour-based friend circles, the users who appear in both types of friend circles are enhanced to build the group profile. Finally, the search results can be personalized by combining the individual profile and the group profile with respect to the current query. Experimental results show a significant improvement of our model compared with existing personalized strategies.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06.

REFERENCES

[1] Matthias Bender, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Ralf Schenkel, and Gerhard Weikum. 2008. Exploiting social relations for query expansion and result ranking. In *ICDE Workshops*. IEEE Computer Society, 501–506.

[2] Paul N Bennett, Krysta Svore, and Susan T Dumais. 2010. Classification-enhanced ranking. In *Proceedings of the WWW'2010*. ACM, 111–120.

[3] 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 SIGIR'2012*. ACM, 185–194.

[4] Truls Amundsen Bjørklund, Michaela Götz, Johannes Gehrke, and Nils Grimsmo. 2011. Workload-aware indexing for keyword search in social networks. In *CIKM*. ACM, 535–544.

[5] Fei Cai, Shangsong Liang, and Maarten De Rijke. 2014. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the SIGIR'2014*. ACM, 835–838.

[6] Mark J. Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards query log based personalization using topic models. In *Proceedings of the CIKM'2010*. 1849–1852.

[7] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. 2009. Personalized social search based on the user's social network. In *CIKM*. ACM, 1227–1236.

[8] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *WWW'2007*. ACM, 581–590.

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *WWW*. ACM, 417–426.

[10] 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 (CIKM '18)*.

[11] Morgan Harvey, Fabio Crestani, and Mark J Carman. 2013. Building user profiles from topic models for personalised search. In *CIKM'2013*. ACM, 2309–2314.

[12] Abhijith Kashyap, Reza Amini, and Vagelis Hristidis. 2012. SonetRank: leveraging social networks to personalize search. In *CIKM*. ACM, 2045–2049.

[13] Lin Li, Zhenglu Yang, Botao Wang, and Masaru Kitsuregawa. 2007. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *Advances in Data and Web Management*. Springer, 228–240.

[14] Xiujun Li, Chenlei Guo, Wei Chu, Ye-Yi Wang, and Jude Shavlik. 2014. Deep learning powered in-session contextual ranking using clickthrough data. In *NIPS'2014*.

[15] Shuqi Lu, Zhicheng Dou, Xu Jun, Jian-Yun Nie, and Ji-Rong Wen. 2019. PSGAN: A Minimax Game for Personalized Search with Limited and Noisy Click Data. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*.

[16] Shuqi Lu, Zhicheng Dou, Chenyan Xiong, Xiaojie Wang, and Ji-Rong Wen. 2020. Knowledge Enhanced Personalized Search. In *SIGIR*. ACM, 709–718.

[17] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).

[18] Meredith Ringel Morris and Eric Horvitz. 2007. SearchTogether: an interface for collaborative web search. In *UIST*. ACM, 3–12.

[19] Nan Mu, Daren Zha, Yuanye He, and Zhihao Tang. 2019. Graph Attention Networks for Neural Social Recommendation. In *ICTAI*. IEEE, 1320–1327.

[20] Ahu Sieg, Bamshad Mobasher, and Robin Burke. 2007. Web search personalization with ontological user profiles. In *CIKM'2007*. ACM, 525–534.

[21] Yang Song, Hongning Wang, and Xiaodong He. 2014. Adapting deep ranknet for personalized search. In *WSDM'2014*. ACM, 83–92.

[22] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2010. Potential for personalization. *ACM Trans. Comput. Hum. Interact.* 17, 1 (2010), 4:1–4:31.

[23] Jaime Teevan, Daniel J Liebling, and Gayathri Ravichandran Geetha. 2011. Understanding and predicting personal navigation. In *WSDM'2011*. ACM, 85–94.

[24] Jaime Teevan, Meredith Ringel Morris, and Steve Bush. 2009. Discovering and using groups to improve personalized search. In *WSDM*. ACM, 15–24.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

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

[27] Maksims Volkovs. 2015. Context models for web search personalization. *arXiv preprint arXiv:1502.00527* (2015).

[28] Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. 2014. Collaborative personalized Twitter search with topic-language models. In *SIGIR*. ACM, 53–62.

[29] Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search personalization with embeddings. In *ECIR'2017*. Springer, 598–604.

[30] Thanh Vu, Dawei Song, Alistair Willis, Son Ngoc Tran, and Jingfei Li. 2014. Improving search personalisation with dynamic group formation. In *SIGIR'2014*. 951–954.

[31] Thanh Vu, Alistair Willis, Son N Tran, and Dawei Song. 2015. Temporal latent topic user profiles for search personalisation. In *ECIR'2015*. Springer, 605–616.

[32] Xiaodong Wang, Zhen Liu, Nana Wang, and Wentao Fan. 2020. Relational Metric Learning with Dual Graph Attention Networks for Social Recommendation. In *PAKDD (1) (Lecture Notes in Computer Science, Vol. 12084)*. Springer, 104–117.

[33] Ryen W White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *WWW'2013*. ACM, 1411–1420.

[34] Ryen W White and Steven M Drucker. 2007. Investigating behavioral variability in web search. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 21–30.

[35] Qiang Wu, Chris JC Burges, Krysta M Svore, and Jianfeng Gao. 2008. *Ranking, boosting, and model adaptation. Tecnical Report.* Technical Report. MSR-TR-2008-109.

[36] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2020. Employing Personal Word Embeddings for Personalized Search. In *SIGIR*. ACM, 1359–1368.

[37] Jing Yao, Zhicheng Dou, Jun Xu, and Ji-Rong Wen. 2020. RLPer: A Reinforcement Learning Model for Personalized Search. In *WWW*. ACM / IW3C2, 2298–2308.

[38] Yukun Zhao, Shangsong Liang, and Jun Ma. 2016. Personalized Re-ranking of Tweets. In *WISE (2) (Lecture Notes in Computer Science, Vol. 10042)*. 70–84.

[39] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding History with Context-aware Representation Learning for Personalized Search. In *SIGIR*. ACM, 1111–1120.

[40] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM*. ACM, 789–797.