USER: A Unified Information Search and Recommendation Model based on Integrated Behavior Sequence

Jing Yao[2], Zhicheng Dou[1]*, Ruobing Xie[3], Yanxiong Lu[3], Zhiping Wang[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
{jing_yao,dou}@ruc.edu.cn,jirong.wen@gmail.com

ABSTRACT

Search and recommendation are the two most common approaches used by people to obtain information. They share the same goal – satisfying the user's information need at the right time. There are already a lot of Internet platforms and Apps providing both search and recommendation services, showing us the demand and opportunity to simultaneously handle both tasks. However, most platforms consider these two tasks independently – they tend to train separate search model and recommendation model, without exploiting the relatedness and dependency between them. In this paper, we argue that jointly modeling these two tasks will benefit both of them and finally improve overall user satisfaction. We investigate the interactions between these two tasks in the specific information content service domain. We propose first integrating the user's behaviors in search and recommendation into a heterogeneous behavior sequence, then utilizing a joint model for handling both tasks based on the unified sequence. More specifically, we design the **U**nified Information **SE**arch and **R**ecommendation model (**USER**), which mines user interests from the integrated sequence and accomplish the two tasks in a unified way. Our unified model has several advantages: (1) Aggregating search and recommendation logs together can alleviate the problem of data sparsity. (2) More accurate user interests can be learned from integrated behavior sequences. (3) We can capture the potential relatedness between the two tasks to essentially promote each other. Experiments on a dataset from a real-world information content service platform verify that our model outperforms separate search and recommendation baselines.

CCS CONCEPTS

• **Information systems** → **Web search engines**; **Personalization**; **Recommender systems**.

KEYWORDS

personalized search; recommendation; unified model

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482489

ACM Reference Format:
Jing Yao[2], Zhicheng Dou[1]*, Ruobing Xie[3], Yanxiong Lu[3], Zhiping Wang[3] and Ji-Rong Wen[4,5]. 2021. USER: A Unified Information Search and Recommendation Model based on Integrated Behavior Sequence. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482489

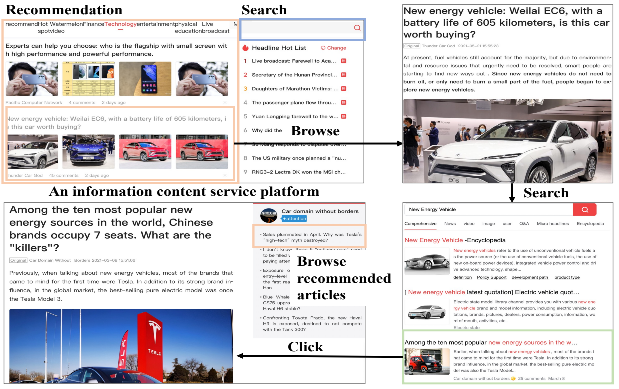
1 INTRODUCTION

On Internet platforms, search and recommendation are two major approaches to help users obtain the required knowledge. In this paper, we mainly focus on the domain of information content service which aims to deliver news feeds, tweets, or web articles to users[1]. In order to improve users' satisfaction with search and recommendation results, a lot of personalized search models and recommendation models have been proposed [5, 10, 13, 14, 20, 36, 37]. These models aim to mine user preferences from their historical behaviors to infer their current intents and generate a personalized document ranking list that can satisfy the current user interest. Typically, many deep learning based personalized search models learn a representation of user interests from her search history to re-rank the candidate documents[13, 20, 21, 43, 44, 48]. Recommendation models also present document ranking lists according to the user's browsing history [14, 36, 37, 39, 41]. However, most existing studies concentrate on only one single task, namely either search or recommendation. They devise a specific model applicable for one task, but rarely consider their combination.

Currently, there are more and more mobile Apps and websites where both information search and recommendation services are available. For the example of Toutiao[2] platform shown in Figure 1, users can not only actively issue queries to seek information, but also browse the recommended articles. Indeed, some early attempts of combining the two services have already been applied. For example, some articles are recommended along with the clicked search results. Queries may also be suggested at the end of a recommended news article. **Therefore, how to effectively aggregate the two tasks together is an essential and valuable problem.**

Actually, some early studies [2] have discussed the similarity between search and recommendation. The two tasks share the same target – helping people get the information they require at the right time. Zamani and Croft [46] propose a vanilla joint learning framework to handle both tasks at the same time. They train two separate

[1] Example information content service platforms are twitter, toutiao, and wechat.
[2] https://www.toutiao.com/


Recommendation
Search
Browse
An information content service platform
Search
Browse recommended articles
Click

recommendHot WatermelonFinanceTechnologyentertainmentphysical Live spotvideo educationbroadcast

Experts can help you choose: who is the flagship with small screen with high performance and powerful performance.

Pacific Computer Network 4 comments 2 days ago

New energy vehicle: Weilai EC6, with a battery life of 605 kilometers, is this car worth buying?

Thunder Car God 45 comments 2 days ago

Headline Hot List Change

1 Live broadcast: Farewell to Aca...
2 Secretary of the Hunan Provinci...
3 Daughters of Marathon Victims: ...
4 The passenger plane flew throu...
5 Yuan Longping farewell to the w...
6 Why did the
7 Su Mang responds to disputes over...
8 The US military once planned a "nu...
9 RNG3-2 Lectra DK won the MSI ch...

New energy vehicle: Weilai EC6, with a battery life of 605 kilometers, is this car worth buying?

Original Thunder Car God 2021-05-21 15:55:23

At present, fuel vehicles still account for the majority, but due to environmental and resource issues that urgently need to be resolved, smart people are starting to find new ways out . Since new energy vehicles do not need to burn oil, or only need to burn a small part of the fuel, people began to explore new energy vehicles.

New Energy Vehicle

Comprehensive News video image user Q&A Micro headlines Encyclopedia

New Energy Vehicle -Encyclopedia

New energy vehicles refer to the use of unconventional vehicle fuels as the power source (or the use of conventional vehicle fuels, the use of new on-board power devices), integrated vehicle power control and drive advanced technology, shape...

definition Policy Support development path product type

[New energy vehicle latest quotation] Electric vehicle quot...

Electric state model library channel provides you with various new energy vehicle brand and model information, including electric vehicle quotations, brands, pictures, dealers, power consumption, information, word of mouth, activities, etc.

Electric state

Among the ten most popular new energy sources in the w...

Earlier, when talking about new energy vehicles , most of the brands that came to mind for the first time were Tesla. In addition to its strong brand influence, in the global market, the best-selling pure electric model was also the Tesla Model...

Car domain without borders 25 comments March 8

Among the ten most popular new energy sources in the world, Chinese brands occupy 7 seats. What are the "killers"?

Original Car Domain Without Borders 2021-03-08 15:51:06

Previously, when talking about new energy vehicles, most of the brands that came to mind for the first time were Tesla. In addition to its strong brand influence, in the global market, the best-selling pure electric model was once the Tesla Model 3.

Car domain without borders +attention

- Sales plummeted in April. Why was Tesla's "high-tech" myth destroyed?
- I don't know these 6 "ordinary cars" need to be filled paying atten
- Exposure o entry-level the first rea Han
- Blue Whale CS75 upgra Haval H6 stable?
- Confronting Toyota Prado, the new Haval H9 is exposed, destined to not compete with the Tank 300?

Figure 1: Illustration of an information content service platform with both search and recommendation services.

models for the two tasks through a joint loss, but neglect the essential relatedness between them in human information-seeking behaviors. Actually, users usually switch between the two services when they are obtaining information from the Web. Let us take the example in Figure 1 for illustration. When a user browses the article list generated by the recommendation system, she is attracted by the article titled "New energy vehicle: Weilai ...?". After reading this article, she switches to the search engine and issues a query to seek more knowledge about "New energy vehicle". Then, she browses the search results and articles recommended along with the clicked document to know more. Such an information-seeking pattern which mixes behaviors made in proactive searches and passive recommendations is common in our surfing process. From the example, we find that the user may switch between the search service and recommender system for a single target, both the search behaviors and browsing behaviors reflect her personalized information need. Therefore, jointly modeling the entire user behavior sequence is expected to discover real user intents more precisely. Besides, some close associations may exist between the two kinds of behaviors that browsing could stimulate search and search might impact browsing in the future. Richer interaction and training data is available. Motivated by this scenario, **we pay attention to jointly modeling both tasks of personalized search and recommendation in the information content domain, exploring the potential relatedness between their corresponding user behaviors to promote each other.**

To begin with, we integrate the user's historical search and browsing behaviors in chronological order, getting a simplified heterogeneous behavior sequence shown in Figure 2. B represents browsed articles, Q indicates queries issued by the user, and D is documents clicked under the corresponding query. Then, we propose a **U**nified Information **SE**arch and **R**ecommendation model (**USER**) to encode the heterogeneous sequence and solve the two tasks in a unified way. We think recommendation and personalized search share the same paradigm: recommendation can be treated as personalized search taking an `EMPTY` query. Hence we design the USER model in a personalized ranking style, to rank candidate documents based on the input query (using empty for recommendation) and the user preferences contained in the integrated behavior sequence. This model has several advantages. **First**, we aggregate the user's search and recommendation logs, alleviating the problem of data sparsity faced by a single task. **Second**, based on the merged behavior sequence, more comprehensive and accurate user profiles can be constructed, improving personalization performance. **Third**, the potential relatedness between search and recommendation can be captured to essentially promote each other.

Specifically, our **USER** model is composed of four modules. **First**, a text encoder is used to learn the representation for the documents and queries. **Second**, the session encoder models the integrated behavior sequence in the current session, captures relatedness between the search and browsing behaviors, and clarifies the user's current intention. As for a search behavior including a query and clicked documents with strong relevance, we employ a co-attention structure [25] to fuse their representations. Then, a transformer layer is constructed to capture the associations between the search and browsing behaviors in the session and fuse the context into the current intention. **Third**, the history encoder learns information from the long-term heterogeneous history sequence as an enhancement. **Finally**, we build a unified task framework to complete the two tasks in a unified way. We first pre-train the unified model with the training data from both tasks, alleviating data sparsity. Then, we make a copy for each task and finetune it with the corresponding task data to fit the individual data distribution. We experiment on a dataset comprised of search and browsing behaviors constructed from a real-world information content service platform with both search and recommendation engines. The results verify that our model outperforms separate baselines and alleviates data sparsity.

Our main contributions are summarized as follows: (1) We pay attention to both tasks of personalized search and recommendation. For the first time, we integrate separate behaviors of the two tasks into a heterogeneous behavior sequence. (2) We model the relatedness between a user's search and browsing behaviors to promote both personalized search and recommendation. (3) We propose a unified search and recommendation model (USER) that accomplishes the two tasks in a unified way with an encoder for the integrated behavior sequence and a unified task framework.

2 RELATED WORK

Personalized Search. Personalized search customizes search results for each user by inferring her personal intents. Early studies relied on features and heuristic methods to analyze user interests. Focusing on click features, Dou et al. [12] proposed P-Click to re-rank documents with their historical click counts. Topic-based features were applied to build user profiles [4, 9, 17, 26, 30, 31, 35]. The Open Directory Project (ODP) [26], learned or latent topic models [6] were used to obtain the topic-based information of a web page. Besides, the user's reading level and location are applied for personalization [3, 11]. Multiple features were combined with a learning to rank method [7, 8] to compute a personalized score [5, 29].

Recently, deep learning was applied to capture potential user preferences. Song et al. [27] leveraged personal data to adapt a general ranking model. Ge et al. [13] devised a hierarchical RNN with query-aware attention to dynamically mine preference information. Lu et at. [20] employed GAN [15] to enhance the training data. Yao et al. [44] adopted reinforcement learning to learn user interests.

Zhou et al. [49] explored re-finding behaviors with a memory network. The latest studies were committed to disambiguating the query by introducing entities [21], training personal word embeddings [43], or involving search history as the context [48]. All these models are specially designed for the personalized search task.

Information Recommendation Models. Personalized content recommendation is critical to help users alleviate information overload and find something interesting. Traditional recommendation systems mainly depended on collaborative filtering (CF) [24] and factorization machine (FM) [23]. With the emergence of deep learning, many models combined both low- and high-order feature interactions, such as Wide & Deep [10] and DeepFM [16]. Specially, representation based models have been studied for the recommendation of news articles that have abundant textual information. These models include two modules: a text encoder to obtain article representations and a user encoder to learn user representation from her browsing history. Then, articles are ranked based on their relevance with the user. Okura et al [22] devised an auto-encoder to learn news representations, and used an RNN to generate user representations. Wu et al. [36] learned article vectors from titles, bodies and topic categories. User representation was a weighted sum of the browsed news vectors. Wu et al. [37] set user embeddings to generate personalized attention to calculate the article and user representations. They also exploited multi-head self-attention [28] to capture contextual information [39]. LSTUR [1] kept both short-term and long-term user profiles. To enhance text representations, entities in the article and their neighbors in the knowledge graph are considered [19, 32, 33]. The GNN structure [40] was also adopted to model high-order relatedness between users and articles [14, 18]. In these models, only the recommendation task is discussed.

Joint Search and Recommendation. Some studies considered both the search and recommendation tasks. In e-commerce, an early work [34] built a unified recommendation and search system by merging their features. Zamani et al. [46] proposed a joint learning framework that simultaneously trains a search model and a recommendation model by optimizing a joint loss. For the situation with only recommendation data but not search logs, a multi-task framework was trained on browsing interactions [47]. These joint methods simply combined the two tasks and train two separate models through multi-task learning or joint loss, without exploring more essential dependency between them. Search history was also used to help generate recommendations for the users with little browsing history [38, 45]. This model just targeted one single task with data from the other task as complementary information. In this paper, we propose a unified model to solve the two tasks at the same time, mining the relatedness between their corresponding user behaviors to promote each other.

3 PROBLEM DEFINITION

Search and recommendation are two main approaches to help people obtain information. Many separate personalized search models and recommendation models have been proposed. As analyzed in Section 1, people usually achieve their information targets through a mixture of proactive searches and passive recommendation, which is popular on information content service platforms with both search and recommendation engines. Both kinds of behaviors reflect the user's information need and preferences. Thus, compared to existing separate approaches, jointly modeling the two tasks and exploiting the relatedness between them might have the potential to promote each other. In this paper, we integrate the user's search and browsing behaviors into a sequence to discover more accurate user interests, then design a unified model to solve the two tasks in a unified way. Next, we define the new problem to be handled.

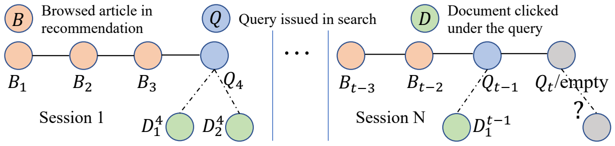


Figure 2: Illustration of the integrated behavior sequence. The target behavior is personalized search with a query Q_t or recommendation with an empty query.

Recall that we focus on the information content domain, let us formulate a user's behaviors with notations. On an information content service platform with both search engine and recommendation engine, the user u could browse articles B in the recommendation system, issue queries Q to seek for information and click satisfied documents D in the search engine. All these behaviors are sequential, so we integrate them into a heterogeneous behavior sequence in chronological order. Referring to existing session segmentation methods [13, 20], we divide the user's whole behavior sequence into several **sessions** with 30 minutes of inactivity as the interval. Past behaviors in the current session are viewed as the **short-term** history. The other previous sessions constitute the **long-term** history. Specifically, we denote the user's history sequence as $H = \{H^l, H^s\} = \{\{S_1, \ldots, S_{N-1}\}, S_N\}$, where N is the number of sessions. Each session S_i corresponds to a sub-sequence with both behaviors, such as $\{B_1, B_2, (Q_3, D_1^3, D_2^3), \ldots, \}$.

We illustrate the whole behavior sequence in Figure 2. The horizontal edges indicate the sequential relationship between two consecutive actions, while the slanted edges point to the documents clicked under the corresponding query. The blue vertical lines separate sessions. For example, in the current Session N, the user first browses two articles in the recommendation system. Then, she enters a query in the search engine and clicks a document under this query. At the current moment t, the user would perform a target behavior, either search with an issued query Q_t or browsing. For both tasks, we are supposed to infer the user's intent and return a personalized document list. Due to the same paradigm, we regard the recommendation task as personalized search with an empty query, and complete the two tasks in a unified personalized ranking style. Facing Q_t or an empty query, the model is required to return a personalized document list based on the query and the user interests learned from the user's integrated behavior sequence.

4 USER: THE UNIFIED MODEL

The architecture of our USER model is shown in Figure 3. First, the text encoder is used to learn representations for documents and queries. Second, the session encoder models the user's integrated behavior sequence within the current session to clarify her information need. Then, the history encoder enhances the user's intent

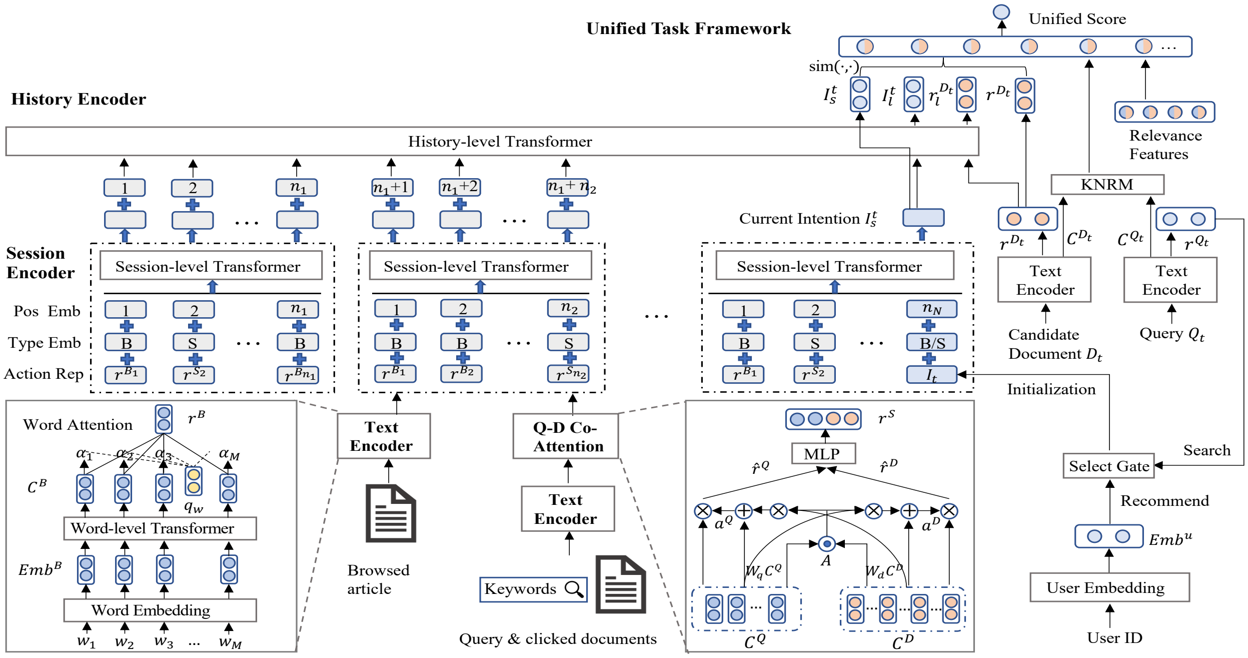


Figure 3: The architecture of our USER model. There are four major components: the text encoder to learn representations for queries and documents; the session encoder to model integrated behaviors in the current session; the history encoder to mine information from the long-term behavior sequence; the unified task framework to complete both tasks in a unified way.

representation by mining information from the long-term history. Finally, we design a unified task framework to complete personalized search and recommendation in a unified way. We present the details of each module in the remaining parts of this section.

4.1 Text Encoder

For each query Q, clicked document D and browsed article B, we apply the text encoder to learn their semantic representations. Taking the calculation of a browsed article B as an example, $B = [w_1, w_2, \ldots, w_M]$ where M is the number of words in the article, the complete text encoder can be divided into three sub-layers. The first is the word embedding layer that converts the word sequence into a matrix with word vectors, i.e. $\text{Emb}^B = [v_1, v_2, \ldots, v_M] \in R^{dim \times M}$. v_i corresponds to the low-dimensional word vector of w_i. In addition, contexts within the article are also helpful for users to figure out the true meaning of a word. For example, the different meanings of "Apple" in "Apple fruit" and "Apple company" can be distinguished based on the different contextual words "fruit" and "company". Therefore, we set a word-level transformer [28] as the second sub-layer to obtain the context-aware word representations $C^B \in R^{dim \times M}$ by capturing interactions between words.

$$C^B = \text{Transformer}_w(\text{Emb}^B). \tag{1}$$

The details about transformer can be referred to [28].

The last sub-layer is a word-level attention layer. In a piece of text, different words contribute different informativeness for expressing the semantics of this text. For instance, in the sequence 'symptoms of novel coronavirus pneumonia', the word 'symptoms' is very informative for learning the text representation, while 'of' has little information. To highlight important words in a text sequence, we exploit a word-level attention mechanism to give them larger weights. We set a trainable vector q_w as the query in the attention mechanism. The weights $\alpha \in R^M$ for all words are computed as:

$$\alpha = \text{softmax}(q_w^T \tanh(W_v \times C^B + b_v)), \tag{2}$$

where W_v and b_v are parameters. The final contextual representation of the browsed document $r^B \in R^{dim \times 1}$ is the weighted sum of all the word vectors, i.e. $r^B = \sum_{i=1}^{M} \alpha_i C_i^B$.

Contextual representations of the query r^Q and clicked document r^D are computed in the same way.

4.2 Session Encoder

At the current time t, the user u has a target action, either search or browsing. We represent her intention with a vector I_t. If the user issues a query Q_t for search, the intention I_t is initialized with the text representation of this query r^{Q_t} computed by the text encoder. Otherwise, we use the corresponding trainable user embedding Emb^u as initialization. This step is realized by a select gate, as:

$$I_t = \begin{cases} r^{Q_t} & \text{if the target behavior is search} \\ \text{Emb}^u & \text{if the target behavior is browsing} \end{cases} \tag{3}$$

Then, we mine information from the user's history comprised of search and browsing behaviors to clarify her personal intent I_t.

According to existing studies [13, 48], it is thought that behaviors within a session show consistency in the user's information need. Thus, the user's past behaviors during the current session could provide rich contextual information for deducing her current intention. In the unified search & recommendation scenario we study, there are both search and browsing actions in a session, as shown in Figure 2. We analyze that several possible relationships exist between the behaviors in the heterogeneous sequence: (1) For a document clicked under a query, we think this document satisfies the user's information need to be expressed by this query. It shows strong relevance between the query and the document. (2) After the user browses a series of recommended articles, she might be triggered to seek for more related information through proactive searches. (3) Queries are actively issued by the user, explicitly showing her preferences. With these queries and clicked documents, we can figure out the points of interest the user focuses on when browsing articles. We design a session encoder to capture these associations in the current session and employ the session context to enhance the intent representation.

First, for a historical query and the corresponding clicked documents, we are supposed to learn the strong relevance between them. Clicked documents indicate the user's intention contained in the query keywords, and the query highlights the important words in the documents. Thus, we suggest adopting the co-attention structure [25] to calculate their representation vectors by fusing their interactive information, instead of the vanilla word-attention mechanism. Taking a query Q and the clicked documents $D_1, D_2, \ldots$ as an example, the detailed computing process is as follows. At the first step, we obtain the contextual vector matrices C^Q and C^{D_i} for the query and each document through the word embedding layer and the word-level transformer of our text encoder. Vectors of all clicked documents are concatenated together as $C^D = [C^{D_1}; C^{D_2}; \ldots]$. Then, we compute an affinity matrix A between C^Q and C^D.

$$A = \tanh((C^Q)^T W_l C^D), \tag{4}$$

where $W_l \in R^{dim \times dim}$ is a weight matrix to be learned. The attention weights for the query and documents are calculated based on the interactive features in the affinity matrix, as:

$$H^Q = \tanh(W_q C^Q + (W_d C^D)A), \quad a^Q = \text{softmax}(W_{hq}^T H^Q), \tag{5}$$

$$H^D = \tanh(W_d C^D + (W_q C^Q)A^T), \quad a^D = \text{softmax}(W_{hd}^T H^D). \tag{6}$$

W_q, W_d, W_{hq}, W_{hd} are parameters. a^Q and a^D are the attention weights for query keywords and document terms respectively. We calculate the attended representation for the query and documents as the weighted sum of the contextual vectors C^Q and C^D.

$$\hat{r}^Q = \sum_{i=1}^{M} a_i^Q C_i^Q, \qquad \hat{r}^D = \sum_{i=1}^{M} a_i^D C_i^D. \tag{7}$$

The two vectors are concatenated to generate the representation of a historical search behavior r^S through an MLP layer, i.e. $r^S = \text{MLP}([r^Q; r^D])$. For a browsing behavior made in recommendation, it corresponds to only a browsed article B. Thus, its representation is just the article representation r^B calculated by the text encoder.

With the representation of all past behaviors in the current session calculated, $H^s = \{r^{B_1}, r^{S_2}, \ldots\}$, we could capture the relationships between the search and browsing behaviors, and fuse the session context into the user's current intention. We combine H^s with the target intention I_t and pass them through a session-level transformer for interaction. On account of the behaviors are sequential and heterogeneous, we add the position and type information of each behavior for clarification. The action type includes search (S) and browsing (B). Finally, the output of the last position I_t^s represents the user's current intention fusing the session context.

$$I_t^s = \text{Transformer}_s^{\text{last}}([H^s, I_t] + [H^s, I_t]_P + [H^s, I_t]_T). \tag{8}$$

$[H^s, I_t]_P$, $[H^s, I_t]_T$ are the position embedding and type embedding. $\text{Transformer}_s^{\text{last}}(\cdot)$ means taking the output of the last position.

4.3 History Encoder

With the session encoder described above, we clarify the user's current information need under the help of the short-term history, obtaining I_t^s. But for the situation with little session history, it is still ambiguous due to the lack of session context. The user's long-term behavior history often reflects relatively stable interests, which also provides some assistant information. Thus, we further model the long-term history to enhance the user's intent representation based on I_t^s. At first, we process each historical session with the session encoder to capture the connections between search and browsing behaviors, getting the contextual representation for all historical behaviors, $H^l = [\{r_s^{B_1}, r_s^{S_2}, \ldots\}, \{r_s^{B_1}, \ldots\}, \ldots]$. We concatenate all session sub-sequences as a long behavior sequence and combine it with the target action as $[H^l, I_t^s]$. Then, a history-level transformer module is conducted on the long-term heterogeneous sequence to fuse the history information into the current intention. To preserve the sequential information between actions, we involve the position of each behavior $\{1, 2, \ldots t\}$. In the final, we take the output of the last position as the user's intent representation enhanced by the long-term behavior history, denoted as I_t^l.

$$I_t^l = \text{Transformer}_h^{\text{last}}([H^l, I_t^s] + [H^l, I_t^s]_P), \tag{9}$$

where $[H^l, I_t^s]_P$ is the position embedding.

Motivated by some news recommendation models [37, 39], the user's attention to a document is also impacted by her interests. Besides, the user might intend to find a specific document that appeared in the history, as analyzed in [49]. Thus, for the candidate document D_t, we can use the long-term history to enhance its representation r^{D_t} calculated by the text encoder in the same way as the target intent, getting $r_l^{D_t}$

$$r_l^{D_t} = \text{Transformer}_h^{\text{last}}([H^l, r^{D_t}] + [H^l, r^{D_t}]_P). \tag{10}$$

We will use $r_l^{D_t}$ together with r^{D_t} to calculate the personalized ranking score for the candidate document in the unified task framework that will be introduced in the next part.

4.4 Unified Task Framework

As for the personalized search and recommendation tasks in the information content domain, the main difference between them is whether there is an issued query. In the problem definition, we claim to unify the two different tasks as a unified problem by regarding the recommendation task as personalized search with an empty query. We represent the user's current intention as I_t that is initialized with the issued query Q_t for search or the user embedding Emb^u for recommendation. The unified problem is to rank the candidate document D_t based on the personalized relevance that is calculated with the current intention I_t, the query Q_t (empty for recommendation) and the user history H. The personalized relevance is denoted as $p^{\texttt{unified}}(D_t | I_t, Q_t, H)$.

Table 1: Basic statistics of the dataset.

Item	Statistic	Item	Statistic
#days	92	#search records	203,190
#users	100,000	#recommend records	840,867
#sessions	515,247	avg. session length	3.58

Through the text encoder, session encoder and history encoder, we get the representations of the user's current intention and candidate document, i.e. I_t^s, I_t^l, r^{D_t} and $r_l^{D_t}$. We calculate the relevance between each pair of them by cosine similarity $\mathrm{sim}(\cdot,\cdot)$. Moreover, for the personalized search task, the correlation between the candidate document and the query keywords is also critical. Thus, we additionally pay attention to the interactive features between the context-aware representations of the query and document, i.e. C^{Q_t} and C^{D_t}. We exploit the interaction-based component KNRM [42] to calculate the interactive score $\mathrm{inter}(C^{Q_t}, C^{D_t})$. The detailed calculation process can be found in [42]. Besides, following [13, 20], we also extract several relevance-based features $F_{q,d}$ for personalized search. When calculating the relevance for articles in recommendation, the interaction score and features are all empty. Finally, the score for the candidate document is calculated by aggregating all these scores and features with an MLP layer, as:

$$f^{\texttt{unified}} = [\mathrm{sim}(I_t^s, r^{D_t}), \mathrm{sim}(I_t^l, r^{D_t}), \mathrm{sim}(I_t^s, r_l^{D_t}), \mathrm{sim}(I_t^l, r_l^{D_t}), \mathrm{inter}(C^{Q_t}, C^{D_t}), F_{q,d}], \quad (11)$$

$$p^{\texttt{unified}}(D_t|I_t, Q_t, H) = \Phi(f^{\texttt{unified}}). \quad (12)$$

$\Phi()$ represents an MLP layer without an activation function. Whether for the search or recommendation task, we generate personalized document list by calculating relevance scores in this way.

4.5 Training and Optimization

We adopt a pairwise manner to train our USER model. For both personalized search and recommendation tasks, we construct each training sample as a document group comprised of a positive document and K negative documents presented in the same impression, represented as $\{D^+, (D_1^-, \ldots, D_K^-)\}$. For each document group, we aim to maximize the score of the positive document and minimize that of those negative documents. The loss $\mathcal{L}$ is computed as the negative log-likelihood of the positive sample. We have:

$$\mathcal{L} = -\log\Big(\frac{\exp(p^{\texttt{unified}}(D^+))}{\exp(p^{\texttt{unified}}(D^+)) + \sum_{i=1}^{K} \exp(p^{\texttt{unified}}(D_i^-))}\Big). \quad (13)$$

where $p^{\texttt{unified}}(\cdot)$ is the abbreviation of $p^{\texttt{unified}}(\cdot|I_t, Q_t, H)$. We minimize the loss with the Adam optimizer.

In the unified scenario, we have access to both search and recommendation data. Thus, we can train one USER model with data from the two tasks and apply the trained model to both of them. However, there may be a problem that some gaps exist between the data distributions of the search task and recommendation task. The only unified model trained on the data from the two tasks is difficult to achieve the best performance on both of them. Therefore, we propose an alternative training method. We first pre-train a unified model with both task data. Then, we make a copy for each task and finetune it with the corresponding task data to fit the individual data distribution. In this case, the model not only benefits from more training data but also adapts to the specific task.

5 EXPERIMENTAL SETTINGS

5.1 Dataset and Metrics

Dataset There is no public dataset with both search and recommendation logs of a shared set of users in the information content domain. To evaluate our unified model, we construct a dataset comprised of users' search and browsing behaviors from a popular information service platform that has both search and recommendation engines. We randomly sample 100,000 users. Then, we obtain their search logs in its search engine and browsed articles recommended by the recommendation system for three months. The whole log is preprocessed via data masking to protect user privacy.

Each piece of search data includes an anonymous user ID, the action time, a query, top 20 returned documents, click tags and click dwelling time. As for each recommendation record, only a browsed article is kept, without other presented but unclicked documents. We generate pseudo unclicked documents for each browsed article for model training. We rank all documents in the recommendation log based on a weighted score of the popularity measured by the click count and the topic similarity with the browsed article calculated by cosine similarity. Nine negative documents ranked at the top are sampled for each browsed article. The original recommendation list is randomly shuffled. All search and browsing behaviors of each user are merged into a sequence in chronological order.

We separate a user's whole behavior sequence into sessions with 30 minutes of inactivity as the interval [13, 20]. Users' browsing behaviors are usually more frequent than search behaviors, which leads to an unbalance in the dataset. Since we intend to explore the relatedness between the search and browsing behaviors, we sample sessions containing both actions and three sessions before and after these sessions. To guarantee each user has enough history for building user profile, we treat the log data of the first eight weeks as the historical set and the other five weeks log as the experimental data. The experimental data is used for training, validation and testing with 4:1:1 ratio. The statistics are shown in Table 1.

Metrics Referring to existing works [37, 39], the recommendation task is also to re-rank the candidate documents. For both tasks, we take the sat-clicked documents with more than 30 seconds of dwelling time as relevant and the others as irrelevant. We choose common ranking metrics to evaluate our model and baselines, including MAP, MRR, P@1, Avg.C (average position of the clicked documents), NDCG@5 and NDCG@10. For recommendation, we also adopt AUC to measure the click-through rate.

5.2 Baselines

The original search results are returned by the search engine. The original recommendation lists are randomly shuffled. Besides, we compare our model with state-of-the-art personalized search models, news recommendation models and the joint framework [46].

HRNN [13]: A hierarchical RNN model with query-aware attention to dynamically mine relevant history information.

RPMN [49]: This model captures complex re-finding patterns of previous queries or documents with the memory network.

Table 2: Overall performance in various scenarios: personalized search with only search data, recommend with browsing data, unified search & recommend with both data. The relative percentages are computed based on the original rankings. The best results in each scenario are shown with underlines. The overall best results are in bold. "†" indicates significant improvements over all corresponding baselines, with paired t-test at p < 0.05 level. MRR is used for search and AUC for recommendation.

Scenarios	Models	MAP		MRR/AUC		P@1		Avg.C		NDCG@5		NDCG@10	
Personalized Search	Ori.	.6614	-	.6859	-	.5766	-	3.898	-	.6720	-	.7148	-
	HRNN	.6707	+1.41%	.6951	+1.34%	.5800	+0.59%	3.727	+4.39%	.6839	+1.77%	.7258	+1.54%
	RPMN	.6724	+1.66%	.6962	+1.50%	.5795	+0.50%	3.642	+6.57%	.6859	+2.07%	.7296	+2.07%
	PEPS	.6727	+1.71%	.6979	+1.75%	.5811	+0.78%	3.626	+6.98%	.6867	+2.19%	.7303	+2.17%
	HTPS	.6749	+2.04%	.6991	+1.92%	.5821	+0.95%	3.615	+7.26%	.6894	+2.59%	.7326	+2.49%
	USER-S	.6768	+2.33%	.7008	+2.17%	.5831	+1.13%	3.564	+8.57%	.6909	+2.81%	.7341	+2.70%
Recommend	Ori.	.2928	-	.4994	-	.0989	-	5.493	-	.2952	-	.4543	-
	NPA	.4787	+63.49%	.7104	+42.25%	.2679	+170.9%	3.489	+36.48%	.5306	+79.74%	.6037	+32.89%
	LSTUR	.4841	+65.33%	.7131	+42.79%	.2767	+179.8%	3.564	+35.12%	.5268	+78.46%	.6075	+33.72%
	NRMS	.4902	+67.42%	.7124	+42.65%	.2794	+182.5%	3.584	+34.75%	.5263	+78.29%	.6128	+34.89%
	GERL	.4591	+56.80%	.7229	+44.75%	.2417	+144.4%	3.499	+36.30%	.5253	+77.95%	.5885	+29.54%
	USER-R	.4975†	+69.91%	.7326†	+46.70%	.2825†	+185.6%	3.381†	+38.45%	.5386†	+82.45%	.6204†	+36.56%
Unified Search & Recommend	Personalized Search Task												
	HTPS	.6789	+2.65%	.7012	+2.23%	.5851	+1.47%	3.546	+9.03%	.6941	+3.29%	.7369	+3.09%
	JSR-HRNN	.6705	+1.38%	.6954	+1.39%	.5796	+0.52%	3.724	+4.46%	.6843	+1.83%	.7258	+1.54%
	USER	**.6845**†	+3.49%	**.7074**†	+3.13%	**.5892**†	+2.19%	**3.479**†	+10.75%	**.7001**†	+4.18%	**.7422**†	+3.83%
	Recommendation Task												
	NRMS	.4984	+70.22%	.7203	+44.23%	.2948	+187.9%	3.496	+36.35%	.5361	+81.61%	.6147	+35.31%
	JSR-NRMS	.4931	+68.41%	.7251	+45.19%	.2765	+179.6%	3.386	+38.36%	.5432	+84.01%	.6143	+35.22%
	USER	**.5035**†	+71.96%	**.7442**†	+49.02%	**.2910**†	+194.2%	**3.258**†	+40.69%	**.5514**†	+86.79%	**.6222**†	+36.96%

PEPS [43]: Yao et al. claim that different users have different understandings of the same word due to their knowledge. They learn personal word embeddings to clarify the query keywords.

HTPS [48]: It encodes the user's history as the context information to disambiguate the current query. We adapt it to the unified scenario by adding the user's browsed articles into her history.

NPA [37]: The model sets user embeddings to compute personalized word- and news-level attention. It highlights important words and articles to generate informative news and user representations.

NRMS [39]: This model utilizes multi-head self-attention to learn news and user representations by capturing the relatedness between words and browsed articles. By adding documents clicked in the search history, we adapt it into the unified scenario.

LSTUR [1]: It includes the short-term user interests modeled from the recent clicked articles with GRU and the long-term profile corresponding to a trainable user embedding.

GERL [14]: It applies transformer on the user's interaction graph to capture high-order associations between users and news.

JSR [46]: This is a general joint framework that trains a separate search model and a recommendation model by optimizing a joint loss. We select HRNN for search and NRMS for recommendation.

USER: This is the unified model proposed in this paper. **USER-S** and **USER-R** are the variants used in independent search and recommendation scenarios respectively. They share the same structure as **USER** but have access to only the data of that single task.

We conduct multiple sets of experiments to decide the model parameters as follows[3]. The size of word embeddings, pretrained by word2vec on all logs, and user embeddings is 100. Due to users' click decisions are usually made based on titles, we use titles in our experiment, instead of complete articles. For each query or document title, the max sequence length is 30. In the history sequence, we maintain up to 20 sessions and the maximum number of user behaviors in a session is 5. The number of heads in the transformer is 8 and the hidden dimension is 50. The number K of negative samples in each document group is 4. The learning rate is $1e - 3$.

6 EXPERIMENTAL RESULTS

6.1 Overall Performance Comparison

We compare all models in various scenarios: pure personalized search with only search data, pure recommendation with only recommendation data and unified scenario with both data. The results are shown in Table 2. We have several findings:

(1) The comparison of the same model trained with the independent task dataset and the unified dataset. **For HTPS, NRMS and USER, their performance on the unified dataset is better than that on the independent task data.** For example, the personalized search model HTPS trained on the unified history promotes 0.6% in MAP based on that trained with pure search data. The recommendation model NRMS has 1.6% improvement in MAP with the unified data. Consistently, our USER model in the unified scenario also shows improvements over USER-S and USER-R on all metrics. Compared to the independent task data, the unified dataset is comprised of both search and browsing behaviors, from which we analyze the user's preferences. The results demonstrate that a

[3] The code is on https://github.com/jingjyyao/Personalized-Search/tree/main/USER

Table 3: Performance of USER variants. The relative percentages are calculated based on the complete USER model.

Variants	Personalized Search						Recommendation					
	MAP		NDCG@5		NDCG@10		MAP		NDCG@5		NDCG@10	
USER	**.6845**	-	**.7001**	-	**.7422**	-	**.5035**	-	**.5514**	-	**.6222**	-
w/o Session Encoder	.6760	-1.24%	.6934	-0.96%	.7343	-1.06%	.3963	-21.29%	.4504	-18.32%	.5393	-13.32%
w/o History Encoder	.6768	-1.12%	.6937	-0.91%	.7348	-1.00%	.4443	-11.76%	.5026	-8.85%	.5766	-7.33%
w/o Unified Pre-train	.6801	-0.64%	.6953	-0.69%	.7383	-0.53%	.5032	-0.06%	.5510	-0.07%	.6215	-0.11%
w/o Unified Data	.6768	-1.12%	.6909	-1.31%	.7341	-1.09%	.4975	-1.19%	.5386	-2.32%	.6204	-0.29%

more precise user interest profile can be constructed based on the integrated behavior sequence to improve ranking qualities.

(2) The comparison of our USER model and the separate personalized search or recommendation baselines. **The USER-S and USER-R variants achieve better results than the corresponding baselines on independent scenarios. Greater improvements are observed on the complete USER model in the unified case with both data, with paired t-test at p<0.05 level.** Specifically, on the pure personalized search, USER-S outperforms the HTPS. In recommendation, USER-R promotes NRMS on all evaluation metrics. This proves that our history encoders can effectively learn user interests to improve personalized rankings. Furthermore, the complete USER model promotes HTPS more greatly in the unified scenario. We analyze it may because the USER model is pre-trained by both search and recommendation tasks on the unified dataset, which benefits from more training data.

(3) The comparison of our unified model USER and the general joint framework JSR. **Compared to JSR, USER improves the corresponding separate variants (USER-S and USER-R) much better by training with the unified data.** The HRNN and NRMS combined in JSR show similar performance to the original HRNN and NRMS. However, the USER model achieves 1.14% improvements in MAP over the USER-S and 1.20% in MAP over the USER-R. JSR simply combines a personalized search model and a recommendation model through optimizing a joint loss, without exploiting any interactions between them. In the USER model, we integrate the two kinds of behaviors into a heterogeneous sequence and complete both tasks based on this sequence. The results suggest that USER provides a better approach to aggregate the two tasks and capture the associations between them to promote each other.

To conclude, **with the unified data comprised of the user's search and browsing logs, a more comprehensive user profile and more training samples can be obtained for personalization. Besides, the USER model is promising to capture the relatedness between the two tasks to promote each other.**

6.2 Ablation Study

We conduct ablation studies to analyze major modules.

USER w/o Session Encoder: We discard the short-term history and the session encoder for clarification.

USER w/o History Encoder: In this variant, we remove the long-term history and the history-level transformer.

USER w/o Unified Pre-train: We skip pre-training one unified model with the training data from both tasks, but train two separate models from scratch, with integrated history sequences.

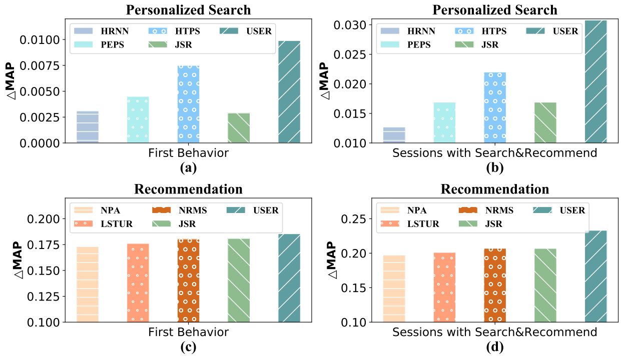


Figure 4: Performance on different data subsets. (a), (c): the first search/recommend behavior of each user; (b), (d): sessions with both search & recommendation.

USER w/o Unified Data: With only separate task data not the unified dataset, USER degrades to USER-S and USER-R respectively.

From the results shown in Table 3, we can observe that:

(1) Removing the session encoder or history encoder and the corresponding behavior history causes a decline in all evaluation metrics for both personalized search and recommendation tasks. This proves that both encoders mine information from the user's history to help personalization. The session encoder captures the user's consistent intention in the current session. The history encoder learns stable user interests in the long-term history. The two parts help clarify the user's current information need together.

(2) There is a decrease in the ranking results when skipping the unified pre-training, especially for the personalized search task. This confirms the benefits of more training samples constructed from both task data in our unified model. It has few impacts on the recommendation task. A possible reason is that the browsing behaviors in recommendation are usually far more frequent than search behaviors, thus the recommendation task has enough training samples. Discarding the unified data leads to a greater decline in both tasks. On a separate task dataset, only one kind of user behavior is available whether in history or training. This decline demonstrates that the integrated behavior sequence is more informative.

6.3 Performance on Specific Set

We further test our model and baselines on different subsets: the first search/recommend behavior of each user, and sessions with search & recommendation. The results are shown in Figure 4, using the improvement of MAP over the original ranking as the metric.

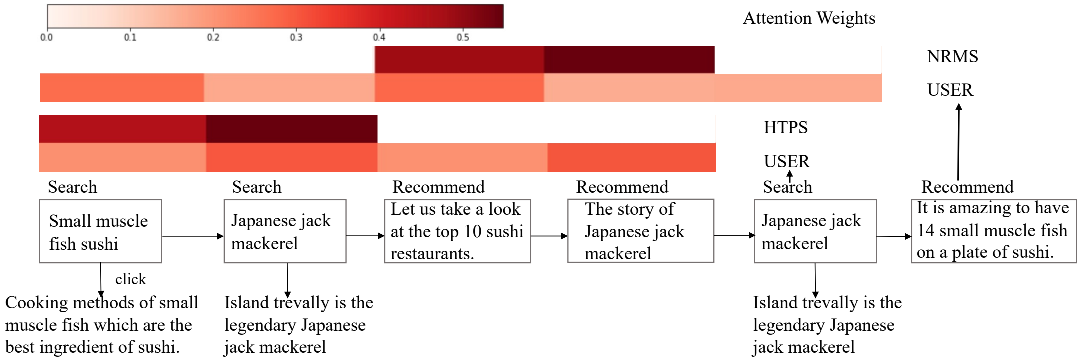

0.0
0.1
0.2
0.3
0.4
0.5
Attention Weights
NRMS
USER
HTPS
USER
Search
Search
Recommend
Recommend
Search
Recommend
Small muscle fish sushi
Japanese jack mackerel
Let us take a look at the top 10 sushi restaurants.
The story of Japanese jack mackerel
Japanese jack mackerel
It is amazing to have 14 small muscle fish on a plate of sushi.
click
Cooking methods of small muscle fish which are the best ingredient of sushi.
Island trevally is the legendary Japanese jack mackerel
Island trevally is the legendary Japanese jack mackerel

Figure 5: Illustration of the heterogeneous behavior sequence in a session and the attention weights of the current action over historical behaviors in different models. A darker area indicates a larger weight.

First Search/Recommend Behavior. We claim that USER is promising to alleviate data sparsity by merging the user's search and recommendation logs. To verify this effectiveness of USER, we sample each user's first search record and recommendation record in the testing data to construct a subset. In this subset, there is little search history for each piece of search data, and little browsing history for each recommendation record. It is a cold-start case for separate personalized search and recommendation tasks.

From Figure 4 (a) and (c), we find that USER trained on the unified dataset outperforms the corresponding baselines with only separate search or recommendation data. In the unified situation, for the user's first search behavior with little search history, the browsing history can be a supplement for mining the user's preferences. As for the first recommendation sample, the search history can also be used as auxiliary information. Thus, we think that combining the two tasks as well as the corresponding behaviors indeed eliminates the problem of user data sparsity and the cold-start challenge.

Sessions with Search & Recommendation. In this paper, we intend to explore the relatedness between the user's search and browsing behaviors to promote the two tasks. Therefore, we sample a subset comprised of sessions with both behaviors. We select several independent baselines and JSR for comparison.

From Figure 4 (b) and (d), we find that USER achieves the best on both tasks. The other joint model JSR that consists of HRNN and NRMS shows similar performance to the separate models. In USER model, we deduce the user's intent based on the integrated behavior sequence. Thus, the potential relatedness between the two kinds of behaviors can be captured to promote personalization, especially for these sessions with both behaviors. However, JSR trains two separate models through a joint loss, which might have difficulty learning the interactions between the two tasks. These results also suggest that USER copes with the unified scenario better than JSR.

6.4 Case Study

In this paper, we focus on the situation with both search and recommendation services in the information content domain. We design a unified model (USER) to jointly handle the two tasks. To illustrate the advantages of our model more intuitively, we conduct a case study to analyze the user's mixed behaviors within a session. Moreover, we discuss the impacts of the user's historical behaviors on the current action in USER, HTPS and NRMS. The impacts are indicated by the attention weights. The results are in Figure 5.

Observing the user's behaviors in the session, we find the user's preferences reflected by the search behaviors and browsing behaviors are consistent, probably about sushi, small muscle fish and Japanese jack mackerel. Besides, there is some relatedness between the two kinds of behaviors. For example, the user browses the article titled "The story of Japanese jack mackerel" in recommendation, followed by a query "Japanese jack mackerel" to seek more relevant information. Thus, integrating the two tasks together has the potential to promote each other. With the aggregated behaviors, we can mine more precise information about the user's interests to help the current ranking. Let us take the last search query "Japanese jack mackerel" as an example. Obviously, this query is strongly relevant to both the historical query "Japanese jack mackerel" and the browsed article "The story of Japanese jack mackerel". USER pays high attention to both the two strongly relevant behaviors. However, HTPS, which is proposed for the independent search case, can only attend to the historical queries, without any information about the browsing actions. With regard to the last recommendation, the historical query "small muscle fish" also reflects relevant user interests, which will be highlighted in USER. **This case study fully proves the value of aggregating the two separate tasks together and our proposal of the unified model**.

7 CONCLUSION

In this paper, we focus on the connections between the personalized search and recommendation in the information content domain, and explore an effective approach to jointly model them together. We integrate the user's search and browsing behaviors into a heterogeneous behavior sequence. Then, we propose the unified model USER. It includes encoders to mine information from the heterogeneous behavior sequence for personalization and a unified task framework to solve both tasks in a unified ranking style. We experiment with a dataset comprised of both behaviors constructed from a real-world commercial platform. The results confirm that our model outperforms the state-of-the-art separate baselines on both tasks. In the future, we will combine the two tasks better.

ACKNOWLEDGEMENTS

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, Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. I also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of RUC.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *Proceedings of ACL 2019*. Association for Computational Linguistics, 336–345.

[2] Nicholas J. Belkin and W. Bruce Croft. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Commun. ACM* 35, 12 (1992), 29–38.

[3] Paul N. Bennett, Filip Radlinski, Ryen W. White, and Emine Yilmaz. 2011. Inferring and using location metadata to personalize web search. In *Proceeding of the 34th International ACM SIGIR 2011*. 135–144.

[4] Paul N. Bennett, Krysta Marie Svore, and Susan T. Dumais. 2010. Classification-enhanced ranking. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*. 111–120.

[5] 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 *ACM SIGIR '12*. 185–194.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent Dirichlet Allocation. In *NIPS 2001*. 601–608.

[7] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *(ICML 2005)*. 89–96.

[8] Chris J. C. Burges, Krysta M. Svore, Qiang Wu, and Jianfeng Gao. 2008. *Ranking, Boosting, and Model Adaptation*. Technical Report MSR-TR-2008-109. 18 pages.

[9] Mark James Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. 2010. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM CIKM 2010*. 1849–1852.

[10] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of DLRS@RecSys 2016*. ACM, 7–10.

[11] Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Sebastian de la Chica, and David Sontag. 2011. Personalizing web search results by reading level. In *Proceedings of the 20th ACM CIKM 2011*. 403–412.

[12] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*.

[13] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In *Proceedings of the CIKM 2018*.

[14] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph Enhanced Representation Learning for News Recommendation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2863–2869.

[15] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *CoRR* abs/1406.2661 (2014).

[16] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of IJCAI 2017*. ijcai.org, 1725–1731.

[17] Morgan Harvey, Fabio Crestani, and Mark James Carman. 2013. Building user profiles from topic models for personalised search. In *22nd ACM CIKM'13*. 2309–2314.

[18] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *Proceedings of ACL 2020*. Association for Computational Linguistics, 4255–4264.

[19] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-Aware Document Representation for News Recommendations. In *RecSys 2020*. ACM, 200–209.

[20] 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 SIGIR 2019*. 555–564.

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

[22] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *Proceedings of SIGKDD 2017*. ACM, 1933–1942.

[23] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*. IEEE Computer Society, 995–1000.

[24] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*. ACM, 285–295.

[25] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. dEFEND: Explainable Fake News Detection. In *Proceedings of KDD 2019*. ACM, 395–405.

[26] Ahu Sieg, Bamshad Mobasher, and Robin D. Burke. 2007. Web search personalization with ontological user profiles. In *Proceedings of the CIKM 2007*.

[27] Yang Song, Hongning Wang, and Xiaodong He. 2014. Adapting deep RankNet for personalized search. In *WSDM 2014*. 83–92.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*.

[29] Maksims Volkovs. 2015. Context Models For Web Search Personalization. *CoRR* abs/1502.00527 (2015).

[30] Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search Personalization with Embeddings. In *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017*.

[31] Thanh Tien Vu, Alistair Willis, Son Ngoc Tran, and Dawei Song. 2015. Temporal Latent Topic User Profiles for Search Personalisation. In *Advances in Information Retrieval - 37th European Conference on IR Research, ECIR 2015*. 605–616.

[32] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *Proceedings of WWW 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 1835–1844.

[33] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *WWW 2019*. ACM, 3307–3313. https://doi.org/10.1145/3308558.3313417

[34] Jian Wang, Yi Zhang, and Tao Chen. 2012. Unified Recommendation and Search in E-Commerce. In *Information Retrieval Technology, 8th Asia Information Retrieval Societies Conference, AIRS 2012, Tianjin, China, December 17-19, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7675)*. Springer, 296–305.

[35] Ryen W. White, Wei Chu, Ahmed Hassan Awadallah, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *22nd International World Wide Web Conference, WWW '13*. 1411–1420.

[36] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *Proceedings of IJCAI 2019*. ijcai.org, 3863–3869.

[37] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural News Recommendation with Personalized Attention. In *Proceedings of KDD 2019*. ACM, 2576–2584.

[38] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Heterogeneous User Behavior. In *Proceedings of EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 4873–4882.

[39] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *Proceedings of EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 6388–6393.

[40] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019. A Comprehensive Survey on Graph Neural Networks. *CoRR* abs/1901.00596 (2019).

[41] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation. In *Proceedings of IJCAI 2020*. ijcai.org, 2519–2525.

[42] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of SIGIR 2017*. 55–64.

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

[44] Jing Yao, Zhicheng Dou, Jun Xu, and Ji-Rong Wen. 2020. RLPer: A Reinforcement Learning Model for Personalized Search. In *WWW '20*. ACM / IW3C2, 2298–2308.

[45] Jiawei Yao, Jiajun Yao, Rui Yang, and Zhenyu Chen. 2012. Product Recommendation Based on Search Keywords. In *Ninth Web Information Systems and Applications Conference, WISA 2012, Haikou, Hainan, China, November 16-18, 2012*. IEEE Computer Society, 67–70.

[46] Hamed Zamani and W. Bruce Croft. 2018. Joint Modeling and Optimization of Search and Recommendation. In *Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems, Bertinoro, Italy, August 28-31, 2018 (CEUR Workshop Proceedings, Vol. 2167)*. CEUR-WS.org, 36–41.

[47] Hamed Zamani and W. Bruce Croft. 2020. Learning a Joint Search and Recommendation Model from User-Item Interactions. In *WSDM '20*. ACM, 717–725.

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

[49] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM '20*. ACM, 789–797.