

Attentive Long Short-Term Preference Modeling for Personalized Product Search

YANGYANG GUO, Shandong University, China

ZHIYONG CHENG, Qilu University of Technology (Shandong Academy of Sciences), China

LIQIANG NIE, Shandong University, China

YINGLONG WANG, Qilu University of Technology (Shandong Academy of Sciences), China

JUN MA, Shandong University, China

MOHAN KANKANHALLI, National University of Singapore, Singapore

E-commerce users may expect different products even for the same query, due to their diverse personal preferences. It is well known that there are two types of preferences: long-term ones and short-term ones. The former refers to users' inherent purchasing bias and evolves slowly. By contrast, the latter reflects users' purchasing inclination in a relatively short period. They both affect users' current purchasing intentions. However, few research efforts have been dedicated to jointly model them for the personalized product search. To this end, we propose a novel Attentive Long Short-Term Preference model, dubbed as ALSTP, for personalized product search. Our model adopts the neural networks approach to learn and integrate the long- and short-term user preferences with the current query for the personalized product search. In particular, two attention networks are designed to distinguish which factors in the short-term as well as long-term user preferences are more relevant to the current query. This unique design enables our model to capture users' current search intentions more accurately. Our work is the first to apply attention mechanisms to integrate both long- and short-term user preferences with the given query for the personalized search. Extensive experiments over four Amazon product datasets show that our model significantly outperforms several state-of-the-art product search methods in terms of different evaluation metrics.

CCS Concepts: • **Information systems** → **Personalization**; *Query representation*; *Learning to rank*;

Additional Key Words and Phrases: Personalized product search, long short-term preference, attention mechanism

This work was finished when Yangyang Guo was a visiting student at the National University of Singapore. The first author claims that this work is under the supervision of Dr. Zhiyong Cheng and Dr. Liqiang Nie.

This work is supported by the National Basic Research Program of China (973 Program), No. 2015CB352502; National Natural Science Foundation of China, No. 61772310, No. 61702300, No. 61672322, and No. 61702302; the Project of Thousand Youth Talents 2016; the Tencent AI Lab Rhino-Bird Joint Research Program, No. JR201805; and the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centre in Singapore Funding Initiative.

Authors' addresses: Y. Guo, L. Nie (corresponding author), and J. Ma, Shandong University, Binhai Road 72, Qingdao, 266200, China; emails: {guoyang.eric, nieliqiang}@gmail.com, majun@sdu.edu.cn; Z. Cheng (corresponding author), Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Artificial Intelligence Institute, Jinan, 250014, China; email: jason.zy.cheng@gmail.com; Y. Wang, Qilu University of Technology (Shandong Academy of Sciences), Keyuan Road 19, Jinan, 250014, China; email: wangyl@sdas.org; M. Kankanhalli, National University of Singapore, 13 Computing Drive, 117417, Singapore; email: mohan@comp.nus.edu.sg.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

1046-8188/2019/01-ART19 \$15.00

<https://doi.org/10.1145/3295822>

ACM Reference format:

Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive Long Short-Term Preference Modeling for Personalized Product Search. *ACM Trans. Inf. Syst.* 37, 2, Article 19 (January 2019), 27 pages.
<https://doi.org/10.1145/3295822>

1 INTRODUCTION

Nowadays, e-commerce has become very popular with the flourishing of the Internet. Its convenient access to an enormous variety of goods enables people to shop almost all products at home. When users on e-commerce websites like Amazon¹ intend to purchase a product, they usually pick their desired ones among millions of items by searching. The standard scenario for the online product search is that a user submits a query, and then the search engine returns a ranked list of products relevant to the given query. Typically, queries submitted by users comprise only a few keywords (e.g., *white T-shirt for men*), which are usually too short or ambiguous to express the users' needs precisely, resulting in unsatisfactory search results. Besides, users' preferences on products could be very diverse (due to different backgrounds, i.e., *age, gender, income*) or strongly affected by the current contexts (e.g., *season, location*). Thereby it is not appropriate to return the same search results to different users for the same query. In the light of this, considering the user's personal intentions under the current contexts and aiming to return relevant products to the given query, the so-called personalized product search, plays a pivotal role in meeting the user's current shopping needs.

The prerequisite for a good personalized product search engine is to accurately model the user preferences and effectively integrate them with the current query. It is well recognized that there are two types of user preferences [3, 63]: long-term ones and short-term ones. The former refers to the user's inherent and relatively stable (evolving slowly) purchasing bias, such as *favorite colors, preferred fashion designs, fitting sizes and consumption level*, which is imperceptibly influenced by the user's personal backgrounds, like *age, upbringing, marriage, education and income*. By contrast, the short-term user preference conveys user's purchasing intention in a relatively short period. It is affected by incidentally transient events, such as *new product release, season change and special personal occasions* like *birthday* [63], which can be inferred from the user's recently purchased products. Compared to the long-term user preference, the short-term one changes more frequently and drastically.

Traditional approaches to product search [17–19, 59] often employ simple matching between the queries and products without harnessing the user's specific attributes. They hardly characterize the user specificity, let alone the heterogeneity. It thus often leads to sub-optimal performance due to the user's diverse expectations. In view of this, personalization becomes quite necessary for the product search. In fact, personalized search has been widely studied in literature over the past few years. Thereinto, the study of personalized web search incorporating the user preference is the most related sub-direction to our work, and it can be roughly divided into two categories: short-term session-based² and long-term profile-based web search. Approaches in the first category [15, 29, 55] capture the short-term user preference from search sessions. Nevertheless, different from the web search whereby a session often contains plenty of queries and rich click results, the session data in the product search³ are usually too sparse to train a good personalized

¹<https://www.amazon.com>.

²A session comprises a set of previous interactions containing past submitted queries and clicked records within a specific time limit [29].

³Pairs of queries and the corresponding purchased products.

search model. Methods in the second category [36, 56] model the long-term user preference based on user's complete browsing behaviors, yet they suffer from two limitations. First, they do not fully exploit user's recent behaviors, containing important contextual information for the current search intention that could reflect the user's recent preference (the short-term preference), e.g., a recently bought *computer* is often followed by accessories like *a mouse* or *a keyboard*. Second, they assume that the long-term user preference is stable yet in fact, it changes over time slightly and slowly [58, 61]. Recently, Ai et al. [1] have presented a personalized product search method, which falls into the second category. They modeled the long-term user preference through a latent space via jointly learning the representations of users, products and queries. This method also bears the aforementioned limitations. As far as we know, literature on the personalized product search is very sparse. To alleviate the problems faced by existing methods, we argue that integrating the relatively stable long-term and time-sensitive short-term user preferences can improve the search accuracy, and thus enhance the user's search experience.

However, an effective integration of the long- and short-term user preferences towards the personalized product search is non-trivial, due to the following facts: (1) Modeling the long- and short-term user preferences is complicated. The former contains multiple manifestations, e.g., *consumption level*, *preferred colors* and *favorite brands*, and gradually evolves with the change of user's background, e.g., *income*. Meanwhile, the short-term one is usually dynamic and time-sensitive, which can be easily affected by transient events. (2) For the user's current need expressed by a query, it is difficult to precisely identify which aspects in the long-term user preference are most relevant. For example, a *T-shirt's* design, instead of its *price*, may impact more on the customers who can afford it. As to the short-term user preference, different products bought recently have different impacts on the user's next purchase. And (3) how to jointly encode the long- and short-term user preferences with the current query is another challenge.

To tackle these problems, we present an Attentive Long Short-Term Preference model (ALSTP for short), to perform the personalized product search by jointly integrating the current given query, the long- and short-term user preferences. Our model comprises three parts: (1) Attentive Short-term Preference Modeling. We establish the short-term user preference based on the products recently bought, rather than the full purchasing history. Subsequently, we leverage an attention mechanism to weight the constituent-wise impact of the short-term user preference given the current query. The short-term user preference is also used for updating the long-term user preference with a small rate. (2) Attentive Long-Term Preference Modeling. The long-term user preference is modeled by using a short-term session-based updating strategy based on the purchased products in chronological order. Similarly, an attention mechanism is used to weight the relevance of each factor in the long-term user preference with respect to the current query. And (3) we fuse the current query, the weighted short- and long-term user preferences to better represent the user's specific intention. Finally, a pairwise learning-to-rank method is used to get the best ranking list.

To verify the effectiveness of our proposed model, we conducted extensive experiments on four public Amazon product datasets. Experimental results demonstrate that our proposed ALSTP model yields better performance as compared to several state-of-the-art methods. It is worth mentioning that our model is applicable to many other scenarios, such as personalized movie search [45] and academic article search [57].

In summary, our main contributions of this article are threefold:

- We present a neural network model for personalized product search by jointly integrating the long- and short-term user preferences, as well as the user's current query.
- We apply the attention mechanism to both the short- and long-term user preferences to weight the importance of different factors in both of them.

- We conducted extensive comparative experiments on four real-world Amazon datasets to thoroughly validate the effectiveness of our ALSTP approach. Moreover, we have released the codes and data to facilitate future research in this direction.⁴

The rest of the article is structured as follows. In Section 2, we briefly review the related work. We detail the scheme of our model and its components in Section 3. Experimental setup and result analysis are presented in Section 4. We finally conclude our work and discuss the future directions in Section 5.

2 RELATED WORK

In this section, we focus on three categories of research directly related to our work: product search, personalized web search and deep learning techniques in information retrieval.

2.1 Product Search

Online product search has become an indispensable part of our daily life, as more and more users search and purchase products on the Internet [31]. There are basically two lines of research working on product search: click prediction and purchase prediction tasks. In view of clicks providing a strong signal of a user's interest in an item, the methods in the first line [16, 21, 43, 44, 68] focus on the click prediction of e-commerce users. For example, Yu et al. [68] proposed a Latent Dirichlet Allocation (LDA) based method for diversified product search. Approaches in [12, 21] analyze the impact of images on user clicks. Recently, a challenge track named "Personalized E-Commerce Search Challenge" was sponsored by CIKM Cup 2016 to provide a unique opportunity for academia and industry researchers to test new ideas for personalized e-commerce search, which requires competitors to correctly predict the user clicks. Several groups [40, 62] reported their proposed ideas and experimental results. However, the user clicks only indicate that the user may be interested in an item [12]. Thus, relying on the click information to predict the user intention for the submitted query is dangerous when users accidentally click on wrong items or they are attracted by some unrelated items due to curiosity. To capture the user's main purpose for the submitted query, in this article, we mainly focus on the purchase prediction.

Typically the e-commerce product inventory information is structured and stored in relational databases. As a result, associating the free-form user queries with these structured data (e.g., product entity specifications and detailed information) can be hard for the search engine system. Some efforts have been dedicated to solving this problem [17–19]. For instance, to fill the big gap between the keyword queries and the structured product entities, Duan et al. [18, 19] proposed a probabilistic retrieval model to optimize the ranking of product entities by mining and analyzing the product search log data. Although the language model is conceptually simple and computationally tractable, the major issue in the product search is that it overlooks the critical information from the entity space and the connection between the queries and product entities. Duan et al. [17] noticed it and learned the query intent representation collectively in both the query space and the structured entity space. With the popularity of the online review platforms in recent years, researchers have attempted to extract information from reviews to represent products [7, 8, 10] by using representation learning techniques (e.g., word2vec [32, 33, 39]). Gysel et al. [59] introduced a latent semantic entity model to learn the distributed representations of words and entities (i.e., products) to solve the semantic mismatching problem in product search.

However, a good product search engine is far beyond the pure relevance between the queries and products. In the field of product search, besides the relevance, the user's personal preference

⁴<https://github.com/guoyang9/ALSTP>.

for products is also critical to persuade users to purchase. Recently, Ai et al. [1] proposed a personalized product search model and took into consideration the users' preference. They highlighted the fact that the purchasing behavior can be highly personal in online shopping and extended the model in Reference [59] by mapping the user into the same latent space with the query and product. Concurrently, there is another work [23] aiming to combine the visual preference and textual preference for product search. Nevertheless, they both ignored the short-term user preference and assumed that the long-term user preference is stable. In this article, we emphasize the importance of integrating the long- and short-term user preferences with the current query in product search.

2.2 Personalized Web Search

Due to the information overload on the web and the diversity of user interests, personalization has been well recognized as an important way to improve the search experience in web search [36]. Instead of giving a complete review on the personalized web search, we mainly recap the related studies falling into the following two categories: short-term session-based and long-term profile-based web search, both being closely related to our work.

Short-term session-based approaches infer user's local preference by tracking user's behaviors in a short period, which is defined as a session. It involves two important problems, namely, *how to determine the duration of a session* and *how to learn the user preference in a session*. To solve these problems, Sriram et al. [55] used temporal closeness and probabilistic similarities between queries to determine the duration of a session; Daoud et al. [14, 15] leveraged a predefined ontology of semantic concepts to construct the short-term user profile; methods in References [58, 61] determine the extent of using context (e.g., prior queries and clicks) and investigate the combination of the query and the context to model the user intention; and Shen et al. [52] captured users' interests by analyzing the immediate contexts and implicit feedback.

Long-term profile-based approaches model the long-term user preference based on the user's overall search logs. Approaches in References [54, 56] apply a probabilistic statistical language modeling technique to discover the relevant context of the current query from the search history. Richardson et al. [49] analyzed the long-term query logs to learn more about the user behavior. Matthijs et al. [36] adopted NLP techniques, such as web page structure parsing, term extraction and part of speech tagging, to extract noun phrases to mine the user profile. Bennett et al. [3] leveraged a simple time decay mechanism to combine the long- and short-term user profiles and pointed out that the long-term behavior provides substantial benefits at the start of a search session while the short-term session behavior contributes the majority of gains in an extended search session.

The decision mechanism that underlies the process of buying a product is different from locating relevant documents or objects [31]. In product search, simply returning something relevant to the user's submitted query may not lead to the purchasing behavior. For example, a returned relevant product which is far beyond a user's consumption capability would not be purchased by this user. Fortunately, such attributes (e.g., consumption capability) of users can be reflected in the user's long-term profile. Meanwhile, short-term behaviors usually provide evidence of user's local purchasing inclinations. Therefore, it is necessary to consider both the long- and short-term user preferences in the product search. Approaches in personalized web search scarcely consider such problems, and thus cannot be directly applied to the product search.

2.3 Deep Learning in Information Retrieval

With the success of deep learning techniques in many domains, such as Computer Vision, Natural Language Processing and Speech Recognition, researchers also applied them to Information Retrieval (IR) and Recommender Systems, and achieved promising performance.

Deep Learning in IR. The deep learning techniques in IR can be roughly classified into two categories: the semantic-oriented and the relevance-oriented. The former mainly concerns the semantic matching between the queries and products. The basic idea under this category is that they first process the text of both the queries and documents, and then build the interaction between them. For example, Huang et al. [28] leveraged a deep neural network (DNN) to project the queries and documents into a common low-dimensional space, and then calculate the relevance of each document with respect to the given query in this space. Shen et al. [53] introduced a convolutional neural network (CNN) to capture the fine-grained contextual structures of entities (queries and documents). Other semantic matching methods like ARC-1 [27], ARC-2 [27], Match-SRNN [60], and MatchPyramid [41, 51] also share the similar idea. Guo et al. [22] pointed out that the aforementioned methods only focus on the semantic matching between the queries and documents. However, ad-hoc retrieval task is mainly about the relevance matching. They proposed a jointly deep architecture to model three key factors of relevance matching. Nevertheless, the proposed model failed to explicitly model the relevance generation process and capture the important IR characteristics. Pang et al. [42] extended the method in Reference [22] to better capture the intrinsic relevance and simulate the human judgment process. In addition, Borisov et al. [4] introduced a neural click model to better understand the user browsing behaviour for the web search.

Deep Learning in Recommendation. However, the aforementioned approaches have not taken personalization into consideration, yet it is vital for the product search. In fact, deep learning has been successfully applied in recommender systems to model the user preference [9, 25, 35]. In particular, the attention mechanism is usually adopted in these systems to model the user's preference more accurately. Attention mechanisms have shown its efficiency in various tasks such as image captioning [6, 66], visual question answering [65, 67], machine translation [2] and information retrieval [5, 34, 47, 64]. A survey on the attention mechanism on those domain is out of the scope of this article. Here we only focus on the attention mechanism on user preference modeling. To name just a few, in the field of recommender systems, Li et al. [34] introduced a neural attentive model to the session-based recommendation. They explored a hybrid encoder with an attention mechanism to model user's sequential behaviors and capture their emphases in the current session. Chen et al. [5] proposed a two-level attention mechanism (i.e., item- and component-level) for the multimedia recommendation. Based on this, a user-based collaborative filtering approach to modeling user preference on both the item-level and component-level is seamlessly applied on implicit feedback. Xiao et al. [64] improved Factorization Machine (FM) by discriminating the importance of different feature interactions.

In this article, we apply deep learning techniques to model the user preferences in the product search. In particular, we use the attention mechanism to model the relation among the query, long- and short-term user preferences to better represent the user's search intention (as we take user's relatively stable purchasing bias and recent purchasing intention into consideration). To the best of our knowledge, we are the first to apply the attention mechanism in both the long- and short-term user preferences to precisely capture the user's search intention. The experimental results demonstrate that the attention mechanism is essential to improve the retrieval performance.

3 OUR PROPOSED MODEL

3.1 Preliminaries

Research Problem. Given a query q of a user u , our goal is to return a ranked list of products based on their matching scores with respect to u 's current search intention. It is difficult for a query q , typically consisting of a few keywords, to precisely describe the user's search intention. Towards this end, we augment the current query by integrating the user's *long-term preference* and

short-term preference. Consequently, the key problem is divided into two subproblems: (1) how to model the long- and short-term user preferences and (2) how to effectively integrate them with the current query to capture the user's search intention.

Model Intuition. Before describing the proposed model, we would like to introduce the underlying intuitions. We start with the definition of two key concepts. User's **long-term preference (LTP)** refers to user's general interests or inclinations on products over a long time period, such as user's favored *clothing style*, *sports*, and *genres of movies*. It is inherently correlated with the user's background (e.g., *gender*, *age*, *career*, or *consumption capability*). Besides, the long-term user preference would change only gradually at a rather slow rate. On the contrary, the **short-term preference (STP)** expresses user's local shopping intention, which is usually situation- or event-dependent, such as *weather* (e.g., *winter* or *summer*) and *recent events* (e.g., *wedding*). For example, users often purchase *T-shirt* and *shorts* together, and buy accessories (e.g., *mouse* or *keyboard*) after purchasing a device (e.g., *computer*). Usually, the recent purchasing behaviors provide some evidence of the current short-term user preference. In summary, LTP reflects user's overall interests and is slowly evolving (due to the changing backgrounds); and in contrast, STP represents user's local purchasing inclinations. Therefore, we model LTP using user's overall transaction history with a slow updating strategy and model STP based on user's previous purchasing behaviors in a recent short time window.

Let us use an example to illustrate why the integration of LTP and STP can help us to infer the user's purchasing intention. Given a "*phone case*" as the query of a user, which is quite ambiguous, since *cases* could be of different *colors*, *styles*, and for different *phones*. From LTP, we could know the user's preferred *price*, *color* and *style*; while from STP we might know the specific model of the case, e.g., *the user recently bought an iPhone 7*. Meanwhile, we should notice that LTP covers the user preference for products from different aspects, which is not only limited to the information related to the current query, such as user preferences over other products. Similarly, STP could also contain information quite different from the current search intention, i.e., not all products purchased recently are directly related to the current query. Therefore, identifying the specific parts in LTP and STP with respect to the current query could capture the user's search intention more accurately and avoid the risk of "query drift" [71]. To achieve the goal, before fusing LTP, STP and the current query, we design two attention mechanisms for LTP and STP, respectively.

Formally, let $\mathcal{P}_u = \{p_1, p_2, \dots, p_N\}$ denote all of the user u 's purchased products list in a chronological order, and $\mathcal{Q}_u = \{q_1, q_2, \dots, q_N\}$ denote the corresponding query list. As queries and the purchased products correspondingly appear in pairs (i.e., $\langle q_i, p_i \rangle$)⁵ and both can indicate user's local search intention or preference, we will use them exchangeably to describe the user preferences. In our model, we infer user u 's STP for a query q_n by referring to his/her recent m queries (i.e., $\{q_{n-m}, \dots, q_{n-1}\}$) and corresponding products (i.e., $\{p_{n-m}, \dots, p_{n-1}\}$) (detailed in Section 3.2.2). For the LTP modeling, the first several purchased products are used to initialize LTP, which is then gradually updated based on STP with a small rate (detailed in Section 3.2.3). The main notations used in this article are summarized in Table 1.

Query and Product Representation. The PV-DM model [30] is adopted for the latent vector representation learning of queries and products. PV-DM is an unsupervised method to learn the continuous distributed vector representations for textual documents. It takes word sequence into consideration and can preserve the semantic features of words. PV-DM takes text documents as inputs and outputs their vector representations in a latent semantic space. In our context, the reviews of products and textual queries are mapped into the same latent space via PV-DM to

⁵Notice that in real scenarios, a query q_i could lead to the purchase of several products (e.g., p_l and p_m). In our model, it can be regarded as two search processes as $\langle q_i, p_l \rangle$ and $\langle q_i, p_m \rangle$.

Table 1. Main Notations Used in This Paper

Q_u	user u 's issued query list in chronological order
\mathcal{P}_u	user u 's purchased product list corresponding to his/her query list
q, p	latent representation of query and product, respectively
\hat{q}, \hat{p}	transformed latent representation of query and product, respectively
g	latent representation of the long-term user preference
h_i, h_{n-1}	the hidden representation of the time step i and the final representation from <i>GRU</i>
c^l, c^g	the attentive short- and long-term user preference representations, respectively
c, c_l	the unified user intent representation and its l th layer projection, respectively
W, b	weight matrix and bias vector, respectively
v	the weight vector for computing the short-term attention score
ϕ	activation function
k	the dimension of both queries and products
m	the short-term window size
β	the long-term user preference updating rate
a_i^l, α_i^l	the short-term attention score before and after normalization, respectively
a_i^g, α_i^g	the long-term attention score before and after normalization, respectively
$\langle q, p \rangle$	query product pair

learn their vector representations,⁶ which are then used as inputs in our model. Without loss of generality, we assume the vector representations of the long-term user preference, products and queries have the same dimension k . Let $p_i \in \mathbb{R}^k$ be the product p_i 's representation and $q_j \in \mathbb{R}^k$ be the query q_j 's representation.

3.2 Model Description

3.2.1 Overall Framework. Figure 1 shows the structure of the proposed ALSTP model. The model comprises three parts: *Attentive Short-Term Preference Modeling (ASTPM)*, *Attentive Long-Term Preference Modeling (ALTPM)*, and *Query Representation Integration (QRI)*.

- **ASTPM.** This part is to extract the short-term user preference based on m immediately preceding queries and their corresponding purchased products, and learn the relations between the current query and those queries by an attention mechanism.
- **ALTPM.** This part models the long-term user preference via a gradually updating strategy based on the historical purchased products in chronological order, and it uses an attention mechanism to associate the current query with the long-term user preference.
- **QRI.** With the attentive modeling of the short- and long-term user preferences, the last part integrates them with the current query to represent the user's current search intention.

In the following, we detail the three parts sequentially.

3.2.2 Attentive Short-Term Preference Modeling. For each query q and product p , we map them into the same latent space through a fully connected layer as

⁶The inputs to PV-DM model are each products' reviews and textual queries (To keep the semantic relationship between queries and reviews). After the unsupervised training of PV-DM, each of these inputs has its own output, e.g., equal-length vector, we take this vector as the representation of each product and query. Afterwards, when a new query comes, we can easily obtain its vector representation by inputting this textual query to the trained PV-DM model. And then a ranked product list can be returned to this query.

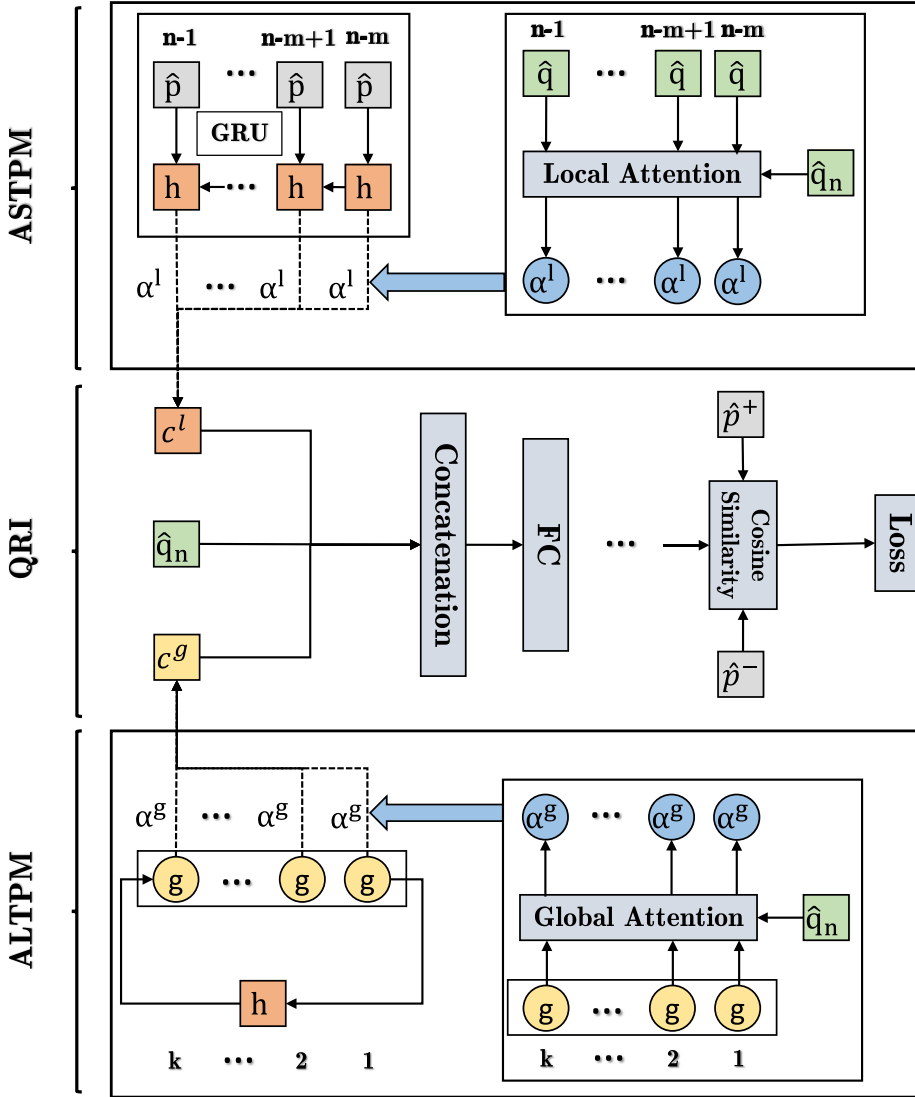


Fig. 1. The architecture of our proposed ALSTP framework. Symbol \hat{p} denotes a product's vector representation after the transformation, and \hat{q} is a query's vector representation. The top part is the Attentive Short-Term Preference Modeling (ASTPM), which is useful to obtain the attentive short-term user preference c^l . The bottom part Attentive Long-Term Preference Modeling (ALTPM) is to learn the attentive long-term user preference c^g . Finally, the middle part Query Representation Integration (QRI) fuses these two with the current query to better represent user's shopping intention and leverages a comparative learning method to obtain the final loss.

$$\hat{p} = \phi(W_c p + b_c), \quad (1)$$

$$\hat{q} = \phi(W_c q + b_c), \quad (2)$$

where q and p are vector representations of the query q and product p obtained from PV-DM (as described in the last subsection). $W_c \in \mathbb{R}^{k \times k}$ and $b_c \in \mathbb{R}^k$ are the weight matrix and bias vector,

respectively. $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ are the product and query representations after transformation, respectively. $\phi(\bullet)$ is the activation function. In our implementation, *ELU* [13] is adopted.⁷

Given the current query q_n , considering that the recently purchased products may relate to the current search intention, several recent queries and the products purchased accordingly are used to capture the short-term user preference. Specifically, let m be the time window size.⁸ To model the influence imposed by the previous queries (or products) on the next purchase, we adopt a Recurrent Neural Network (RNN) model equipped with Gated Recurrent Units (GRU) [11] to model the short-term user preference, as it has been successfully applied in the session-based product recommendation [26, 34].

GRU was proposed in Reference [11]. The activation of GRU is a linear interpolation between the previous activation \mathbf{h}_{t-1} and the candidate activation $\hat{\mathbf{h}}_t$,

$$\mathbf{h}_t = (1 - z_t)\mathbf{h}_{t-1} + z_t\hat{\mathbf{h}}_t, \quad (3)$$

where the update gate z_t is given by

$$z_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}). \quad (4)$$

The candidate activation function $\hat{\mathbf{h}}_t$ is computed as

$$\hat{\mathbf{h}}_t = \tanh[\mathbf{W}_x \mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})], \quad (5)$$

where the rest gate \mathbf{r}_t is given by

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}). \quad (6)$$

The RNN module takes the recent m product representations $[\hat{\mathbf{p}}_{n-m}, \dots, \hat{\mathbf{p}}_{n-1}]$ as inputs. We thus can obtain a set of high dimensional hidden representations $[\mathbf{h}_{n-m}, \dots, \mathbf{h}_{n-1}]$, denoting the short-term user preference. The hidden state of RNN is initialized by the long-term user preference \mathbf{g} , which is introduced in the next section.

Attention Mechanism. Notice that the previous queries (or purchased products) are not all closely relevant to the current query (targeted products). For example, previous queries about *mouse* and *keyboard* are more relevant than the *phone* with the current query of *display screen*. Considering that these previous queries have different relevance levels to the current query, we propose to use an attention mechanism to capture their relevance scores or attentions based on the representations of queries. Let $\hat{\mathbf{q}}_n$ be the representation of the current query and $[\hat{\mathbf{q}}_{n-m}, \dots, \hat{\mathbf{q}}_{n-1}]$ denote the representations of the previous query list corresponding to the previous product list of $[\hat{\mathbf{p}}_{n-m}, \dots, \hat{\mathbf{p}}_{n-1}]$. As shown in Figure 2, we apply a two-layer neural network to estimate the attentions $[a_{n-m}^l, \dots, a_{n-1}^l]$ based on $\hat{\mathbf{q}}_n$ and $[\hat{\mathbf{q}}_{n-m}, \dots, \hat{\mathbf{q}}_{n-1}]$,

$$a_j^l = \mathbf{v}^T \phi(\mathbf{W}_0^l \mathbf{q}_n + \mathbf{W}_1^l \mathbf{q}_j + \mathbf{b}^l), \quad (7)$$

where $\mathbf{W}_0^l \in \mathbb{R}^{f \times k}$ and $\mathbf{W}_1^l \in \mathbb{R}^{f \times k}$ (f is a scalar hyper-parameter) are matrix parameters of the first layer, $\mathbf{b}^l \in \mathbb{R}^k$ is the bias, $\mathbf{v} \in \mathbb{R}^{f \times 1}$ is the vector parameter of the second layer, and ϕ is the activation function *ELU*. Subsequently, the attention weights are further normalized by applying a *softmax* function:

$$\alpha_j^l = \frac{\exp(a_j^l)}{\sum_{i=n-m}^{n-1} \exp(a_i^l)}. \quad (8)$$

⁷The activation functions could be *sigmoid*, hyperbolic tangent (*tanh*), rectified linear unit (*ReLU*), leaky *ReLU* or exponential linear unit (*ELU*). We tested all those activation function in experiments. And the *ELU* function achieved better performance.

⁸As users' purchasing transactions are usually too sparse to learn, for example, some users have no purchase transactions in a specific time period (e.g., a month). Thus, we choose to use a short-term time window (e.g., the recently purchased four products) to learn the short-term user preference.

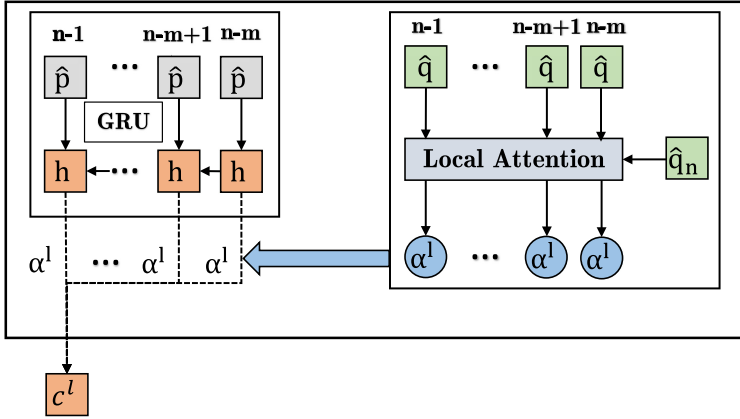


Fig. 2. Illustration of attentive short-term preference modeling.

Accordingly, the previously purchased products of these queries contribute differently to the final short-term user preference modeling. The role of α^l is to determine which products bought before should be emphasized or ignored to capture the attentive short-term user preference. In this way, the attentive short-term user preference is modeled as a linear combination of the short-term user preference weighted by the corresponding attentions obtained by Equation (8),

$$c^l = \sum_{i=n-m}^{n-1} \alpha_i^l h_i, \quad (9)$$

where c^l is the final representation of the attentive short-term user preference. It is expected that c^l captures the information in the short-term user preference related to the current query. The RNN-based module can update the short-term user preference immediately and effectively.

3.2.3 Attentive Long-Term Preference Modeling. By contrast, the long-term user preference is relatively stable and updates slowly. Although it is modeled based on the overall purchasing history of the user, in our model, we initialize it by a set of products purchased at early times and then update it by the products bought subsequently. Let $\mathbf{g} \in \mathbb{R}^k$ denote the long-term user preference, which corresponds to latent factors representing user's inherent purchasing bias (e.g., favorite colors, preferred fashion design in *Clothing* dataset). For simplicity, we use the first m products to set it up, and then update it with every m products purchased (evolving gradually). In this setting, \mathbf{g} is updated based on a session of m products, which is also the window size of modeling the short-term user preference. Therefore, \mathbf{g} is updated via

$$\mathbf{g} = (1 - \beta)\mathbf{g} + \beta\mathbf{h}'_{n-1}, \quad (10)$$

where \mathbf{h}'_{n-1} is the final hidden representation of the short-term user preference based on the previous m products and β is a hyper-parameter denoting the updating rate.

Attention Mechanism. We argue that users can perform distinctively towards different queries based on their specific long-term preference. Therefore, the attention mechanism for long-term user preference is to find out which factors in \mathbf{g} are more important for the current query. For example, if a user can afford most of the laptops on the e-commerce website, and then a machine's performance or appearance, instead of its price, will affect the user's final purchase decision more. Because of that, a good product search engine should learn how to emphasize the influential factors in the long-term user preference and depress the inessential ones. Thus, in this article, similar

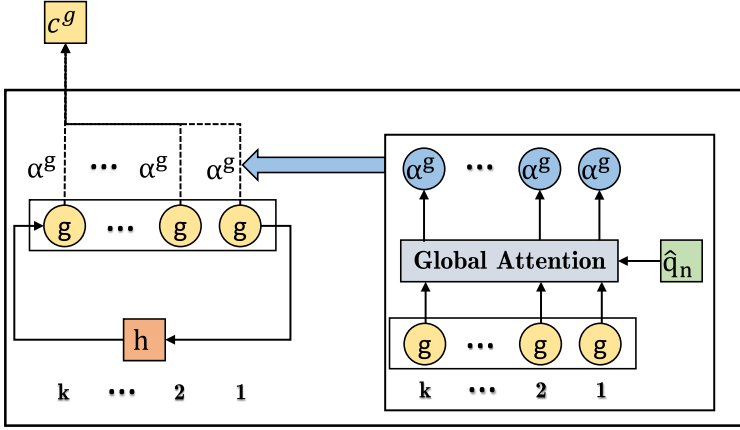


Fig. 3. Illustration of attentive long-term preference modeling.

to the attentive short-term preference modeling, we also use an attention mechanism to measure the relevance between different aspects in the long-term user preference and the current query,

$$a_j^g = g_j \mathbf{w}^g \hat{\mathbf{q}}_n + b^g, \quad (11)$$

where g_j is the j th element of the long-term user preference, $\mathbf{w}^g \in \mathbb{R}^{1 \times k}$ is the weight parameter, and b^g is the bias. Figure 3 illustrates the main idea of the computation of these attention weights. The final attention $\boldsymbol{\alpha}^g = [\alpha_1^g, \dots, \alpha_k^g]$ is obtained by normalization via a *softmax* function,

$$\alpha_j^g = \frac{\exp(a_j^g)}{\sum_{i=1}^k \exp(a_i^g)}. \quad (12)$$

Finally, we compute the element-wise product of the long-term user preference and the attention weights by,

$$\mathbf{c}^g = \mathbf{g} \odot \boldsymbol{\alpha}^g, \quad (13)$$

where \mathbf{c}^g is the representation of the attentive long-term user preference. It is expected to capture the information in the long-term user preference related to the current query.

3.2.4 Query Representation Integration. With respect to the query q , \mathbf{c}_g has summarized the current attentions on the long-term user preference; and \mathbf{c}_l adaptively captures the relevant purchasing intentions in the short-term user preference. Besides, the query is the most direct representation of the user's current need. The basic idea of this work is to learn a product search model that considers both the short- and long-term user preferences to identify the current purchasing intention. Based on this idea, as shown in Figure 4, we fuse the above three parts into a unified representation \mathbf{c} by concatenation,

$$\mathbf{c} = [\mathbf{q}_n; \mathbf{c}^l; \mathbf{c}^g]. \quad (14)$$

We then map this unified representation into the same latent space with the products for direct matching. To model the interactions between these three parts and obtain better fusion features, we refer to Deep Neural Networks (DNNs), which introduces multi-layer of non-linear interactions and has been proven to be very effective in the feature fusion tasks [70]. Specifically, the fully

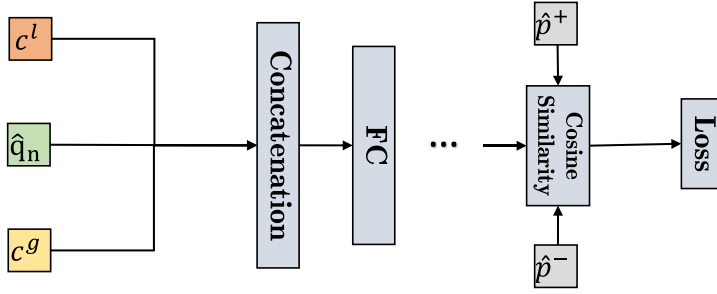


Fig. 4. Illustration of query representation integration.

connected layers are

$$\begin{aligned}
 c_1 &= \phi(W_1 c + b_1), \\
 c_2 &= \phi(W_2 c_1 + b_2), \\
 &\dots, \\
 c_L &= \phi(W_L c_{L-1} + b_L),
 \end{aligned} \tag{15}$$

where W_l and b_l denote the weight matrix and bias vector for the l th fully connected layer, respectively. $\phi(\cdot)$ is the activation function *ELU*. Besides, our network structure follows a tower pattern, where the bottom layer is the widest and each successive layer has a smaller number of neurons [25]. Ultimately, the output from the last layer c_L has the dimension of k , equal to the query latent representation size.

With the current upgraded query representation c_L and product representation \hat{p} , the relevance score between the query and product p_j is computed as

$$s_j = \varphi(c_L, \hat{p}_j), \tag{16}$$

where φ is a distance function that can be cosine similarity, dot product, Euclidean or Manhattan distances. In our experiments, we observed that cosine similarity yields relatively better performance. After that, all products are ranked in a descending order based on their relevance scores with respect to the current query. The top ranked products are returned to users.

3.3 Optimization

We use a pairwise learning method to train the model. All the observed query-product pairs are treated as positive pairs. For each positive pair (q, p^+) , we randomly sample N_s negative products p^- . A query with a positive product and a negative product constructs a triplet (q, p^+, p^-) for training. The BPR loss is adopted, which is a pairwise method and has been widely used in the recommender systems towards personalized ranking [48].

$$Loss(c_L; \hat{p}^+, \hat{p}^-) = -\log(\sigma(s_p^+ - s_p^-)) + \lambda(\|\Theta\|^2), \tag{17}$$

where \hat{p}^+ and \hat{p}^- denote the representations of the positive and negative samples, respectively. And s_p^+ and s_p^- are the corresponding positive and negative sample scores. λ is the ℓ_2 regularization hyper-parameter, σ is the sigmoid function, and Θ denotes all the parameters in our model. To optimize the objective function, we employ the stochastic gradient descent (SGD), a universal solver for optimizing the neural network models. The steps for training the network are summarized in Algorithm 1.

ALGORITHM 1: Attentive Long- and Short-term Preference Modeling.

Input: (Query, Product) latent representation pairs in chronological order.
Output: Latent representation of the long-term user preference and all the learned parameters.

```

1 Initialize all parameters  $\Theta$  with xavier [20] distribution;
2 for each user  $u$  do
3   Initialize long-term user preference  $\mathbf{g}$  with zeros;
4   for each  $m$  (query, product) pairs do
5     Transform them into the same latent space according to Equation (1) and Equation (2);
6     Compute the product latent representations  $[\hat{\mathbf{p}}_{n-m}, \dots, \hat{\mathbf{p}}_{n-1}]$ ;
7     Compute local attention weights  $\alpha^l$  according to Equation (7) and Equation (8);
8     Compute attentive short-term preference  $\mathbf{c}^l$  according to Equation (9);
9     Compute global attention weights  $\alpha^g$  according to Equation (11) and Equation (12);
10    Compute attentive long-term preference  $\mathbf{c}^g$  according to Equation (13);
11    Compute final query representation  $\mathbf{c}_L$  according to Equation (15);
12    for each sampled negative product do
13      Compute the final loss according to Equation (17);
14      update  $\Theta$ ;
15    end
16    if  $m$  products purchased then
17      Update  $\mathbf{g}$  according to Equation (10);
18    end
19  end
20 end

```

4 EXPERIMENTS

We conducted extensive experiments on several datasets to thoroughly justify the effectiveness of our model. In particular, our experiments mainly answer the following research questions:

- **RQ1:** Can our model outperform the state-of-the-art product search methods? (Section 4.2)
- **RQ2:** How does each component in our model affect the final performance, including the short-term user preference and long-term one? (Section 4.3)
- **RQ3:** Are the attention mechanisms and long-term user preference updating strategy helpful to the final product search results? (Section 4.3)
- **RQ4:** How do the important parameters influence the performance of our model, including the window size of the short-term user preference modeling and the embedding size of the user preferences and product representations? (Section 4.4)
- **RQ5:** Can our attentive short-term preference module capture the impact of the previous transactions on the current purchasing intention? (Section 4.5)

In the following, we first described the experimental setup. Then the performance comparisons of the proposed model ALSTP with several state-of-the-art product search methods are reported. Thereafter, we analyzed the utility of each model component by the performance comparisons among different variants of our ALSTP model, followed by the analysis of the effectiveness of the attention mechanism and the long-term user preference updating strategy. We then studied the influence of the two important parameters involved in our model and finally illustrated the effectiveness of the attention mechanism with some case studies.

Table 2. Statistics over the Evaluation Datasets

Datasets	#Users	#Products	#Feedback	#Queries	Avg. Query Length	Avg. Review Length
Phones	3,216	9,018	47,139	107	6.61	135.39
Clothing	5,200	20,424	72,142	1,325	7.64	35.88
Toys	4,188	11,526	74,423	371	7.05	103.09
Electronics	45,225	61,918	773,502	839	6.89	160.83

4.1 Experimental Setup

4.1.1 Dataset. We experimented on the public Amazon product dataset.⁹ The dataset contains product reviews and metadata from Amazon including hundreds of millions of reviews spanning from May, 1996 to July, 2014. It is organized into 24 product categories. In our experiments, we adopted the five-core version provided by McAuley et al. [37], whereby the remaining users and products have at least five reviews, respectively. Besides, we selected four categories with different sizes: *Phones*, *Toys*, *Clothing*, and *Electronics*. Following the strategy in References [24, 38], we extracted the users' product purchasing behaviors based on their reviews, i.e., the products they reviewed are the ones they purchased. Our model uses the previously purchased products in a neighboring window size to model the short-term user preference. To study the influence of the window size on our model, we further filtered the dataset to make sure each user has at least 10 purchased products (i.e., 10 reviews). And the words with low frequency were removed in our experiment. The basic statistics of these processed subsets are shown in Table 2. Note that the number of feedback equals to that of the query-product pairs.

Query Extraction. As Rowley described in Reference [50], a typical scenario of user searching a product is to use *a producer's name, a brand or a set of terms that describe the category of the product* as the query in retrieval. Based on this observation and following the strategy of References [1, 59], for each product a user purchased, we extracted the corresponding search query from the categories to which the product belongs. The extraction of the textual representation for queries of the topics based on the categories is detailed as follows. We first extracted the category information for each product from its metadata. And then we concatenated the terms from a single hierarchy of categories to form a string of topics. Finally, we removed the punctuation, stop words and duplicate words from this topic string. To eliminate the duplicate words, we maintained the terms from sub-categories in view of these terms carrying more specific information compared to their parent-categories. For example, in the dataset of *Phones*, the extracted query for *Cell Phones & Accessories* → *Accessories* → *Batteries* → *Internal Batteries* would be "cell phones accessories internal batteries." Finally, for the convenience of modeling the sequential pattern, we randomly selected one as the final query for the products with multiple queries.

4.1.2 Baseline Methods. We compared the proposed ALSTP model with a logistic regression based method and different retrieval approaches from two categories: (1) traditional methods based on bag-of-words representations, such as *Query Likelihood Model* [69] and *Extended Query Likelihood with User Models* [1] and (2) representation learning approaches based on latent space modeling, such as *Latent Semantic Entity* [59] and *Hierarchical Embedding Model (HEM)* [1]. It is worth noting that the recently proposed HEM is the state-of-the-art method for personalized product search. We introduced these models in the following. To ensure a fair comparison, we carefully tuned these models and reported the best performance.

⁹<http://jmcauley.ucsd.edu/data/amazon/>.

Table 3. Performance Comparisons between ALSTP and Baselines over Four Amazon Datasets

Dataset	Metric	LRS	QL	UQL	LSE	HEM	ALSTP
Phones	HR	0.050	0.058	0.058	0.094	0.176	0.194*
	MRR	0.011	0.015	0.015	0.025	0.056	0.065[†]
	NDCG	0.022	0.024	0.024	0.040	0.083	0.096[†]
Clothing	HR	0.076	0.079	0.082	0.040	0.076	0.109[†]
	MRR	0.015	0.019	0.019	0.012	0.014	0.037
	NDCG	0.027	0.032	0.033	0.023	0.027	0.052[†]
Toys	HR	0.150	0.105	0.112	0.065	0.188	0.202[†]
	MRR	0.042	0.030	0.031	0.013	0.050	0.061[†]
	NDCG	0.076	0.047	0.049	0.024	0.084	0.094[†]
Electronics	HR	0.069	0.080	0.080	0.110	0.179	0.198[†]
	MRR	0.016	0.021	0.021	0.023	0.055	0.064*
	NDCG	0.028	0.034	0.034	0.042	0.083	0.091[†]

Symbols * and [†] denote the statistical significance with two-sided t-test of $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline. The best performance is highlighted in bold.

Logistic Regression Search Model (LRS). This method is the variant of the winner of CIKM Cup 2016 competition. The original method is designed for click prediction tasks, we adapted it to the same setting with our proposed method as well as other baselines. Due to the lack of abundant auxiliary information, we adopted the second best model in the original article [62], of which the final performance is 0.4175 of NDCG (The best model is 0.4238) on the competition dataset.¹⁰ The query representations obtained via the PV-DM model are used to represent the query feature, and the product search task is converted to a classification problem. Impressive performance was observed (Details can be found in Table 3).

Query Likelihood Model (QL). This method is a language modeling approach. It first estimates a language model for each document, and then ranks the documents by the likelihood of generating the query according to the estimated model [69]. Formally, given a query Q , the retrieval score of a document D is defined as

$$P_{QL}(Q|D) = \sum_{w \in Q} \log \frac{tf_{w,D} + \mu P(w|C)}{|D| + \mu}, \quad (18)$$

where $tf_{w,D}$ is the frequency of the word w in D , $|D|$ is the length of D , μ is a Dirichlet smoothing parameter, and $P(w|C)$ is a collection language model that can be computed as the frequency of w in the corpus C divided by the size of C . The optimal value of μ varies from collection to collection. In our experiment, we tested different values from 2,000 to 10,000 with a step size of 4,000.

Extended Query Likelihood with User Models (UQL). This model is first introduced to the personalized product search by Ai et al. [1]. Specifically, let U be the set of the most frequent words¹¹ of reviews submitted by the user u , and then the likelihood model of the user-query pair (U, Q) for document D is

$$P_{UQL} = \lambda P_{QL}(Q|D) + (1 - \lambda) P_{QL}(U|D), \quad (19)$$

¹⁰<https://competitions.codalab.org/competitions/11161>.

¹¹Here, we define the words appearing more than 50 times as the most frequent words.

where λ is a coefficient parameter controlling the weights of U in the search. We searched λ in $[0, 1]$ with a step size of 0.2. When $\lambda = 1$, it becomes the same as the QL method.

Latent Semantic Entity (LSE). This method is specially designed for product search [59]. It maps words and products into the same latent space and learns a mapping function f_{LSE} between them by

$$f_{LSE}(s) = \tanh\left(W\left(\mathbf{w}_v \frac{1}{|s|} \sum_{w_i \in s} \delta_i\right) + \mathbf{b}\right), \quad (20)$$

where s is a n -gram string extracted from the review of an item or a user-issued query, w_i is the i th constituent word in s , and f_{LSE} is the learned representation of s in the latent space. The objective is to directly maximize the similarity between the vector representation \mathbf{e}_i of the product i and its corresponding projected n -gram s_i 's latent representation $f_{LSE}(s_i)$. For simplicity, we set the word embedding and product embedding to the same size; and for the n -gram window size n , we tuned it exponentially $\{2^i \mid 2 \leq i \leq 4\}$.

Hierarchical Embedding Model (HEM). This model (HEM) proposed in Reference [1] is the state-of-the-art approach for the personalized product search. It extends LSE [59] by adding the element of user preference to the product search. Similar to UQL, HEM also uses a coefficient to control the weight between the query model \mathbf{q} and the user model \mathbf{u} by

$$\mathbf{M}_{uq} = \lambda \mathbf{q} + (1 - \lambda) \mathbf{u}. \quad (21)$$

HEM learns the distributed representations of queries, users and products by maximizing the likelihood of observed user–query–product triplets. We tuned the query model weight λ from 0.0 to 1.0 with a step size of 0.2.

4.1.3 Evaluation Metrics. We applied three standard metrics to measure the performance of our model and the baselines from distinct perspectives: Hit Ratio (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

- **HR** measures whether the top 20 results contain the correct items. In our setting, it indicates the percentage of queries to which the method hits a correct item in the list.
- **MRR** is a popular metric in the information retrieval field. It is the average of reciprocal ranks of the desired products.
- **NDCG** is widely used for measuring the rank accuracy, as it takes into account the position of positive items in the rank list by assigning a higher score to the item at a higher position.

Without special specification, we truncated the rank list at 20 for all the three metrics.

4.1.4 Experimental Settings. Dataset Split. We partitioned each of the four datasets into three sets: training, validation and testing sets. We first extracted the user-product pairs from users' reviews, and then extracted the queries for these products. We finally got the user–query–product triplets. For each dataset, the last purchasing transaction of each user is held for the testing set, the second last for the validation set, and the rest for the training set. Moreover, we hid the reviews of the validation and testing sets in the training phase to simulate the real-world scenarios. For the ALSTP model, we trained it on the training set, tuned the parameters on the validation set, and reported the final results on the testing set based on the optimal parameter settings.

Parameter Settings. In the training procedure of ALSTP, the parameters are initialized by the *xavier* method [20] and then optimized with the standard Stochastic Gradient Descent (SGD) with the momentum value 0.9. The layers of GRU is fixed to 1, the learning rate is tuned in the range of $[0.00001, 0.0005, 0.0001, 0.001, 0.01]$, and the regularizer is $[0.000001, 0.00001, 0.0001, 0.001]$. The number of negative samples (i.e., N_s) for each positive training data is set to 5 for LSE, HEM, and

our model. Besides, to avoid the unstable gradient update, we clipped the global norm of parameter gradients with 5 [46].

4.2 Performance Comparison (RQ1)

Table 3 summarizes the comparison results between our model and all the baselines over the four datasets regarding all the three metrics. We also conducted pairwise significance test (i.e., t-test) between our model and the baseline with the best performance. The main observations from Table 3 are as follows:

- Across all the four datasets, our method can outperform all the competitors significantly. This reveals the integration of the current query with the attentive short and long-term user preferences can better express the user's shopping need. Compared to HEM, our method can achieve a larger improvement on the *Clothing* dataset (the absolute improvement of NDCG over *Clothing* is 0.025 (93% relative improvement) while on *Phone* is 0.013 (16%), *Toys* is 0.010 (12%), and *Electronics* is 0.008 (10%)), which is mainly because of the more frequent behaviors of purchasing clothes. Besides, *fashion* and *seasonal changes* exert more influence on user's local choices on clothes, revealing the importance of the short-term user preference.
- For both the traditional bag-of-words (i.e., QL and UQL) and state-of-the-art representation learning (LSE and HEM) methods, personalized product search consistently outperforms the non-personalized ones. This indicates that the user's shopping intention regarding the same query can be much diverse as well as the importance of personalization in the product search.
- For the logistic regression based method LRS, on the *Toys* dataset, it surpasses the QL and UQL with a large margin, and achieved better performance than LSE and HEM on the *Clothing* dataset. This is mainly because the number of products per query on these two datasets is much shorter than the other two datasets. So that the concise logistic regression method can generalize well on these two datasets.
- On the *Clothing* dataset, the performance of both bag-of-words methods (QL and UQL) exceed the two representation learning methods (LSE and HEM), which is mainly because the average review length is much smaller than that of the other three datasets (*Clothing* is 36, while on the other three are 135, 103, 161, respectively). This indicates that less review words make the user preference modeling more difficult for the representation learning methods. Our model not only leverages the review word embedding to learn the product representation, but also considers the recently purchased products to have more insights the user's purchasing intention. Therefore, our ALSTP model can consistently outperform the QL and UQL.

4.3 Model Ablation (RQ2 and RQ3)

To study the utility of each component in our model, we decomposed our proposed ALSTP model with a set of variants, including the following:

- **Without Preference Modeling (WoPM):** It prunes the user preferences (including the long-term and short-term ones) from ALSTP;
- **Short-Term Preference Modeling (STPM):** It is a variant of ALSTP without considering the long-term user preference or the attention mechanism. In addition, it leverages the recently bought products to denote the short-term user preference;

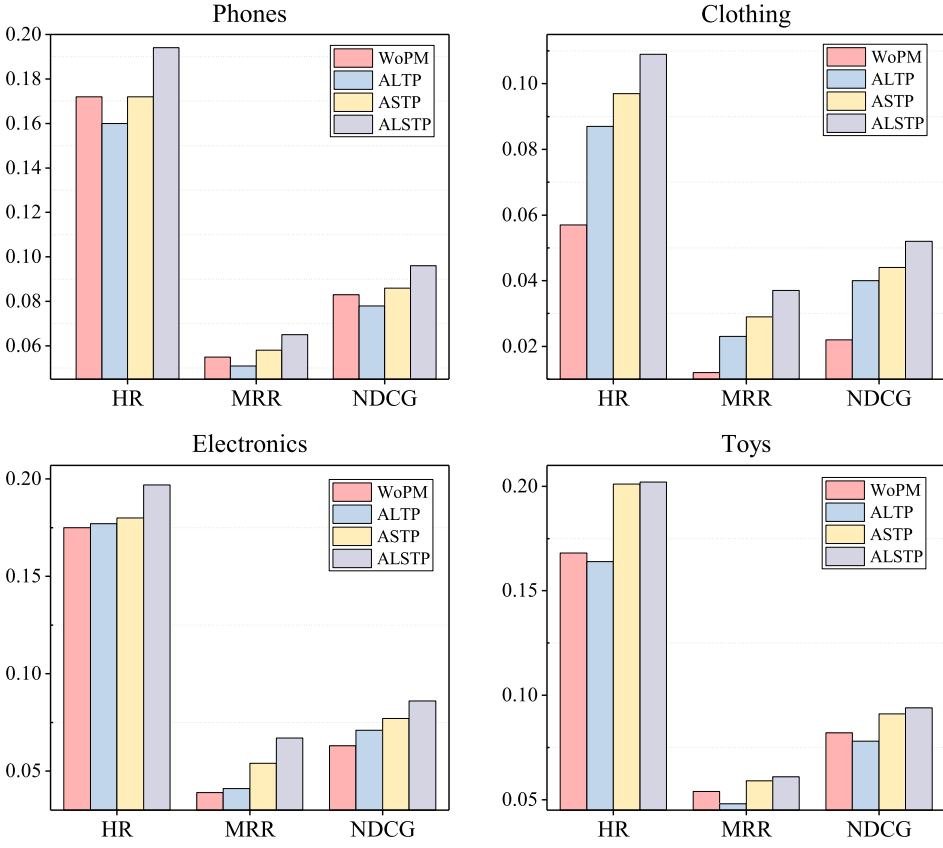


Fig. 5. Comparison of variants of our model.

- **Attentive Short-Term Preference Modeling (ASTP):** It is a variant of ALSTP without considering the long-term user preference but the attention mechanism for the short-term preference;
- **Long-Term Preference Modeling (LTPM):** It is a variant of ALSTP without considering the short-term user preference or the attention mechanism. It leverages the user's whole transaction history to learn the long-term user preference;
- **Attentive Long-Term Preference Modeling (ALTP):** It is a variant of ALSTP without considering the short-term user preference but the attention mechanism for the long-term user preference;
- **Long Short-Term Preference Modeling (LSTP):** It prunes the attention mechanism from our proposed ALSTP model.

Comparison Among Preferences. We compared the ALSTP with three variants: WoPM, ASTP, ALTP, representing product search without preferences, with short-term user preference, and with long-term user preference, respectively. The performance of these three variants and our final model is shown in Figure 5. The main observations are threefold:

- With the consideration of the user's shopping interest in a short term, i.e., ASTP, we observed it has better performance than WoPM and ALTP on the four datasets. It reveals the

Table 4. Influence of Attention Mechanism

Dataset	Metric	ASTPM		ALTPM		ALSTP	
		STPM	ASTP	LTPM	ALTP	LSTP	ALSTP
Phones	HR	0.173	0.173*	0.158	0.160*	0.183	0.194*
	MRR	0.056	0.058	0.048	0.051 [†]	0.061	0.065 [†]
	NDCG	0.085	0.086*	0.075	0.073 [†]	0.090	0.096 [†]
Clothing	HR	0.092	0.097*	0.083	0.087 [†]	0.100	0.109*
	MRR	0.029	0.027*	0.021	0.023 [†]	0.033	0.037 [†]
	NDCG	0.042	0.044*	0.034	0.040	0.047	0.052
Toys	HR	0.192	0.201*	0.159	0.164 [†]	0.200	0.202 [†]
	MRR	0.059	0.061*	0.045	0.048*	0.059	0.061 [†]
	NDCG	0.091	0.094	0.072	0.078 [†]	0.092	0.094
Electronics	HR	0.173	0.180 [†]	0.174	0.178 [†]	0.190	0.198 [†]
	MRR	0.054	0.056 [†]	0.042	0.045*	0.052	0.064 [†]
	NDCG	0.080	0.083 [†]	0.068	0.070*	0.079	0.091 [†]

Symbols * and [†] denote the statistical significance with two-sided t-test of $p < 0.05$ and $p < 0.01$, respectively, compared to the methods without attention mechanism.

influence of user's previous purchases on the next purchasing intention, and the importance of considering user's short-term preference in the product search.

- On datasets of *Phones* and *Toys*, ALTP (with the consideration of the long-term preferences) performs worse than WoPM. This is mainly because the incidently transient events (which can be modeled through the short-term user preference) affect user's decisions more and the long-term user preference has not updated promptly. Therefore, the long-term user preference would disturb the current query and may cause the drift of the query intention.
- For WoPM, though it simply matches the learned representations of queries and products without personalization, it can still perform well comparing to the baselines in Table 3. This may be credited to the power of the representation learning of our neural-net framework.

Utility of Attention Mechanism. We analyzed the utility of the attention mechanism for ASTPM, ALTPM and ALSTP, corresponding to the short-term user preference modeling, long-term preference user modeling and our final model, respectively. As we can see from Table 4, the models with attention perform better than those without attention in most cases. As we mentioned, the long-term preference contains diverse manifestations. Without the specific emphasis on relevant aspects in the long-term user preference, it may also lead to the “query drift” problem. For the short-term user preference, the previously bought products do not contribute equally to the next purchase. Hence, it is also important to treat them differently regarding the current query. For the proposed ALSTP model, the attention mechanism is necessary. It helps us capture not only the previous important local factors in the short-term user preference, but also the triggering aspects in the long-term user preference. For example, on the *Electronics* dataset, the relative improvement of ALSTP over LSTP is 23% for MRR, ASTP over STPM is 4%, and ALTP over LTPM is 7%.

Utility of Long-term User Preference Updating. We studied the utility of the long-term user preference updating on the four datasets with respect to the three metrics by tuning the updating rate β ¹² and reported the performance with the long-term user preference updating and the performance without updating. As shown in Figure 6, the performance with the long-term user

¹²Although the consideration of updating rate can obtain better performance, there is no uniform trends observed with the changing of β . Besides, the optimal value of β are different from dataset to dataset. For simplicity, we reported the results

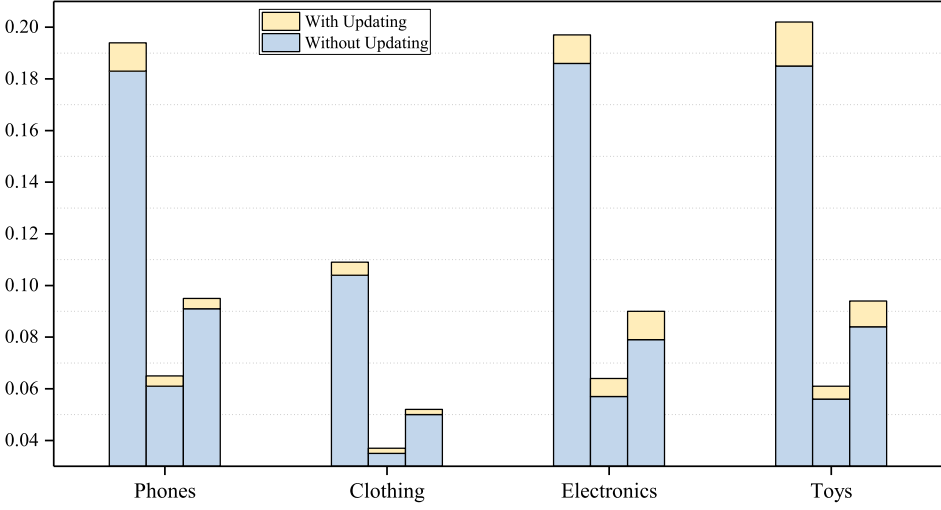


Fig. 6. Influence of the long-term user preference updating.

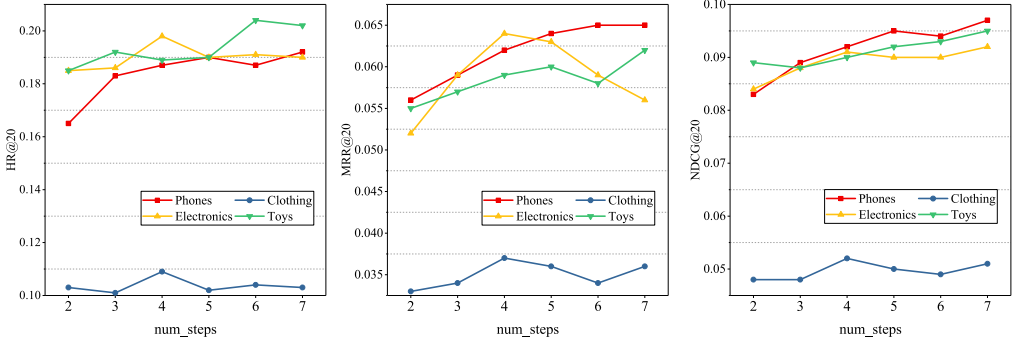


Fig. 7. Performance of ALSTP on four Amazon datasets by varying the numbers of previous queries with respect to HR, MRR and NDCG.

preference updating surpasses the one without updating. This is in accordance with our assumption that the long-term user preference should update gradually.

4.4 Influence of Important Parameters (RQ4)

In this section, we studied the influence of two parts: (1) *the number of previous purchase queries/items in short-term preference modeling* and (2) *the embedding size of the user preference on the final performance of our model*.

Number of Previous Queries. As show in Figure 7, the effect of the number of previous queries varies considerably on different datasets. For the datasets like *Phones* and *Toys*, user's purchasing behaviors are relatively sparse. For example, a user will not buy phones, phone accessories, or dolls frequently in a short time. Therefore, considering more previous queries could provide more information to infer the user's current preference (i.e., *Phones* for seven and *Kindle* for six). While for the other two datasets, there are more short-term patterns in the user's purchase behaviors.

based on a fixed value of β (i.e., $\beta = 0.5$), based on which a relatively good performance can be observed for all the four datasets.

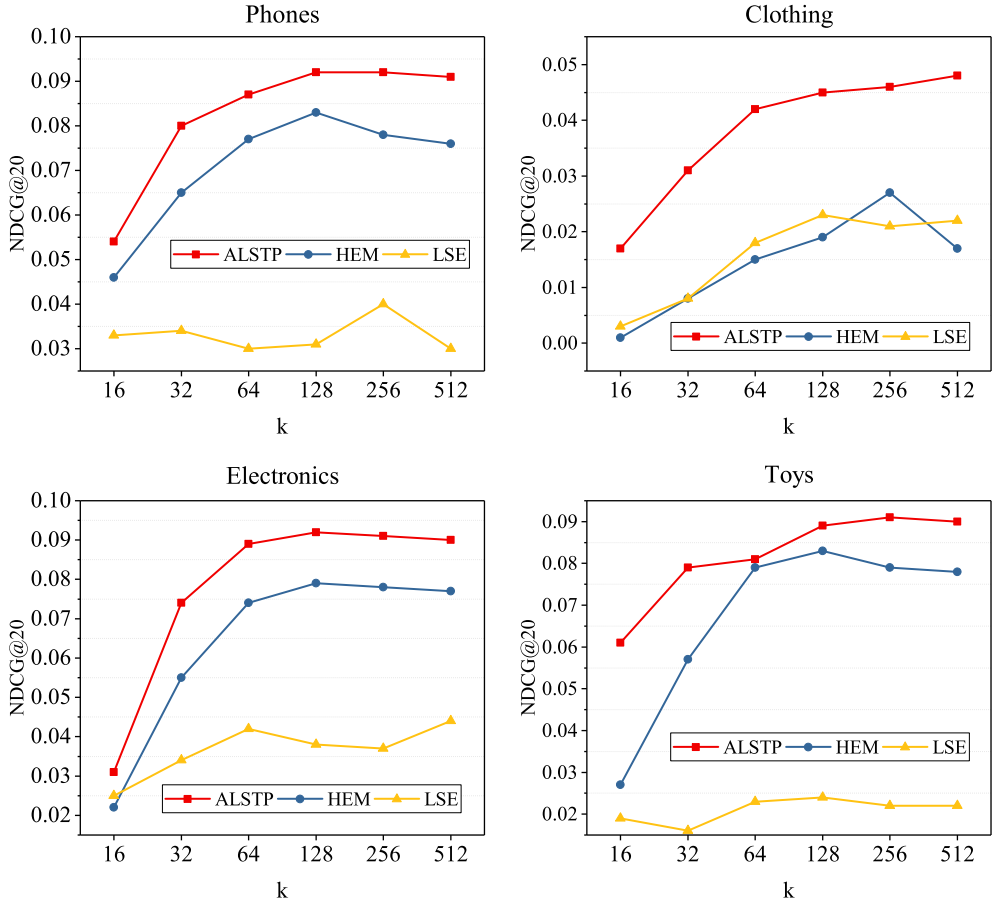


Fig. 8. Performance of LSE, HEM and ALSTP on four Amazon datasets by varying the embedding size of user preference with respect to NDCG.

Taking *Clothing* for example, users tend to buy a set of clothes in a short period, showing the obvious local preference. Due to a forthcoming event (e.g., *wedding* or *new season*), they will change to purchase a new style of clothes.

Embedding Size of User Preference. To analyze the effect of the embedding size on the baselines of LSE and HEM, as well as our proposed ALSTP model, we show the results of these methods with distinct embedding sizes over the four datasets. Figure 8 shows the performance variation with the increasing of the embedding sizes. It can be observed that for all the three methods, with the increasing of the embedding size, the performance improves firstly, and then starts to deteriorate. Generally, the larger embedding size will lead to the better representation capability, while it will result in over-fitting when the embedding size is too large. From the Figure 8, we can see that 256 is a proper embedding size for our model.

4.5 Short-term Preference Attention Visualization (RQ5)

To demonstrate the effectiveness of our attention mechanism on the short-term user preference, we illustrated a few examples. We randomly chose one example from each dataset. As shown in Figure 9, although our attention weights are computed based on the extracted queries

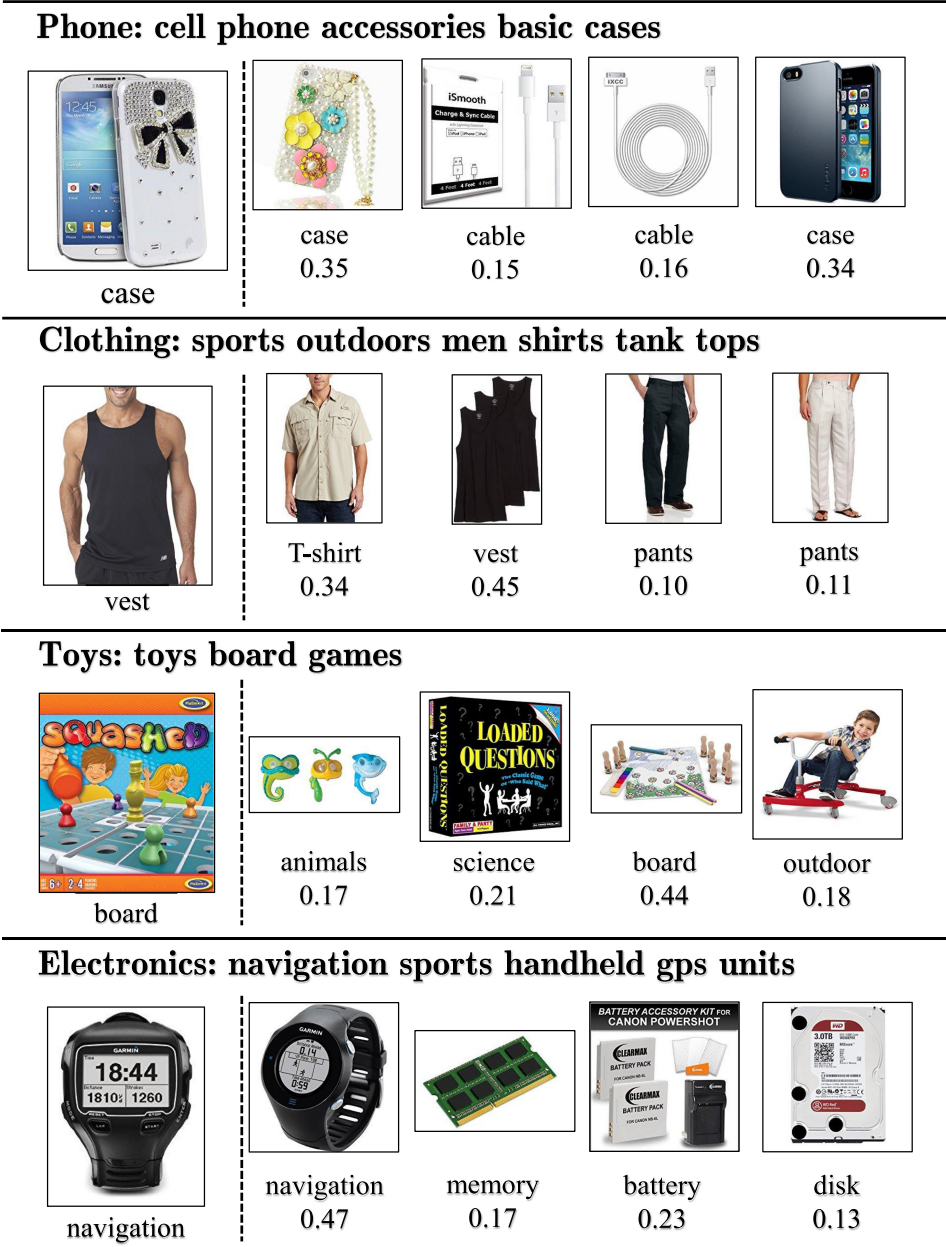


Fig. 9. Visualization results of the ASTPM model over four datasets. Here, the first line shows the chosen queries of each dataset, and we removed some words (i.e., words of category information) in the query for simplicity. The picture on the left side is the current product that the user would like to buy; the products on the right are immediately previously purchased products; values in the last line are the corresponding attention weights.

(corresponding to each product), we could still observe the relevance between the desired product and the previously purchased products. For the first query, the user wants to search *cell phone basic cases*, based on his/her former purchased *phone cases* and *phone cable*, we can see the two previously purchased *phone cases* are more related to the desired product than the others. In the second example, for the currently targeted search product *men's vest*, the previously purchased product like *men's long pants* and *men's T-shirt* are less related, while the recently bought *vest* is more useful on suggesting the targeted product. The user in the third example shows diverse interests in *toys & games*, and intends to find *board games*. Obviously, the third item is more related compared to the other three ones. In the last example, the user intends to buy a *navigation sports handheld* instead of *computer memory*, *battery* or *hard disk*, thus the purchased *navigation* is more informative in guiding the current choice.

Based upon the above observations and analysis, our attentive short-term preference modeling module is verified to be quite useful on inferring user's current search intention.

5 CONCLUSION AND FUTURE WORK

In this article, we present an ALSTP model to better capture the user preference for personalized product search. We argue that it is critical to consider the user's long- and short-term preferences simultaneously in the product search. To this end, we introduced a novel ALSTP model to capture the attentions of the long- and short-term user preferences on the current query, and then attentively integrated them with the current query to better represent the user's search intention. To the best of our knowledge, ALSTP is the first model that exploits the attention mechanism on both user's short and long-term preferences for personalized product search. We conducted extensive experiments on four real-world Amazon Datasets and demonstrated that ALSTP can remarkably outperform several state-of-the-art competitors in the product search. Besides, we analyzed the effectiveness of each component in our model and visualized the attentions in the short-term preference modeling. In the future, we plan to leverage a pre-trained *RNN* to better model the short-term user preference. Moreover, encoding user's demographic information into the long-term user preference modeling and semantic understanding of recently purchased products into the short-term user preference modeling may lead to the better performance.

REFERENCES

- [1] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 645–654.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *arXiv preprint arXiv:1409.0473*.
- [3] Paul N. Bennett, Ryan W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisjuk, and Xiaoyuan Cui. 2012. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 185–194.
- [4] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *Proceedings of the 25th International Conference on World Wide Web*. ACM, 531–541.
- [5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 335–344.
- [6] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE, 6298–6306.
- [7] Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. 2019. Review-driven answer generation for product-related questions in E-Commerce. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. ACM.

- [8] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Mohan Kankanhalli. 2018. MMALFM: Explainable recommendation by leveraging reviews and images. *arXiv preprint arXiv:1811.05318*.
- [9] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S. Kankanhalli. 2018. A³NCF: An adaptive aspect attention model for rating prediction. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI, 3748–3754.
- [10] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-Aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. ACM, 639–648.
- [11] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Syntax, Semantics and Structure in Statistical Translation* (2014), 103.
- [12] Sung H. Chung, Anjan Goswami, Honglak Lee, and Junling Hu. 2012. The impact of images on user clicks in product search. In *Proceedings of the 12th International Workshop on Multimedia Data Mining*. ACM, 25–33.
- [13] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). In *arXiv preprint arXiv:1511.07289*.
- [14] Mariam Daoud, Lynda Tamine-Lechani, and Mohand Boughanem. 2008. Learning user interests for a session-based personalized search. In *Proceedings of the 2nd International Symposium on Information Interaction in Context*. ACM, 57–64.
- [15] Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, and Bilal Chebaro. 2009. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM Symposium on Applied Computing*. ACM, 1732–1736.
- [16] Atish Das Sarma, Nish Parikh, and Neel Sundaresan. 2014. E-commerce product search: Personalization, diversification, and beyond. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 189–190.
- [17] Huizhong Duan and ChengXiang Zhai. 2015. Mining coordinated intent representation for entity search and recommendation. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM, 333–342.
- [18] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. A probabilistic mixture model for mining and analyzing product search log. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. ACM, 2179–2188.
- [19] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. Supporting keyword search in product database: A probabilistic approach. *Proc. VLDB Endow.* 6, 14 (2013), 1786–1797.
- [20] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. MIT, 249–256.
- [21] Anjan Goswami, Naren Chittar, and Chung H. Sung. 2011. A study on the impact of product images on user clicks for online shopping. In *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 45–46.
- [22] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, 55–64.
- [23] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. 2018. Multi-modal preference modeling for product search. In *Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1865–1873.
- [24] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. ACM, 507–517.
- [25] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. ACM, 173–182.
- [26] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Dávid Szepesvári. 2015. Session-based recommendations with recurrent neural networks. In *arXiv preprint arXiv:1511.06939*.
- [27] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*. MIT, 2042–2050.
- [28] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. ACM, 2333–2338.
- [29] Ameni Kacem, Mohand Boughanem, and Rim Faiz. 2017. Emphasizing temporal-based user profile modeling in the context of session search. In *Proceedings of the Symposium on Applied Computing*. ACM, 925–930.
- [30] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. ACM, 1188–1196.
- [31] Beibei Li, Anindya Ghose, and Panagiotis G. Ipeirotis. 2011. Towards a theory model for product search. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, 327–336.

- [32] Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Trans. Inf. Syst.* 36, 2 (2017), 11.
- [33] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 165–174.
- [34] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1419–1428.
- [35] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Information and Knowledge Management*. ACM, 811–820.
- [36] Nicolaas Mattheijs and Filip Radlinski. 2011. Personalizing web search using long term browsing history. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 25–34.
- [37] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.
- [38] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.
- [39] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.
- [40] Joao Palotti. 2016. *Learning to Rank for Personalized e-commerce Search at CIKM Cup 2016*. Technical Report. Tech. rep.
- [41] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A study of matchpyramid models on ad-hoc retrieval. In *arXiv preprint arXiv:1606.04648*.
- [42] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. DeepRank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 257–266.
- [43] Nish Parikh and Neel Sundaresan. 2011. Beyond relevance in marketplace search. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2109–2112.
- [44] Nish Parikh and Neel Sundaresan. 2011. A user-tunable approach to marketplace search. In *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 245–248.
- [45] Seung-Taek Park and David M. Pennock. 2007. Applying collaborative filtering techniques to movie search for better ranking and browsing. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 550–559.
- [46] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*. ACM, 1310–1318.
- [47] Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2017. NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1667–1676.
- [48] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI, 452–461.
- [49] Matthew Richardson. 2008. Learning about the world through long-term query logs. *ACM Trans. Web* 2, 4 (2008), 21.
- [50] Jennifer Rowley. 2000. Product search in e-shopping: A review and research propositions. *J. Cons. Market.* 17, 1 (2000), 20–35.
- [51] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 373–382.
- [52] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. ACM, 824–831.
- [53] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 373–374.
- [54] David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryan W. White, Susan Dumais, and Bodo Billerbeck. 2012. Probabilistic models for personalizing web search. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*. ACM, 433–442.

- [55] Smitha Sriram, Xuehua Shen, and Chengxiang Zhai. 2004. A session-based search engine. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 492–493.
- [56] Bin Tan, Xuehua Shen, and ChengXiang Zhai. 2006. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 718–723.
- [57] Imad Tbahriti, Christine Chichester, Frédérique Lisacek, and Patrick Ruch. 2006. Using argumentation to retrieve articles with similar citations: An inquiry into improving related articles search in the MEDLINE digital library. *Int. J. Med. Inform.* 75, 6 (2006), 488–495.
- [58] Yury Ustinovskiy and Pavel Serdyukov. 2013. Personalization of web-search using short-term browsing context. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. ACM, 1979–1988.
- [59] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, 165–174.
- [60] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI, 2922–2928.
- [61] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. 2010. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1009–1018.
- [62] Chen Wu, Ming Yan, and Luo Si. 2017. Ensemble methods for personalized e-commerce search challenge at CIKM Cup 2016. In *arXiv preprint arXiv:1708.04479*.
- [63] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 723–732.
- [64] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI.
- [65] Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *Proceedings of the European Conference on Computer Vision*. Springer, 451–466.
- [66] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. ACM, 2048–2057.
- [67] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 21–29.
- [68] Jun Yu, Sunil Mohan, Duangmanee Pew Putthividhya, and Weng-Keen Wong. 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, 463–472.
- [69] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22, 2 (2004), 179–214.
- [70] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 187–196.
- [71] Liron Zighehnic and Oren Kurland. 2008. Query-drift prevention for robust query expansion. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 825–826.

Received May 2018; revised September 2018; accepted November 2018