Multi-Resolution Attention for Personalized Item Search

Furkan Kocayusufoglu
Google Inc., Mountain View, CA, USA
furkank@google.com

Tao Wu
Google Inc., Mountain View, CA, USA
iotao@google.com

Anima Singh
Google Inc., Mountain View, CA, USA
animasingh@google.com

Georgios Roumpos
Google Inc., Mountain View, CA, USA
roumposg@google.com

Heng-Tze Cheng
Google Inc., Mountain View, CA, USA
hengtze@google.com

Sagar Jain
Google Inc., Mountain View, CA, USA
sagarj@google.com

Ed Chi
Google Inc., Mountain View, CA, USA
edchi@google.com

Ambuj Singh
University of California, Santa Barbara, CA, USA
ambuj@cs.ucsb.edu

ABSTRACT

Personalized item search has become an essential tool for online platforms—where users interact with a large corpus of items (*e.g.*, click, purchase, like) via a search query—to provide their users with a more satisfactory search experience. The record (or history) of users' past interactions serves as a valuable asset to achieve personalization. While user history data can span over a long period of time, only a part of the history is relevant to a user's current search intent. Moreover, since historical interactions take place at aperiodic points in time, modeling their relevance to the current search query entangles complex temporal dependencies. We propose *multi-resolution attention* to address these challenges for personalized item search. Our approach captures higher-order temporal relations between user queries and their history across several temporal subspaces (*i.e.*, resolutions), each corresponding to distinct temporal ranges with *adaptive* time boundaries that are also learned directly from data. We achieve this by coupling the conventional multi-head attention module with a differentiable soft-thresholding mechanism, which essentially operates as a masking function in the temporal domain. Comparisons with strong baselines on an open-source benchmark dataset confirm the efficacy of our approach.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Information systems** → **Information retrieval**.

KEYWORDS

item search, personalization, temporal attention, multi-resolution attention, recommender systems

ACM Reference Format:
Furkan Kocayusufoglu, Tao Wu, Anima Singh, Georgios Roumpos, Heng-Tze Cheng, Sagar Jain, Ed Chi, and Ambuj Singh. 2022. Multi-Resolution Attention for Personalized Item Search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498426

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02.
https://doi.org/10.1145/3488560.3498426

1 INTRODUCTION

Recent years have seen rapid growth in popularity and complexity of online e-commerce and content-sharing platforms (*e.g.* music streaming [3], video streaming [6], photo sharing [12]). Consequently, developing a high-quality search engine has become one of the key objectives of these online platforms with millions of active users. Despite their contextual differences, all of these platforms bear the common challenge of retrieving suitable contents from a large searchable database to satisfy their users' search intents.

Users interact with such platforms in highly personalized ways [29]. The same search query entered by different users is likely to carry different search intentions, due to the diverse nature of personal taste and preferences [33, 35]. To that extent, historical interactions of users serve as a great asset for the problem of personalized item search to improve users' search experiences. For example, one can look for *relevant* signals within the user history that can inform users' intent behind their search query. The idea of utilizing users' history to better understand their search needs has been widely studied in the literature and proven valuable for various domains including product search [1, 2, 14], web search [27, 34], microblog search [39], and video search [10, 22].

Researchers have explored various directions to model user history, most of which are naturally formulated as a sequence modeling problem, since user history data often originate as a sequence of (ordered) interactions (e.g. purchases, watches, likes). Amongst the previously proposed mechanisms, self-attention [38] has gradually become a key component in sequence modeling tasks, leading to state-of-the-art results across many domains, including natural language processing [11], speech recognition [9], and recommender systems [19]. The self-attention mechanism has also proven useful in personalized item search, thanks to its ability to detect attention weights from the input event sequence with respect to the given context (in this case, search query). Such attention weight distribution intrinsically carries a notion of relevance between the search query and user history, leading to contextualized personalization.

Most of the self-attention based and other sequential models by design account for sequential signals rather than temporal signals.

However, the latter aspect has significant implications for personalized item search. Since the user interactions take place at aperiodic points in time [7], there can be gaps between the sequential patterns and temporal patterns of user behaviors (as illustrated in Figure 1b). This entangles different explanatory factors unique to personalized item search. Intrinsically, the time spans between the search query and user history items directly affect their degree of relevance (e.g., interaction occurred a day ago vs. a month ago). While there are emerging efforts to incorporate temporal information into neural sequential recommendation models [17, 23, 41, 43, 44], where the goal is to recommend items that are likely to be of interest to users solely based on their past interactions (without the existence of a search query), this research direction has not been adequately studied for the setting of personalized item search.

In this work, we propose *multi-resolution attention* for personalized item search. Multi-resolution attention effectively retrieves relevant items for users while accounting for higher-order temporal dependencies between their search query and item history. The key innovative idea behind our method is to compute the relevance between the search query and the history items over various temporal regions (or subspaces), which in turn can recognize and incorporate users' interests from various temporal resolutions. We achieve this by a novel multi-head attention formulation that explicitly enforces different attention heads to cover parts of the sequence that belong to distinct temporal regions with adaptive time boundaries, which are also learned jointly with the rest of the model. Our approach comes in two variants, each designed to accommodate different temporal densities of real-world data.

We evaluate the proposed approach on a public benchmark dataset and compare it with strong baseline approaches, including the adaptations of two state-of-the-art temporal models [23, 41] originally proposed for the sequential recommendation task. Our experiments showcase that multi-resolution attention consistently achieves superior performance across five different domains, outperforming the best baseline by up to 4.7%. Moreover, our method is a parameter efficient alternative to existing embedding-based [23] and kernel-based [41] methods, providing a new perspective on modeling the complex temporal nature of user history.

2 RELATED WORK

Personalized item search is a generic concept aiming to improve users' search experience by retrieving personalized items from a large searchable database. Conceivably, the most popular application domain for personalized item search is online e-commerce, wherein the term product is generally used as a substitute for the term item. Studies on personalized *product search* [1, 2, 4, 5, 14, 25, 42] essentially aim to link search queries with products (often via their contextual information) while taking into account the users' previous action logs within the platform.

Being the earliest study to investigate personalization in product search, authors in [2] proposed a hierarchical embedding model to learn latent semantic representations of users, products and queries with their associated language data (e.g. review), and retrieve products according to the similarities directly measured in this latent space. In another study [14], authors proposed a technique consisting of two attention networks, each designed to independently capture short and long-term user interests. However, their method defines the short/long-term interest solely based on the sequential order of interactions (*e.g.* last m interactions), which can not capture the rich temporal patterns found in data. Considering that the previous studies often model user history and their search query as separate signals, authors in [1] more recently argued that the two signals are tightly connected, and investigated the potential of personalization with respect to query characteristics. While their findings highlight the need for query-aware personalization in product search, their method essentially treats all previous interactions as a set, ignoring their order, let alone temporal dynamics.

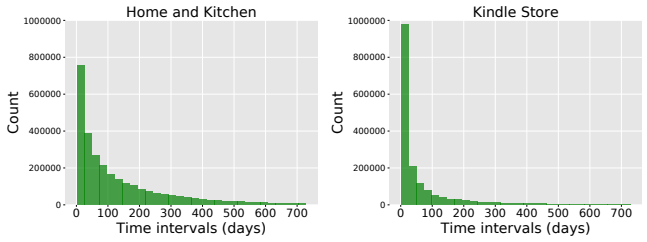
The pursuit of personalization has also become one of the main pillars in the development of search engines for content-sharing platforms [13, 18]. The diverse contextual nature of such platforms imposes unique representational challenges. In the lack of descriptive information, authors in [18] combined the discrete user history signal with the corresponding provider information to perform personalized item retrieval. Another work [13] utilizes multiple in-session (clicks, contacts) and meta-data signals to jointly learn user and listing embeddings for personalized home listing search.

Despite their great success, the aforementioned studies for personalized item search do not leverage the rich temporal signals found in data. Being the first to address this problem, our work aims to recognize and capture higher-order temporal dependencies between users' search queries and item history using a novel multi-resolution attention mechanism.

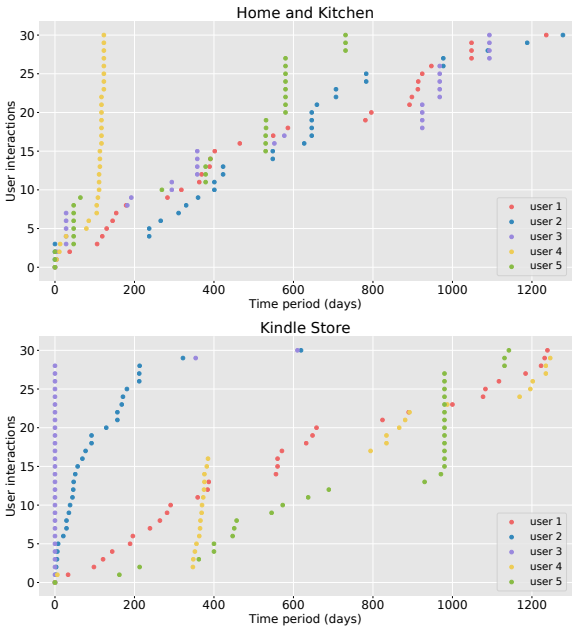
Sequential recommendation is another line of related work to ours, with a prominent difference that the users can not specify their information needs explicitly. Therefore, the goal is to recommend items that are likely to be of interest to users solely based on their history. Amongst the pioneer studies, authors in [19] leveraged the self-attention mechanism for adaptive summarization of user history. Memory networks are adopted in [8] to memorize the anchor items that drive future user actions. Another study [26] proposed a hierarchical gating network to adaptively control which latent features of items will contribute to the downstream task.

There have been recent studies to model the *temporal aspect* of user history in the context of sequential recommendation [17, 23, 41, 43, 44]. Amongst these studies, authors in [23] proposed a time-aware self-attention module, which—in addition to relative position representations [32]— learns relative time interval representations to jointly capture both the sequential and the temporal nature of user interactions. The idea of modeling time intervals between user interactions is also studied in a recent work [41], wherein the authors instead employed a combination of different time kernels to calibrate the attention weights between user interactions based on their relative time intervals.

These studies fundamentally differ from the setting of personalized item search because the presence of a search query marks a pivotal point in time, which has distinct indications in modeling the temporal signals with respect to user history. Needless to say, different search queries entered by users have varying temporal relations to their interaction history, the extent of which also depends on the context of the query (*e.g.* "sports watch" vs "Fitbit Versa 3"). Nonetheless, our experiments include adaptations of the aforementioned temporal recommendation models [23, 41] as baselines to (1) better assess the efficacy of our proposed approach, and (2) bridge the methodological gap between the two problem settings.



(a) Histogram of time intervals between consecutive interactions of users.



(b) Last thirty interactions of the same set of five users who actively interact with both categories. Y axis represents the users' interactions in chronological order, while the X axis represents the time period (in days) with respect to their latest interaction.

Figure 1: An analysis on temporal resolution of user interactions with two broad categories: Home and Kitchen, Kindle Store.

3 TEMPORAL RESOLUTION OF USER INTERACTIONS

We first provide insights into the complex temporal dynamics of user interactions observed in a real-world setting [7] and further draw connections to key motivations behind our proposed method.

In the context of personalized item search, there exist numerous factors contributing to the relationship between a search query and the recorded user interactions (*i.e.*, history). Such relationships are often tightly connected to the temporal aspect of the user histories [20, 21, 23, 24], as well as the users' search intents [1] (which, in turn, are reflected on their query formulations [15, 33]). For instance, while some search queries (e.g. "action movies") might exhibit longer-term dependencies on user histories, some others (e.g. "humidifier") are triggered by users' short-term interests, hence might not depend on their longer-term history data. Moreover, such dependencies might display strong periodic patterns, as in the case of recurrent purchases of grocery products or cleaning supplies.

To study the complex and non-linear nature of these temporal dependencies in data, one needs to examine the temporal dynamics of user interactions. Figure 1a plots the histogram of time intervals between two consecutive interactions (of the same user) for two broad product categories, showing that the users' interaction patterns vary depending on the context of their search. Furthermore, to observe users' interaction patterns more closely for both categories, we also plot the most recent interactions of the same set of five active users, together with the temporal information of their interactions. Figure 1b shows that even the same user might exhibit vastly different temporal patterns when interacting with different categories (*e.g.*, users 3 and 4), revealing one of the major challenges in personalized item search. We also observe that user interactions tend to be concentrated (or grouped) over multiple temporal spans, each corresponding to a relatively narrow time period (e.g. 2-3 days). On the other hand, the gaps between such grouped interactions can be very large (e.g., over a year). For the remaining of this paper, we refer to such grouped temporal patterns as the ***temporal resolutions*** of user interactions.

We further argue that the potential for personalization differs across these resolutions. With that being our key intuition, we propose a novel approach (*multi-resolution attention*) that can (i) adaptively recognize such temporal resolutions and (ii) accurately model the diverse dependency patterns between search queries and user histories across these resolutions. The next section explains the details of our method, which is illustrated in Figure 2.

4 PROPOSED METHOD

4.1 Problem Setting

We start with formally introducing our problem and the notations used in this paper. Let $\mathcal{I}$ denote the set of items, $\mathcal{U}$ denote the set of users. For each user $u \in \mathcal{U}$, we are given the following inputs: *(i)* a recent search query q^u, *(ii)* a time-ordered list of previously interacted items $S^u = (v_1^u, \cdots, v_{|S^u|}^u)$ where $v_i^u \in \mathcal{I}$, and *(iii)* a list of timestamps $T^u = (t_1^u, \cdots, t_{|T^u|}^u)$ corresponding to each interaction, where $t_1^u \leq \cdots \leq t_{|S^u|}^u \leq t_{q^u}$, with t_{q^u} being the timestamp of q^u, and $|S^u|$ (or $|T^u|$) denoting the number of interactions the user u previously had with the system. Our goal is to predict the $(|S^u|+1)$th item that the user u will interact with.

The main notations used in our paper are summarized in Table 1.

Notation	Description
$\mathcal{I}, \mathcal{U}$	item and user set
Q^u	user u's query sequence (in chronological order)
S^u	user u's item sequence (corresp. to Q^u)
T^u	user u's timestamp sequence (corresp. to Q^u, S^u)
N	maximum sequence length
d, d_q	latent dimensions
h	number of attention heads
$\mathbf{q}, \mathbf{w}$	query and word embedding vectors, respectively
$\mathbf{M}, \mathbf{P}$	item and position embedding matrices, respectively
$\mathbf{T}$	relative time interval matrix
$\widehat{\mathbf{E}}$	input embeddings (corresp. to S^u)
$\mathbf{E}^{(l)}$	input embeddings after (l)th self-attention block
$\mathbf{E}^{\mathbf{q}}$	query embeddings (corresp. to Q^u)
$\mathbf{E}^{\mathbf{h}}$	input history encoding
$\mathbf{E}^{\mathbf{qh}}$	query-aware history encoding
$\mathbf{H}$	output representation of the model
Δ_i	time boundary for ith attention head

Table 1: Main notations used in this paper.

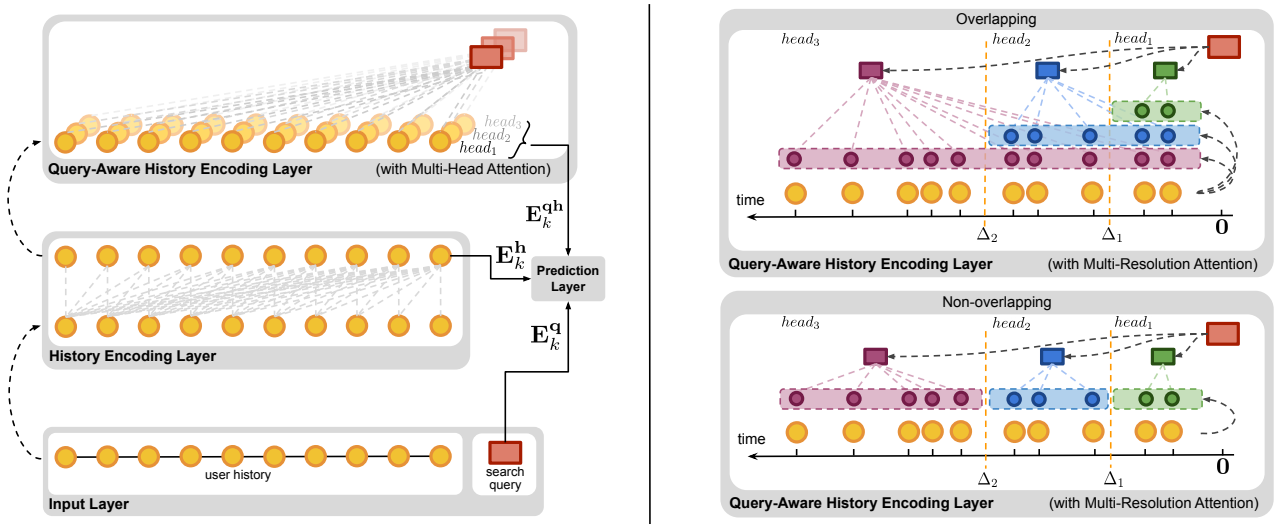


Figure 2: An illustration of our proposed method architecture. The left side shows the high-level components of our method, while the right part shows the proposed *multi-resolution attention* variants: *overlapping* (top) and *non-overlapping* (bottom).

4.2 Preliminaries

This section introduces preliminary techniques that serve as the building blocks of our framework, which are explained in the context of recommender systems to maintain contextual consistency.

The **self-attention mechanism** [38] aims to capture the importance (attention) weights of the sequential inputs (in our case, items) that are identified through the inner products of item representations. The items with higher attention weights have more contributions to the final output representation, and consequently, to the final downstream task. Such mechanism in principle assumes that the output of a given sequential input is relevant to only part of the sequence, which makes it a natural and desired instrument for recommendation tasks [1, 8, 19, 23, 36]. It is formally defined as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_{attn}}})V \tag{1}$$

where Q, K, and V respectively denote the queries, keys, and values of items in the sequence. Here it is important to note that the above term "query" is domain-agnostic, *i.e.*, it is not tied to the notion of "search query" in the context of our work. This mechanism relies on the positional embeddings to recognize and capture the sequential order of items. Hence, in applications, the vector representation for each position is combined with the corresponding item embeddings.

Causality. Due to the nature of our problem, the model should only take into account the previous items when predicting the next item. Therefore, we need to prevent leftward information flow (leak) in self-attention computation. This is achieved by masking the upper triangular entries of $QK^\top$, that is, $(QK^\top)_{i,j} = -\infty \; \forall i < j$.

The **self-attention block (SAB)** is defined as a combination of self-attention and point-wise feed-forward network (FFN) layers:

$$\text{SAB}(X) = \text{FFN}(\text{Attention}(XW^Q, XW^K, XW^V)) \tag{2}$$

where $W^Q \in \mathbb{R}^{d \times d_{attn}}$, $W^K \in \mathbb{R}^{d \times d_{attn}}$, $W^V \in \mathbb{R}^{d \times d}$ are the (linear) projection weight matrices. FFN is essentially a two-layer MLP with ReLU activation, applied *independently* to each position of the input.

4.3 Query-Aware Personalization

This section focuses on building the base of our architecture by leveraging several neural components introduced in Section 4.2. The subsequent section is dedicated to the novel components of our method, explaining how we leverage the temporal signal.

For training purposes, we transform the input sequence of each user into a fixed-length sequence of N interactions $S^u=(v_1^u, \cdots, v_N^u)$. If the input sequence has more than N items, we only consider the most recent N interactions and omit the remaining items, while if the sequence length is less than N, we left pad the sequence until it reaches the length N. Same procedure is also applied to the corresponding sequence of timestamps $T_u = (t_1^u, \cdots, t_N^u)$ and the sequence of queries $Q^u = (q_1^u, \cdots, q_N^u)$. Each query q_i^u results in interaction with item v_i^u. For simplicity, we assume that both q_i^u and v_i^u have the same timestamp (t_i^u), hence once can define an event $e_i^u = (q_i^u, v_i^u, t_i^u)$. Here we note that the Q^u is formed solely for notational convenience, and our method (1) does not require each history item to be associated with a particular query, and (2) does not consider the previous query signals in its next item predictions.

4.3.1 ***Embedding Layer:*** Next, we create sets of *learnable embeddings* for items, positions, and search queries, which are then processed by a series of self-attention blocks.

Item Embeddings. An item embedding matrix is denoted as $\mathbf{M} \in \mathbb{R}^{|I| \times d}$. The row vector $\mathbf{M_v} \in \mathbb{R}^d$ represents the embedding of an item $v \in I$. Zero vector is used for the padding items.

Position Embeddings. A learnable position embedding matrix is denoted as $\mathbf{P} \in \mathbb{R}^{N \times d}$. The row vector $\mathbf{P_k} \in \mathbb{R}^d$ represents the embedding of a position $k \in [1, \cdots, N]$.

Query Embeddings. One of the standard ways to form a query embedding is to compute the average of its word embeddings [40]: $\mathbf{q} = \frac{\sum_{w \in q} \mathbf{w}}{|q|}$, where $\mathbf{q} \in \mathbb{R}^{d_q}$ and $\mathbf{w} \in \mathbb{R}^{d_q}$ respectively denote the query and word embeddings, and $|q|$ is the length of the query q.

4.3.2 ***Input Layer:*** Given an input sequence of interacted items $S^u = (v_1^u, \cdots, v_N^u)$ and queries $Q^u = (q_1^u, \cdots, q_N^u)$, we first right

shift the item sequence by one index $\hat{S}^u = (<pad>, v_1^u, \cdots, v_{N-1}^u)$ and then map both sequences to their embedding forms. For $\hat{S}^u$, we combine the embeddings of items and their absolute positions to form input embeddings ($\widehat{\mathbf{E}}$):

$$\widehat{\mathbf{E}} = \begin{bmatrix} \mathbf{0} + \mathbf{P_1} \\ \mathbf{M_{v_1^u}} + \mathbf{P_2} \\ \cdots \\ \mathbf{M_{v_{N-1}^u}} + \mathbf{P_N} \end{bmatrix}, \quad \mathbf{E^q} = \begin{bmatrix} \mathbf{q_1^u} \\ \mathbf{q_2^u} \\ \cdots \\ \mathbf{q_N^u} \end{bmatrix} \tag{3}$$

where $\widehat{\mathbf{E}} \in \mathbb{R}^{N \times d}$, and $\mathbf{0}$ is the padding vector. $\mathbf{E^q} \in \mathbb{R}^{N \times d}$ represents the query embedding matrix.

4.3.3 ***History Encoding Layer:*** Next, to capture item-item relations, a stack of L self-attention blocks are employed to transform the input embeddings ($\widehat{\mathbf{E}}$) to another latent representation $\mathbf{E^h}$:

$$\begin{aligned} \mathbf{E}^{(0)} &= \widehat{\mathbf{E}} \\ \mathbf{E}^{(l+1)} &= \text{SAB}(\mathbf{E}^{(l)}), \quad \forall\, l \in [0, \cdots, L-1] \\ \mathbf{E^h} &= \mathbf{E}^{(L)} + \widehat{\mathbf{E}} \end{aligned} \tag{4}$$

where $\mathbf{E^h} \in \mathbb{R}^{N \times d}$ is the output of the Lth self-attention block with skip connection to the input embeddings. Being referred as the *item history encoding*, it is essentially a non-linear transformation of input embeddings, where the kth representation ($\mathbf{E_k^h} \in \mathbb{R}^d$) can be seen as a compact summary of the first k interactions, and be used to predict the $(k+1)$th interacted item [19]. However, such representation alone is not sufficient to fully capture the user intent, since it is still unaware of the search query.

4.3.4 ***Query-Aware History Encoding Layer:*** Our next component summarizes the query-relevant parts of user history by capturing query-item relations rooted in data. It consists of an additional attention module, in which the attention weights over the outputs of the history encoding are computed with respect to the search queries. Specifically, we employ a multi-head attention layer [38], which learns attention distributions in h different d/h-dimensional representation subspaces:

$$\begin{aligned} &\mathbf{E^{qh}} = Concat(head_1, ..., head_h) \\ &\text{where } head_i = \text{Attention}(\mathbf{E^q} W_i^Q, \mathbf{E^h} W_i^K, \mathbf{E^h} W_i^V) \end{aligned} \tag{5}$$

where $W_i^Q \in \mathbb{R}^{d_q \times (d/h)}$, $W_i^K \in \mathbb{R}^{d \times (d/h)}$, $W_i^V \in \mathbb{R}^{d \times (d/h)}$ are projection matrices for each head, and h is the number of attention heads. $\mathbf{E^{qh}} \in \mathbb{R}^{N \times d}$ is called *query-aware history encoding*, summarizing the parts of the interaction history that are most relevant to the search query. Note that we further extend this layer in Section 4.4 to incorporate temporal information using a novel multi-resolution attention module.

4.3.5 ***Prediction Layer:*** Leveraging all the components introduced so far, we now can predict the next item based on the previous $k-1$ items and the kth search query. In more detail, we combine the representations of the query, the query-aware history encoding and the item history encoding to form a final latent representation:

$$\mathbf{H} = ReLU(Concat(\mathbf{E^q}, \mathbf{E^{qh}}, \mathbf{E^h}))W^H \tag{6}$$

where the weight matrix $W^H \in \mathbb{R}^{(d_q+2d) \times d}$ projects the combined representation back into d dimensions. Finally, we measure the relevance score ($r_{k,v_i} \in \mathbb{R}$) of the kth interacted item being $v_i \in \mathcal{I}$ by:

$$r_{k,v_i} = \mathbf{H_k} \mathbf{M}_{\mathbf{v_i}}^\top \tag{7}$$

where $\mathbf{H_k} \in \mathbb{R}^d$ is the kth row vector of $\mathbf{H}$, and $\mathbf{M_{v_i}} \in \mathbb{R}^d$ is the embedding of item v_i. Intuitively, items with higher relevance scores are more likely to be interacted, thus we can generate recommendations by ranking the items based on their relevance scores.

4.3.6 ***Optimization:*** Recall that we convert the input sequence S^u into a fixed N-length sequence, shifted to the right by one index; $\hat{S}^u = (<pad>, v_1^u, \cdots, v_{N-1}^u)$ with the expected output (prediction) sequence being $O^u = S^u = (v_1^u, \cdots, v_N^u)$. In order to learn accurate relevance scores of expected outputs, we use the cross-entropy loss with 100 negative samples at each step:

$$\mathcal{L} = -\sum_{\{\mathcal{E}^u | u \in \mathcal{U}\}} \sum_{k=2}^{N} \left[log(\sigma(r_{k,O_k^u})) + \sum_{v_j \notin O^u} log(1 - \sigma(r_{k,v_j})) \right] \tag{8}$$

where σ is the *sigmoid* function and $\mathcal{E}^u = (\hat{S}^u, Q^u, T^u, O^u)$ includes the model inputs and expected output for user u. Note that we ignore the first index due to padding. More details on training and implementation are provided in Section 5.3.

4.4 Multi-Resolution Attention

Modeling input sequences as a combination of item ids and their absolute positions assumes a homogenous temporal resolution across the entire sequence, i.e., time intervals between all adjacent items are the same. However, this is rarely the case in real-world applications [23] as we also demonstrated in Figure 1b. Motivated by these observations, we now propose a novel approach to incorporate the rich temporal resolution of user history in the setting of personalized item search. To emphasize, we are interested in temporal dependencies between the search query and the past interacted items, unlike the query-less setting where the temporal dependencies are studied solely within the item domain [23, 24, 41, 43, 44].

We introduce a new attention layer—*MultiResAttn*—that is designed to capture asymmetric query-item relations across multiple time resolutions. The main intuition behind our approach is to explicitly guide multiple attention heads to focus on parts of the item sequence that belong to distinct temporal ranges (*i.e., resolutions*).

We first define an attention function $\tilde{A}$ (an adaption of Eq. 1) as:

$$\tilde{A}(Q, K, V, C) = \text{Softmax}(\frac{QK^\top + C}{\sqrt{d_{attn}}})V \tag{9}$$

where $C \in \mathbb{R}^{N \times N}$ is an additive input to the softmax function, allowing flexibility for controlling (or scaling) the attention weights between queries and items. Note that all upper triangular elements of C are set to $-\infty$ by default to avoid future information leakage. We now explain how we leverage this adaption in our query-aware history encoding layer by modifying Equation 5 to take the form:

$$\begin{aligned} &\mathbf{E^{qh}} = MultiResAttn(\mathbf{E^q}, \mathbf{E^h}, \mathbf{E^h}, \mathbf{T}) \\ &\text{where } MultiResAttn(Q, K, V, T) = Concat(head_1, ..., head_h) \\ &\text{and } head_i = \tilde{A}(QW_i^Q, KW_i^K, VW_i^V, \Phi_i(\mathbf{T})) \end{aligned} \tag{10}$$

with $\Phi_i : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$ and $\mathbf{T} \in \mathbb{R}^{N \times N}$. $\mathbf{T}$ is a lower triangular matrix including relative time intervals between the search queries and the items; $\mathbf{T}_{k,j} = t_k^q - t_j^v$ (when $k \geq j$), with t_k^q and t_j^v being

the timestamps of kth query and jth item, respectively. Recall that $t^q_k = t^v_{k+1}, \forall k \in [1, ..., N-1]$, due to shifted item sequence (Eq. 3).

With the help of $\Phi_i(\cdot)$ function, we can enforce certain constraints on $head_i$'s attention distribution, based on $\mathbf{T}$. To this end, we consider two different variants: *(i)* **non-overlapping** and *(ii)* **overlapping** multi-resolution attention. These variants are also illustrated in Figure 2 (right). As the names suggest, for the former variant, the time ranges that attention heads cover do not coincide, i.e., $head_1$ covers $[\Delta_0, \Delta_1)$, $head_2$ covers $[\Delta_1, \Delta_2)$ and so on. For the latter variant, each head instead covers an extended range, i.e., $head_1$ covers $[\Delta_0, \Delta_1)$, $head_2$ covers $[\Delta_0, \Delta_2)$ and $head_h$ covers $[\Delta_0, \Delta_h)$. The following $\Phi_i(\mathbf{T})$ function achieves this by masking the items that are out of the desired temporal ranges for $head_i$:

$$\text{overlapping}: \quad \Phi_i(\mathbf{T})_{k,j} = \begin{cases} 0 & \Delta_0 \leq \mathbf{T}_{k,j} < \Delta_i \\ -\infty & otherwise \end{cases} \tag{11}$$

$$\text{non-overlapping}: \quad \Phi_i(\mathbf{T})_{k,j} = \begin{cases} 0 & \Delta_{i-1} \leq \mathbf{T}_{k,j} < \Delta_i \\ -\infty & otherwise \end{cases} \tag{12}$$

$$\text{s.t. } \Delta_{i-1} < \Delta_i, \ \forall i \in [1, ..., h] \text{ and } \Delta_0 = 0$$

where $\Phi_i(\mathbf{T})_{k,j}$ represents the (k, j)th entry of $\Phi_i(\mathbf{T})$. Note that the time boundaries of attention heads (Δ_i) can be seen as temporal cut-off points in time, which in turn decides on how much representational power is allocated to the respective temporal ranges. A natural choice is to favor most recent interactions with shorter ranges from the search query since they tend to carry a higher influence on users' next interactions [23]. This can be achieved by computing Δs using some form of an exponential function, such as $\Delta_i = ab^i$, where the hyper-parameters $a, b \in \mathbb{R}^+$ are of the same time units as $\mathbf{T}$ (*e.g.* hours, days). While such formulation complies with the exponentially decaying influence phenomenon commonly observed in the literature [28, 41], by varying $\{a, b\}$, one can adapt Δs to different domains with varying temporal resolutions.

Finding good Δs by hyper-parameter tuning can be challenging and may require excessive computational effort. Next, we take our idea a step further and propose a more flexible and adaptive approach. Our goal is to *learn* Δs jointly with the rest of the model. However, the hard-thresholding mechanism (Eq. 11 & 12) is not differentiable and prevents the model from learning Δs through back-propagation. To sidestep this issue, we propose the following softer-thresholding reparameterization, which remains differentiable with respect to Δs:

$$\Phi_i(\mathbf{T})_{k,j} = \begin{cases} log(\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau})) & \text{overlapping} \\ log(\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau})) + log(\sigma(\frac{\mathbf{T}_{k,j} - \Delta_{i-1}}{\tau})) & \text{non-overlapping} \end{cases} \tag{13}$$

where log and σ denote the natural logarithm and the sigmoid function, while $\tau \in \mathbb{R}^+$ is the temperature scaling parameter.

Taking a closer look into the overlapping variant, the respective item is *masked* when $\Delta_i - \mathbf{T}_{k,j} << -\tau$ (that is, $\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau}) \approx 0$ and $log(\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau})) \approx -\inf$). Conversely, it is kept when $\Delta_i - \mathbf{T}_{k,j} >> \tau$ (that is, $\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau}) \approx 1$ and $log(\sigma(\frac{\Delta_i - \mathbf{T}_{k,j}}{\tau})) \approx 0$). In other words, the items that are far from Δ_i are either kept or masked based on whether they fall inside or outside of the corresponding boundary. We note that $\partial head_i / \partial \Delta_i \approx 0$ for such items, hence they do not contribute to the learning of Δ_i. On the other hand, the "near boundary" items (*i.e.* $|\Delta_i - \mathbf{T}_{k,j}| \sim \tau$) may or may not be masked depending on their contribution to the final loss, which in turn generates either a pull or a push force on Δ_i. Furthermore, it is straightforward to apply the same logic to the non-overlapping variant, where the second term further masks the items that are already covered by the previous head with boundary Δ_{i-1}. Lastly, some attention heads may have no coverage for certain users who have no interactions within (or near) the respective temporal regions (see Figure 1b). In such cases, we set $head_i$ to zero vector to indicate the lack of interactions for that particular resolution.

In practice, we initialize Δs using the aforementioned exponential function for faster adaptation and further update them during training. That said, the proposed module is generic and one can choose any increasing function for initializing Δs. More details on training and hyper-parameters are given in Section 5.3.

5 EXPERIMENTS

This section introduces our experimental setup, and presents an empirical analysis of our proposed approach. The experiments aim at quantitatively evaluating the contributions of each introduced model component (illustrated in Figure 2), as well as comparing our proposed variants with alternative techniques in the literature.

5.1 Datasets and Evaluation

Datasets: We evaluate the performance of our method on an open-source benchmark dataset from Amazon [30]. The 5-core version of the dataset is used, where all users and items with less than 5 reviews are removed. Following [2, 19], we treat the presence of a review as an interaction and use the respective timestamps to determine the temporal order of interactions. All other contextual information of items is disregarded. We follow the common practice (outlined in [2, 37]) to extract realistic queries for each user-item interaction based on the respective items' hierarchical category information. Although these queries are shown to be similar to real user query formulations in e-commerce platforms [31], we observe that they lead to memorization issues in our setting because each item is always associated with the same query across all users. To alleviate this issue and make the problem more challenging, we randomly drop 50% of the words from the associated query for each user-item interaction recorded in data, leading to more diverse query formulations of the same item across different users.

The following diverse range of categories are employed in our experiments: *Home and Kitchen, Kindle Store, Movies and TV, Pet Supplies, Grocery and Food*. Due to computational constraints, we further remove items with less than 15 interactions for *Home and Kitchen* and less than 10 interactions for *Grocery and Food* category. Dataset statistics are given in Table 2. We follow the same prepossessing steps mentioned in [19]. For users who interacted with at least three items, we use their second last interaction for validation and their last interaction for testing, while the remaining interactions are used for training.

Evaluation Metrics: We evaluate ranking performance by computing Hit@K and NDCG@K with $K \in \{3, 10\}$. Hit@K is a recall-focused metric measuring the percentage of times that the ground-truth next item is among the top K items, while NDCG@K is a

	Home & Kitchen	*Kindle Store*	*Movies & TV*	*Pet Supplies*	*Grocery & Food*
Number of users	229,210	161,790	250,893	243,690	147,474
Number of items	97,100	153,242	65,860	71,457	44,672
Number of query words	2,946	151	650	1,644	951
Avg. interactions per user	12.94	14.48	10.33	8.62	7.67
Avg. words per query	6.65	4.82	3.39	5.90	4.21

Table 2: Statistics of dataset categories.

position-aware metric which assigns larger weights on higher positions. Following [16, 36], for each user, we sample 100 negative items based on their popularity—excluding the previously interacted items—and rank them together with the ground-truth item.

5.2 Baselines

We experiment with a variety of baseline methods ranging from (i) rather simple non-personalized methods to (ii) more sophisticated deep learning based methods for personalized item search, and to (iii) state-of-the-art temporal models adapted from the sequential recommendation literature. These methods are listed below:

POP$_i$: A simple statistical model that ranks items according to their popularity in the training split across all users.

Query only (Q): A query-only approach that ranks items solely based on their respective query embeddings ($\mathbf{E^q}$). We refer to this *non-personalized* approach as Q for simplicity.

SasRec (H) [19]: A position-based self-attention model [19] that ranks items solely based on the item history encodings ($\mathbf{E^h}$). Since it only leverages the user history, it is referred as H for simplicity.

SasRec+Q (HQ): A query-aware approach that ranks items based on the combined signals of **h**istory encodings and **q**uery embeddings, simply referred as HQ (Equation 6 without $\mathbf{E^{qh}}$).

The next set of baseline approaches target the modeling of query-aware history encoding ($\mathbf{E}^{qh}$), each extending the HQ variant mentioned above. To that end, we employ two strong approaches [1, 38] for personalized item search and adapt two recently proposed temporal models [23, 41] for sequential recommendation:

HQ w/*MultiHeadAttn* [38]: A benchmark approach that employs standard multi-head attention layer [38].

HQ w/*ZeroAttn* [1]: An approach that employs zero attention mechanism [1] to provide the model with the flexibility of ignoring the user history, allowing for more adaptive personalization.

HQ w/*TiSasRec* [23]: An adaptation of recently proposed TiSasRec [23] model. We combine *learnable* relative time interval embeddings with history encodings prior to computing attention weights between the user query and the user history.

HQ w/*Dejavu* [41]: An adaptation of Dejavu [41] model, which is the state-of-the-art for temporal sequential recommendation. Following [41], we employ a mix of time kernels to calibrate the influence (attention weights) of historical actions with respect to the search query, based on the temporal gaps between the two.

As for the proposed approach, we experiment with the following two variants introduced in Section 4.4:

HQ w/*MultiResAttn–O*: A multi-resolution attention variant where attention heads cover ***overlapping*** temporal ranges (Eq. 11).

HQ w/*MultiResAttn*: A multi-resolution attention variant where attention heads cover ***non-overlapping*** temporal ranges (Eq. 12).

5.3 Model Configurations

We implement the proposed methods and all the baseline approaches using Tensorflow. The following settings are applied to each method for fair comparisons. The parameters are learned using mini-batch SGD with Adam optimizer. The {batch size, learning rate, sequence length (N), latent dimensions (d, d_q)} are set to {128, 1e-3, 50, 60}, respectively. The vocab size of search queries is determined based on statistics shown in Table 2. The unit of time is set to *days* for all applicable methods. For the proposed variants, we set the Δ initialization parameters $\{a, b\}$ to {1,5} and the temperature scaling parameter τ to 5, which are observed to work well across all datasets. Furthermore, we apply a grid search over the following hyper-parameters on all datasets and applicable methods: number of self-attention blocks (L) in {1,2}, number of attention heads (h) in {1,2,3,4,5}, the vocab size of time embeddings (for *TiSasRecAttn*) in {256,512}, and the number of exponential decay time kernels (for *Dejavu*) in {3,5,10}. The remaining hyper-parameters for baseline approaches are set based on the suggestions made by the authors in their respective papers. Lastly, the best models are selected by early stopping based on the NDCG@10 score on the validation set, with a patience of 20 epochs. All results are reported on the test set.

5.4 Experimental Results

Table 3 shows the overall performance of baselines and our proposed method variants on all five dataset categories. In this section, unless otherwise stated, the relative performance measures between methods are computed with respect to the NDCG@3 metric.

Ablation results. The first set of four baselines—H, Q, HQ, HQ w/*MulHeadAttn*—helps to assess the incremental contributions of each model component presented in Section 4.3, while providing insights into the characteristics of each dataset category. We observe that the query signal alone is more valuable to our task than the user history signal for four of the categories, except Kindle Store. Combining the user history and query signals (see HQ baseline) leads to major improvements compared to the strongest signal of the two across all categories. The largest gain is 13.8% for Movies and TV, while the average gain is 6.4%. HQ w/*MulHeadAttn* baseline leads to further improvements with up to 8.9% relative gain compared to the HQ variant, while the average gain across all categories is 4.1%. This rather sophisticated approach serves as a strong baseline, granted it does not take the temporal aspect of user interactions into account. These results demonstrate the importance of query-aware history summarization for personalized item search and motivate us to investigate further gains when the temporal aspect is considered. For the remaining, we drop the term 'HQ w/' in our referrals to the corresponding methods for simplicity.

Temporal component. Our results reveal a clear trend of approaches with temporal flavor outperforming others that purely rely on sequential patterns. Moreover, our proposed approach consistently achieves best performance across all categories and evaluation metrics. To put this in perspective, among the time-aware methods, our proposed approach outperforms *TiSaSRecAttn* by up to 4.9% and *Dejavu* by up to 6.5%. When compared to *MultiHeadAttn*, we achieve up to 12.9% improvement on ranking performance (*MultiResAttn-O* on Kindle Store). When our best performing variant for each dataset is considered, they collectively provide 6.9% improvement on average compared to *MultiHeadAttn*, which is

Datasets	Metrics	Baselines							Ours	
		POP_i	SasRec (H)	Query only (Q)	SasRec+Q (HQ)	HQ w/ *MultiHeadAttn*	HQ w/ *ZeroAttn*	HQ w/ *TiSasRec*	HQ w/ *Dejavu*	HQ w/ *MultiResAttn*	HQ w/ *MultiResAttn–O*
Home and Kitchen	Hit@3	0.018	0.117	0.553	0.572	0.578	0.583	0.582	0.594	0.584	**0.601**
	Hit@10	0.067	0.227	0.688	0.715	0.713	0.725	0.720	**0.733**	0.717	0.730
	NDCG@3	0.013	0.096	0.471	0.487	0.497	0.505	0.499	0.508	0.506	**0.519**
	NDCG@10	0.031	0.134	0.521	0.540	0.547	0.557	0.550	0.559	0.555	**0.564**
Kindle Store	Hit@3	0.037	0.484	0.225	0.513	0.542	0.574	0.579	0.572	0.585	**0.596**
	Hit@10	0.107	0.668	0.405	0.728	0.742	0.766	0.764	0.773	0.777	**0.785**
	NDCG@3	0.026	0.408	0.175	0.422	0.447	0.482	0.481	0.474	0.490	**0.505**
	NDCG@10	0.051	0.475	0.239	0.501	0.520	0.552	0.550	0.548	0.561	**0.573**
Movies and TV	Hit@3	0.035	0.290	0.522	0.583	0.625	0.643	0.639	0.636	**0.663**	0.655
	Hit@10	0.118	0.447	0.778	0.815	0.832	0.842	0.847	0.846	**0.852**	0.848
	NDCG@3	0.025	0.244	0.420	0.478	0.521	0.538	0.536	0.533	**0.556**	0.553
	NDCG@10	0.053	0.301	0.514	0.564	0.597	0.611	0.614	0.610	**0.627**	0.623
Pet Supplies	Hit@3	0.022	0.236	0.528	0.566	0.575	0.579	0.591	0.581	0.592	**0.602**
	Hit@10	0.074	0.392	0.714	0.753	0.759	0.776	0.777	0.766	0.771	**0.783**
	NDCG@3	0.015	0.201	0.436	0.471	0.480	0.483	0.496	0.484	0.497	**0.506**
	NDCG@10	0.034	0.256	0.505	0.539	0.548	0.556	0.564	0.553	0.563	**0.573**
Grocery and Food	Hit@3	0.019	0.240	0.675	0.695	0.711	0.720	0.713	0.719	0.728	**0.734**
	Hit@10	0.073	0.354	0.842	0.859	0.865	0.875	0.874	0.872	0.880	**0.883**
	NDCG@3	0.014	0.205	0.570	0.591	0.601	0.619	0.610	0.616	0.625	**0.631**
	NDCG@10	0.033	0.245	0.631	0.652	0.657	0.676	0.671	0.673	0.681	**0.686**

Table 3: The ranking performance of baseline and proposed approaches on all five categories. The best performance is highlighted in boldface, while the second best performance is underlined. Results show that our proposed variants consistently outperform the baselines.

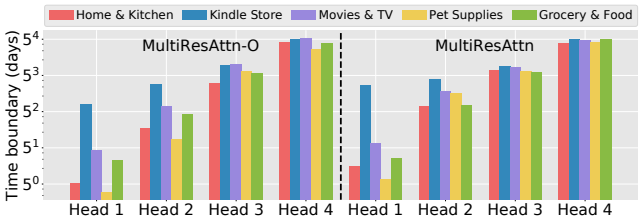


Figure 3: Learned time boundaries (Δs) with two proposed variants: MultiResAttn-*O* (left) and MultiResAttn (right), both with $h = 4$. Time boundaries are plotted on $\log_5$ scale.

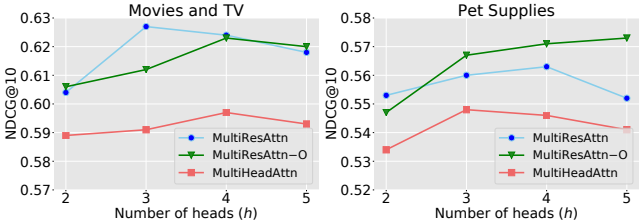


Figure 4: Effect of attention head count on ranking performance.

more than two times the average improvements achieved by *TiSasRecAttn* (3.1%) and *Dejavu* (2.7%) baselines across all datasets. Despite ignoring the temporal signal, *ZeroAttn* overall shows comparable performance to *TiSasRecAttn* and *Dejavu*. Furthermore, both proposed variants also outperform *ZeroAttn* in every comparison.

Between the two proposed variants, *MultiResAttn–O* performs the best for four categories. Movies and TV is the only category where *MultiResAttn* variant achieves a slightly higher ranking. To further investigate the potential motives behind our findings, we take a closer look into the temporal resolutions captured by our proposed variants. Figure 3 plots the time boundaries learned by both variants (with h=4) across all categories. We observe higher variations in the learned boundaries for attention heads covering the most recent history (*e.g.*, Heads 1 and 2). In particular, the first time boundary ranges from less than a day (for Pet Supplies) to over a month (for Kindle Store). Moreover, the non-overlapping variant (right) tends to learn slightly longer temporal spans compared to the overlapping variant (left). We conclude that different categories have varying temporal dynamics and densities, and our approach can adaptively recognize such temporal differences found in data.

Sensitivity analysis. Figure 4 shows the performance of proposed variants based on the number of attention heads (h). We also include the *MultiHeadAttn* baseline in our analysis for better comparison. For the Movies and TV, the highest score is obtained by the *MultiResAttn* variant with h=3, suggesting that the temporal dependencies are better captured across non-overlapping time spans. On the other hand, the Pet Supplies category favors the *MultiResAttn–O* variant with larger h, implying that the temporal dependencies reach gradually longer time spans that overlap, presumably due to the recurring user needs for this particular category.

6 CONCLUSION

We propose a Multi-Resolution Attention model for personalized item search. The key component of our architecture is the query-aware history encoding layer, which enables our method to exploit higher-order temporal dependencies between users' search queries and item history. This is achieved by a novel attention module consisting of multiple attention heads, each assigned to recognize and capture users' interests within designated temporal resolutions. The proposed method comes in two variants (*overlapping* and *non-overlapping*) to accommodate different temporal densities of real-world data. Both proposed variants are thoroughly examined by experiments using a large real-world dataset with five different item category domains. Our findings not only demonstrate the efficacy of Multi-Resolution Attention but also provide insights into the varying temporal dynamics captured across different domains.

REFERENCES

[1] Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of ACM CIKM'19*. 379–388.

[2] 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 ACM SIGIR'17*. 645–654.

[3] Ashton Anderson et al. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of TheWebConf'20*. 2155–2165.

[4] Keping Bi, Qingyao Ai, and W Bruce Croft. 2021. Learning a Fine-Grained Review-based Transformer Model for Personalized Product Search. In *Proceedings of ACM SIGIR'21*. 123–132.

[5] Xuxiao Bu, Jihua Zhu, Xueming Qian, and Member IEEE. 2020. Personalized product search based on user transaction history and hypergraph learning. *Multimedia Tools and Applications* 79 (2020), 22157–22175.

[6] Jean Burgess and Joshua Green. 2018. *YouTube: Online video and participatory culture*. John Wiley & Sons.

[7] Pedro G Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction* 24, 1 (2014), 67–119.

[8] Xu Chen et al. 2018. Sequential recommendation with user memory networks. In *Proceedings of ACM WSDM'18*. 108–116.

[9] Chung-Cheng Chiu et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *ICASSP'18*. IEEE, 4774–4778.

[10] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of ACM RecSys'16*. 191–198.

[11] Jacob Devlin et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Chantat Eksombatchai et al. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of TheWebConf'18*.

[13] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of ACM SIGKDD'18*. 311–320.

[14] Yangyang Guo et al. 2019. Attentive long short-term preference modeling for personalized product search. *ACM TOIS'19* 37, 2 (2019), 1–27.

[15] Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. 2020. Query reformulation in E-commerce search. In *Proceedings of ACM SIGIR'20*.

[16] Jin Huang et al. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of ACM SIGIR'18*. 505–514.

[17] Wendi Ji et al. 2020. Sequential Recommender via Time-aware Attentive Memory Network. In *Proceedings of ACM CIKM'20*. 565–574.

[18] Jyun-Yu Jiang, Tao Wu, Georgios Roumpos, Heng-Tze Cheng, Xinyang Yi, Ed Chi, Harish Ganapathy, Nitin Jindal, Pei Cao, and Wei Wang. 2020. End-to-End Deep Attentive Personalized Item Retrieval for Online Content-sharing Platforms. In *Proceedings of TheWebConf'20*. 2870–2877.

[19] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM'18*. IEEE, 197–206.

[20] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings ACM SIGKDD'09*. 447–456.

[21] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *Proceedings of ACM SIGIR'18*. 95–104.

[22] Sudarshan Lamkhede and Sudeep Das. 2019. Challenges in search on streaming services: netflix case study. In *Proceedings of ACM SIGIR'19*. 1371–1374.

[23] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *Proceedings of WSDM'20*. 322–330.

[24] Yang Li, Nan Du, and Samy Bengio. 2017. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065* (2017).

[25] Shang Liu, Wanli Gu, Gao Cong, and Fuzheng Zhang. 2020. Structural Relationship Representation Learning with Graph Embedding for Personalized Product Search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 915–924.

[26] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of ACM SIGKDD'19*. 825–833.

[27] Nicolaas Matthijs and Filip Radlinski. 2011. Personalizing web search using long term browsing history. In *Proceedings of ACM WSDM'11*. 25–34.

[28] Hongyuan Mei and Jason M Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. *Advances in Neural Information Processing Systems* 30 (2017).

[29] Wendy W Moe. 2003. Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. *Journal of consumer psychology* 13, 1-2 (2003), 29–39.

[30] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of EMNLP-IJCNLP'19*. 188–197.

[31] Jennifer Rowley. 2000. Product search in e-shopping: a review and research propositions. *Journal of consumer marketing* (2000).

[32] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of NAACL'18*. 464–468.

[33] Parikshit Sondhi et al. 2018. A Taxonomy of Queries for E-commerce Search. In *Proceedings of ACM SIGIR'18*. 1245–1248.

[34] David Sontag, Kevyn Collins-Thompson, Paul N Bennett, Ryen W White, Susan Dumais, and Bodo Billerbeck. 2012. Probabilistic models for personalizing web search. In *Proceedings of ACM WSDM'12*. 433–442.

[35] Ning Su, Jiyin He, Yiqun Liu, Min Zhang, and Shaoping Ma. 2018. User intent, behaviour, and perceived satisfaction in product search. In *Proceedings of ACM WSDM'18*. 547–555.

[36] Fei Sun et al. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of ACM CIKM'19*. 1441–1450.

[37] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *CIKM'16*. 165–174.

[38] Ashish Vaswani et al. 2017. Attention is all you need. In *NeurIPS'17*.

[39] Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. 2014. Collaborative personalized twitter search with topic-language models. In *Proceedings of ACM SIGIR'14*. 53–62.

[40] Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of ACM SIGIR'15*. 363–372.

[41] Jibang Wu, Renqin Cai, and Hongning Wang. 2020. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*. 2199–2209.

[42] Teng Xiao, Jiaxin Ren, Zaiqiao Meng, Huan Sun, and Shangsong Liang. 2019. Dynamic bayesian metric learning for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1693–1702.

[43] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2019. Self-attention with functional time representation learning. In *NeurIPS'19*.

[44] Yu Zhu et al. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM.. In *IJCAI'17*, Vol. 17. 3602–3608.