Sequential Modeling of Hierarchical User Intention and Preference for Next-item Recommendation

Nengjun Zhu*
Shanghai Jiao Tong University,
Shanghai, CN
zhu_nj@sjtu.edu.cn

Jian Cao†
Shanghai Jiao Tong University
Shanghai, CN
cao-jian@sjtu.edu.cn

Yanchi Liu
Rutgers University
New Jersey, USA
yanchi.liu@rutgers.edu

Yang Yang
Nanjing University
Nanjing, CN
yangy@lamda.nju.edu.cn

Haochao Ying
Zhejiang University
Shanghai, CN
haochaoying@zju.edu.cn

Hui Xiong
Rutgers University
New Jersey, USA
hxiong@rutgers.edu

ABSTRACT

The next-item recommendation has attracted great research interests with both static and dynamic users' preferences considered. Existing approaches typically utilize user-item binary relations, and assume a flat preference distribution over items for each user. However, this assumption neglects the hierarchical discrimination between user intentions and user preferences, causing the methods have limited capacity to depict intention-specific preference. In fact, a consumer's purchasing behavior involves a natural sequential process, i.e., he/she first has an intention to buy one type of items, followed by choosing a specific item according to his/her preference under this intention. To this end, we propose a novel key-array memory network (KA-MemNN), which takes both user intentions and preferences into account for next-item recommendation. Specifically, the user behavioral intention tendency is determined through key addressing. Further, each array outputs an intention-specific preference representation of a user. Then, the degree of user's behavioral intention tendency and intention-specific preference representation are combined to form a hierarchical representation of a user. This representation is further utilized to replace the static profile of users in traditional matrix factorization for the purposes of reasoning. The experimental results on real-world data demonstrate the advantages of our approach over state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Representation learning; memory networks; intention modeling

*This work was done while the author was visiting Rutgers University.
†Corresponding author.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371840

ACM Reference Format:
Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. 2020. Sequential Modeling of Hierarchical User Intention and Preference for Next-item Recommendation. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371840

1 INTRODUCTION

Recommender systems (RSs) can help users find their preferred items from a vast number of choices, thus can enhance user experiences. Moreover, they allow platforms, such as Amazon, Tmall, and Uber, to increase user engagement and derive new business value. In fact, user behavior data (e.g., click, purchase, and check-in) is growing exponentially and has been sequentially collected by the platforms as a knowledge source to be fed into RSs. For instance, 62 million user trips were accumulated in July 2016 by Uber, and more than 10 billion check-ins were generated by over 50 million users at Foursquare [34]. As a result, next-item RSs, which recommend the next item to users according to their interests, are becoming a hot research topic in recent years [10, 18, 30, 33].

Along this line, efforts have been made to combine information on users' previously (also called long-term) and recently (also called short-term) accessed items in different ways. This combination has been proven effective for next-item recommendation tasks, since it can depict the static and dynamic preferences of a user simultaneously [7]. Existing next-item recommendation approaches usually model users' preferences by utilizing user-item binary relations and basing item sequences. As a result, they have a limited modeling capacity in capturing sequential characteristics over context/side information, such as the sequences of item categories [12]. Besides, existing methods assume a flat preference distribution over items for each user and neglects the hierarchical distinction between user intentions and user preferences, which makes it hard to fully exploit users' structural decision patterns for user preferences learning.

To address the above issues, we introduce the concept of **user intention** and utilize item category information to assist in modelling user preferences. Although user intention modeling has been a research topic for a long time in marketing [22], it is hard to transfer these modeling methods to RSs. Thus, instead of trying to investigate the behavioral intentions, we use the category of items

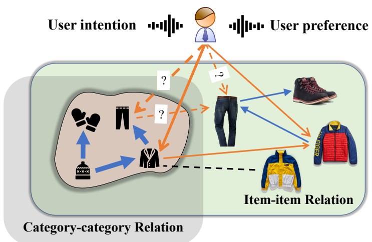


Figure 1: An illustration of the user's hierarchical model, which extends user-item binary relations to user-intention-item triadic relations, where the intention is indicated by the product category to be bought in this case, and the intention-item relations imply user preferences.

the user desires to represent them directly. Some examples are listed as follows: different product categories such as clothing and electronics, different service types such as dining and entertainment, and different area types such as tourism region and commercial district. Despite its simplicity, category information provides an intuitive way to represent users' behavioral intentions.

Generally, a user makes decisions with consideration of both his/her intentions and preferences [18], and decision-making involves a natural sequential process. For instance, to purchase an item or visit a point-of-interest (POI), the decision process can be roughly divided into two steps. The first step is related to his intentions, i.e., a high-level selection, and the second step relies on his preferences to make a specific selection. A specific example of this circumstance would be, after a person purchases a pair of trousers, she may also have an intention to select one top which matches the trousers to complete the whole look. Then, she would consider which top is preferred for purchase. This decision process can be based on user-intention-item triadic relations, as shown in Figure 1. Compared to user-item binary relations, the user-intention-item triadic relations can describe more elaborate decision processes of users because the flat user preference distribution over items is refined into specific categories/intentions.

To this end, in this paper, we propose a novel model, namely **K**ey-**A**rray **Mem**ory **N**eural **N**etwork (KA-MemNN), in which user behavioral intentions and user preferences are combined to form hierarchical and well-rounded representations of users. Specifically, we first generate user-specific category/intention embeddings through an attention network, which is fed with related items and guided by the target user. Then, we evaluate users' intention tendencies through category addressing, followed by an aggregation of the items retrieved from a memory bank (MB) to summarize the intention-specific preference of users. All intention-specific preference representations are further summarized to output a more well-rounded one. This representation is adjusted according to the long- and short-term MBs. Finally, to learn the parameters, a pairwise loss function is applied to conventional matrix factorization (MF), but the static representations of users are replaced with the outputs of KA-MemNN. Our contributions are summarized as follows: *1)* We utilize hierarchical and well-rounded representations of users based on user-intention-item triadic relations for next-item recommendation. *2)* We design a memory network, i.e., KA-MemNN, to model user behavioral intentions and preferences based on both long- and short-term information. *3)* We perform experiments on real-world data, and the results show that our model consistently outperforms state-of-the-art methods in terms of AUC (area under the curve) and Recall evaluation metrics.

2 RELATED WORK

Our KA-MemNN model addresses the next-item recommendation problems, and it conducts representation learning for users through a memory neural network (MemNN). Therefore, our method is related to state-of-the-art next-item approaches, i.e., Markov chains (MCs) and neural networks (NNs) including representation learning and MemNNs.

MCs have become classic tools to handle users' sequentially generated behaviors through sequential pattern mining [24, 32] and transition modeling [21, 36]. For instance, based on the insight that frequent patterns can be utilized to predict the next items users will access, [32] emphasizes personalities in their model by discovering user-specific frequent patterns. In contrast, [21] proposed FPMC which directly utilizes matrix factorization (MF) to capture users' general tastes and combines MF with a first-order MC to predict the next item based on the recent items by learning a transition graph over the items. To alleviate problems caused by sparsity issues and long-tailed distributions, [7] proposed FOSSIL, which further integrates FPMC with a similarity-based method, i.e. FISM [14]. These methods attempt to combine user long-term and short-term preferences. Nevertheless, the sequential factors and personalities are not adequately utilized to mine item-item relations, which may impair the recommendation performance [27].

In recent years, researchers have begun to embrace NNs in RSs since deep learning has achieved tremendous success in many tasks [5, 9, 23]. For instance, [8] and [29] designed B-Interaction and attentional pooling layers respectively, to automatically learn second-order feature interaction based on traditional factorization machine technology. [28] and [16] employed recurrent neural networks (RNNs) to conduct user long-time visit prediction and real-time location prediction by exploring trajectory data. However, items in a session may not follow a rigidly sequential order in many real scenarios, e.g., items in a shopping cart, where RNN is not applicable. Beyond that, hierarchical representation learning has attracted immense attention. These methods usually combine users' previously and recently accessed items differently to learn high-level representations. For instance, [10] and [27] both employed a two-layer structure to construct a hybrid representation over users and items. [34] argued that the weights of different components should not be fixed and thus proposed an attention-based SHAN (the sequential hierarchical attention network). Our work follows this pipeline and constructs hierarchical representations for users, but the main difference being that we utilize two types of sessions simultaneously, (i.e. item and category sessions), by considering user-category-item triadic relations.

MemNNs have been proven useful for a variety of document reading and question answering (QA) tasks, such as end-end memory neural networks (EE-MemNNs) [25] and key-value memory neural networks (KV-MemNNs) [19]. The memory component, i.e., the memory bank, of MemNNs can increase modeling capacity as well as generate a more informative representation of historical knowledge to track long-term dependencies. However, the applications of MemNNs in RSs usually base item sequences [2]. Limited work utilizes MemNNs with taxonomy information. Recently, a non-session approach CMN (a collaborative memory network) [4] was proposed. CMN takes the users who have co-click/co-visitation behaviors with the target user on a specific item as the neighborhoods. These neighborhoods are stored into internal memory to accumulate local neighborhood-based information. However, CMN learns static representations of users, and thus, it is inadequate for users' dynamic preference modeling. Our method is also related to TMRN [12] in which memory neural networks have been first combined with taxonomy information. However, TMRN attempts to encode the hierarchical category semantics without incorporating item information, which cannot fully exploit the link information from user-category-item triadic relations.

3 THE PROPOSED METHOD

3.1 Problem Formulation

Let $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{G}$ represent the sets of users, items, and categories (categories and intentions have no distinction in this paper), respectively. $\mathcal{I}_p \subseteq \mathcal{I}$ is an item subset which records the items belonging to the category $p \in \mathcal{G}$. An item $a \in \mathcal{I}$ may belong to multiple categories which are denoted by the category subset $\mathcal{G}_a \subseteq \mathcal{G}$. A user $i \in \mathcal{U}$ has two related behavior sequences: item sessions and category sessions. $\mathcal{S}^i = \{\mathcal{S}^i_1, \mathcal{S}^i_2, \cdots, \mathcal{S}^i_t | \mathcal{S}^i_t \subseteq \mathcal{I}\}$ denotes the item sessions and $\mathcal{A}^i = \{\mathcal{A}^i_1, \mathcal{A}^i_2, \cdots, \mathcal{A}^i_t | \mathcal{A}^i_t \subseteq \mathcal{G}\}$ denotes the category sessions, where $\mathcal{A}^i_t = \cup_{a \in \mathcal{S}^i_t} \mathcal{G}_a$, and $\mathcal{S}^i_t$ (resp. $\mathcal{A}^i_t$) is the t-th item session (resp. category session) following the timestamps. For a fixed time t, the session $\mathcal{A}^i_s = \mathcal{A}^i_t$ and $\mathcal{S}^i_s = \mathcal{S}^i_t$ can reflect user i's short-term intentions and preferences, while the sessions before the timestamp t denoted by $\mathcal{A}^i_l = \cup_{T \in [1,t-1]} \mathcal{A}^i_T$ and $\mathcal{S}^i_l = \cup_{T \in [1,t-1]} \mathcal{S}^i_T$ can reflect user i's long-term intentions and preferences. Formally, given a user and two item sessions $\mathcal{S}^i_l$ and $\mathcal{S}^i_s$ along with two corresponding category sessions $\mathcal{A}^i_l$ and $\mathcal{A}^i_s$, we aim to recommend the next items through mining user-intention-item triadic relations.

3.2 Overview

Our KA-MemNN shown as Figure 2 has three core components: the attention neural network (ANN), the key-array memory bank (KA-MB), and the hierarchical weighting unit (HWU). ANN takes the target user and related visited items as input to generate user-specific category embeddings. The HWU learns user-category-item relations. The KA-MB, which organizes a user's past behaviors, supports the HWU as an input. More specifically, two KA-MBs, i.e., the long- and short-term KA-MBs, organize the pair of $(\mathcal{S}^i_l, \mathcal{A}^i_l)$ and the pair of $(\mathcal{S}^i_s, \mathcal{A}^i_s)$, respectively. By querying and reading the corresponding KA-MB, the HWU in each hop generates hybrid embeddings for users. Then, through two hops of accumulation, we obtain users' final preference representations. Theoretically, we can construct one KA-MB for each item session in $\mathcal{S}^i$ along with its corresponding category session in $\mathcal{A}^i$. In practice however, due to the sparsity of user's behavior data and to simplify the process of investigating users' sequential behaviors, we only construct the aforementioned KA-MBs. Next, we introduce each part of our model in detail.

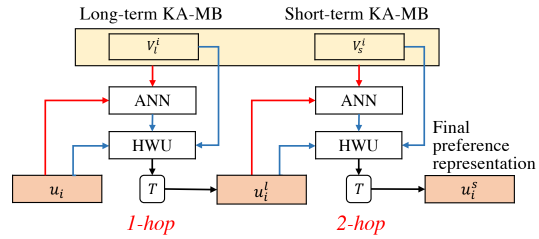


Figure 2: The overview of the network architecture.

3.3 General Embedding Construction

Since the basic IDs of users and items (i.e., one-hot representations) have a very restricted representation capability, we first employ two separate fully connected layers with the weight matrices $\boldsymbol{U} \in \mathbb{R}^{N \times K}$ and $\boldsymbol{V} \in \mathbb{R}^{M \times K}$ to construct two continuous low-dimensional embeddings of users and items, respectively, where N and M are the numbers of users and items, while K is the dimensionality of vectors. Here, we only feed the fully connected layer network with one-hot representations of users (or items), and then the network outputs the corresponding embeddings denoted as the bold lowercase $\boldsymbol{u}_i \in \mathbb{R}^K$ for user i, i.e., the i-th row of $\boldsymbol{U}$ (or $\boldsymbol{v}_a \in \mathbb{R}^K$ for item a, i.e., the a-th row of $\boldsymbol{V}$). Both $\boldsymbol{u}_i$ and $\boldsymbol{v}_a$ are reshaped into a column vector.

We call these $\boldsymbol{u}_i$ and $\boldsymbol{v}_a$ as the general/static embeddings of users and items since they only reflect users' general preferences and do not differentiate between users' new and old behaviors. This is the critical limitation of non-session RSs, such as BPR [20] and CMN [4], and we address this problem by introducing the session-based KA-MB, which will be discussed later.

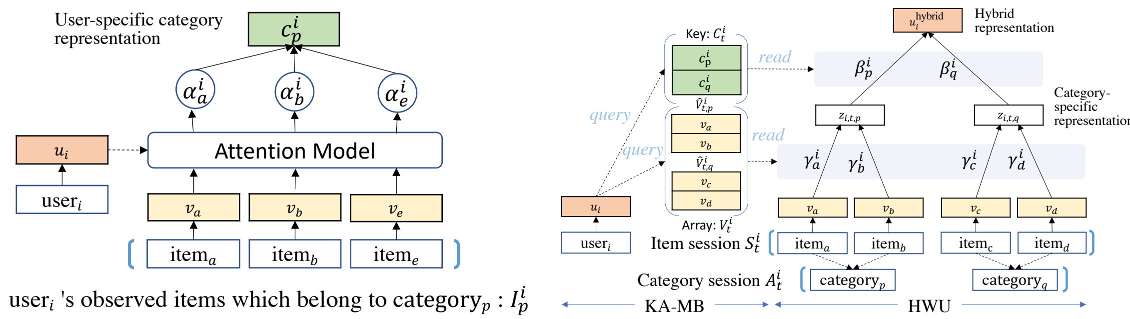
3.4 Semantic Category Embedding Construction

Instead of constructing the embeddings of categories based on primary IDs, in this paper, we generate a more informative representation using taxonomy information by employing an attention network, which is based on the users' and items' embeddings. The attention mechanism shares the intuition that humans pay attention to only the most important parts of the target, and has been successfully applied in many applications such as machine translation [1], image captioning [26], and textual content modeling [31]. Recently, some RSs incorporate a neural attention layer in their models to learn representations with limited knowledge of the whole context [11, 35]. However, these learned representations have no relations with the categories of items, and thus cannot draw user-specific category embeddings based on items.

In Figure 3a, we construct semantic category embeddings for each user by designing ANN as follows:

$$\boldsymbol{c}^i_p = ANN(\boldsymbol{u}_i, \{\boldsymbol{v}_a | a \in \mathcal{I}^i_p\}) \qquad (1)$$

Next, we implement ANN by introducing an attention function and an aggregation function.



(a) ANN. **(b) KA-MB & HWU.**

Figure 3: The components of KA-MemNN.

Given a target representation $\boldsymbol{u}_i \in \mathbb{R}^K$, an attention function f maps $|\mathcal{I}_p^i|$ source representations with dimension K, i.e., $\{\boldsymbol{v}_a | a \in \mathcal{I}_p^i\}$, to a weight vector with dimension $|\mathcal{I}_p^i|$, i.e., $\boldsymbol{\alpha}^i \in \mathbb{R}^{|\mathcal{I}_p^i|}$, where $\mathcal{I}_p^i$ records user i's observed items which belong to category p. Each weight α_a^i in $\boldsymbol{\alpha}^i$ reflects the importance of $\boldsymbol{v}_a \in \mathbb{R}^K$ for $\boldsymbol{u}_i$. Then, an aggregation function g is adopted to generate a summarized representation $\boldsymbol{c}_p^i \in \mathbb{R}^K$.

$$(\alpha_1^i, \alpha_2^i, \cdots, \alpha_{|\mathcal{I}_p^i|}^i) = f(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{|\mathcal{I}_p^i|} | \boldsymbol{u}_i) \tag{2}$$

$$\boldsymbol{c}_p^i = g(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{|\mathcal{I}_p^i|} | \alpha_1^i, \alpha_2^i, \cdots, \alpha_{|\mathcal{I}_p^i|}^i) \tag{3}$$

We implement f by a neural network as follows:

$$\boldsymbol{h}_a = \phi(\boldsymbol{W}\boldsymbol{v}_a + \boldsymbol{b}_0), \tag{4}$$

$$\alpha_a^i = \frac{exp(\boldsymbol{u}_i^\top \boldsymbol{h}_a)}{\sum_{a \in \mathcal{I}_p^i} exp(\boldsymbol{u}_i^\top \boldsymbol{h}_a)} \tag{5}$$

where $\boldsymbol{W} \in \mathbb{R}^{K \times K}$ and $\boldsymbol{b}_0 \in \mathbb{R}^K$ are the model parameters. We first feed the dense embedding vector of items through a multi-layer perceptron (MLP) [6] to get the hidden representation $\boldsymbol{h}_a$. Then, $\boldsymbol{h}_a$ is utilized with the target representation $\boldsymbol{u}_i$ to generate the weights. $\phi(\cdot)$ is the activation function and we utilize two RELU functions[17] to enhance the nonlinear capability and filter out some trivial information. Finally, we generate the user-specific representation of categories as a sum of the item embeddings weighted by the importance scores as follows:

$$\boldsymbol{c}_p^i = \sum_{a \in \mathcal{I}_p^i} \alpha_a^i \cdot \boldsymbol{v}_a \tag{6}$$

$\boldsymbol{c}_p^i$ incorporates the information of all observed items belonging to category p of user i.

3.5 KA-MB & HWU

After obtaining the embeddings of users, items, and categories, we can schedule them into memory to extract vital information. The static representation $\boldsymbol{u}_i$ is often utilized to make recommendations in conventional approaches. However, in addition to the immutable characteristics in users' preferences, there are many influential and dynamic factors affecting users' decisions, such as fashion trends and promotion activities. Thus, we should update users' representations based on the static representation $\boldsymbol{u}_i$ and the user's recently accessed categories and items.

We define two KA-MBs as two pairs of memory slots, i.e., $(\boldsymbol{C}_t^i, \boldsymbol{V}_t^i)$, $t \in \{l, s\}$, to store user i's long- and short-term sessions:

$$\begin{aligned} \boldsymbol{C}_t^i &= [\boldsymbol{c}_1^i, \boldsymbol{c}_2^i, \cdots, \boldsymbol{c}_p^i, \cdots]^\top, p \in \mathcal{A}_t^i \\ \boldsymbol{V}_t^i &= [\hat{\boldsymbol{V}}_{t,1}^i, \hat{\boldsymbol{V}}_{t,2}^i, \cdots, \hat{\boldsymbol{V}}_{t,p}^i, \cdots], p \in \mathcal{A}_t^i \\ \hat{\boldsymbol{V}}_{t,p}^i &= [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_a, \cdots]^\top, a \in \mathcal{I}_p \cap \mathcal{S}_t^i \end{aligned} \tag{7}$$

where $\boldsymbol{C}_t^i \in \mathbb{R}^{|\mathcal{A}_t^i| \times K}$ and $\hat{\boldsymbol{V}}_{t,p}^i \in \mathbb{R}^{|\mathcal{I}_p \cap \mathcal{S}_t^i| \times K}$. Then, we can conduct operation functions, such as *query* and *read*, on these matrices.

The relation between categories and items has a hierarchical structure, which is analogous to users' sequential decision-making process. Therefore, we design HWU to aggregate the information from observed items differently according to taxonomy, shown as Figure 3b. HWU is an extension of key-value memory networks (KV-MemNNs) [19]. For a standard KV-MemNN, assume we have a query vector $\boldsymbol{u}_i \in \mathbb{R}^K$, then the task is to reconstruct $\boldsymbol{u}_i$ to a more informative representation by retrieving $|\mathcal{A}_t^i|$ key-value pairs: $\boldsymbol{C}_t^i \in \mathbb{R}^{|\mathcal{A}_t^i| \times K}$ and $\tilde{\boldsymbol{V}}_t^i \in \mathbb{R}^{|\mathcal{A}_t^i| \times K}$ from the memory bank (MB).

$$kv(\boldsymbol{u}_i, \boldsymbol{C}_t^i, \tilde{\boldsymbol{V}}_t^i | \omega) = \omega(\boldsymbol{C}_t^i \boldsymbol{u}_i)^\top \tilde{\boldsymbol{V}}_t^i \tag{8}$$

where $kv(\cdot)$ is a KV-MemeNN function, and $\omega(\boldsymbol{C}_t^i \boldsymbol{u}_i) \in \mathbb{R}^{|\mathcal{A}_t^i|}$ is a weight vector, or affinity vector describing the similarity between the target *query* and the corpus. $\omega(\cdot)$ is usually an activation function like RELU and Softmax. Then, the $\boldsymbol{u}_i$ is reconstructed by a weighted sum of $\tilde{\boldsymbol{V}}_t^i$, where each *value* gets the weight from the similarity between its corresponding *key* and the target *query*. When the *key* and *value* become the same, i.e., $\boldsymbol{C}_t^i = \tilde{\boldsymbol{V}}_t^i$, the KV-MemNN degrades to the end-end memory network (EE-MemNN) [25].

Instead, our HWU generalizes KV-MemNN by replacing the single *value* with a variable-length *array* which records multiple *values*, i.e., the $\tilde{\boldsymbol{V}}_t^i$ is replaced by $\boldsymbol{V}_t^i$. The *values* in the same array have the same property which makes them have a shared *key*, such as all of them belong to the same category. We refer to this data structure as having a key-array data type. Formally, given the *query* $\boldsymbol{u}_i$, the *key* $\boldsymbol{C}_t^i$, and the *array* $\boldsymbol{V}_t^i$, HWU reconstructs a hybrid representation $\boldsymbol{u}_i^{hybrid}$ of $\boldsymbol{u}_i$ as follows:

$$\boldsymbol{u}_i^{hybrid} = HWU(\boldsymbol{u}_i, \boldsymbol{C}_t^i, \boldsymbol{V}_t^i) \tag{9}$$

Next, we implement HWU hierarchically by introducing two operations on KA-MBs, i.e., array aggregation and key addressing.

Array Aggregation: The purpose of this step is to generate category-specific representations shown as $\boldsymbol{z}_{i,t,p}$ in Figure 3b. The aggregation operation is a set function, which maps a block of a matrix, i.e., $\hat{\boldsymbol{V}}_{t,p}^i$, to one single vector, i.e., $\boldsymbol{z}_{i,t,p}$. In this work, we study three typical aggregation operations on each matrix block $\hat{\boldsymbol{V}}_{t,p}^i$ as follows.

∗ *average pooling:* It is a linear and nondistinctive combination of the *values* in the *array.*

$$\boldsymbol{z}_{i,t,p} = average(\hat{\boldsymbol{V}}_{t,p}^i) \tag{10}$$

where $\boldsymbol{z}_{i,t,p} \in \mathbb{R}^K$ and the function $average(\cdot)$ computes the mean over all row vectors. In such a case, all weights in $\boldsymbol{\gamma}^i \in \mathbb{R}^{|\mathcal{I}_p \cap \mathcal{S}_t^i|}$ shown in Figure 3b are equal to the value of $\frac{1}{|\mathcal{I}_p \cap \mathcal{S}_t^i|}$.

∗ *weighted average pooling:* Similar to average pooling, however the impact from each item is distinguished by the linking strength between the item and the current status of the user. Weighted average pooling is set as the default aggregation operation in our model unless otherwise stated. This is because it achieves the best performance over other pooling approaches.

$$\boldsymbol{\gamma}^i = Softmax(\hat{\boldsymbol{V}}_{t,p}^i \boldsymbol{u}_i) \tag{11}$$
$$\boldsymbol{z}_{i,t,p} = \boldsymbol{\gamma}^{i\top} \cdot \hat{\boldsymbol{V}}_{t,p}^i$$

∗ *max pooling:* It extracts the maximum value of each dimension over all elements' embeddings read from the KA-MB as the the value of the new vector.

$$\boldsymbol{z}_{i,t,p} = max(\hat{\boldsymbol{V}}_{t,p}^i) \tag{12}$$

where the function $max(\cdot)$ extracts the maximum value of each column vector.

Key Addressing: During key addressing, each category p in the $\mathcal{A}_t^i$ is assigned a relevance probability, e.g., β_p^i in Figure 3b, by comparing the target user's embedding to each category's embedding.

$$\boldsymbol{\beta}^i = Softmax(\boldsymbol{C}_t^i \boldsymbol{u}_i) \tag{13}$$

Combining with $\boldsymbol{z}_{i,t,p}$, we obtain a hybrid representation as follows:

$$\boldsymbol{u}_i^{hybrid} = \sum_{p \in \mathcal{A}_t^i} \beta_p^i \boldsymbol{z}_{i,t,p} \tag{14}$$

3.6 KA-MemNN

The framework of KA-MemNN is shown as Figure 2. Besides the KA-MBs and the HWU, we have two additional user-independent time-aware units shown as Ts in the figure just like [19]. These units are implemented as RELU functions with the parameters $\boldsymbol{R}_l \in \mathbb{R}^{K\times K}$ and $\boldsymbol{R}_s \in \mathbb{R}^{K\times K}$, and two corresponding bias vectors $\boldsymbol{b}_l \in \mathbb{R}^K$ and $\boldsymbol{b}_s \in \mathbb{R}^K$. Then, all operations are listed as follows:

$$\begin{aligned} \boldsymbol{u}_i^l &= HWU(RELU(\boldsymbol{R}_l \boldsymbol{u}_i + \boldsymbol{b}_l), \boldsymbol{C}_l^i, \boldsymbol{V}_l^i) \\ \boldsymbol{u}_i^s &= HWU(RELU(\boldsymbol{R}_s \boldsymbol{u}_i^l + \boldsymbol{b}_s), \boldsymbol{C}_s^i, \boldsymbol{V}_s^i) \end{aligned} \tag{15}$$

Algorithm 1: KA-MemNN

Input: User sessions $\mathcal{S}^i$ and $\mathcal{A}^i$, initial learning rate η, regularization λ, and dimension K.
Output: model parameters Θ.

Init Θ from Normal Distribution $\mathcal{N}(0, 0.01)$;.
repeat
  shuffle the set of observations $\{(i, \mathcal{S}_l^i, \mathcal{S}_s^i, a^+, a^-)\}$
  for *each observation* $(i, \mathcal{S}_l^i, \mathcal{S}_s^i, a^+, a^-)$ **do**
    compute $\boldsymbol{u}_i^s$ according to Eq. (15)
    compute $\hat{r}_{i,a^+}$ and $\hat{r}_{i,a^-}$ according to Eq. (16)
    update Θ by the gradient descent optimization
  end for
until convergence
return Θ

The $\boldsymbol{u}_i^s$ is the final representation that does not only contain the information of the user's general preference, but also has a dynamic response to users' current behaviors. Moreover, it is learned by uncovering the relations over the users, categories, and items.

3.7 Model Learning

After the final user representation has been learned, we further utilize it with item representations to predict the scores from a user to items as follows:

$$\hat{r}_{i,a} = \boldsymbol{u}_i^{s\top} \boldsymbol{v}_a \tag{16}$$

Next, we employ a pairwise loss function, which is introduced in [20], to train our model. We randomly select a positive example a^+ from the current visited session of user i, i.e., $a^+ \in \mathcal{S}_s^i$, and a negative example a^- from unvisited items which have a same category with the positive example, i.e., $a^- \in (\cup_{p \in \mathcal{G}_{a^+}} \mathcal{I}_p) \backslash (\mathcal{S}_l^i \cup \mathcal{S}_s^i)$. Then we define a pairwise order $\hat{r}_{i,a^+} > \hat{r}_{i,a^-}$ based on the assumption that users prefer observed item a^+ rather than unobserved item a^- when both of them belong to the same category. The final optimization function is as follows:

$$\begin{aligned} \arg\min_{\Theta} & \sum_{(i,\mathcal{S}_l^i,\mathcal{S}_s^i,a^+,a^-) \in \mathcal{D}} -\ln\sigma(\hat{r}_{i,a^+} - \hat{r}_{i,a^-}) \\ & + \lambda_{uv} \parallel \Theta_{uv} \parallel + \lambda_{rwb} \parallel \Theta_{rwb} \parallel \end{aligned} \tag{17}$$

where $\mathcal{D}$ is the set of examples, $\Theta_{uv} = \{\boldsymbol{U}, \boldsymbol{V}\}$ and $\Theta_{rwb} = \{\boldsymbol{R}_l, \boldsymbol{R}_s, \boldsymbol{b}_l, \boldsymbol{b}_s, \boldsymbol{W}, \boldsymbol{b}_0\}$ are the sets of model parameters, $\lambda = \{\lambda_{uv}, \lambda_{rwb}\}$ is the set of regularization parameters, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is a logistic function. The detailed learning algorithm is presented in Algorithm 1. Once the model has been learned, we recommend the items with the largest scores $\hat{r}_{i,a}$ to user i.

4 EXPERIMENTS

In this section, we empirically evaluate that our proposed approach outperforms start-of-the-art baselines as well as validate that the components in our model can improve the recommendation performances with respect to multiple metrics.

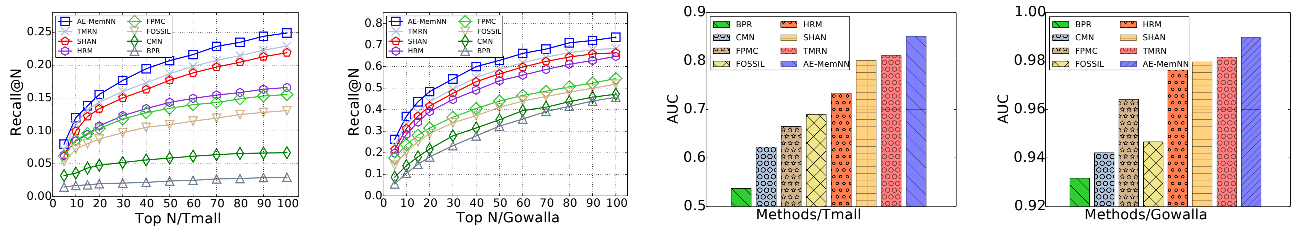


Figure 4: Performance Comparison of Methods for next-item recommendation according to item-based metrics.

Table 1: Statistics of datasets

Dataset	Tmall	Gowalla
#user	20,202	15,063
#item	24,774	11,897
#category / #region	769	4,544
avg. session length	2.72	3.02
#train session	70,895	128,250
#test session	4,040	3,012
user-item matrix density	0.039%	0.149%
user-intention matrix density	0.978%	0.251%

4.1 Experimental Setup

Datasets. Tmall dataset [10] and Gowalla dataset [3] are employed to conduct our experiments. Tmall, which is the IJCAI-15 competition dataset[1], records users' historical purchase behaviors on China's largest online shopping platform[2], while Gowalla[3] collects users' check-in information including the time and locations of check-ins. We choose purchase behaviors in Tmall and check-in behaviors in Gowalla to explore users' preferences. For both datasets, we extract the data generated in the last seven months and items which have been observed by more than 20 users to form the final datasets. Similar to [34], user behaviors in each day are treated as a session, while all singleton sessions, i.e., containing only one item are also excluded. We treat the last sessions of 20% of the users randomly selected as test sessions and randomly remove one item in each test session as the next item to be predicted. Then, all sessions, including already processed ones are split into long- and short-term sessions to train the model. The descriptive statistics of both datasets are summarized in Table 1. Note that the categories in Tmall and the region grids of size $1 \times 1km^2$ in Gowalla, which is divided according to the Geographic Information System (GIS) coordinates, are treated as behavioral intentions, respectively.

Baselines. We compare our model with the following baseline algorithms, including non-session recommendation, session-based recommendation, and hierarchical representation approaches. Models are tuned for best performance through tuning of parameters, such as the dimensionality K is set to 50, $\lambda_{uv} = \lambda_{rwb} = 0.001$ for Tmall dataset and $\lambda_{uv} = 0.05, \lambda_{rwb} = 0.001$ for Gowalla dataset. We utilize Adam [15] to optimize the training process, with initial learning rates are set to 0.05 for both datasets.

1) **BPR** [20]. A traditional latent factor CF (collaborative filtering) model which optimizes a pairwise loss function. *2)* **CMN** [4]. A non-session memory network, which takes users' neighborhoods as the values in the memory bank. We set the number of the hops to 2 since it achieves the best performance on the datasets. *3)* **FPMC** [21]. This method models user preferences by combining MF, which captures users' general preference and a first-order MC to predict the user' next action. *4)* **FOSSIL** [7]. This method integrates factored item similarity with MC to model a user's long- and short-term preferences. Note that we set μ_u and μ as single scalar since the length of each session is variable. *5)* **HRM** [27]. This method generates a hierarchical user representation to capture sequential information and general tastes. We use max pooling as the aggregation operation because this achieves the best result. *6)* **SHAN** [34]. This model employs two attention networks to mine users' long- and short-term preferences. *7)* **TMRN** [12]. also a memory neural network. It combines taxonomy information and encodes the hierarchical category semantics without incorporating item information. In this paper, we set the number of hops to one as we do not have detailed taxonomy information.

Metrics. We employ two commonly used metrics, Recall and AUC, to evaluate the performance of methods for next-item recommendation. The first metric evaluates the fraction of ground truth items that have been retrieved over the total amount of ground truth items, while the second metric evaluates how highly ground truth items have been ranked over all items. The larger the values of both Recall and AUC metrics, the better the performance.

4.2 Comparison of Performance

Figure 4 compares the performances of our KA-MemNN model and the baselines on the Tmall and Gowalla datasets regarding Recall and AUC metrics. From the figure, we make the following key observations.

1) Our proposed method KA-MemNN outperforms all baselines including non-session approaches, i.e., BPR and CMN, traditional MC-based next item recommendation methods, i.e., FPMC and FOSSIL, the state-of-the-art hierarchical representation methods, i.e., HRM and SHAN, and a taxonomy-aware MemNN method, i.e., TMRN, on both two datasets. For example, at Recall@20, KA-MemNN improves 8.67% and 10.83% compared with the second-best method, i.e., TMRN, on the Tmall and Gowalla datasets, respectively, although TMRN also achieves excellent performance. This

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=47
[2]www.tmall.com
[3]https://snap.stanford.edu/data/loc-gowalla.html

observation empirically verifies the superiority of our proposed KA-MemNN concerning next recommendation problems with respect to Recall and AUC metrics. Since only KA-MemNN and TMRN utilize taxonomy information, the reason behind this improvement might lie in the fact that the auxiliary information can enrich the representations of users. Even so, KA-MemNN outperforms TMRN, which maybe because TMRN cannot fully exploit the link information from user-category-relations as aforementioned.

2) The performances of session-based methods including KA-MemNN, TMRN, SHAN, and HRM are better than those of BPR and CMN methods, which neglect the sequential information, depicted by a substantial improvement gap, e.g., 13.58% on Tmall and 30.30% on Gowalla in relation to Recall@20 between KA-MemNN and BPR. This indicates that the combination of users' long- and short-term behaviors is necessary to promote recommendation performance, and users' current intentions and preferences have a large influence on users' next behaviors.

3) The performance of CMN is better than that of BPR, although both of them are non-session approaches. The reason for this may lie in the fact CMN accumulates useful knowledge from the memory banks to form informative representations of users, which verifies the effectiveness of MemNNs as well.

4.3 Influence of Components

Influence of the Category Embedding. To gain further insight into the influence of ANN, we propose two simplified versions of KA-MemNN: a general category embeddings version called GC-MemNN and a no category embeddings version called NC-MemNN. GC-MemNN eliminates the attention layer between categories and items and generates an additional embedding matrix for categories, such as user embeddings and item embeddings. In contrast to GC-MemNN which still retains the relationship between categories and items, NC-MemNN excludes category information and directly aggregates item embeddings without key addressing. This method is more like traditional EE-MemNNs. We also apply $Recall@N = \frac{\|Rec@N \cap \mathcal{G}_{a^+}\|}{\|\mathcal{G}_{a^+}\|}$ to evaluate the performance on category predictions, where $Rec@N$ is the set of the categories which the top-N recommended items belong to, and a^+ is the observed item in the test dataset. Due to space limitation, we show the results under the metric of Recall@20 [34]. In addition, we evaluate the score of each category as an average of the scores of its items. This way the intention-based AUC can also reflect the performance of item recommendations.

Table 2 shows that on both datasets, KA-MemNN has the best performance regarding item-based and intention-based Recall@20 and AUC, followed by GC-MemNN. This indicates that the architecture of KA-MemNN is effective, and the user intention information enhances the performance when conducting next-item recommendation. This conclusion is further validated by the observation that the performance of GC-MemNN is better than those of NC-MemNN and TMRN (TMRN is selected since it achieves the best performance over the other baselines). Although GC-MemNN learns a general category representation for all users, it still considers the information of categories compared to NC-MemNN, which promotes the performance. We argue the reason why the performance of GC-MemNN is not as good as KA-MemNN is because of the latter's attention layer, which can learn personalized category embeddings as well as personalized relations between categories and items for each user. Moreover, KA-MemNN has a more significant performance improvement of intention-based metrics compared to the improvement of item-based metrics. On online platforms, the former is vital for improving user experiences since it can lower the risk of recommending unacceptable items.

Table 2: Influence of the category embedding.

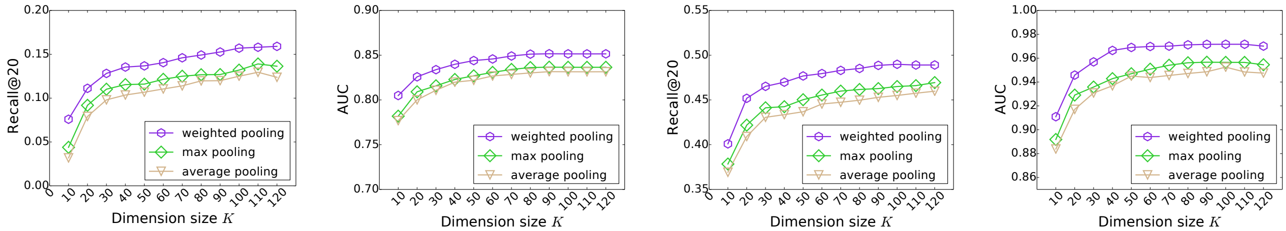
Tmall	**Item**		**Category**	
Method	Recall@20	AUC	Recall@20	AUC
TMRN	0.1438	0.8012	0.2598	0.8432
NC-MemNN	0.1386	0.8009	0.2583	0.8359
GC-MemNN	0.1492	0.8256	0.3028	0.8501
KA-MemNN	**0.1554**	**0.8512**	**0.3469**	**0.8597**
Gowalla	**Item**		**Category**	
Method	Recall@20	AUC	Recall@20	AUC
TMRN	0.4365	0.9796	0.5123	0.9734
NC-MemNN	0.4293	0.9665	0.5084	0.9805
GC-MemNN	0.4536	0.9889	0.5531	0.9884
KA-MemNN	**0.4838**	**0.9896**	**0.5729**	**0.9901**

Table 3: Influence of long- and short-term MBs.

Tmall	**Item**		**Category**	
Method	Recall@20	AUC	Recall@20	AUC
L-MemNN	0.1173	0.7381	0.2283	0.7899
S-MemNN	**0.1596**	0.8403	0.3391	0.8438
LS-MemNN	0.1453	0.8398	0.3212	0.8376
KA-MemNN	0.1554	**0.8512**	**0.3469**	**0.8597**
Gowalla	**Item**		**Category**	
Method	Recall@20	AUC	Recall@20	AUC
L-MemNN	0.4034	0.9011	0.4150	0.8945
S-MemNN	0.4368	0.9529	0.5314	0.9642
LS-MemNN	0.4569	0.9794	0.5427	0.9788
KA-MemNN	**0.4838**	**0.9896**	**0.5729**	**0.9901**

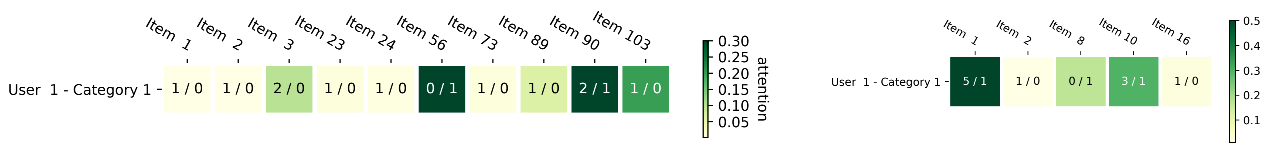
Influence of Long- and Short-term MBs. To explore the influence of long- and short-term MBs, we further propose the additional three simplified versions of KA-MemNN, namely L-MemNN, S-MemNN, and LS-MemNN. Compared to the framework of KA-MemNN, the simplified versions only have one hop, e.g., *1-hop* in Figure 2. More specifically, the framework of L-MemNN and S-MemNN only has long-term KA-MB and short-term KA-MB, respectively. For LS-MemNN, we will fill one KA-MB with all sessions of the target user, i.e., $\mathcal{S}_l^i \cup \mathcal{S}_s^i$ and $\mathcal{A}_l^i \cup \mathcal{A}_s^i$.

Table 3 shows that the approaches fed with more or newer users' behavior data generally perform better. For example, the performances of S-MemNN and LS-MemNN are better than that of L-MemNN on both datasets in terms of the item- and category-based Recall@20 and AUC. Additionally, we can guess that the preferences of users on Tmall are more sensitive to time changes compared to that of users on Gowalla, since the performance of S-MemNN on Tmall is even better than that of LS-MemNN, although the latter



(a) Tmall dataset **(b) Gowalla dataset**

Figure 5: Experimental results for exploring the impact of dimension size K in terms of item-based Recall@20 and AUC.



(a) Tmall dataset **(b) Gowalla dataset**

Figure 6: The heat map of attention which is utilized to construct category embeddings. The color scale indicates the value of the weights, darker representing a higher weight and lighter a lower weight. The inside annotations use the format as "the visited times in long-term sessions / the visited times in short-term sessions". For example, "1 / 0" means this item has and only has been visited once before the current session.

employs more users' behavior data. But on Gowalla dataset, LS-MemNN achieves better performance than S-MemNN, which means the long-term sessions can make the model more robust. Even so, KA-MemNN delivers better performance than LS-MemNN since it can utilize both long- and short-term information and distinguish the importance of items in different sessions.

4.4 Influence of Hyper-parameters

Figure 4 shows the values of item-based Recall@20 and AUC for KA-MemNN across different number of dimensions with size K, and also shows the performances of KA-MemNN when it employs different aggregation operations, i.e., average pooling, weighted average pooling, and max pooling. Accordingly, we have two major observations. (1) a larger value of K can improve the performance since it increases the representation capability of the model. (2) weighted average pooling achieves the best performance compared to the other two aggregation operations, since it makes our model weight the items twice; the lower layer's weights are the linking strength between the items and the target user, and the upper layer's weights are the linking strength between the categories and the target user. This weighting mechanism depicts user-intention-item relations better and exploits the fully modeling capacity of HWU.

4.5 Visualization of Attention

We sample two users from the datasets and visualize the attention between one selected category and the observed items, which belong to the category for each user. Then the attention heat maps on Tmall and Gowalla are shown as Figure 6. We can observe that the frequently visited items usually obtain a relatively larger weight when generating their category embeddings. Also, the items, which are visited in a recent session, also have a larger impact on their category embeddings compared to previously visited items. This phenomena may be reasonable since category-specific users' preferences are reflected on the frequently visited items that belong to this category, and users' current intentions usually have a strong relation with recently visited items.

5 CONCLUSION

In this paper, to address the next-item recommendation problem, we took both user intentions and user preferences into consideration and proposed the KA-MemNN model to capture users' dynamic information in a more comprehensive way. Specifically, the degrees of intention tendencies and the representations of the items which are viewed by users were jointly utilized to form users' hierarchical representations. Then, by assembling an attention model, the representation of intentions and the relationship between intentions and items were personalized for each user. Finally, the experiments on real-world data validated that our model outperformed the state-of-the-art approaches with a significant margin in terms of Recall and AUC metrics. An interesting point of future work is that we can infuse users' search queries and discriminative behaviors to construct more informative user intentions [13].

ACKNOWLEDGEMENT

This work is partially supported by National Key Research and Development Plan (No. 2018YFB1003800) and NSFC (No. 71531001).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv Preprint arXiv:1409.0473* (2014).

[2] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 108–116.

[3] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1082–1090.

[4] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *Proceedings of the International SIGIR Conference on Research & Development in Information Retrieval*. ACM, 515–524.

[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He, and Zhenhua Dong. 2018. DeepFM: An End-to-End Wide & Deep Learning Framework for CTR Prediction. *arXiv preprint arXiv:1804.04950* (2018).

[6] Simon Haykin and Neural Network. 2004. A comprehensive foundation. *Neural Networks* 2, 2004 (2004), 41.

[7] Ruining He and Julian McAuley. 2016. Fusing similarity models with Markov chains for sparse sequential recommendation. In *Proceedings of the International Conference on Data Mining (ICDM)*. IEEE, 191–200.

[8] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the International SIGIR Conference on Research & Development in Information Retrieval*. ACM, 355–364.

[9] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the ACM Conference on Recommender Systems*. ACM, 241–248.

[10] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 1858–1864.

[11] Liang Hu and Soingkui Jian. 2018. Interpretable Recommendation via Attraction Modeling: Learning Multilevel Attractiveness over Multimodal Movie Contents.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 3400–3406.

[12] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 573–581.

[13] Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. 2018. Improving Entity Recommendation with Search Log and Multi-Task Learning.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 4107–4114.

[14] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 659–667.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980* (2014).

[16] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 2341–2347.

[17] Yuanzhi Li and Yang Yuan. 2017. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*. 597–607.

[18] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1734–1743.

[19] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv Preprint arXiv:1606.03126* (2016).

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the WWW Conference on the World Wide Web*. ACM, 811–820.

[22] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. 2015. Interactive intent modeling: Information discovery beyond search. *Commun. ACM* 58, 1 (2015), 86–92.

[23] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the International Conference on Machine Learning*. ACM, 791–798.

[24] Shuo Shang, Ruogu Ding, Kai Zheng, Christian S Jensen, Panos Kalnis, and Xiaofang Zhou. 2014. Personalized trajectory matching in spatial networks. *The VLDB Journal—The International Journal on Very Large Data Bases* 23, 3 (2014), 449–468.

[25] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*. 2440–2448.

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 3156–3164.

[27] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the International SIGIR Conference on Research & Development in Information Retrieval*. ACM, 403–412.

[28] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. ACM, 495–503.

[29] 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. *arXiv Preprint arXiv:1708.04617* (2017).

[30] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent Convolutional Neural Network for Sequential Recommendation. In *Proceedings of the WWW Conference on World Wide Web*. ACM, 3398–3404.

[31] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[32] Ghim-Eng Yap, Xiao-Li Li, and S Yu Philip. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer, 48–64.

[33] Haochao Ying, Jian Wu, Guandong Xu, Yanchi Liu, Tingting Liang, Xiao Zhang, and Hui Xiong. 2019. Time-aware metric embedding with asymmetric projection for successive POI recommendation. *World Wide Web* 22, 5 (2019), 2209–2224.

[34] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press.

[35] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. 2019. Next Item Recommendation with Self-Attentive Metric Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

[36] Hongke Zhao, Qi Liu, Hengshu Zhu, Yong Ge, Enhong Chen, Yan Zhu, and Junping Du. 2018. A sequential approach to market state modeling and analysis in online p2p lending. *IEEE Transactions on Systems, Man and Cybernetics: Systems* 48, 1 (2018), 21–33.