Received April 14, 2020, accepted May 28, 2020, date of publication June 24, 2020, date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004656

Towards a Deep Attention-Based Sequential Recommender System

SHAHPAR YAKHCHI, AMIN BEHESHTI, SEYED-MOHSSEN GHAFARI, MEHMET A. ORGUN, (Senior Member, IEEE), AND GUANFENG LIU, (Member, IEEE)

Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

Corresponding author: Shahpar Yakhchi (shahpar.yakhchi@hdr.mq.edu.au)

This work was supported by the AI-Enabled Processes (AIP Research Centre).

ABSTRACT With the availability of a large amount of user-generated online data, discovering users' sequential behaviour has become an integral part of a Sequential Recommender System (SRS). Combining the recent observed items (i.e., short-term preferences) with prior interacted items (i.e., long-term preferences) has gained increasing attention in recent years. However, the existing methods mostly assume that all the adjacent items in a sequence are highly dependent, which may not be practical in real-world scenarios due to the uncertainty of customers' shopping behaviours. A user-item interaction sequence may contain some irrelevant items which may in turn lead to false dependencies between items. Moreover, current studies usually assign a static representation to each item when modeling a user's long-term preferences. Therefore, they cannot differentiate the contributions of the items. Specifically, these two types of users' preferences have been separately modeled and then linearly combined, which may fail to model complicated user-item interactions. In order to overcome the above mentioned problems, we propose a novel Deep Attention-based Sequential (DAS) model. DAS consists of three different blocks, (*i*) *an embedding block:* which embeds users and items into low-dimensional spaces; (*ii*) *an attention block:* which aims to discriminatively learn dependencies among items in both users' long-term and short-term item sets; and (*iii*) *a fully-connected block*: which first learns a mixture of users' preferences representation through a nonlinear way and then combines it with users' embeddings to have a personalized recommendation. Extensive experiments demonstrate the superiority of our proposed model compared to the state-of-the-art approaches in SRSs.

INDEX TERMS Attention network, dependency modeling, sequential recommender systems.

I. INTRODUCTION

With the rapid growth of online platforms, many companies have started building their e-commerce websites and smartphone applications to encourage their customers to keep interacting with products and services. These platforms can be extremely helpful for users to narrow down their options, while a huge amount of interaction information can be generated. For instance, around 62 million trips with Uber[1] have been recorded in July 2016 [1]. By analyzing the huge amount of information about users' historical sequential behaviour, Sequential Recommender Systems (SRSs) can predict the next interacted items. This can help users, with their decision-making process as well as increasing business profits for companies.

Compared to the classic Recommender Systems (RSs), new challenges have emerged in SRSs. First, an aggregated form of user-item interaction in SRSs is a type of implicit feedback, (e.g., check-in behaviours and purchased items). It is also quite hard for approaches based on the implicit data to predict whether a user is not interested in unpurchased items or just is not aware of them. Therefore, a traditional RS which treats recommendation as a prediction task and only optimizes one-class prediction score (i.e., '1' or '0') may not be appropriate [1], [2]. Second, although a user's sequential behaviour (i.e., short-term preferences) reflects recently observed items, a user's general taste (i.e., long-term preferences) also plays an important role in forming a user's preferences [3]. However, the existing studies either do not take both two types of users' preferences into account or mostly combine them linearly. As an example, Factorizing Personalized Markov Chain (FPMC) proposed by Rendel *et al.* [4], integrates Markov Chain (MC) as a commonly-used method

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

[1]https://www.uber.com/au/en/

in SRSs to model sequential behaviours and Matrix Factorization (MF) for modeling users' long-term preferences. FPMC just utilizes a linear aggregation function to combine these two different types of users' preferences. The Hierarchical Representation Model (HRM) partially solves the problem of modeling high-level user-item interactions through adopting a nonlinear aggregation function [5]. However, these methods have the same overly strong assumption that each item is contextually dependent on the next one and has the same contribution in predicting the next target item. Therefore, they underestimate the impact of irrelevant items in dependency modeling and neglect the context of each item in a shopping basket, which can be defined by selecting a set of items in a transaction. Moreover, the current approaches mainly learn users' long-term preferences through a static low-dimensional representation, which means that a user's general taste always remains the same. However, this is not practical in many real-world cases.

Consider the following example to explain the mentioned problem. Assume $S_1 = \{burger, coke, chips\}$ is a customer shopping list, which at the last minute of shopping, a *shampoo* is added to $S_1 = \{burger, coke, chips, shampoo\}$. While the first three purchased items are strongly correlated, the last item is pretty much irrelevant to them and may generate interference for the next predicted items. In this example, the customer may be more willing to buy a *ketchup sauce* as the next item which is contextually dependent on *burger*, *coke*, and *chips*. In this transaction history, if we consider the first three purchased items as a context, a *shampoo* has no correlation with them and does not share a similar context. The existing methods that may not distinguish the irrelevant items within a contextual sequence may recommend a *conditioner* as a next item due to being highly correlated to the last purchased item. This example can demonstrate the importance of identifying irrelevant items in a transaction for next-item recommendation task. Hence, this can explain that user-item interaction sequences may not follow a strict order and shopping behaviours may contain noisy items which are not contextually dependent on the previous items.

In order to handle the problem of user-item interaction sequences with noise in long- and short-term item sets, we propose a novel Deep Attention-based Sequential recommender system (DAS). DAS consists of three different blocks: (*i*) *an embedding block:* in this block, we aim to take raw user-item and item-item interactions as inputs and embed them into low-dimensional spaces; (*ii*) *an attention block:* here, we use two attention networks to differentiate the importance of each item in both long- and short-term users' preferences. The main purpose of using the attention network is to assign different weights to the items. This can emphasize those strongly context-relevant items and downplay weakly correlated ones in a user-item interaction sequence; and (*iii*) *a fully-connected block:* in this block, we employ two deep network structures, which we first feed the combination of the outputs of the two attention networks to the first deep network in order to learn a mixture of users' preferences. Next, we combine the learned users' mixture preferences with the users' embedding vectors and project them to the second deep network to produce a personalized recommendation.

In particular, the contributions of our work are summarized as follows:

- Introducing an attention network to discriminate the importance of items in long- and short-term users' preferences;
- Designing a deep network structure consisting of three different blocks to generate a mixture of long- and short-term users' preferences; and
- Conducting empirical evaluations on two real-world datasets. The results demonstrate the superiority of our method compared with the state-of-the-art methods in terms of Area Under Curve (AUC), Precision, Recall and novelty evaluation metrics.

II. RELATED WORK

RSs have been known as the filtering tools which help customers with their decision making process and provide them with the relevant items in which they may interested. Generally speaking, there are two different types of RSs in the literature, general recommender systems and sequential recommender systems. The first type of systems mainly focuses on modeling users' long-term preferences from user-item interactions item set, while the second type of systems leverages users' short-term preferences. In the following, we use users' long-term preference or general taste and users' short-term preference or sequential behaviour interchangeably.

A. GENERAL RECOMMENDER SYSTEMS

General recommender systems can be roughly categorised into two main categories: collaborative filtering [6], [7] and content-based models [8]. Matrix Factorization (MF) is one of the most widely used methods in general recommender systems, where users' preferences are learned through user-item latent vectors [9]. MF-based methods use two different types of data: *explicit feedback* and *implicit feedback*. Models that focus on explicit feedback deal with the rating prediction problem where users explicitly express their preferences through providing ratings to the specific items [9]. While approaches based on implicit feedback, (e.g., clicks) aim to formulate recommendation as a ranking problem, which highly depends on the selection of an objective loss function to optimize [10]. Despite the great success of MF-based methods in RSs, they may fail due to the data sparsity problem, where there is a lack of available information regarding user-item interactions.

Although general recommender systems have shown the advantages of capturing users' long-term preferences, their performance may decrease as the users' short-term preferences are ignored. Therefore, it is hard for general recommenders to recommend items which comply with the recent user-item interactions.

B. SEQUENTIAL RECOMMENDER SYSTEMS

Different from general recommender systems, SRSs use the current user-item interaction sequences in order to model users' sequential behaviours. MC has been known as a straightforward method to model sequential dependency in SRSs. For instance, Shani *et al.* [11] examine the relation between a pair of items and then predict the probability of the next set of items for interaction. Attention mechanism is also applied in SRSs, FDSA for instance, use this technique to utilize transition patterns between features of items [12].

Recently, Recurrent Neural Networks (RNN), have been known as one of the most used deep learning-based methods in SRSs. Following this, Zhang *et al.* [13] proposed a novel RNN-based model is proposed to capture users' sequential behaviours for click prediction. Different from traditional RNN, several works have been introduced to modify classic RNN in order to better capture the whole historical user-item interaction sequences [14], [15].

C. UNIFIED RECOMMENDER SYSTEMS

Recently, a few studies focused on building a unified recommender by taking both users' general preferences and sequential behaviours into account. Inspired by the great capability of the MF technique to model users' long-term preferences, FPMC is proposed to fuse classic MF with first-order MC to better capture both users' long-term and short-term preferences [4]. Following that, we have witnessed the rising trend of taking the both types of users' preferences into account. Inspired by the word embedding technique [16], Prod2Vec is proposed to utilize information from a sequence of interacted items to improve MF performance [17]. HRM with a hierarchical structure partially solves the problem of modeling abstract user-item interactions through adopting a non-linear aggregation function, while they may lose much information due to employing the max pooling function as an aggregation function [5]. However, one of the shortcomings of MC-based approaches is that they can only model local sequential patterns between two sequences. Moreover, as they consider a fixed-weight for different items, they may fail to capture complex user-item interactions.

Inspired by Convolutional Neural Networks (CNNs), Caser [18] has been introduced as a sequential recommender which treats user-item interactions as an image and then learns sequential patterns as local features of the image by using convolutional filters. Furthermore, RNNs also have attracted more attention in modeling sequential dependencies in SRSs [14], [19]. For instance, SLi-Rec improves the classic RNN structure such as Long Short-Term Memory (LSTM) by proposing time-aware and content-aware controllers to fully exploit user modeling and then attention-based framework is applied to combine general and sequential recommenders [20]. Due to the great success of both CNN and RNN in capturing local sequential patterns and complex long-term dependencies, respectively, Xu *et al.* [21] have proposed a novel Recurrent Convolutional Neural Network model (RCNN) to better generate the recommendation.

Apart from basic RNNs, improved architectures such as Gated Recurrent Unit (GRU) [22] and Long-Short Term Memory (LSTM) [23] have been developed to model sequential dependencies. Although, both GRU and LSTM have shown great success in SRSs, they have difficulties to model long-range dependent patterns.

Lately, researchers have employed an attention mechanism due to its powerful capability in focusing on selective parts [24]. Although incorporating an attention network presents the superior performance in context learning in the work by Wang *et al.* [25], this model ignores users' general taste. Instead, a two-layer hierarchical design called, SHAN, has been proposed by Ying *et al.* [1] as an attention-based SRS to incorporate both users' general tastes and short-term preferences in a unified manner. The main difference between our work and SHAN can be be highlighted from three different aspects: firstly, in modeling each of a user's long and short-term preferences, SHAN calculates the attention score which is guided by the user embedding. Therefore, it may not completely discover the contributions of each item, and it may not be able to find noisy items. Secondly, SHAN ignores the context of users' shopping basket, which in turn can play an important role in predicting the next-item recommendation. Thirdly, unlike SHAN, we add the user's embedding vector at the final layer which as stated in [26], can improve the model performance through using the pre-training model's parameters.

To sum up, most of the existing studies assume that there is a rigid order between two adjacent items in a sequence. However, this may not be true in many real-world cases and there may be some noisy items in a sequence which generate fake dependencies.

III. DEEP ATTENTION-BASED SEQUENTIAL RECOMMENDER SYSTEM (DAS)

In this section, we formulate our problem and define notations used throughout the paper and then explain the details of our model.

A. PROBLEM STATEMENT

Let $U = \{u_1, u_2, \ldots, u_{|u|}\}$ denote the user set and $V = \{v_1, v_2, \ldots, v_{|v|}\}$ indicate the item set, where $|u|$ and $|v|$ are the total number of users and items, respectively. For a given user u, $Q^u = G^u_{t-1} \cup S^u_t$ represents the total users' sessions, where G^u_{t-1} and S^u_t are explained as follows. At a certain timestamp t, $S^u_t = \{v_1, v_2, \ldots, v_i\}$ is u's sequential behaviour, where $i \in V$, reflecting the user's short-term preference. Following that, $G^u_{t-1} = \{S^u_1, S^u_2, \ldots, S^u_{t-1}\}$ is a set of interacted items at timestamp $t - 1$ (e.g., purchasing history, clicked items, and check-in behaviours), which represents a user's general taste (i.e., long-term preference). In this paper, for simplification, G^u_{t-1} and S^u_t represent the u's long-term and short-term interacted item sets, respectively.

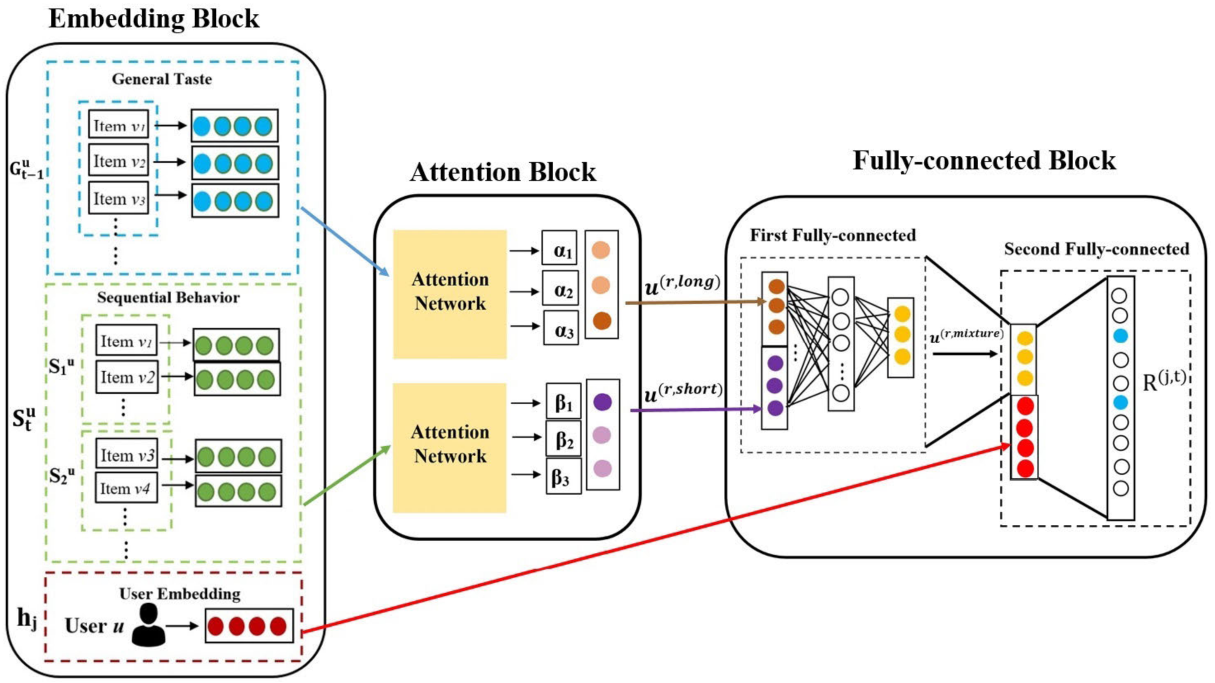


FIGURE 1. The architecture of DAS, which learns sequential dependencies attentively and discriminatively over user-item interaction sequences.

B. THE NETWORK ARCHITECTURE

As it is shown in Figure 1, our attention-based model, DAS, consists of three main blocks with different purposes. From the left to the right, (*i*) *an embedding block:* takes raw user-item and item-item interactions data as inputs and embeds them into low-dimensional spaces. (*ii*) *An attention block:* includes two attention networks to identify the most relevant items simultaneously in both long-term and short-term users' item sets based on the embedded user-item interactions. The main idea behind employing an attention mechanism is to pay more attention to the relevant items and less attention to those items that are irrelevant. The attention mechanism by assigning different weights to different items is able to focus more on contextual items in order to make a user's preferences more personalized, and (*iii*) *a fully-connected block:* which consists of two deep network structures to first learn a mixture of a user's preferences through one fully-connected network. Then, a concatenation of a user's mixture preferences with a user embedding vector is passed to the second fully-connected network to design a personalized SRS.

C. EMBEDDING BLOCK

In SRSs, predicting the next interesting items in a session is similar to predicting a relevant word in Natural Language Processing (NLP), in which the size of the items and the size of the vocabularies are too large. Inspired by NLP techniques, if we consider words as items or users, which are indexed by meaningless IDs, we need to transform them from IDs space to a more informative representation space rather than IDs. Therefore, a fully connected layer is employed to embed user and item IDs (i.e., one-hot representations) into two continuous low-dimensional spaces as $U \in \mathbb{R}^{k\times|u|}$ and $V \in \mathbb{R}^{k\times|v|}$, where k is the latent dimension of embedding spaces.

1) ITEM EMBEDDING

Given the original one-hot encoded item set as an input (where in this vector, item i ($i \in V$) at position i equals to 1, and the rest of the positions are set to 0) may limit derived information. Therefore, in DAS, we use the embedding block to map these sparse vectors into an informative representation. We have a vector with length $|v|$, which represents the context of each item. We use $h_i \in \mathbb{R}^k$ to represent the context of item i, where k is the latent dimension, namely:

$$h_i = \sigma(W^1_{:,i}), \tag{1}$$

where $W^1_{:,i} \in \mathbb{R}^{k\times|v|}$ is a weight matrix, and the i^{th} column of matrix $W^1_{:,i}$ endows one-hot vector item i to its embedding h_i. Here, we use logistic function $\sigma(.)$ as our activation function to better model the non-linearities.

2) USER EMBEDDING

Similar to item embedding, we take user j's one-hot vector ($j \in U$) at this block to embed it into $h_j \in \mathbb{R}^k$. Formally:

$$h_j = \sigma(W^2_{:,j}), \tag{2}$$

where $W^2_{:,j} \in \mathbb{R}^{k\times|u|}$ is a weight matrix and the j^{th} column endows one-hot vector of user j to its embedding h_j. Due to the simplification, we set the same size for user and item embedding.

D. ATTENTION BLOCK

Inspired by a great success of attention networks in many tasks, such as machine translation [27], image captioning [28], and recommendation [29], we use an attention network to discriminate the contributions of different items in predicting the next items. Despite the fact that most of the existing studies assume all items are contextually dependent and there is no noise in a session, in this block we employ two attention networks to identify the highly context-dependent items in both long- and short-term users' preferences. The attention network automatically assigns different weights to different items to downplay the impact of irrelevant items which may overwhelm the impact of the relevant ones.

1) LONG-TERM USER'S PREFERENCE REPRESENTATION

A set of interacted items in the long-term item set G^u_{t-1} can reflect a user's general taste. However, the consideration of a fixed weight for modeling a user's general taste may not reflect the user's real preferences due to a simple assumption that each item plays an equal role in long-term users' preferences. To fulfill the above requirements, for a given user u, we first measure the alignment of item v_i's embedding vector, h_i, with respect to the context matrix W^α. Then we compute the attention score α_{ip} through Equation 3, which indicates the level of contribution of contextual item v_i in the occurrence of item v_p in a long-term interacted item set. Then, the softmax function is applied to normalize the attention score α_{ip}, where the higher score represents the more item contribution. Finally, a weighted sum over the attentive context embeddings in a long-term item set G^u_{t-1} is computed to build a user's long-term representation. Formally;

$$\alpha_{ip} = \frac{exp(e(h_i))}{\sum_{p\in G^u_{t-1}} exp(e(h_p))} \tag{3}$$

$$e(h_i) = W^\alpha h_i^T \tag{4}$$

$$u^{(r,long)} = \sum_{i\in G^u_{t-1}} \alpha_{ip} h_i \quad \text{w.r.t} \quad \sum_{i\in G^u_{t-1}} \alpha_{ip} = 1, \tag{5}$$

where W^α is a shared weight over the first attention network which is randomly initialized and then will be learned during the training process. Similar to the work by Wang *et al.* [25], we consider W^α as an item-level context matrix shared by all the items to find the more informative items among a set of items like the one used in memory network [30]. Finally, the long-term users' preference representation, $u^{(r,long)}$, can be calculated with the help of attentive context which is represented by Equation 4.

2) SHORT-TERM USER'S PREFERENCE REPRESENTATION

Rather than long-term users' preferences, the most recent interacted items in a session, which form short-term users' preferences, are also important for predicting the next items. Therefore, similar to users' long-term item sets, an attention network has been applied here to more focus on key items in users' short-term item sets.

$$\beta_{sn} = \frac{exp(e(h_s))}{\sum_{n\in S^u_t} exp(e(h_n))} \tag{6}$$

$$e(h_s) = W^\beta h_s^T, \tag{7}$$

where similar to the long-term users' preferences, we first measure the similarity weights of each item v_s embedding as h_s, with the context matrix W^β, to find the most relevant items in users' short-term item set. Then, we normalize this attentive context through the softmax function to find the level of contribution of item v_s's context with regards to the target item v_n [31]. Finally, to form a user's short-term representation, $u^{(r,short)}$, a weighted sum over contextual item embedding in short-term item set S^u_t is computed, where the weights can be inferred from the attention network.

$$u^{(r,short)} = \sum_{s\in S^u_t} \beta_{sn} h_s \quad \text{w.r.t} \quad \sum_{s\in S^u_t} \beta_{sn} = 1 \tag{8}$$

E. FULLY-CONNECTED BLOCK

In this block, we adopt two deep network structures; one for learning a mixture of users' preferences and the other for making a personalized item recommendation. Unlike most of the existing studies that linearly combine long- and short-term users' preferences, which may limit model performance [4], [32], we concatenate the outputs from the attention block and feed it to the first deep network in order to learn a mixture of users' preferences.

$$u^{mixture} = \phi_a(W \begin{bmatrix} u^{(r,long)} \\ u^{(r,short)} \end{bmatrix} + b), \tag{9}$$

where ϕ_a is an activation function, $W \in \mathbb{R}^{d\times 2d}$ and $b \in \mathbb{R}^d$ are the weight matrix and bias, respectively, and d is the dimension of hidden layers. We call $u^{mixture} \in \mathbb{R}^d$ the mixture of long- and short-term users' preferences, which encodes two types of users' preferences in user-item interaction item set Q^u. Next, when $u^{mixture}$ is ready, we combine it with a user's embedding vector h_j and project it into the final output network with $|I|$ number of nodes, to capture the high-level representation of user u_j, namely:

$$R^{(i,t)} = W' \begin{bmatrix} u^{mixture} \\ h_j \end{bmatrix} + b', \tag{10}$$

where $W' \in \mathbb{R}^{|I|\times 2d}$ and $b' \in \mathbb{R}^{|I|}$ are the weight matrix and bias in the final deep network, respectively. At this layer, $R^{(i,t)}$ denotes the probability that a user will interact with item i at time t. Similar to the work by Tang and Wang [18], we add the user's embedding vector h_j at the final layer for two main reasons: (1) to make a more personalized recommendation; and (2) to improve model performance by using pre-training model's parameters, which is stated in [26].

F. NETWORK TRAINING

Since in this work we focus on implicit feedback, (e.g., check-ins and purchase transactions), our goal is to provide a ranked-list over items [2]. Therefore, after computing users' mixture preferences, we propose to use a pair-wise ranking objective function to rank observed entries higher than unobserved ones [18]. To do so, we transform the output of our final deep network, $R^{(i,t)}$, to the probabilities score.

$$p(S_t^u|S_1^u, S_2^u, \ldots, S_{t-1}^u) = \sigma(R^{(i,t)}), \tag{11}$$

where σ is sigmoid function, $\sigma = 1/(1 + e^{(-x)})$. For each positive pair (u, i), in $I^+_{(u,t)} = \{i|R^{(i,t)} = 1\}$, we randomly keep one item in each session as an unobserved item at time t which need to be predicted by our model $I^-_{(u,t)}$. Then, we update the binary cross-entropy loss function as:

$$l = -\sum_{(u,t)} \Big(\sum_{i \in I^+_{(u,t)}} log\sigma(R^{(ui,t)}) \sum_{i' \in I^-_{(u,t)}} log(1 - \sigma(R^{(ui',t)}) \Big) + \lambda_{ui}||\Theta_{ui}||^2 + \lambda_{at}||\Theta_{at}||^2, \tag{12}$$

where $\Theta = \{\Theta_{ui}, \Theta_{at}, \Theta_d, b, b'\}$ is the set of model parameters, $\Theta_{ui} = \{W^1, W^2\}$ is the set of weights for item and user embeddings, respectively, and $\Theta_{at} = \{W^\alpha, W^\beta\}$ is the set of weights in attention networks. Further, $\Theta_d = \{W, W'\}$ is the set of weights for two deep networks, and $\lambda = \{\lambda_{ui}, \lambda_{at}\}$ is a set of our regularization parameters. We also adopt Stochastic Gradient Descent (SGD) for updating our parameters. Algorithm 1 presents the details of our proposed model.

Algorithm 1 DAS Algorithm

```
Input: long- and short-term item sets{$G^u_{t-1}$, $S^u_t$}, learning rate $\eta$, $\lambda$, $K$
Output: a set of parameter $\Theta$ Initialize $\Theta_{ui}$, $\Theta_d$ with Normal Distribution $N(0, 0.01)$
Initialize $\Theta_{at}$ with Uniform Distribution$[-\sqrt{\frac{3}{K}}, \sqrt{\frac{3}{K}}]$
while convergent do
    for $\forall$ $u \in U$ do Randomly pick an item $i$ from $Q^u$
        $h_i$, $h_j$ $\leftarrow$ arrange embeddings (Equation 1,2)
        compute a user's long-term representation $u^{(r,long)}$ based on the Equation 3-5
        compute a user's short-term representation $u^{(r,short)}$ based on the Equation 6-8
        compute a user's mixture of preferences representation $u^{mixture}$ based on the Equation 9
        $R^{(i,t)} \leftarrow \sigma$ ($u^{mixture}$.Concat $h_j$) based on the Equation 10-11
        update $\Theta$ with gradient descent
    end for
end while
return $\Theta$
```

1) RECOMMENDATION
We feed our network with the user's embedding vector h_j and user-item interaction set $Q^u = G^u_{t-1} \cup S^u_t$. Next, we predict $R^{(i,t)}$, which is the probabilities of the user's interaction with item v_i at timestamp t. Then, N items with the highest values will be recommended to this user. Therefore, the complexity of our model for making a recommendation to all users is $O(|U||I|d)$.

IV. EXPERIMENTS AND EVALUATION
In this section, we introduce the datasets, evaluation metrics and baseline methods used in our experiments. Next, we discuss the impact of different hyper-parameters on our proposed model, DAS.

A. EXPERIMENTAL SETUP
1) DATASETS
We evaluate our model on two real-world datasets, Gowalla [33] and Tmall [34], to compare the performance of DAS with the baseline approaches. Gowalla aggregates users' check-ins information from location-based social networking website, Gowalla. While Tmall records users' transactions in the largest online shopping website in China, where each session (transaction) consists of multiple items.

2) DATA PREPOSSESSING
Note that, similar to the work by Ying *et al.* [1] in both the datasets, we only consider the data in the last seven months and remove the sessions with only one item and items with less than 20 observations. Then, to better represent users' sequential behaviours (i.e., short-term preferences), transactions in one day are considered as one session. After the prepossessing step, the characteristics of our datasets are shown in Table 2. As in the relevant works [34] and [1], we randomly divide our datasets into 20% and 80% for tests and training, respectively. To better evaluate DAS, we randomly keep one item in each session to be predicted by our model.

3) PARAMETER SETTING
We set the size of the item and user embedding to 100, which are initialized randomly with Normal distribution $N(0, 0.01)$ and the weight parameters in attention network are initialized from the uniform distribution $U(-\sqrt{\frac{3}{K}}, \sqrt{\frac{3}{K}})$. We use Stochastic Gradient Descent (SGD) as our optimization technique to update our parameters, the learning rate η is set to 0.001, and the training's epoch size is 10. We also empirically set the batch size to 50 and consider $\lambda_{uv} = \{0.01, 0.001, 0.0001\}$ as our user and item embedding regularization, and $\lambda_a = \{0, 1, 10, 50\}$ as our attention network regularization.

4) EVALUATION METRICS
We test the performance of DAS in terms of the recommendation accuracy and novelty. Therefore, we use three widely adopted metrics for measuring the recommendation accuracy in SRSs such as Recall@N, Precision@N and AUC, where the larger value indicates better performance. Recall and Precision evaluate the ability of our model to find all the relevant top-N items within a dataset, where $N \in \{5, 100\}$, while AUC shows how well our model can rank ground truth items.

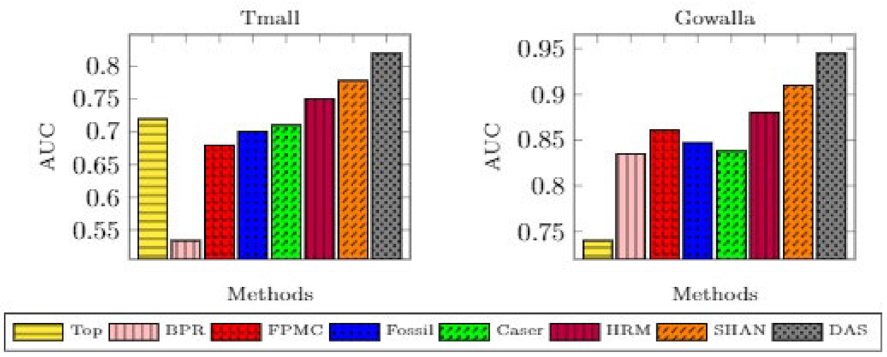


FIGURE 2. Performance comparison on the Gowalla and Tmall datasets in terms of AUC.

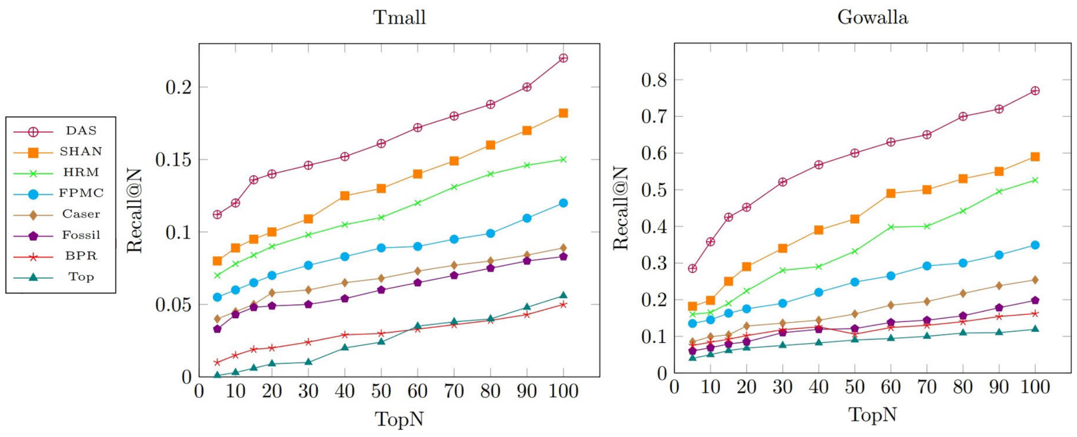


FIGURE 3. Performance comparison on the Gowalla and Tmall datasets in terms of Recall@N.

5) BASELINES

Here, we introduce the baseline methods for comparison, including classic next item recommendation, combining long- and short-term users' preferences in SRSs and attention-based SRSs. For our evaluations we use the same datasets as SHAN, FPMC, Caser, Fossil and HRM. So, for implementing these approaches we consider the corresponding experimental setup as explained in their papers. Furthermore, for implementing BPR and Top approaches we tune them in a way to reach their best performance in order to have a fair comparison.

- SHAN: This is a state-of-the-art method in SRSs, which combines sequential behaviours with users' general taste through a hierarchical attention-based structure [1].
- FPMC: Linearly combines MF and MC model to learn users' preferences [4].
- BPR: This is a state-of-the-art method for the next item recommendation with a pairwise ranking loss through implicit feedback [35].
- Top. Recommends the top popular items which have been determined during training.
- FOSSIL: This method uses the high-order MC and the similarity model for modeling sequential behaviour and user's general preference, respectively. Then it linearly combines them to make the next item recommendation [32].
- Caser: This is a state-of-the-art model, which uses CNN for sequence embedding [18].
- HRM: Introduces a hierarchical representation to learn both the users' general taste and sequential behaviour [5].

B. PERFORMANCE EVALUATION

Based on Figures 3 and 2 and Table 1, we have the following findings: (*i*) DAS significantly and consistently outperforms all the compared approaches with respect to Precision, Recall and AUC evaluation metrics in both Gowalla and Tmall datasets. From the results we can see the superiority of DAS compared to the SHAN which is the state-of-the-art method in SRSs. There may be two reasons behind this result. First, as we stated in Section III-D we employ the attention mechanism in DAS to measure the importance of each item as the similarity of its embedding with the item level context vector which is shared by all contextual items. While SHAN computes the attention score as the similarity between the embedding of item and user. Therefore, unlike SHAN, our model which treats all the contextual items as a whole may better model the complex dependency relations.

TABLE 1. Performance comparison on the Gowalla and Tmall datasets in terms of precision.

Dataset	Metric	DAS	SHAN	HRM	FPMC	Caser	Fossil	BPR	Top
Gowalla	Prec@1	**0.2887**	0.1941	0.1401	0.1341	0.1131	0.0917	0.0884	0.0655
	Prec@5	**0.2984**	0.1962	0.1410	0.1374	0.1148	0.0951	0.0890	0.0671
	Prec@10	**0.3012**	0.1977	0.1483	0.1398	0.1194	0.0974	0.0899	0.0698
Tmall	Prec@1	**0.2541**	0.1812	0.1324	0.1211	0.0998	0.0811	0.0711	0.0601
	Prec@5	**0.2712**	0.1847	0.1371	0.1284	0.1001	0.0841	0.0725	0.0611
	Prec@10	**0.2917**	0.1853	0.1394	0.1296	0.1131	0.0891	0.0741	0.0642

TABLE 2. Statistics of our datasets.

Dataset	Tmall	Gowalla
Number of users	20,716	15,254
Number of items	25,143	13,052
avg.session length	2.81	2.99
train session	71,998	128,115
test session	3565	3611

Second, we add the embedding of user at the final fully connected layer to make a more personalized recommendation. Therefore, compared with SHAN which is the second-best model, DAS has shown 21.5% and 35.5% improvements on the Tmall dataset with regards to Recall@20, and Precision@10, respectively. We also observe that on the Gowalla dataset, DAS shows promising results, where it achieves 37.5% and 0.35.4% improvements in terms of Recall@20 and Precesion@10, respectively. According to Table1, where the best result in each row is highlighted in boldface, DAS has shown the best performance compared to the others. This demonstrates the effectiveness of DAS in capturing the most important and relevant items in long- and short-term users' preferences through employing an attention mechanism; (*ii*) compared to the other hybrid methods (e.g., HRM, FPMC, Caser and Fossil) which combine long- and short-term users' preferences, after DAS, SHAN outperforms others like HRM, FPMC, Caser and Fossil, with a performance improvement of around 16%, 27% compared to the HRM at Recall@50 in Tmall and Gowalla datasets, respectively. These observations demonstrate that selecting an attention mechanism as an aggregation function performs well despite using a simple max pooling operation. Compared to the HRM, the performance of FPMC is limited, due to the use of a linear aggregation function. The performance of Caser and Fossil are close, but Caser has shown a slight improvement with regards to the Precision and Recall evaluation metrics, which may be because of using various convolutional filters. DAS achieves better performance compared to the all the mentioned approaches, indicating the superiority of DAS in truly capturing users' preferences; and (*iii*) among all the compared methods, those approaches which combine users' sequential behaviours with their general taste (e.g., DAS, SHAN, HRM, FPMC, Caser and Fossil), generally outperform the traditional methods which ignore users' short-term preferences (i.e., BPR and Top). This can indicate the importance of users' sequential data. Surprisingly, on the Tmall dataset, the Top method achieves better performance than BPR with respect to Recall, especially when N starts increasing from 60. According to Figure 2, a minor improvement is recorded for BRP compared to the FPMC in terms of AUC. This indicates the trend of purchasing popular items in online shopping. In contrast to the Tmall dataset, Top achieves the lowest Recall in different Ns on the Gowalla dataset, possibly because of the property that the Gowalla dataset has more personalized information. Finally, based on Table 1, our approach surpasses all the compared baselines in terms of Recall, Precision and AUC on both of the Gowalla and Tmall datasets.

C. EVALUATING USER-ITEM INTERACTION SEQUENCES WITH NOISE

1) THE EFFECT OF SESSION LENGTHS

Here, we examine the performance of DAS compared to the other baselines to show the advantages of our model to handle user-item interaction sequences with noise in different sequence lengths. To do so, we test our model under the different size of session lengths, which we call SLen for simplicity, which denotes the number of existing items in one session. For example, SLen-5 denotes that there are five contextual items in a session. The longer session lengths may contain more irrelevant items which in turn can reduce recommendation accuracy, if those items are not recognized. According to Figure 4, we can observe that DAS consistently outperforms the other approaches under different session lengths. Moreover, increasing the session lengths results in better recommendation accuracy, where SLen-5 achieves the best results with respect to evaluation metrics Recall@10 and Precision@10. We only test the effect of different session lengths on DAS and SHAN, which are sensitive to the session lengths and leave other models out as they mostly model first-order dependencies. Moreover, we only examine the effect of session lengths on the Gowalla dataset, which clearly from Table 2, on average has longer session lengths.

2) THE EFFECT OF DISORDER ITEMS

DAS is able to capture attentive context and thus focus more on the most important items in a session and ignore irrelevant

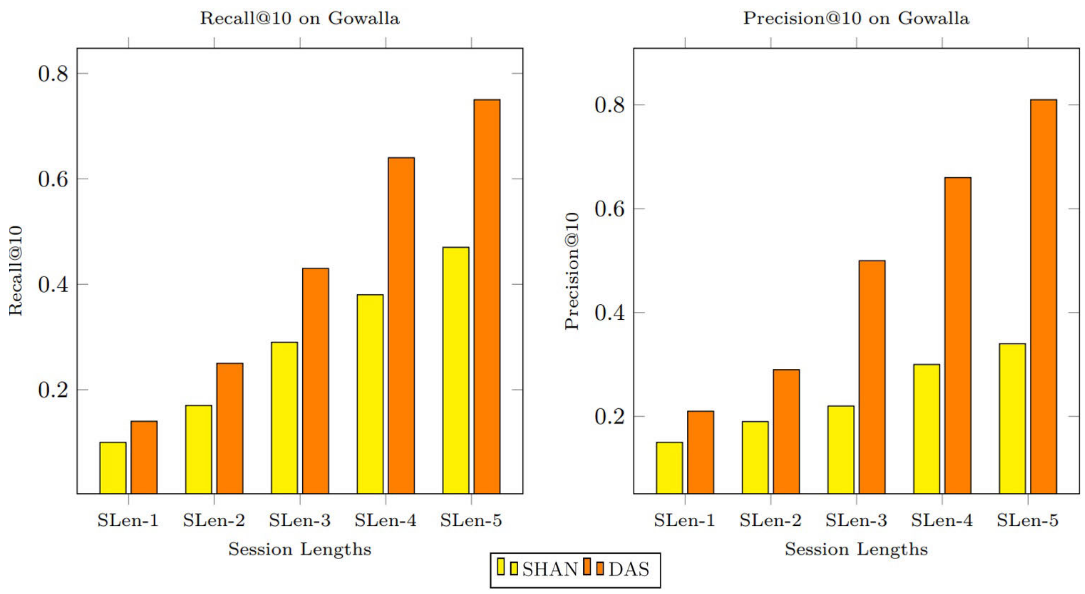


FIGURE 4. **The effect of different session lengths.**

ones regardless of their order. To evaluate this, we make the Tmall dataset disordered and test the performance of DAS under this condition. To do so, similar to [25], we randomize the default item order on the Tmall datasest to create a new disordered dataset. Table 3 demonstrates the superiority of DAS on the new dataset. Based on Table 3, unlike DAS, the performances of the rest of the compared methods are decreased on a disordered dataset, indicating the strong ability of DAS to emphasize on the important items no matter where they are in a transaction.

TABLE 3. **Performance evaluation on disordered Tmall.**

Model	AUC	Recall@10	Precision@10
DAS	**0.8102**	**0.1043**	**0.2802**
SHAN	0.6851	0.0701	0.1684
HRM	0.6114	0.0541	0.1211
FPMC	0.5885	0.0521	0.1124
Caser	0.5311	0.0321	0.1021
Fossil	0.5047	0.0294	0.0747
BPR	0.4987	0.01020	0.0665
TOP	0.3851	0.0019	0.06024

3) NOVELTY EVALUATION

Except for the mentioned evaluation metrics, we also compare DAS with the existing approaches in terms of the novelty metric. Considering that recommending items similar to those that a user has already purchased, may not be satisfactory and users may be more willing to be recommended by new items. Hence, similar to the work by Wang *et al.* [25], we consider novelty metric as another evaluation parameter to show the capability of our model in recommending novel items. The novelty metric is able to measure the difference between contextual items in a shopping basket and a set of recommended items, where the larger difference can represent the novel items [36]. Our proposed model by discovering an attentive context and measuring the contributions of each item in both long- and short-term item set, can provide users with new unseen items and avoid recommending duplicated items.

4) MCAN@K

In our model, items which are purchased in the context c can be used for recommendation R. An increasing between recommended and purchased items, means the less novelty. Subsequently, the novelty measures the mean of unseen items corresponding to the context c over all N top-K recommended items.

$$MCAN = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|R_i \cap c_i|}{|R_i|}) \qquad (13)$$

Figure 5 demonstrates the performance of DAS compared to the other methods in terms of the novelty metric on both the Tmall and Gowalla datasets. From the Figure 5 we can see that both Tmall and Gowalla datasets show a similar trend. Among approaches that only consider users' long-term preferences (e.g., BPR and Top), Top recommends the top popular items to a user and hence it is more likely that those items have been already observed by the user. Thus TOP achieves the lowest novelty score. Although FPMC take both types of users' preferences into account, it can not well learn the parameters on such sparse datasets. Therefore, the recommended items may be relatively random ones and accordingly users can not experience a novel item by FPMC method, resulting in low novelty score. Except SHAN, compared to the hybrid methods which combine both types of preferences,

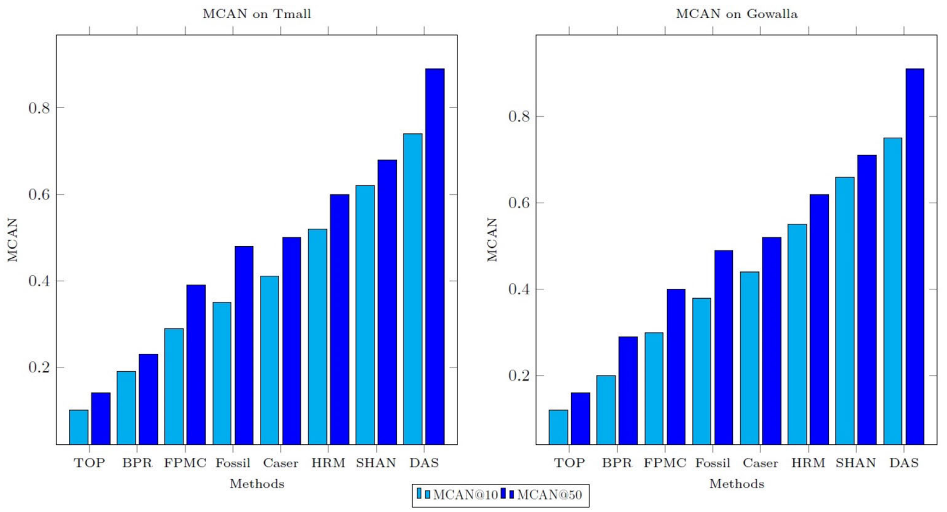


FIGURE 5. The evaluation under novelty metric.

TABLE 4. Impcat of different type of preferences at Recall@20.

(a) Tmall

Methods	AUC	Recall@20
DAS-long	0.720	0.032
DAS-short	0.781	**0.175**
DAS	**0.821**	0.149

(b) Gowalla

Methods	AUC	Recall@20
DAS-long	0.910	0.245
DAS-short	0.934	0.425
DAS	**0.945**	**0.443**

HRM reports a high level of novelty may be due to using different aggregation functions as well as optimization criteria. After DAS, SHAN has a higher novelty score compared to the all aforementioned methods due to employing attention mechanism that considers dynamic properties in users' long and short-term preferences. To sum up, the experimental results indicate that DAS may better capture the most influential and contextual items through the attentive context in both long- and short-term users' preferences, and thus represent a promising result in SRSs.

D. IMPACT OF DIFFERENT TYPE OF PREFERENCES

In this section, we aim to evaluate the impact of each of long- and short-term users' preferences on the next item recommendation problem, separately. Therefore, we consider a version of DAS, called DAS-long, for situations in which only a user's long-term preference is modelled and another version, called DAS-short, which only considers the user's short-term preference. We compare the performance of DAS with DAS-long and DAS-short in Table 4. Compared to the approaches which only model user's general taste, DAS-long performs better than BPR by 17% and 13% with respect to Recall@20 on the Tmall and Gowalla datasets, respectively. This can explain that employing an attention mechanism can truly identify a set of context-relevant items in the user's general taste, leading to better present user's mixture of preferences. In both the datasets, DAS-short achieves better performance than DAS-long, indicating the importance of sequential behaviours in the next item recommendation problem. Remarkably, DAS-short represents a better performance than SHAN on both the datasets, as SHAN achieves 0.778 and 0.901 AUC on the Tmall and Gowalla datasets, respectively. This may explain that although the main focus of DAS-short is on identifying the most important items in a short-term interacted item set, a part of the user's general taste is learned during the training process.

To short, Table 4 indicates the superior performance of DAS compared to the DAS-long and DAS-short. This demonstrates that considering both types of users' preferences can result in a better users' preference modeling and increasing recommendation accuracy accordingly.

E. IMPACT OF HYPER-PARAMETER

In this section, we investigate the impact of hyper-parameters on the performance of DAS. Similar to the work by Ying *et al.* [1], we demonstrate the results just under Recall@20. The sizes of the item and user embeddings are the same and set to 100. We also empirically set the batch size to 50. We consider $\lambda_{uv} = \{0.01, 0.001, 0.0001\}$ as our user and item embedding regularization, and $\lambda_a = \{0, 1, 10, 50\}$ as our attention network regularization. As shown in Table 5, the performance of DAS gradually is improved when λ_a start moving from 0 to 50 on both the Tmall and Gowalla datasets, which means that employing an attention mechanism can help us to better discover users' preferences.

TABLE 5. Impcat of different regularization at Recall@20.

(a) Tmall

λ_{α} \ λ_{uv}	0.01	0.001	0.0001
0	0.091	0.084	0.077
1	0.132	0.122	0.109
10	0.142	0.131	0.119
50	0.149	0.139	0.128

(b) Gowalla

λ_{α} \ λ_{uv}	0.01	0.001	0.0001
0	0.242	0.365	0.384
1	0.347	0.422	0.458
10	0.354	0.414	0.449
50	0.361	0.443	0.452

V. CONCLUSION

In this paper, we have proposed a Deep Attention-based Sequential model, DAS, for the next item recommendation problem. We have first embedded users and items into the latent dimensional spaces and then passed them into two attention networks to discriminate the contribution of each item in both long-and short-term users' preferences. Next, we have combined these two types of users' preferences to learn a mixture of users' preference through a deep network. Finally, we have fed a concatenation of a learned users' mixture preferences with users' embedding to the final deep neural network to make a personalized item recommendation. Specifically, our method was able to model complex and abstract user-item interactions through the nonlinear aggregation function. Extensive experiments on two real-world datasets have demonstrated the superiority of our model compared to the state-of-the-art methods in terms of defined evaluation metrics.

REFERENCES

[1] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 3926–3932.

[2] I. Bayer, X. He, B. Kanagal, and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," in *Proc. 26th Int. Conf. World Wide Web*, Sydney, NSW, Australia, Apr. 2017, pp. 1341–1350.

[3] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: Challenges, progress and prospects," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Shanghai, China, Aug. 2019, pp. 6332–6338.

[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, New York, NY, USA, 2010, pp. 811–820.

[5] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for NextBasket recommendation," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Santiago, Chile, 2015, pp. 403–412.

[6] J. B. Schafer, "Collaborative filtering recommender systems," in *The Adaptive Web, Methods and Strategies of Web Personalization*, vol. 4321, P. Brusilovsky, Ed. Berlin, Germany: Springer, 2007, pp. 291–324.

[7] M. D. Ekstrand, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 175–243, 2011.

[8] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web, Methods and Strategies of Web Personalization*, vol. 4321, P. Brusilovsky, Ed. Berlin, Germany: Springer, 2007, pp. 325–341.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[10] A. Karatzoglou, L. Baltrunas, Y. Shi, "Learning to rank for recommender systems," in *Proc. Conf. RecSys*, Beijing, China, 2013, pp. 493–494.

[11] G. Shani, D. Heckerman, and R. I. Brafman, "An MDP-based recommender system," *J. Mach. Learn. Res.*, vol. 6, pp. 1265–1295, Sep. 2005.

[12] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, "Feature-level deeper self-attention network for sequential recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2019, pp. 4320–4326.

[13] Y. Zhang, "Sequential click prediction for sponsored search with recurrent neural networks," in *Proc. AAAI*, Toronto, Canada, 2014, pp. 1369–1375.

[14] B. Hidasi, "Session-based recommendations with recurrent neural networks," in *Proc. Conf. Learn. Represent. (ICLR)*, Puerto Rico, Territory, 2016, pp. 1–4.

[15] R. Devooght and H. Bersini, "Long and short-term recommendations with recurrent neural networks," in *Proc. UMAP*, M. Bieliková, Ed, 2017, pp. 13–21.

[16] T. Mikolov I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, New York, NY, USA, 2013, pp. 3111–3119.

[17] T. Mikolov, I. Sutskever, K. Chen, and G. S. Corrado, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. Conf. Recommender Syst.*, New York, NY, USA, 2016, pp. 59–66.

[18] J. Tang and K. Wang, "Personalized Top-N sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, 2018, pp. 565–573.

[19] B. Twardowski, "Modelling contextual information in session-aware recommender systems with neural networks," in *Proc. 10th Conf. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 273–276.

[20] Z. Yu, J. Lian, A. Mahmoody, G. Liu, and X. Xie, "Adaptive user modeling with long and short-term preferences for personalized recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2019, pp. 4213–4219.

[21] C. Xu, P. Zhao, Y. Liu, J. Xu, V. S. S. S. Sheng, Z. Cui, X. Zhou, and H. Xiong, "Recurrent convolutional neural network for sequential recommendation," in *Proc. World Wide Web Conf.*, New York, NY, USA, 2019, pp. 3398–3404.

[22] K. Cho, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, pp. 1724–1734, Oct. 2014.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Conf. Neural Inf. Process Syst.*, Ottawa, Canada, 2014, pp. 3104–3112.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, New York, NY, USA, 2015, pp. 1–15.

[25] S. Wang, "Attention-based transactional context embedding for next-item recommendation," in *Proc. AAAI*, New York, NY, USA, 2018, pp. 2532–2539.

[26] X. He, "Neural collaborative filtering," in *Proc. WWW*, Sydney, NSW, Australia, 2017, pp. 173–182.

[27] K. Chen, "Syntax-directed attention for neural machine translation," in *Proc. AAAI*, New York, NY, USA, 2018, pp. 4792–4799.

[28] L. Chen, H. Zhang, J. Xiao, and L. Nie, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," *CoRR*, vol. abs/1611.05594, pp. 5659–5667, Mar. 2016.

[29] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018.

[30] A. Kumar, "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. ICML*, New York, NY, USA, 2016, pp. 1378–1387.

[31] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2016, pp. 1480–1489.

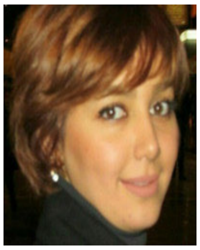
[32] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Madrid, Spain, Dec. 2016, pp. 191–200.

[33] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2011, pp. 1082–1090.

[34] L. Hu, L. Cao, S. Wang, G. Xu, J. Cao, and Z. Gu, "Diversifying personalized recommendation with user-session context," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Sydney, NSW, Australia, Aug. 2017, pp. 1858–1864.

[35] S. Rendle, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. UAI*, Toronto, Canada, 2009, pp. 452–461.

[36] S. Wang, "Perceiving the next choice with comprehensive transaction embeddings for online recommendation," in *Proc. ECML*, vol. 10535, M. Ceci Ed. Skopje, Macedonia: Springer, 2017, pp. 285–302.



SHAHPAR YAKHCHI received the bachelor's and master's degrees (Hons.) in computer software engineering. She is currently pursuing the Ph.D. degree with Macquarie University, Sydney, Australia.

Her research interests are data analysis and recommender systems.

AMIN BEHESHTI is the Director of the AI-Enabled Processes (AIP) Research Centre, the Head of the Data Analytics Research Lab, and a Senior Lecturer in data science with the Department of Computing, Macquarie University. He is the leading author of the *Process Analytics* book, has been recognized as a high-quality researcher in big data/data/process analytics, and served as a keynote speaker, a general-chair, a PC chair, an organization chair, and a program committee member of top international conferences.

SEYED-MOHSSEN GHAFARI received the bachelor's and master's degrees (Hons.) in computer software engineering, and the Ph.D. degree from Macquarie University, Sydney, Australia.

His research interest is data analysis, especially in online social networks.

MEHMET A. ORGUN (Senior Member, IEEE) is working with the Division of Information and Communication Sciences, Department of Computing, Macquarie University, Sydney, NSW, Australia. His current research interests include computational intelligence, multiagent systems, trust and security, temporal reasoning, and formal methods.

GUANFENG LIU (Member, IEEE) received the Ph.D. degree in computer science from Macquarie University, Sydney, Australia, in 2013. He is a Lecturer/Assistant Professor with the Department of Computing, Macquarie University. His research interests include service computing, trust computing, and social network mining.