Information Processing and Management 59 (2022) 102755

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A Convolutional Attention Network for Unifying General and Sequential Recommenders

Shahpar Yakhchi [a,*], Amin Behehsti [a], Seyed-mohssen Ghafari [a], Imran Razzak [b], Mehmet Orgun [a], Mehdi Elahi [c]

[a] *Macquarie University, Sydney, Australia*
[b] *Deakin University, Geelong, Australia*
[c] *Bergen University, Bergen, Norway*

ARTICLE INFO

Keywords:
General recommenders
Sequential recommenders
User purpose modeling
Personal preference modeling
Attention mechanism
Convolutional neural network

ABSTRACT

General recommenders and sequential recommenders are two modeling paradigms of recommender. The main focus of a general recommender is to identify long-term user preferences, while the user's sequential behaviors are ignored and sequential recommenders try to capture short-term user preferences by exploring item-to-item relations, failing to consider general user preferences. Recently, better performance improvement is reported by combining these two types of recommenders. However, most of the previous works typically treat each item separately and assume that each user–item interaction in a sequence is independent. This may be a too simplistic assumption, since there may be a particular purpose behind buying the successive item in a sequence. In fact, a user makes a decision through two sequential processes, i.e., start shopping with a particular intention and then select a specific item which satisfies her/his preferences under this intention. Moreover, different users usually have different purposes and preferences, and the same user may have various intentions. Thus, different users may click on the same items with an attention on a different purpose. Therefore, a user's behavior pattern is not completely exploited in most of the current methods and they neglect the distinction between users' purposes and their preferences. To alleviate those problems, we propose a novel method named, CAN, which takes both users' purposes and preferences into account for the next-item recommendation. We propose to use Purpose-Specific Attention Unit (PSAU) in order to discriminately learn the representations of user purpose and preference. The experimental results on real-world datasets demonstrate the advantages of our approach over the state-of-the-art methods.

1. Introduction

Due to the information explosion, people are surrounded by too many options and services. Therefore, there is a need for a tool to help customers with their decision-making process, find their interested items and alleviate the information overload problem. Recommendation systems have emerged as a platform which automatically recommends a small set of items in order to help users find their desired items in online services. Based on how the users' preferences are modeled, there are two types of

[*] Corresponding author.
E-mail addresses: shahpar.yakhchi@hdr.mq.edu.au (S. Yakhchi), amin.behehsti@mq.edu.au (A. Behehsti), seyed-mohssen.ghafari@hdr.mq.edu.au (S.-m. Ghafari), imran.razzak@deakin.edu.au (I. Razzak), mehmet.orgun@mq.edu.au (M. Orgun), mehdi.elahi@uib.no (M. Elahi).

https://doi.org/10.1016/j.ipm.2021.102755
Received 27 April 2021; Received in revised form 28 August 2021; Accepted 5 September 2021
Available online 7 October 2021

recommenders: general recommenders and sequential recommenders (Dong, Zheng, Zhang, & Wang, 2018; Rendle, Freudenthaler, & Schmidt-Thieme, 2010; Wang et al., 2015).

General recommenders aim to learn what items a user is typically interested in. Matrix factorization is one of the most widely used methods in this setting, which learns user–item interactions in a latent vector space to model the general user preferences (Koren, Bell, & Volinsky, 2009). While sequential recommenders try to capture sequential patterns from previously visited items. Markov Chains-based classic sequential recommenders assume that the next visited item highly depends on the only most recent visited items (Grbovic et al., 2015).

Soon after, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have become dominant paradigms in modeling complex relations over user–item interaction sequences (Hidasi, Karatzoglou, Baltrunas & Tikk, 2016; Tang & Wang, 2018; Yuan, Karatzoglou, Arapakis, Jose, & He, 2019; Zogan, Razzak, Jameel, & Xu, 2021; Zogan, Razzak, Wang, Jameel, & Xu, 2020). Lately, an attention-based approaches such as SHAN (Ying et al., 2018) can surpass the traditional methods due to the strong capability of attention mechanism in highlighting the selective parts in a user and item interaction sequence (Bahdanau, Cho, & Bengio, 2015).

Both the aforementioned classes of approaches have their strengths and shortcomings (Wang et al., 2015). Although general recommenders have been widely adopted to capture long-term user preferences, their performance is limited due to ignoring short-term user preferences. A major advantage of the sequential recommenders is their capability to model sequential dependencies, e.g. a customer who has recently purchased an iPhone is more likely to buy an iWatch next. However, sequential recommenders discard prior user–item interactions within user behaviors, and thus failing to capture general user preferences (Dong et al., 2018).

Based on the above observation, it is better to build a recommender system which benefits from the advantages of both general- and sequential recommenders. FPMC as an example, is a combination of MC and MF, in which instead of using the same transition matrix for all users, an individual transition matrix is used for each user (Rendle et al., 2010). FPMC can well capture both sequential behavior and general taste of the users and then linearly combine them (Rendle et al., 2010). HRM takes one step forward to make progress by using different types of aggregation operations, especially non-linearity into its model (Wang et al., 2015). However, users decision-making pattern is not exploited thoroughly by the existing models as they mainly take each user–item interaction independently and consider each item in a sequence as a separated entity. Hence, the current studies may fail to capture local contexts in a session and ignore a user's purpose which is reflected by a set of clicked successive items in a session. The same user may have various purposes and different users may have different purposes by clicking on the same items. Furthermore, different items within a session may also have different informativeness for revealing purposes and preferences of different users. Therefore, the previous works neglect the hierarchical distinction between user purposes and user preferences, which in turn makes it a challenging task to fully exploit users' decision-making patterns.

Usually, a user's decision-making process is a combination of two sequential steps; a user's main purpose and his/her preference. Taking the shopping event of a user as an example, she/he starts shopping with a specific purpose and then keeps looking into different items until she/he finds items that satisfy her/his preference. Suppose Alice is a Ph.D. student and her previous actions are mostly related to her field of study such as looking for a workshop, and finding an article. Alice has a plan to travel overseas for presenting her work in an international conference. She starts booking her flight and hotel and her next action may be visiting some universities or institutions. While current systems may recommend tourist attractions or car rental companies to her because many users may look for them after booking a hotel and a flight, ignoring her educational purpose of this travel which is hidden inside her long-term interacted item set. Based on this observation, we can see that the user's main purpose may be hidden inside her/his very previous actions, while analyzing her/his very current actions can show her/his preferences on particular items.

The above illustrations reveal the difficulty of capturing collective dependency in a session. In the other words, the next choice of item may not be only affected by a part of current session, but all items need to be taken into consideration as a collective of interacted items may have a particular purpose. Moreover, most of these works have taken user–item relationships into consideration from the static views and the dynamic property of users' preferences are ignored. More importantly, the users' main purposes are not only forgotten, but also there is no difference between the contributions of the same items in modeling preferences of different users. Therefore, how to fully exploit users decision-making process and completely take both the users' motivations along with their current interests are still largely unexplored.

To address the above issues, we propose a novel model called CAN, A convolutional attention network for unifying general and sequential recommenders, which unifies the benefits of both general- and sequential recommenders. CAN consists of two main modules: purpose encoder and preference encoder. In the purpose encoder we first embed users and items into low-dimensional vectors and then use the CNN network to identify user purposes by capturing the local and high-level information of the long-term interacted item set. Then, we propose to use a Purpose-Specific Attention Unit (PSAU) to differently attend to different items and fully exploit different informativeness of different items. Next, at preference encoder we also utilize PSAU in order to learn the items' informativeness in the short-term interacted item set to better understand users' preferences. Lastly, the final user representation is learned through coupling user long-term and short-term preferences. The model's parameters are learned by employing the Bayesian personalized ranking optimization criterion to generate a pair-wise loss function (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2009). From the experiments, we can observe the superiority of our model over the state-of-the-art algorithms on two datasets. The **key contributions** of the paper are summarized as follows:

- We introduce a unified framework, named CAN, integrating a CNN network and attention-based PSAU module to model the users' purposes and personal preferences.

- We propose a Purpose-Specific Attention Unit, PSAU, which takes user embedding as the query vector of the purpose- and personal preference-level attention networks to differentially attend to important items according to user purposes and preferences.
- We use the PSAU in both the long- and short-term interacted item set to generate a high-level hybrid user representation.
- We conduct extensive experiments on two real-world datasets. The experimental results demonstrate the superiority of our proposed model compared to the state-of-the-art methods.

The rest of the paper is organized as follows: we discuss the related works in Section 2. The proposed methodology and our experiments are presented in Section 3 and Section 4, respectively, before we conclude the paper in Section 5.

2. Related work

Based on different aspects of user behavior, there are two types of paradigms that are applied to recommendation tasks: general recommender and sequential recommender. Both paradigms have strengths and weaknesses, which in the following discussion, we will analyze each paradigms.

2.1. General recommender

The main goal of general recommenders is to discover the users' long-term preferences by exploiting their past items interactions. Early works on this kind of recommenders mostly use Collaborative Filtering (CF) to model users' preferences (Koren & Bell, 2011; Sarwar, Karypis, Konstan, & Riedl, 2001). Matrix factorization (MF) is one of the widely adopted techniques in CF, which aims to learn user and item latent vectors in order to compute a user's preference on an item (Koren et al., 2009; Salakhutdinov & Mnih, 2007). Basically there are two different types of data with which MF-based approaches deal: explicit feedback, e.g., given ratings, and implicit feedback, e.g., mouse clicking. The first one treats making a recommendation as a rating prediction problem, referring to the approaches that try to predict users' preference scores by utilizing their rating patterns (Koren et al., 2009). Unlike approaches belonging to the first class, implicit feedback oriented methods formulate making a recommendation as a ranking problem based on the idea of the Learning to-Rank technique (Karatzoglou, Baltrunas, & Shi, 2013). Although general recommenders may better model the long-term user preferences, their performance is limited due to ignoring short-term user preferences.

2.2. Sequential recommender

Different from general recommenders, sequential recommenders try to understand the sequential user behaviors and model the short-term user preferences (Wang et al., 2019). Markov chain (MC) has been known as a typical solution in this setting. For instance, SPMC exploits both sequential and social information to make a more personalized recommendation model (Cai, He, & McAuley, 2017). In the past few years, deep learning methods have shown their great capability in modeling the complex interactions between users and items. Among deep neural networks techniques, Recurrent Neural Network (RNN) is one of the widely adopted methods in sequential recommenders due to its capability in sequence modeling. Apart from using basic RNN (Hidasi, Karatzoglou et al., 2016; Zhang et al., 2014), improved architectures like long short-term-memory (LSTM) (Wu, Ahmed, Beutel, Smola, & Jing, 2017) and gated recurrent unit (GRU) (Hidasi, Quadrana, Karatzoglou & Tikk, 2016) have also been introduced to better model dependencies in a longer sequence. Different from RNN, Convolutional Neural Network (CNN) stores the embedding of the user–item interaction sequences in a matrix and then treats this matrix as an image (Tang & Wang, 2018; Yuan et al., 2019). Although the basic deep neural networks (i.e., RNN, CNN) have shown a great success in modeling sequential dependencies, they may have some shortcomings in modeling complex relations between users and items. Thus, three advanced models have been introduced to overcome this problem: (i)*attention mechanism*: by more focusing on relevant and important interactions in a sequence (Kang, Wan, & McAuley, 2018; Ying et al., 2018); (ii) *memory networks*: by incorporating an external memory matrix (Chen et al., 2018; Hu, He, Sha, & Niu, 2019); and (iii) *mixture models*: by combining the strength of the current deep neural models (Tang et al., 2019).

Inspired by the outperformance of Transformer (Naseem, Razzak, Khushi, Eklund, & Kim, 2021; Naseem, Razzak, Musial, & Imran, 2020; Wolf et al., 2020; Zogan et al., 2021) in NLP tasks, SRs have motivated to use self-attention technique to better capture sequential dependency. BERT4Rec (Sun et al., 2019) for instance, has used the deep bidirectional self-attention algorithm to model the sequences of users' behaviors. Except these methods, Graph Neural Network (GNN) has emerged as a solid structure With the strong capability of modeling complex transition patterns of items (Wu, Zhang, Sun, & Cui, 2020). SURGE is an example of GNN-based model,in which different types of users' preferences are modeled. The authors have also used graph network to model users' dynamic behavior.

While sequential recommender models are good at capturing the sequential dependency, they mostly recommend items similar to those that a user currently visited and the general user preference is ignored.

2.3. Unified recommender

There are some recent attempts to combine both general- and sequential recommenders in a unified system. For instance, FPMC is one of the pioneering works in the literature which fuse MF and MC into one model in order to learn the both users' long- and short-term preferences (Rendle et al., 2010). Soon after, Hierarchical Representation Model (HRM) is proposed by Wang et al. (2015) which non-linearly models both sequential behaviors and users' general taste to make a better recommendation. While FPMC and HRM have exploited user long-term preferences to improve the performance of sequential recommenders, CoFactor benefits from integrating a co-occurrence item-to-item matrix into an MF model (Liang, Altosaar, Charlin, & Blei, 2016). BINN which is proposed by Li et al. (2018), is another attempt in unifying both types of users' preferences. The authors have stated that different types of users' actions (e.g., browse, click, collect, cart, and purchase) need to be treated differently. Their proposed model consists of two main components: Neural Item Embedding and Discriminative Behaviors Learning. At first component, BINN tries to find the items' similarities by analyzing users' sequential behaviors. While at second component, two alignments Session Behaviors Learning (SBL) and Preference Behaviors Learning (PBL) are introduced to learn discriminative behaviors (Li et al., 2018). Although BINN can record a significant improvement over several state-of-the-art models, it uses LSTM for discriminative behaviors learning part, which may limit the performance of their recommender system as it may not be able to capture the dynamic property of users' preferences. Moreover, BINN only considers purchase behavior for modeling users' historical preferences. This may not only cause in losing some useful information by exploiting other types of users' behaviors (e.g., click, add to cart, and etc.), but also may fail to learn latent users' purposes which is hidden in a collection of successive user–item interaction

Our model falls under this category and the difference of our method over the existing works can be seen in three different aspects. First, the main purpose of a user's shopping behavior is ignored in most of the current unified recommenders, which in turn may lead to performance degradation as it plays an important role in the user's decision-making. Second, current methods mostly consider the same informativeness for clicked items in the sequence of user–item interaction, which may result in uncompleted exploited short-term users' preferences. Third, we propose to use a PSAU component to apply in both long-and short term interacted item set in order to dynamically recognize important items for recommendation based on user preferences.

3. Proposed methodology: Convolutional attention network

Before introducing the details of our proposed model, we first define and formulate the research problem and basic concepts and then we present the optimization procedures.

3.1. Notations and problem formulation

In this section, we investigate the next-item recommendation problem with implicit feedback data. Let us consider $U = \{u_1, u_2, \dots, u_{|u|}\}$ as the user set and $V = \{v_1, v_2, \dots, v_{|v|}\}$ as the item set, where $|u|$ and $|v|$ are the total number of users and items, respectively. For each user u, we define $G^u = \{S_1^u, S_2^u, \dots, S_T^u\}$ as her/his transaction history, where T is the total number of sessions and each session $S_t^u \subseteq V(t \in [1, T])$, where S_t^u represents a set of interacted items for users u at time step t. We denote S_t^u as the short-term preference of user u (i.e., her/his sequential behavior) at specific time step t. In addition to short-term preference, long-term preference of user u is also important for identifying items that users will interact in the near future. Therefore, we consider $G_{t-1}^u = \bigcup_{t=1}^{t-1} S_t^u$ to reflect the long-term preference of user u (i.e., general preference), where G_{t-1}^u is a set of interacted item sets before time step t. For the rest of this paper, we call G_{t-1}^u and S_t^u as the long- and short-term interacted item sets regarding time step t, respectively. Given user u transaction history G^u, we aim to predict the next items which the user will likely purchase by learning her/his long- and short-term preferences.

3.2. Modeling and learning

The framework of CAN is illustrated in Fig. 1. As shown in Fig. 1, our proposed model consists of two main modules: (1) *the purpose encoder* and (2) *the preference encoder*. The first module aims to learn the main purpose of the long-term interacted item set for the users. It takes a set of user–item interactions in the long-term item set and embeds them into low-dimensional vector representations, and then these vectors are passed to a CNN network to effectively capture the local contextual information of the sequence in order to identify a user's main purpose. Then, we propose to use a Purpose-Specific Attention Unit (PSAU) to differentially attend to the users' main purposes. The reason behind applying PSAU is that different users may have different purposes of buying the same items. For instance, both users *a* and *b* buy item *i*, while user *a* buys this item as a souvenir for her friend, but user *b* is interested in this item for herself. Then, we propose to use a Purpose-Specific Attention Unit (PSAU) to differentially attend to the users' main purposes. The reason behind applying PSAU is that different users may have different purposes of buying the same items. For instance, both users *a* and *b* buy item *i*, while user *a* buys this item as a souvenir for her friend, but user *b* is interested in this item for herself. Thus, we propose to use PSAU in order to incorporate the informativeness of purchasing the same items for different users. The next module is (2) *the preference encoder*, which aims to learn the users' current preferences. The same user may have different preferences and each item may be more or less informative for that specific preference. Hence, PSAU is also applied here to discriminate each item informativeness.

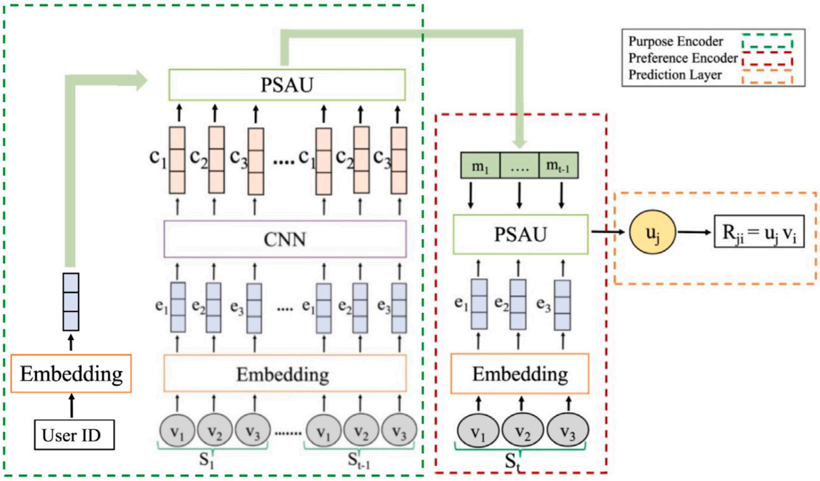

Fig. 1. The architecture of CAN, which consists of two main modules purpose encoder and preference encoder.

3.3. Purpose encoder

Our purpose encoder module has three core components: (i) embedding look-up, (ii) convolutional neural network and (iii) Purpose-Specific Attention Unit (PSAU). Usually users' decision-making process consists of two sequential vital steps, namely, users' main purposes and users' preferences. Normally, people start shopping with an intention and then view different items until they find interesting items that satisfy their preferences. In this block, we aim to first convert a session of items into a sequence of low-dimensional dense vectors. Then, we use a convolutional neural network for capturing local information. Since local contexts within a set of interacted items may imply a user's purpose. For instance, Julia wants to have a Halloween party. She goes to a shopping and puts a set of {*hanging ghost*, *pumpkin*, *lollipop*, *plastic blood bag*} together. In this collection of items, the local combination of the "hanging ghost", and "plastic blood bag" may be more important to show the user's main intention of this shopping. Therefore, we use a CNN network here to learn contextual representations of a set of items. Finally, at this block, PSAU unit is applied to distinguish the level of informativeness of different items in revealing the users' motivations of purchasing a set of items together. The reason behind using PSAU unit in purpose encoder is that different items may have different level of contributions in presenting a user's main purpose, and the same words may have different informativeness for the recommendation of different users. Based on this observation, we need to identify important items in demonstrating shopping's purpose of different users, and thus the personalized attention-based network is proposed to apply in this block.

Embedding Look-up. First, we use embedding look-up to embed user and item IDs (i.e., one-hot representations) into two continuous low-dimensional spaces, where e_i represents the item embedding vector of item i, and u_j denotes the user embedding vector of user j. The embedding matrix is denoted by $E = [e_1, e_2, \dots, e_i]$, $E \in R^{|V| \times D}$, where D and $|V|$ represent the embedding dimension and the total number of items, respectively. The matrix $U \in R^{D \times |U|}$ is the user embedding matrix, where u_j denotes the user embedding vector of user j.

Convolutional Neural Network (CNN). Second, we employ CNN to learn contextual information of user–item interactions (Kim, 2014). CNN is one of the deep learning techniques with a great capability in capturing local information (Wu, Wu, Liu et al., 2019). Therefore, we use CNN to capture the user's main purpose in the long-term item set. Next, we perform a convolution operator on the matrix E as the concatenation of the items' embedding vectors. Let $K_w \in R^{N_f \times (2K+1)D}$, and $b_w \in R^{N_f}$ denote the parameters of CNN network, in which K_w is the kernel and b_w represents the bias parameters. N_f is the number of CNN filters, and $2K + 1$ is the window size of CNN. Then, c_i illustrates the contextual representation of item i:

$$c_i = ReLU(K_w \times e_{\lfloor i-k \rfloor : \lfloor i+k \rfloor} + b_w), \tag{1}$$

where $e_{\lfloor i-k \rfloor : \lfloor i+k \rfloor} \in G^u_{t-1}$ is the combination of the embedding vectors of items from position $\lfloor i - k \rfloor$ to position $\lfloor i + k \rfloor$. We use ReLU as our non-linear activation function.

Purpose-Specific Attention Unit (PSAU). The last component in the *purpose encoder* is the Purpose-Specific Attention Unit (PSAU), to differentially attend to important items according to user purposes. In a sequence of user–item interactions, each item may be more or less informative for learning users' purpose representation. For instance, imagine {*pizza bread*, *pepperoni*, *cheese*} as a set of purchased items together for making a pizza. In this shopping basket, *pizza bread* is more informative to represent the users' purposes than *cheese*. Furthermore, different users may purchase the same items for a different purpose. Therefore, based on these observations, identifying the contributions of different items for different users play an important role in personalized recommendation. However, most of the current approaches use a classic attention network which computes attention score as a weighted sum over the embeddings of items and a fixed attention query vector, ignoring users' main purposes. To learn the informativeness of each item for different users, we propose to employ the PSAU cell to identify the most informative items related to the users' main purpose within a user–item interaction sequence. PSAU first takes the embedded user-ID vector $u_j^{'} \in R^{D_u}$, where D_u is the user embedding dimension. Then, we use a dense non-linear layer to transform the embedding vector $u_j^{'}$ to the purpose-level user preference vector p_j, which is formulated as:

$$p_j = ReLU(W_1 \times u_j^{'} + b_1), \tag{2}$$

where $W_1 \in R^{D_u \times D_p}$ and $b_1 \in R^{D_p \times 1}$ are model parameters, and D_p is the preference vector dimension. Next, we denote α_j as the attention score of item j, which can extract the level of informativeness of each item according to the users' main purpose. The attention score α_j, is calculated based on the interaction between the user preference vector and the contextual item representations, which is shown as :

$$a_i = c_i^T tanh(W_2 \times p_j + b_2), \tag{3}$$

$$\alpha_i = \frac{exp(a_i)}{\sum_{i \in G_{t-1}^u} exp(a_i)}, \tag{4}$$

where $W_2 \in R^{D_p \times N_f}$ and $b_2 \in R^{N_f \times 1}$ are model parameters. Next, the user's main purpose representation m_i is modeled as a weighted sum of the contextual representation of item i with their attention scores. Formally, this representation can be formulated as follows:

$$m_i = \sum_{i \in G_{t-1}^u} \alpha_i c_i \tag{5}$$

3.4. Preference encoder

As it is clear from Fig. 1, PSAU is also employed in the preference encoder module in order to learn an informative user short-term preference representation. Different users may have different preferences by clicking on the same items and different items are more or less informative for modeling user preferences. Hence, we use PSAU here as well to model the different informativeness of the same items for different users. Hence, we first take the item embedding $e_i \in S_t^u$ in a short-term interacted item set to model a user preference vector p_d, which is shown as:

$$p_d = ReLU(W_3 \times e_i + b_3), \tag{6}$$

where $W_3 \in R^{D_u \times D_q}$ and $b_3 \in R^{D_q \times 1}$, and D_q is the preference query size. Next, the attention weight α_i' represents the level of informativeness of item i in the short-term user preference, which can be computed by the interactions between the user's purpose representation and user preference vector. Then, the softmax function is used to normalize the attention weight, which is calculated as follows:

$$a_i' = m_i^T tanh(W_4 \times p_d + b_4), \tag{7}$$

$$\alpha_i' = \frac{exp(a_i)}{\sum_{i \in S_t^u} exp(a_i)} \tag{8}$$

where $W_4 \in R^{D_q \times N_f}$ and $b_4 \in R^{N_f \times 1}$ are model parameters. Finally, the contextual user representation u_j is computed as follows:

$$u_j = \sum_{i \in S_t^u} a_i' m_i \tag{9}$$

3.5. Prediction layer

After the final user representation u_j has been learned, we calculate the inner product of it and item representation v_i in order to compute the user preference score R_{ij} as follows:

$$R_{ij} = u_j v_i \tag{10}$$

Next, followed by Rendle et al. (2009), we utilize a pair-wise loss function in order to train our model. We aim to provide a ranked list of the next items to be recommended, where observed items should have higher score than unobserved ones. Let $D = \{(u, v_i, v_j) : u \in U, v_i \in G^u, v_j \in V/G^u\}$ denote the set of pair-wise training instances. Then we train our model by maximizing a posterior (MAP) as follows:

$$\arg\min_{\Theta} \sum_{(u,v_i,v_j) \in D} - \ln \sigma(R_i^u - R_j^u) + \lambda_{uv} ||\theta_{uv}||^2 + \lambda_a ||\theta_a||^2 \tag{11}$$

where θ_{uv} ={U, V} is the set of user and item embedding parameters, $\theta_a = \{W_1, W_2, W_3, W_4\}$ is the set of weights of attention networks, λ_{uv} and λ_a are the regularization parameters, and σ is a logistic function.

Table 1
Statistics of our datasets.

Dataset	Users	Items	Sessions length	Training sessions	Testing sessions	Interactions
Tmall	20,716	25,143	2.81	71,998	3565	85,432
Gowalla	15,254	13,052	2.99	128,115	3611	94,654

4. Experiments

In this section, we present experimental evaluation of proposed recommender and compare the performance with state-of-the-art baseline methods such as BPR (Rendle et al., 2009),FOSSIL (He & McAuley, 2016), Caser (Tang & Wang, 2018), FPMC (Rendle et al., 2010), HRM (Wang et al., 2015), GRU4Rec (Hidasi, Karatzoglou et al., 2016), NARM (Li et al., 2017), SHAN (Ying et al., 2018), and MEANS (Hu et al., 2019).

4.1. Datasets and experimental setting

We conduct our experiments on two widely used datasets Tmall[1] and Gowalla[2] The Tmall dataset records the user's consumption and browsing behavior during the user's shopping process. It has too many interactions of 424,170 users on 1,090,390 items within six months. In this dataset there are four kinds of activities: click, collect, add-to-cart and purchase. Following the settings in Ying et al. (2018) and Hu et al. (2017) we only consider the users' purchase activities in our experiment. The Gowalla aggregates the users' check-in information from the location-based social networking website, Gowalla from February 2009 to October 2010. Gowalla consists of 6,442,890 number of total check-ins, where each record consists of user id, timestamp, GPS location and POI id. We follow the same preprocessing procedure as in SHAN (Ying et al., 2018) and we treat user transactions or check-ins in one day as a session. Sessions with only one item and items with less than 20 time observations are removed from datasets. We randomly select the sessions in the last week as a test set, and the rest are used for training. In addition, we randomly keep one item in each session as the next item to be predicted. The statistics of the datasets after the preprocessing stage are illustrated in Table 1.

Baselines: To demonstrate the effectiveness of our method, we compare it with the following representative state-of-the-art recommender systems built on various frameworks including RNN, CNN, attention models and memory networks:

- TOP: This method identifies the top popular items based on the number of occurrences in each session in the training data, and then recommends those items in test data.
- BPR (Rendle et al., 2009): This is a state-of-the-art baseline for binary implicit feedback through pairwise learning to rank.
- FOSSIL (He & McAuley, 2016): This method integrates factored item similarity with a Markov chain to model the user's long- and short-term preferences.
- Caser (Tang & Wang, 2018): This is a state-of-the-art model, which uses CNN for sequence embedding.
- FPMC (Rendle et al., 2010): This is a combination of MF and MC model in order to learn user preferences.
- HRM (Wang et al., 2015): This model non-linearly learns both sequential behavior and users' general taste to make a better recommendation.
- GRU4Rec (Hidasi, Karatzoglou et al., 2016): This is a state-of-the-art sequential recommender, which applies modern recurrent neural network (GRU) to be able to model the whole session.
- NARM (Li et al., 2017): This is a sequential recommender which combines a recurrent neural network with an attention network.
- SHAN (Ying et al., 2018): This is a state-of-the-art sequential recommender, which employs a two-layer hierarchical attention network to learn long- and short-term preference.
- MEANS (Hu et al., 2019): This model first operates a max-pooling technique on the most recent sessions and the results are stored into an external memory. Then the attention mechanism is applied to learn long-term user preference. Finally, at prediction layer a recommendation is made by learning a mixture of long- and short-term preference.

Evaluation Metrics. Similar to the previous work (Ying et al., 2018), we also adopt several widely used evaluation metrics AUC, Recall@N, and Precision@N to evaluate the performance of our model, where $N \in \{5, 10, 20\}$. Recall measures the proportion of the right ranked items overall top-k recommendation items in a list, while Precision measures the proportion of results which are relevant. Different from both above metrics, AUC computes how highly predicted items are ranked over all items. The larger metric scores show better model performance. Due to the space limitation, we name Recall and Precision as Re and Pre in the rest of the paper, respectively.

Parameter Settings. We set the item embedding and user embedding dimensions, D, to 100, which is a trade-off between the performance of recommendation and the computation cost for both datasets. Similar to the Wu, Wu, An et al. (2019), we set the number of CNN kernels N_f and the window size to 400 and 3, respectively. We apply dropout strategy (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) to each layer of CNN in order to avoid overfitting. The dropout rate is set to 0.2, the

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=53

[2] https://snap.stanford.edu/data/loc-gowalla.html

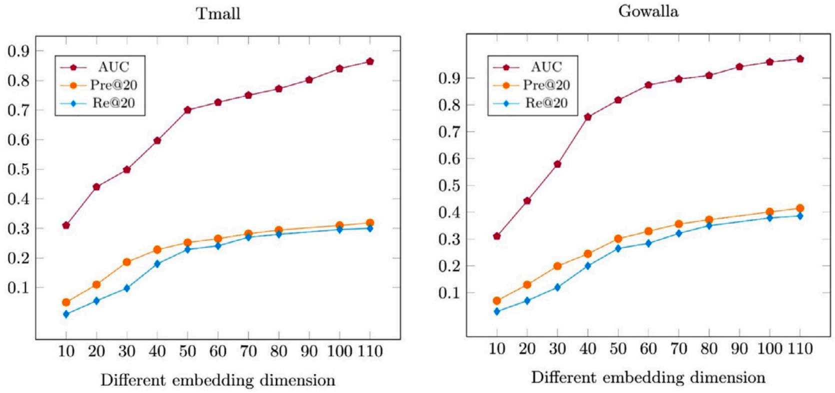

Fig. 2. Impact of different embedding dimension on Gowalla and Tmall datasets. In each figure, we have shown the impact of different embedding sizes on three evaluation metrics AUC, Precision and Recall.

Table 2
Impact of different regularization at Recall@20.

Dataset	λ_{uv} λ_α	0	1	10	50
Tmall	0.01	0.085	0.126	0.143	0.146
	0.001	0.079	0.124	0.138	0.139
	0.0001	0.073	0.111	0.129	0.133
Gowalla	0.01	0.250	0.344	0.355	0.372
	0.001	0.321	0.397	0.423	0.432
	0.0001	0.342	0.421	0.452	0.461

Table 3
Impact of different session lengths.

Tmall			
Methods	AUC	Re@20	Pre@20
CAN-S	0.745	0.196	0.213
CAN-L	0.889	0.221	0.282

Gowalla			
Methods	AUC	Re@20	Pre@20
CAN-S	0.814	0.219	0.263
CAN-L	0.916	0.298	0.342

batch size is empirically set to 50, the sizes of both the user purpose query D_p and preference query D_q are set to 200. The learning rate η is 0.01. Items and users dimensions are randomly initialized with normal distribution $N(0, 0.01)$ and then learned during the training process. The attention parameters are initialized with the $U(-\sqrt{\frac{3}{k}}, \sqrt{\frac{3}{k}})$.

4.2. Impact of hyper-parameters

In this subsection, we investigate the impact of hyper-parameters on the performance of CAN. 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. Based on Table 2, the performance of CAN is gradually increased when $\lambda_a > 0$ in both Tmall and Gowalla datasets, which indicates the effectiveness of applying attention mechanism in our model. We also test the impact of different embedding dimensions, D, related to the user, item and hidden layer parameters in attention network. As it is clear from Fig. 2, the higher embedding dimension can result in better AUC, Recall@20, and Precision@20 as it can learn more latent features form user and item as well as their interactions through attention mechanism. From this figure, a slight improvement is recorded while the embedding dimension is increased from 100, and thus we set the embedding size to 100.

4.3. Impact of different sessions lengths

We examine the performance of CAN under different sequence lengths as the local features captured by CNN network may be different. Table 3 demonstrates the results of our investigation. We consider sessions with less than 3 items as a short session and

Table 4
Impcat of CAN modules.

Tmall			
Methods	AUC	Re@20	Pre@20
CAN-PurEn	0.817	0.256	0.278
CAN-PreEn	0.781	0.210	0.264
CAN	**0.915**	**0.317**	**0.322**

Gowalla			
Methods	AUC	Re@20	Pre@20
CAN-PurEn	0.924	0.284	0.312
CAN-PreEn	0.899	0.256	0.299
CAN	**0.989**	**0.392**	**0.401**

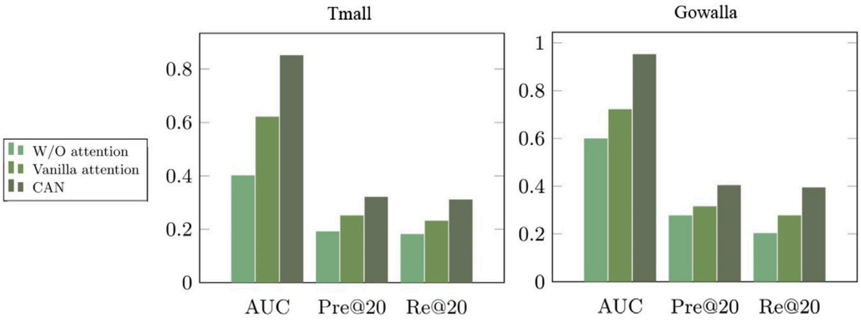

Fig. 3. Impact of PSAU on Gowalla and Tmall datasets. W/O attention means no attention mechanism is used.

treat sessions with more than 3 items as a long session. The percentage of short and long sessions are 90%, 10% and 83%, 17% in both Tmall and Gowalla datasets, respectively. In Table 3, CAN-S refers to a situation where short sessions are modeled, while only long sessions are considered in CAN-L. From this table, we can have several observation. First, the performance of both CAN-L and CAN-S are too close. Second, CAN-L performs slightly better than CAN-S with respect to AUC, Pre@20, and Re@20 in both Tmall and Gowalla datasets. This is probably because of capturing the more contextual features through long sessions. Third, the performance of CAN-L is still too close to the overall performance of our model.

4.4. Impact of CAN modules

In this experiment, we aim to test the performance of two modules, i.e., purpose encoder and preference encoder in Table 4. CAN-PurEn means only user purpose module is used, while CAN-PreEn only considers a user's preference. According to Table 4, we can have several observations. First, the CAN-PurEn can effectively improve the performance of our approach, as it can help our model CAN to achieve the higher performance compared to the state-of-the-art models. This may be due to the capturing the local patterns in a long-term interacted item set through CNN and highlighting the important items according to user preferences by PSAU cell. Second, the CAN-PreEn is also another effective module in our model, which indicates a significant improvement in the performance of CAN. This is probably because items in a short-term interacted item set usually have different informativeness and recognizing the important items can help better modeling user representations. Third, generally CAN performs better than two single modules. It demonstrates that combining these two modules is helpful in learning user representation and predicting next items.

4.5. Impact of PSAU component

In order to verify the effectiveness of the PSAU component in our model, we compare the performance of our model in the presence and absence of the PSAU cell. As it is clear from Fig. 3, we have different findings: (1) applying attention mechanism can show better performance compared to the model without attention. The reason behind this observation may be because of assigning different weight to different items, and attention mechanism can discover the important items in a user–item interaction; (2) our model CAN consistently outperforms the model without attention mechanism and vanilla attention. The reason behind this observation may be because of assigning different score to the same items for modeling different users, while vanilla attention assigns a fixed score and thus is not able to differentiate the importance of the same items in modeling the different user preferences. Attention mechanism pays same attention to each item by computing the attention weights only based on the input representation

Table 5
The performance of different methods regarding the evaluation metrics in Tmall dataset.

Datasets	Tmall						
Metrics	Re5	Re10	Re20	Pre5	Pre10	Pre20	AUC
Top	0.021	0.052	0.084	0.051	0.062	0.074	0.392
BPR	0.024	0.090	0.122	0.062	0.069	0.074	0.481
Fossil	0.110	0.120	0.125	0.083	0.088	0.092	0.691
Caser	0.041	0.049	0.052	0.100	0.108	0.115	0.701
FPMC	0.050	0.055	0.061	0.118	0.125	0.130	0.742
HRM	0.060	0.065	0.070	0.121	0.129	0.133	0.751
GRU4Rec	0.062	0.065	0.069	0.138	0.145	0.149	0.762
NARM	0.063	0.068	0.073	0.141	0.149	0.159	0.781
SHAN	0.071	0.076	0.079	0.155	0.160	0.166	0.789
MEANS	0.074	0.079	0.082	0.163	0.172	0.177	0.790
CAN	**0.201**	**0.278**	**0.317**	**0.200**	**0.260**	**0.322**	**0.915**

Table 6
The performance of different methods regarding the evaluation metrics in Gowalla dataset.

Datasets	Gowalla						
Metrics	Re5	Re10	Re20	Pre5	Pre10	Pre20	AUC
Top	0.038	0.048	0.059	0.061	0.066	0.071	0.711
BPR	0.069	0.074	0.081	0.077	0.082	0.089	0.800
Fossil	0.215	0.298	0.312	0.091	0.095	0.099	0.810
Caser	0.075	0.083	0.089	0.114	0.119	0.124	0.815
FPMC	0.115	0.129	0.138	0.127	0.133	0.142	0.820
HRM	0.119	0.125	0.145	0.150	0.157	0.161	0.824
GRU4Rec	0.121	0.135	0.141	0.155	0.160	0.165	0.828
NARM	0.130	0.136	0.140	0.156	0.159	0.163	0.830
SHAN	0.135	0.140	0.144	0.163	0.169	0.175	0.832
MEANS	0.142	0.150	0.158	0.170	0.175	0.180	0.840
CAN	**0.250**	**0.312**	**0.392**	**0.360**	**0.399**	**0.401**	**0.989**

sequence via a fixed vector, and thus the user preferences are not incorporated. While in contrast to vanilla attention, the attention scores in PSAU are computed based on the interaction between the user preference vector and the contextual item representations. Therefore, our model can highlight important items in user's purpose according to her/his personal preference, which in turn can help in better user representation learning. Based on these results, we can validate the effectiveness of the PSAU cell in our approach.

4.6. Overall performance comparison

In this subsection, we compare the results of our model with the other state-of-the-art approaches in both Tmall and Gowalla datasets, which is summarized in Tables 5 and 6. This table illustrates that:

1. According to Tables 5 and 6, where the best result in each row is highlighted in boldface, our proposed model significantly and consistently outperforms all state-of-the-art models in terms of Precision@N, Recall@N and AUC in different Ns in both Tmall and Gowalla datasets. Specifically, compared to MEANS which is the best baseline in terms of all evaluation metrics, CAN has shown 14% and 16% improvements with respect to the AUC on Tmall and Gowalla datasets, respectively.This indicates the effectiveness of CAN, which can recognize important items in users' purposes according to their preferences through CNN network and PSAU component.
2. Deep learning methods using attention network (CAN, MEANS, SHAN, and NARM) show better performance compared with the methods without attention mechanism. The reason may be due to the capability of attention mechanism in recognizing the most important items in user and item interaction.
3. Overall, all unified approaches (CAN, MEANS, SHAN, NARM, HRM, FPMC, and Fossil) outperform the best general- and sequential recommenders such as BPR and GRU4Rec, respectively.
4. Among all unified approaches, after CAN, MEANS outperforms others like SHAN, NARM, HRM, FPMC, and Fossil. While the performance of MEANS and SHAN are too close, MEANS can achieve around 5% and 9% improvement compared to SHAN at Recall@20 in Tmall and Gowalla datasets, respectively. This indicates the effect of using external memory to store long-term user and item interaction after a max-pooling operation. However, MEANS cannot effectively model the local contexts in the long term user preference, and is not able to find important items for revealing purposes and preferences of different users. Moreover, although MEANS uses attention mechanism, it cannot model the informativeness of different items. Different from all mentioned approaches, our proposed model can dynamically find important items according to user purposes and preferences.

5. Conclusion

In this paper, we propose a novel unified recommendation approach which consists of a Purpose-Specific Attention Unit (PSAU). In our approach, CAN, we learn the users' purposes in long-term interacted item set by using CNN. We use PSAU cell to recognize important items in users' purposes according to their preferences. Since same items may have different informativeness for different users, we use PSAU in short-term interacted item set as well to model users' preferences. The extensive experimental results on the real-world datasets validate the effectiveness of our approach compared to other state-of-the-art methods. As our future work, we aim to take contextual information into sequential recommenders in order to make a more accurate recommendation. Furthermore, modeling different heterogeneous actions can be another direction for our future work.

CRediT authorship contribution statement

Shahpar Yakhchi: Carried out experiment. **Amin Behehsti:** Worked on writing the script. **Seyed-mohssen Ghafari:** Worked on writing the script. **Imran Razzak:** Worked on analysis and writing. **Mehmet Orgun:** Worked on analysis and writing. **Mehdi Elahi:** Worked on analysis and writing.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Third int. conf. on learning representations*.

Cai, C., He, R., & McAuley, J. J. (2017). SPMC: socially-aware personalized Markov chains for sparse sequential recommendation. CoRR.

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., et al. (2018). Sequential recommendation with user memory networks. In *Proc. of the 11th int. conf. on web search and data mining* (pp. 108–116). ACM.

Dong, D., Zheng, X., Zhang, R., & Wang, Y. (2018). Recurrent collaborative filtering for unifying general and sequential recommender. In *Proceedings of the 27th int. joint conf. on artificial intelligence* (pp. 3350–3356). ijcai.org.

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., et al. (2015). E-commerce in your inbox: Product recommendations at scale. In *Proc. of the 21th SIGKDD int. conf. on knowledge discovery and data mining, Australia* (pp. 1809–1818). ACM.

He, R., & McAuley, J. J. (2016). Fusing similarity models with Markov chains for sparse sequential recommendation. CoRR, arXiv:1609.09152.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *4th int. conf.on learning representations*.

Hidasi, B., Quadrana, M., Karatzoglou, A., & Tikk, D. (2016). Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proc. of the 10th ACM conf. on recommender systems* (pp. 241–248). ACM.

Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., & Gu, Z. (2017). Diversifying personalized recommendation with user-session context. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence* (pp. 1858–1864). ijcai.org.

Hu, C., He, P., Sha, C., & Niu, J. (2019). Memory-augmented attention network for sequential recommendation. In R. Cheng, N. Mamoulis, Y. Sun, & X. Huang (Eds.), *11881*, *Int. conf. on the web information systems engineering* (pp. 228–242). Springer.

Kang, W., Wan, M., & McAuley, J. J. (2018). Recommendation through mixtures of heterogeneous item relationships. CoRR, arXiv:1808.10031.

Karatzoglou, A., Baltrunas, L., & Shi, Y. (2013). Learning to rank for recommender systems. In *Seventh ACM conf. on recommender systems* (pp. 493–494). ACM.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proc. of the 2014 conf. on empirical methods in natural language processing* (pp. 1746–1751). ACL.

Koren, Y., & Bell, R. M. (2011). *Recommender systems handbook* (pp. 145–186). Springer.

Koren, Y., Bell, R. M., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Journal of Computer*, *42*(8), 30–37.

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *Proc. of the conf. on information and knowledge management* (pp. 1419–1428). ACM.

Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., & Chen, E. (2018). Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. CoRR, arXiv:1808.01075.

Liang, D., Altosaar, J., Charlin, L., & Blei, D. M. (2016). Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proc. of the 10th ACM conf. on recommender systems* (pp. 59–66). ACM.

Naseem, U., Razzak, I., Khushi, M., Eklund, P. W., & Kim, J. (2021). Covidsenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis. *IEEE Transactions on Computational Social Systems*.

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, *113*, 58–69.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of the 25th conf. on uncertainty in artificial intelligence* (pp. 452–461). AUAI Press.

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In *Proc. of the 19th int. conf. on world wide web* (pp. 811–820). ACM.

Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. In *Proc. of the 21th annual conf. on neural information processing systems* (pp. 1257–1264). Curran Associates, Inc..

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th int. world wide web conf.* (pp. 285–295). ACM.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450).

Tang, J., Belletti, F., Jain, S., Chen, M., Beutel, A., Xu, C., et al. (2019). Towards neural mixture recommender for long range dependent user sequences. In *The world wide web conf.* (pp. 1782–1793). ACM.

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proc. of the 11th int. conf. on web search and data mining* (pp. 565–573). ACM.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. A. (2019). Sequential recommender systems: Challenges, progress and prospects. In *Proc. of the 28th int. joint conf. on artificial intelligence* (pp. 6332–6338). ijcai.org.

Wang, P., et al. (2015). Learning hierarchical representation model for NextBasket recommendation. In *Proc. of the 38th int. SIGIR conf. on research and development in information retrievalaug* (pp. 403–412).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45).

Wu, C., Ahmed, A., Beutel, A., Smola, A. J., & Jing, H. (2017). Recurrent recommender networks. In *Proc. of the 10^{th} int. conf. on web search and data mining* (pp. 495–503). ACM.

Wu, C., Wu, F., An, M., Huang, J., Huang, Y., & Xie, X. (2019). NPA: neural news recommendation with personalized attention. In *Proc. of the 25^{th} SIGKDD int. conf. on knowledge discovery and data mining* (pp. 2576–2584). ACM.

Wu, C., Wu, F., Liu, J., He, S., Huang, Y., & Xie, X. (2019). Neural demographic prediction using search query. In *12^{th} ACM int. wsdm conf.* (pp. 654–662). ACM.

Wu, S., Zhang, W., Sun, F., & Cui, B. (2020). Graph neural networks in recommender systems: A survey. CoRR, arXiv:2011.02260.

Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xu, G., Xie, X., et al. (2018). Sequential recommender system based on hierarchical attention networks. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 3926–3932). ijcai.org.

Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J. M., & He, X. (2019). A simple convolutional generative network for next item recommendation. In *Proc. of the 12^{th} int. conf. on web search and data mining* (pp. 582–590). ACM.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., et al. (2014). Sequential click prediction for sponsored search with recurrent neural networks. CoRR.

Zogan, H., Razzak, I., Jameel, S., & Xu, G. (2021). DepressionNet: Learning multi-modalities with user post summarization for depression detection on social media. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 133–142).

Zogan, H., Razzak, I., Wang, X., Jameel, S., & Xu, G. (2020). Explainable depression detection with multi-modalities using a hybrid deep learning model on social media. arXiv preprint arXiv:2007.02847.