A Dynamic Co-attention Network for Session-based Recommendation

Wanyu Chen[1,2] Fei Cai[1,*] Honghui Chen[1] Maarten de Rijke[2]

[1] Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
wanyuchen, caifei, chenhonghui@nudt.edu.cn

[2] Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

Session-based recommendation is the task of recommending the next item a user might be interested in given partially known session information, e.g., part of a session or recent historical sessions. An effective session-based recommender should be able to exploit a user's evolving preferences, which we assume to be a mixture of her short- and long-term interests. Existing session-based recommendation methods often embed a user's long-term preference into a static representation, which plays a fixed role when dealing with her current short-term interests. This is problematic because long-term preferences may be more or less important for predicting the next conversion depending on the user's short-term interests.

We propose a Dynamic Co-attention Network for Session-based Recommendation (DCN-SR). DCN-SR applies a co-attention network to capture the dynamic interactions between the user's long- and short-term interaction behavior and generates co-dependent representations of the user's long- and short-term interests. For modeling a user's short-term interaction behavior, we design a Contextual Gated Recurrent Unit (CGRU) network to take actions like "click", "collect" and "buy" into account. Experiments on e-commerce datasets show significant improvements of DCN-SR over state-of-the-art session-based recommendation methods, with improvements of up to 2.58% on the Tmall dataset and 3.08% on the Tianchi dataset in terms of Recall@10. MRR@10 improvements are 3.78% and 4.05%, respectively. We also investigate the scalability and sensitivity of DCN-SR. The improvements of DCN-SR over state-of-the-art baselines are especially noticeable for short sessions and active users with many historical interactions.

CCS CONCEPTS

• **Recommender systems**;

KEYWORDS

Dynamic co-attention network, Representation learning, Behavior modeling, Session-based recommendation, E-commerce

ACM Reference Format:
Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2019. A Dynamic Co-attention Network for Session-based Recommendation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357964

1 INTRODUCTION

Recommender systems are an effective solution to help people cope with an increasingly complex information landscape [5, 37]. Conventional recommender systems often discard sequential information and focus on mining the static relevancy between users and items from interactions [10, 27, 38]. For instance, a typical conventional recommender system based on matrix factorization [18] may be effective at modeling a user's general preferences by learning from their entire interaction history but it does not model the order of the user's interactions. Unlike conventional recommender systems, session-based recommender systems model the evolution of a user's short-term preference implied by sequential interactions in a session with the aim of recommending the next item a user may be interested in [34]. Popular modeling choices for session-based recommender systems include Markov chains and Recurrent Neural Networks (RNNs) [8]. For instance, the Factorizing Personalized Markov Chain (FPMC) model combines Markov chains with matrix factorization to achieve good recommendation performance [25]. Wang et al. [32] propose a Hierarchical Representation Model (HRM) model that extends FPMC by employing a two-layer structure to construct a hybrid representation. Markov chain-based methods only model local sequential patterns between adjacent interactions. RNN-based models can model multi-step sequential behaviors: the Hierarchical Recurrent Neural Network (HRNN) model [23] and Dynamic REcurrent bAsket Model (DREAM) [36] model embed all of a user's historical interactions into the final hidden state of an RNN to represent their current preferences; both achieve significant improvements over HRM and FPMC.

Today's session-based recommender systems successfully capture users' short-term decision making process. But they do not capture variations in the relative importance of a user's long-term vs. short-term interests for session-based recommendation. Users

*Corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11...$15.00
https://doi.org/10.1145/3357384.3357964

with different shopping preferences may prefer different next items even under the same session context. Thus, how to better capture individual users' dynamic consumption motivations is critical [16, 33].

Our working hypothesis is that the relative importance of events in a user's long-term interaction history depends on events in their short-term interaction history, and vice versa. Let us consider an example. Take a user who has searched for a camera in the current session; her long-term interactions related to electronic products should probably be given a higher weight than her interactions related to clothing when deciding what to recommend next. Conversely, if the user's past interactions indicate a strong interest in the Sony brand, then, during the current session, interactions related to this brand may be more important than others when predicting the next item. But there is more that should be modeled than the relation between past and present interactions. Different user actions, e.g., clicks, add-to-cart, or buy, provide different types of information about the user's interest and, hence, should trigger different follow-up actions. For example, a click on a camera may indicate that the current recommendation is not satisfactory so that substitute offerings should be recommended; adding an item to the cart may show a strong consumption motivation of a user for the item; and while repeat purchases are important [24], a purchase action involving a camera should probably be followed by a recommendation of complementary items [30].

In summary, the main challenges facing session-based recommendation are [16, 33]:

- *How to incorporate user's long-term as well as short -term preferences for session-based recommendation?* and
- *How to capture users' dynamic preferences with implicit preference data?*

To address these questions, we propose a *Dynamic Co-attention Network for Session-based Recommendation* (DCN-SR). DCN-SR has three main components:

(1) The first is a Contextual Gated Recurrent Unit (CGRU) network to model a user's short-term preferences, which we represent as a combination of hidden states of interactions in the current session.
(2) The second is a Multi-Layer Perceptron (MLP) to deal with a user's historical interactions and infer long-term preferences.
(3) The third is a co-attention network that uses the outputs of the first two components to capture interactions between actions in a user's long-term and short-term interaction histories and generate co-dependent representations of their long-term and short-term preferences.

To the best of our knowledge, in the field of session-based recommender systems, ours is the first attempt to use a co-attention network to exploit the relation between a user's long-term and short-term preferences learned from their long-term and short-term interaction history.

Experiments on two e-commerce datasets, the Tmall dataset and the Tianchi dataset, show that DCN-SR outperforms state-of-the-art baselines in prediction accuracy. In addition, we investigate the scalability and sensitivity of DCN-SR with different lengths of search sessions and different numbers of user historical interactions.

In summary, our key technical contributionss in this paper are:

(1) We design a dynamic co-attention network model for session-based recommendation (DCN-SR) that is able to integrate users' long-term and short-term preferences.
(2) We design a contextual gated recurrent unit CGRU to incorporate different types of short-term user actions so as to better estimate a user's next consumption interests.
(3) We analyse the recommendation performance of DCN-SR and find that DCN-SR consistently meets or beats the state-of-the-art, especially with short sessions and active users.

2 RELATED WORK

We summarize related work in two areas – sequential recommender systems and attention-based models.

2.1 Sequential recommendation models

Interactive systems log users' behavior along with the associated timestamps [14]. Many models have been proposed to leverage this kind of sequential data for modeling users' dynamic preferences and for sequential recommendation. Markov chains have been a popular choice. Following the Factorizing Personalized Markov Chain (FPMC) [25] model, Feng et al. [6] apply metric embeddings with a low dimensional vector for playlist and successive location recommendation. He and McAuley [8] fuse similarity models with Markov chains for sequential recommendation to solve sparse recommendation problems. In order to better capture both users' general taste and sequential behavior, Wang et al. [32] extend FPMC by using a hierarchical structure to learn user representations (HRM). Those Markov chain-based methods only model the local sequential patterns between adjacent interaction events.

Deep neural networks have improved the performance on the sequential recommendation task. Hidasi et al. [13] propose an RNN-based model for session-based recommendation that consists of Gated Recurrent Unit (GRU) units and uses a session-parallel mini-batch training process. With user profiles available, Quadrana et al. [23] develop hierarchical RNNs with cross-session information transfer and Yu et al. [36] propose a dynamic recurrent basket model (DREAM) to capture global sequential patterns for learning a user's dynamic interest representations based on RNNs, which outperforms HRM and FPMC. DREAM embeds all of a users' historical interactions into the final hidden state of an RNN to represent their current preferences. To improve the performance of RNN-based approaches to sequential recommendation, Tan et al. [26] adopt data augmentation and a method to account for shifts in the input data distribution. The RNN-based approaches listed above usually implicitly encode a user's long-term and short-term interactions into a latent factor or hidden state without distinguishing between the roles that each event may play when making recommendations.

Memory-based approaches leverage user memory networks to store and manipulate a user's previous interactions. Chen et al. [4] propose a Recommendation with User Memory Network (RUM) model to leverage external memory networks integrated with collaborative filtering. It uses a static latent vector to represent users' general preferences and the memory can only store and distinguish users' short-term interactions with a fixed size, which ignores the possibility that different historical interactions may have different degrees of importance.

Our approach to sequential recommendation differs from the work listed above because we do not only exploit the benefits of incorporating long-term and short-term interests, but also consider dynamic aspects of the relation between a user's long-term and short-term preferences. In addition, unlike the work listed above, we explore the information contained in users' different actions.

2.2 Neural attention based models

Attention mechanisms have been applied to recommendation tasks to help models exploit users' preferences [11, 22, 28]. Li et al. [19] propose a neural attentive session-based recommendation machine (Neural Attentive Recommendation Machine (NARM)) that takes the last hidden state from the session-based RNN as the sequential behavior, and uses the other hidden states of previous clicks for computing attention to capture users' current preferences in a given session. Although NARM achieves significant improvements over traditional RNN-based approaches, it does not consider users' long-term preferences based on their historical interactions. Ying et al. [35] adopt a hierarchical attention network for sequential recommendation (SHAN). The first attention layer in SHAN learns users' long-term preferences based on the historical purchased item representations, while the second one outputs the final user representation as a combination of the user's long-term and short-term preferences. It is worth pointing out that SHAN generates its attentive representation of user's long-term and short-term preferences independently and thus ignores the relations between them. As to memory-based models, Liu et al. [21] propose a short-term attention memory priority model (STAMP), in which the attention weights are calculated from the session context and enhanced with the final records in the current session.

Our approach to sequential recommendation differs from the aforementioned models in two ways. First, the attention mechanism used in recent sequential recommendation models deals with users' historical and recent interactions separately. In contrast, DCN-SR applies a co-attention network to calculate the correlated importance of actions in both a user's historical and recent interactions, and generates co-dependent representations for their long-term and short-term preferences. Second, previous work considers a user's long-term preferences to be a static vector when dealing with the user's different short-term interests. In contrast, in DCN-SR, an event in a user's historical interactions may have different degrees of importance when combined with different recent sessions.

3 APPROACH

The DCN-SR model we propose has three main components: a short-term preference generator, a long-term preference generator, and a co-attention network with short-term and long-term preferences. As shown in Fig. 1, these three components can be trained in a joint manner and give a predicted score of a user's preference for an item through a trilinear composition. We first describe the notation used and then detail the three components in DCN-SR.

3.1 Problem formulation and notation

Given a user and their sequential interactions, we aim to recommend their next purchase based on her long-term and short-term preferences learned from those interactions.

For a user u, we denote her current session as $Session_u = \{(x_1, a_1), (x_2, a_2), \dots, (x_T, a_T)\}$, where x_i is the i-th item in the session and a_i denotes an action (e.g., click, cart or purchase) along with the item; T denotes the number of events in the current session. In addition, we consider the items that u interacts with in her historical sessions and denote them as $History_u = \{x_1, x_2, \dots, x_N\}$. Here, N denotes the number of events in the user's historical interactions. For exploring the user's long-term preferences, not all actions necessarily depict the user's preference. Therefore, we only retain items with actions that can clearly reveal the user's preference, such as buy or collect. As shown in Fig. 1, there is an embedding layer at the bottom of the network used for generating the item embeddings as well as the action embeddings. We use $\mathbf{x}_i$ and $\mathbf{a}_i$ to indicate the embeddings of x_i and a_i.

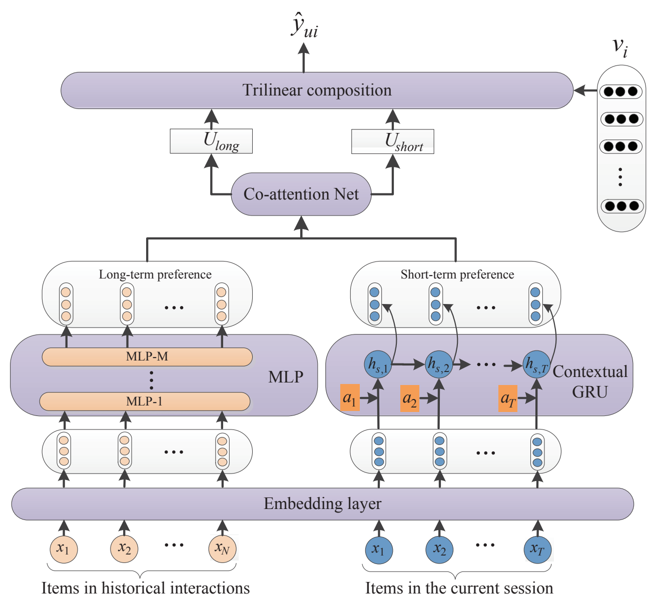


Figure 1: Structure of the DCN-SR model.

3.2 Short-term preference generator

A user's discriminative actions, such as click, collect or purchase, can help to explore sequential interactions as prior knowledge to predict the items that the user is mostly like to access. Since different actions may imply different consumption motivations in a short session, we take all types of actions in the current session into account when learning a user's short-term preferences.

As shown in Fig. 2, we model a user's sequential interactions in a session with a Contextual GRU network (CGRU) considering the action along with each item as a contextual feature. We modify the operations in a traditional GRU cell and add the action embedding $\mathbf{a}_i$ to the input gate, forget gate and update gate, respectively, shown as the purple arrows in Fig. 2. The hidden state $\mathbf{h_t}$ in CGRU can be a linear interpolation between the previous hidden state $\mathbf{h}_{t-1}$ and the candidate hidden state $\hat{\mathbf{h}}_t$:

$$\mathbf{h}_t = \mathbf{z}_t \mathbf{h}_{t-1} + (1 - \mathbf{z}_t)\hat{\mathbf{h}}_t, \tag{1}$$

where the update gate $\mathbf{z}_t$ is given by:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{V}_z \mathbf{a}_t + \mathbf{U}_z \mathbf{h}_{t-1}), \tag{2}$$

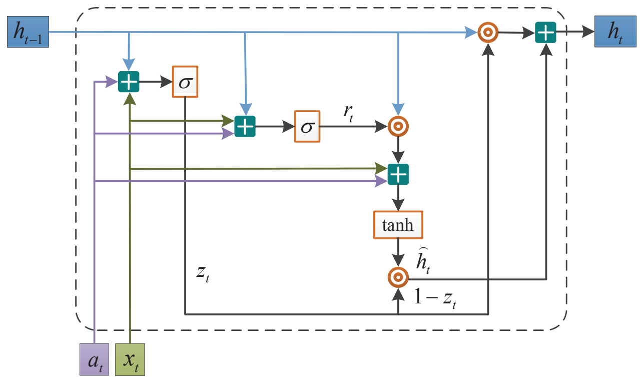


Figure 2: Structure of the Contextual GRU network.

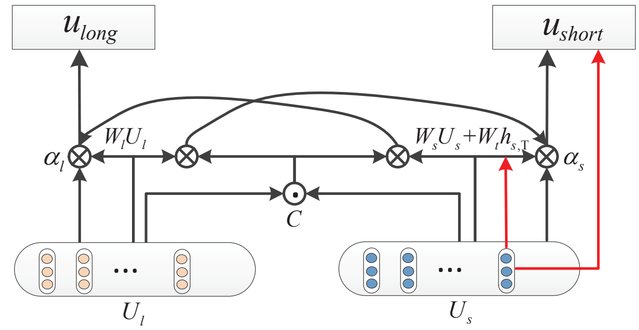


Figure 3: Structure of the co-attention network.

where $\mathbf{W}_z$, $\mathbf{V}_z$ and $\mathbf{U}_z$ are update parameters for $\mathbf{x}_t$, $\mathbf{a}_t$ and $\mathbf{h}_{t-1}$, respectively. The candidate hidden state can be computed as:

$$\hat{\mathbf{h}}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{V}\mathbf{a}_t + \mathbf{r}_t \odot \mathbf{U}\mathbf{h}_{t-1}), \tag{3}$$

where the reset gate $\mathbf{r}_t$ can be calculated by:

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{V}_r\mathbf{a}_t + \mathbf{U}_r\mathbf{h}_{t-1}), \tag{4}$$

where $\mathbf{W}_r$, $\mathbf{V}_r$ and $\mathbf{U}_r$ are reset parameters for $\mathbf{x}_t$, $\mathbf{a}_t$ and $\mathbf{h}_{t-1}$, respectively.

As each hidden state contains the information of a user's search intent in the current session, we use a collection of hidden states to represent a user's initial short-term preference as $\mathbf{U}_s = \{\mathbf{h}_{s,1}, \mathbf{h}_{s,2}, \ldots, \mathbf{h}_{s,T}\}$ and $\mathbf{U}_s \in \mathbb{R}^{D\times T}$, where D is the dimension of each hidden state in $\mathbf{U}_s$. We will future explore the user's interest drift across these hidden states with a co-attention network in Section 3.4.

3.3 Long-term preference generator

As we discussed above, when exploring a user's long-term preference with their historical interactions, we only retain the items with actions that could depict users' preferences (e.g., buy or collect). We feed the dense low-dimensional embedding of each item in a user's historical interactions $History_u = \{x_1, x_2, \ldots, x_N\}$ through a multi-layer perceptron (MLP) to generate hidden representations of those events:

$$\begin{aligned}
\mathbf{z}_{1,i} &= \phi(\mathbf{W}_1\mathbf{x}_i + \mathbf{b}_1) \\
\mathbf{z}_{2,i} &= \phi(\mathbf{W}_2\mathbf{z}_{1,i} + \mathbf{b}_2) \\
&\vdots \\
\mathbf{z}_{M,i} &= \tanh(\mathbf{W}_M\mathbf{z}_{M-1,i} + \mathbf{b}_M) \\
\mathbf{X}_i &= \mathbf{z}_{M,i},
\end{aligned} \tag{5}$$

where $\mathbf{W}_m$, $\mathbf{b}_m$ and ϕ denote the weight matrix, the bias vector and the activation function in the m-th layer. Here, we use a ReLU as the activation function, as it has been shown to be more expressive than others and can deal with the vanishing gradient problem effectively [12, 15]. M indicates the number of layers used in MLP network. The output of the final layer $\mathbf{X}_i$ is the hidden representation of the i-th event. We also apply a collection of these event representations to indicate a user's initial long-term preference as $\mathbf{U}_l = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ and $\mathbf{U}_l \in \mathbb{R}^{D\times N}$. We use a MLP network because of its non-linear modeling capability, which has been applied in many neural collaborative filtering works and shows reliable performance [12].

3.4 Co-attention network

It is beneficial to incorporate the short-term and long-term preference of a user when making recommendations. However, traditional methods treat these two types of preference as independent [35], which ignores the (potential) mutual dependence between them.

In addition, conventional attention mechanisms assign weights for the events in a user's historical and recent interactions separately. We argue that historical interactions and recent interactions can provide context for each other when calculating the importance of each event. Thus, we design a co-attention network to explore correlations between historical and current interactions of a user.

As shown in Fig. 3, after generating a user's initial short-term and long-term preferences, we use them as the inputs of the co-attention network and calculate the affinity matrix $\mathbf{C}$:

$$\mathbf{C} = \tanh(\mathbf{U}_l^{\mathrm{T}}\mathbf{W}_c\mathbf{U}_s), \tag{6}$$

where $\mathbf{W}_c \in \mathbb{R}^{D\times D}$ contains the weights. After computing the affinity matrix, we consider it as a feature and use it to transform the short-term attention space into the long-term attention space with:

$$\begin{aligned}
\mathbf{H}^l &= \tanh(\mathbf{W}_l\mathbf{U}_l + (\mathbf{W}_s\mathbf{U}_s + \mathbf{W}_t\mathbf{h}_{s,T})\mathbf{C}^{\mathrm{T}}) & (7)\\
\alpha_l &= \mathrm{softmax}(\mathbf{w}_{hl}^{\mathrm{T}}\mathbf{H}^l) & (8)
\end{aligned}$$

and vice versa:

$$\begin{aligned}
\mathbf{H}^s &= \tanh(\mathbf{W}_s\mathbf{U}_s + \mathbf{W}_t\mathbf{h}_{s,T} + (\mathbf{W}_l\mathbf{U}_l)\mathbf{C}) & (9)\\
\alpha_s &= \mathrm{softmax}(\mathbf{w}_{hs}^{\mathrm{T}}\mathbf{H}^s), & (10)
\end{aligned}$$

where $\mathbf{W}_l$, $\mathbf{W}_s$, $\mathbf{W}_t \in \mathbb{R}^{K\times D}$, $\mathbf{w}_{hl}$, $\mathbf{w}_{hs} \in \mathbb{R}^K$ are weight parameters for long-term and short-term preferences, respectively. Here, $\alpha_l \in \mathbb{R}^N$ and $\alpha_s \in \mathbb{R}^T$ are the attention probabilities for the events in historical and current interactions, respectively.

It should be noted in Eq. 7 and Eq. 9 that besides the collection of the hidden states in current session, i.e., $\mathbf{U}_s$, we explicitly consider the final hidden state in the current session, i.e., $\mathbf{h}_{s,T}$, shown as the red arrows in Fig. 3. Importantly, $\mathbf{h}_{s,T}$ summarizes the complete sequential behavior, which contains different information from $\mathbf{U}_s$ when exploring user's short-term preferences [19, 21]. Both NARM [19] and STAMP [21] have shown that the explicit use of $\mathbf{h}_{s,T}$ improves the performance of session-based recommendations.

Based on the attention weights, the co-dependent representations of a user's long-term and short-term preferences can be calculated as the weighted sum of their interactions representations:

$$\mathbf{U}_{co-l} = \sum_{n=1}^{N} \alpha_l^n \mathbf{X}_n, \tag{11}$$

and

$$\mathbf{U}_{co-s} = \sum_{t=1}^{T} \alpha_s^t \mathbf{h}_{s,t}. \tag{12}$$

In order to take $\mathbf{h}_{s,T}$ into consideration, we use $\mathbf{U}_{long} = \mathbf{U}_{co-l}$, $\mathbf{U}_{short} = [\mathbf{U}_{co-s}; \mathbf{h}_{s,T}]$ to represent the final representations of a user's long-term and short-term preference. And then, for a given candidate item x_i, the scoring function that produces a prediction can be a trilinear combination:

$$\hat{z}_{ui}^l = \mathbf{v}_i^{\mathrm{T}} \mathbf{B}_l \mathbf{U}_{long} \quad \hat{z}_{ui}^s = \mathbf{v}_i^{\mathrm{T}} \mathbf{B}_s \mathbf{U}_{short} \quad \hat{z}_{ui} = \sigma(\mathbf{B}^{\mathrm{T}}[\hat{z}_{ui}^l; \hat{z}_{ui}^s]), \tag{13}$$

where $\mathbf{B}_l \in \mathbb{R}^{E \times D}$, $\mathbf{B}_s \in \mathbb{R}^{E \times 2D}$ and $\mathbf{B} \in \mathbb{R}^{1 \times 2}$, E is the dimension of each item embedding. The trilinear combination incorporates the user's long-term preferences as well as their short-term preferences towards an item. Moreover, $\hat{z}_{ui}$ represents the unnormalized cosine similarity between the user's preference and the i-th candidate item. We use $\hat{\mathbf{z}}_u \in \mathbb{R}^V$ to denote the vector that consists of $\hat{z}_{ui} (i \in [1, \ldots, V])$, where V is the number of candidate items. It is then processed by a softmax function:

$$\hat{\mathbf{y}}_u = \mathrm{softmax}(\hat{\mathbf{z}}_u), \tag{14}$$

where $\hat{\mathbf{y}}_u$ denotes the output vector of our model, which represents a probability distribution over the candidate items, and each element $\hat{y}_{ui}$ denotes the probability of the item v_i being the next purchase.

We adopt the cross-entropy loss as our loss function:

$$L(\hat{\mathbf{y}}_u) = -\sum_{i=1}^{V} \mathbf{y}_u \log(\hat{\mathbf{y}}_u), \tag{15}$$

where $\mathbf{y}_u$ is the true distribution.

Finally, a Back-Propagation Through Time (BPTT) method with a fixed number of time steps is adopted to train our DCN-SR model based on Eq. 15.

4 MODEL ANALYSIS

To provide insights into DCN-SR, we discuss its connection to previous work on session-based recommendation. By choosing appropriate settings, DCN-SR can subsume several existing methods, including session-based recommendations with recurrent neural networks (GRU4Rec) and an attention-based model, i.e., the Neural Attentive Recommendation Machine (NARM).

4.1 DCN-RS vs. GRU4Rec

GRU4Rec is an RNN-based approach that uses the final hidden state to represent a user's preference:

$$\mathbf{h}_T = GRU_{sess}(\mathbf{v}_T, \mathbf{h}_{T-1}), \tag{16}$$

and predict the score for a candidate item v_i as:

$$\hat{z}_{ui} = \sigma(\mathbf{v}_i^{\mathrm{T}} \mathbf{h}_T). \tag{17}$$

As shown in Fig. 1, when we do not consider the user's historical interactions and different actions, DCN-SR will reduce to an RNN-based approach. To show how our model degenerates to GRU4Rec, we set the historical interactions empty and the weight parameter $\mathbf{w}_{hs} = 0$; then, in the co-attention component, the affinity matrix $\mathbf{C}$, $\mathbf{U}_{co-l}$ and $\mathbf{U}_{co-s}$ will be 0. The user's preferences will be equal to the final hidden state of the session, i.e., $\mathbf{U}_{short} = [\mathbf{U}_{co-s}; \mathbf{h}_{s,T}] = \mathbf{h}_{s,T}$. And the prediction score is calculated as:

$$\hat{z}_{ui} = \sigma(\hat{z}_{ui}^s) = \sigma(\mathbf{v}_i^{\mathrm{T}} \mathbf{U}_{short}) = \sigma(\mathbf{v}_i^{\mathrm{T}} \mathbf{h}_{s,T}). \tag{18}$$

This is the same as the prediction function (Eq. 17) of GRU4Rec. However, by enabling a user's historical interactions, DCN-SR is able to collect valuable information for her long-term preference. In addition, with the weight parameter $\mathbf{w}_{hs}$, DCN-SR can adaptively select important items in the current session to capture the user's short-term interest, which can bring improved performance in the task of sequential recommendation as shown in our experiments.

4.2 DCN-SR vs. NARM

Both DCN-SR and NARM apply an attention mechanism to capture a user's main interest. In NARM, the attention mechanism takes the last hidden state $\mathbf{h}_T$ from the RNN as the sequential behavior, which denotes the global encoder of the current session:

$$\mathbf{c}_g = \mathbf{h}_T. \tag{19}$$

It then uses the hidden states of previous clicks in the current session for computing attention scores, which is a local encoder combining different parts of the sequence:

$$\mathbf{c}_l = \sum_{i=1}^{T} \alpha_i \mathbf{h}_i, \tag{20}$$

where α is the weighted factor calculated by:

$$\alpha_i = \mathbf{v}^{\mathrm{T}} \sigma(\mathbf{A}_1 \mathbf{h}_i + \mathbf{A}_2 \mathbf{h}_T), \tag{21}$$

where σ is an activation function, and $\mathbf{A}_1$ and $\mathbf{A}_2$ are used to transform $\mathbf{h}_i$ and $\mathbf{h}_T$ into a latent space.

By concatenating the global and local encoder, NARM adopts a unified representation $\mathbf{c}$ to model the user's short-term preference:

$$\mathbf{c} = [c_g; c_l] = [\mathbf{h}_T; \sum_{i=1}^{T} \alpha_i \mathbf{h}_i]. \tag{22}$$

The prediction score for a candidate item v_i is calculated as:

$$\hat{z}_{ui} = \mathbf{v}_i^{\mathrm{T}} \mathbf{B} \mathbf{c}, \tag{23}$$

where $\mathbf{B}$ is a latent parameter.

To see the connection between DCN-SR and NARM, we set the user's historical interactions empty and ignore different actions in RNN. Thus DCN-SR can be reduced to:

$$\mathbf{C} = \tanh(\mathbf{U}_l^{\mathrm{T}} \mathbf{W}_c \mathbf{U}_s) = 0, \tag{24}$$

and

$$\begin{aligned} \mathbf{H}^s &= \tanh(\mathbf{W}_s \mathbf{U}_s + \mathbf{W}_t \mathbf{h}_{s,T} + \mathbf{W}_l \mathbf{U}_l \mathbf{C}) \\ &= \tanh(\mathbf{W}_s \mathbf{U}_s + \mathbf{W}_t \mathbf{h}_{s,T}). \end{aligned} \tag{25}$$

Because $\mathbf{U}_s$ is a set of hidden states in RNN, i.e., $\mathbf{U}_s = \{\mathbf{h}_{s,1}, \mathbf{h}_{s,2}, \ldots, \mathbf{h}_{s,T}\}$, we divide Eq. 25 for each hidden state as:

$$\mathbf{H}_i^s = \tanh(\mathbf{W}_s \mathbf{h}_{s,i} + \mathbf{W}_t \mathbf{h}_{s,T}). \tag{26}$$

Then, the attention weight for each event in the current interactions is calculated by:

$$\begin{aligned} \alpha_s^i &= \mathrm{softmax}(\mathbf{w}_{hs}^{\mathrm{T}} \mathbf{H}_i^s) \\ &= \mathrm{softmax}(\mathbf{w}_{hs}^{\mathrm{T}} \tanh(\mathbf{W}_s \mathbf{h}_{s,i} + \mathbf{W}_t \mathbf{h}_{s,T})). \end{aligned} \tag{27}$$

We can see that the attention weights calculated in our model are the same as NARM. Based on these attention weights, we can rewrite a user's final short-term preference as:

$$\mathbf{U}_{short} = [\mathbf{U}_{co-s}; \mathbf{h}_{s,T}] = \left[\sum_{i=1}^{T} \alpha_s^i \mathbf{h}_{s,i}; \mathbf{h}_{s,T}\right]. \tag{28}$$

The prediction score is generated by:

$$\hat{z}_{ui} = \sigma(\hat{z}_{ui}^s) = \sigma(\mathbf{v}_i^{\mathrm{T}} \mathbf{B}_s \mathbf{U}_{short}). \tag{29}$$

According to these derivations, we see that by choosing proper activation functions in Eq. 27 and 29, DCN-SR and NARM have the same representations of users' preferences and prediction function.

Based on our analysis, we see that DCN-SR is a very general model for session-based recommendation. On the one hand, by introducing different settings (i.e., parameters and activation functions) DCN-SR can be seen as a generalization of many existing models. On the other hand, DCN-SR enables us to explore more information from users' historical interactions and the dynamic correlations between long-term and short-term preferences.

5 EXPERIMENTAL SETUP

We design experiments to examine the effectiveness of DCN-SR and focus on the following research questions: (RQ1) Does DCN-SR outperform state-of-the-art baselines for session-based recommendation? (RQ2) Does the Contextual GRU, which incorporates different user actions, contribute to the performance of DCN-SR? (RQ3) How is the performance of DCN-SR impacted by sessions with different lengths? (RQ4) How does the performance of DCN-SR vary across users with different numbers of historical interactions? (RQ5) How can we visualize the co-attention mechanism?

5.1 Model summary

As DCN-SR considers users' long-term and short-term preferences, we mainly compare our method with personalized session-based recommendation models, i.e., HRNN and SHAN. In addition, we also consider some traditional models, i.e., FPMC and Item-pop, as well as neural models with or without an attention mechanism, i.e., NARM, STAMP and GRU4Rec. These are our baselines:

Item-pop A method that ranks items based on the number of interactions, which is a non-personalized approach [1].

FPMC A state-of-the-art hybrid model for sequential recommendation, based on Markov chains and collaborative filtering. Both sequential behaviors and general taste are taken into account [25].

GRU4Rec An RNN-based model for session-based recommendation, which contains GRUs and utilizes session-parallel mini-batches as well as a pair-wise loss function for training [13].

NARM An RNN-based model that applies an attention mechanism to capture users' main purposes from the hidden states and combines it with sequential behavior as final representations of users' current preferences [15].

STAMP A memory-based model with attention mechanism that explicitly considers correlations between each click and the last click in a session. It combines the weighted events and the last click to model users' current preferences [21].

Table 1: Dataset statistics.

Dataset	Tmall	Tianchi
#of users	822	14,080
#of items	5,823	40,886
#of interactions	157,709	3,782,379
#of action types	4	4
Average interactions per user	192.13	272.03
Average interactions per item	27.03	93.68
# interactions in training set	147,735	2,897,330
# interactions in test set	9,974	885,049

HRNN A hierarchical RNN for personalized session-based recommendation which uses a session- and a user-level RNN to model users' short- and long-term preferences [23].

SHAN A personalized session-based recommendation method that adopts a hierarchical attention network, in which the first attention layer learns users' long-term preferences while the second one outputs the final user representation as a combination of the user's long-term and short-term preferences [35].

5.2 Datasets and experimental setup

Datasets. We use two publicly available real-world datasets to evaluate our models and the baselines. *Tmall* is a dataset released by Taobao.[1] It contains records of online transactions, with 884 users, 9,531 brands and 182,880 interactions. Customer action types include click, collect, cart, and purchase. *Tianchi* is a dataset provided by Alibaba.[2] It is based on user-commodity behavior data of Alibaba's M-Commerce platforms. It contains 23,291,027 interactions of 20,000 customers on 4,758,484 items within a month plus category information of each item. Customer actions include click, collect, cart, and purchase.

For the Tmall dataset, we filter out users with fewer than 3 interactions and items that appear less than 3 times [36]. For the Tianchi dataset, we filter out users with fewer than 20 interactions and items with fewer than 50 interactions. The characteristics of the datasets after preprocessing are summarized in Table 1.

Settings and parameters. For evaluation, we divide the Tmall and Tianchi datasets into training and test sets according to the users' search time. The training set consists of all but the last 7 days of interactions; the test set contains the remaining 7 days of interactions. As collaborative filtering methods cannot recommend an item that has not appeared before, we filter out interactions from the test set with items that do not appear in the training set.

For the Tmall and Tianchi datasets, we treat user records in one day as a session to model short-term preferences, following [20, 35]. For the Tmall dataset, although there is no detailed time information beyond one day, the sequential information of user behaviors on items still exists, so we can also model it with an RNN [20].

Unless specified differently, for all the results that we presented, the number of recommendations (N) equals 10 [12, 15]. We use *Recall@10* and *MRR@10* to evaluate the performance of models [19, 21]. Recall@10 is used to evaluate the recall of the recommender

[1] http://102.alibaba.com/competition/addDiscovery/index.htm
[2] https://tianchi.aliyun.com/getStart/information.htm?spm=5176.100067.5678.2.30a8b6d933N6Rr&raceId=231522

system, i.e., whether the test item is contained in the top 10 list. MRR@10 measures the ranking accuracy of the recommender system, i.e., whether the test item is ranked at the top of the list.

We optimize the hyperparameters using Adam [17] with the initial learning rate set to 0.01, and the mini-batch size fixed at 512. The dimension of the item embeddings is set to 50 and we use one GRU layer with 100 hidden units. Optimization is done on a validation set, which is partitioned from the training set with the same procedure as the test set [3].

6 RESULTS AND DISCUSSION

6.1 Overall performance

To answer **RQ1**, we examine the recommendation performance of the baselines and DCN-SR. See Table 2.

Table 2: Performance of recommendation models. The results produced by the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences of DCN-SR vs. the best baseline is determined by a t-test (▲ for α = .01, or △ for α = .05).

	Tmall		Tianchi	
Model	Recall@10	MRR@10	Recall@10	MRR@10
Item-pop	.1058	.0455	.0022	.0011
FPMC	.1813	.1227	.0594	.0377
GRU4Rec	.5852	.5613	.1117	.0875
NARM	.7237	.6781	.3155	.1909
STAMP	<u>.7246</u>	<u>.6872</u>	<u>.3185</u>	<u>.1955</u>
HRNN	.6894	.6617	.1971	.1801
SHAN	.7101	.6687	.2208	.1843
DCN-SR	**.7433**△	**.7132**▲	**.3283**△	**.2034**▲

Let us first consider the baselines. From Table 2, we see that neural-based approaches outperform traditional methods, i.e., Item-pop and FPMC. As to non-personalized session-based approaches, i.e., GRU4Rec, NARM and STAMP, we see that NARM and STAMP both improve over GRU4Rec, which indicates the utility of using an attention mechanism. This result can also be proved by comparing the results of the personalized models, i.e., HRNN and SHAN, where SHAN with a hierarchical attention structure shows better performance than HRNN. The results of HRNN are higher than of a simple RNN-based approach such as GRU4Rec, which means that incorporating users' historical and recent interactions together can help to boost the recommendation performance. STAMP and NARM show better results than SHAN. The explicit use of the last hidden state seems to improve the performance for session-based recommendations, as the last behavior in a short session can reveal users' current consumption motivations better. STAMP outperforms other baselines in terms of Recall@10 and MRR@10. Hence, we use STAMP as our baseline in later experiments.

Next, we compare the baselines against the DCN-SR model. Personalized and non-personalized models, i.e., SHAN and STAMP, both lose against DCN-SR in terms of Recall@10 and MRR@10. This shows that using the co-attention network helps to improve the recommendation performance. This may be due to two factors: one is that with the co-attention network, DCN-SR can capture the mutual dependence between users' historical and recent interactions and learn dynamic representations of users' long- and short-term preferences; the other is that DCN-SR integrates both a user's long- and short-term preferences to predict their next interactions.

The improvements of DCN-SR over the best baseline model in terms of Recall@10 are 2.58% on the Tmall dataset and 3.08% on the Tianchi dataset. MRR@10 improvements are 3.78% on the Tmall dataset and 4.05% on the Tianchi dataset. Significant improvements against the best performing baseline are observed for the DCN-SR model at the $\alpha = .01$ level in terms of MRR@10 on both datasets. For Recall@10, we observe significant improvements at the $\alpha = .05$ level on both datasets. The fact that improvements in terms of MRR@10 are bigger than in terms of Recall@10 suggests that the main effect of DCN-SR's architecture is to boost the ranking of relevant items rather than the number of relevant items found.

6.2 The Contextual GRU network

For **RQ2**, in order to demonstrate the utility of the CGRU network, which considers users' actions as search context in a short session, we examine the recommendation performance of DCN-SR under different settings, i.e., DCN-SR$_{GRU}$ (with a simple GRU network) vs. DCN-SR$_{CGRU}$ (with the Contextual GRU network). Table 3 contrasts their performance against the best baseline model (STAMP), with different numbers of recommended items N.

DCN-SR$_{GRU}$, which lacks users' action information, still beats the best baseline model, i.e., STAMP, which indicates that the dynamic co-attention network helps to improve the performance of sequential recommendations. DCN-SR$_{CGRU}$ consistently achieves improvements over DCN-SR$_{GRU}$, which demonstrates the utility of the Contextual GRU network. Improvements of DCN-SR$_{GRU}$ over STAMP are significant at the $\alpha = .05$ level in terms of MRR and Recall, on both datasets. For DCN-SR$_{CGRU}$, we observe significant improvements at the $\alpha = .05$ level in terms of Recall, and at the $\alpha = .01$ level in terms of MRR on the two datasets.

Regarding different numbers of recommendations, we see that the overall performance in terms of Recall and MRR increases when N ranges from 5 to 15, as a large value of N increases the probability of including a user's preferred item in the list.

The improvements of DCN-SR$_{CGRU}$ in terms of MRR are more significant than those in terms of Recall, as indicated by the relative improvements over DCN-SR$_{GRU}$ with different numbers of recommendations. We further conduct paired t-tests, verifying that these improvements are statistically significant for $\alpha = .05$ in terms of Recall and $\alpha = .01$ in terms of MRR. These improvements can prove that incorporating the information contained in users' different actions helps to learn more accurate representations of users' short-term preferences.

6.3 Session length

In order to understand the scalability of sequential recommendation models when applied with sessions of different lengths, we divide the sessions in the datasets into *short* (no more than 5 queries), *medium* (6 to 15 queries) and *long* sessions (more than 15 queries)

Table 3: Recommendation performance with different numbers of recommended items N. The results produced by the best performer in each row are boldfaced. Statistical significance of pairwise differences is determined by a t-test ($^{\blacktriangle}$ for $\alpha = .01$ and $^{\vartriangle}$ for $\alpha = .05$ when comparing DCN-SR$_{CGRU}$ or DCN-SR$_{GRU}$ vs. STAMP; $^{\bullet}$ for $\alpha = 0.01$ and $^{\circ}$ for $\alpha = 0.05$ when comparing DCN-SR$_{CGRU}$ vs. DCN-SR$_{GRU}$).

		Tmall			Tianchi		
N		STAMP	DCN-SR$_{GRU}$	DCN-SR$_{CGRU}$	STAMP	DCN-SR$_{GRU}$	DCN-SR$_{CGRU}$
5	Recall@5	.7148	.7256$^{\vartriangle}$	**.7272**$^{\vartriangle\circ}$	.2685	.2744$^{\vartriangle}$	**.2752**$^{\vartriangle\circ}$
	MRR@5	.6859	.6964$^{\vartriangle}$	**.6999**$^{\blacktriangle\bullet}$	.1895	.1939$^{\vartriangle}$	**.1955**$^{\blacktriangle\bullet}$
10	Recall@10	.7246	.7395$^{\vartriangle}$	**.7433**$^{\vartriangle\circ}$	.3184	.3278$^{\vartriangle}$	**.3283**$^{\vartriangle\circ}$
	MRR@10	.6872	.7016$^{\vartriangle}$	**.7132**$^{\blacktriangle\bullet}$	.1955	.2016$^{\vartriangle}$	**.2034**$^{\blacktriangle\bullet}$
15	Recall@15	.7314	.7470$^{\vartriangle}$	**.7501**$^{\vartriangle\circ}$	.3473	.3575$^{\vartriangle}$	**.3581**$^{\vartriangle\circ}$
	MRR@15	.6878	.7032$^{\vartriangle}$	**.7147**$^{\blacktriangle\bullet}$	.1985	.2046$^{\vartriangle}$	**.2067**$^{\blacktriangle\bullet}$

on the test set and report separate results in Fig. 4. We do not Item-pop and FPMC in the comparison, as their performance is worse than that of the RNN-based models, especially with short sessions.

From Fig. 4 we can see that as the session length increases, the performance of all models improves on the Tmall dataset while it decreases on the Tianchi dataset. The DCN-SR model always achieves the highest scores, on both datasets, across different session lengths. Specifically, for Recall@10, as shown in Fig. 4a and Fig. 4c, among the baselines, NARM and STAMP perform better than the model without attention mechanism, i.e., GRU4Rec, across all three session lengths. As for the personalized methods, although both have a hierarchical structure, SHAN shows better performance than HRNN across all session lengths, which demonstrates the utility of an attention mechanism that combines long- and short-term preferences. STAMP outperforms NARM except when applied with short sessions; this may be due to the fact that short sessions contain less information than long sessions, thus RNN-based model, i.e., NARM, can provide positional and sequential information as a supplementary for recommendation, while STAMP lacks information for predicting users' preferences with few interactions.

For MRR@10, a similar trend is shown in Fig. 4b and 4d. Particularly, DCN-SR shows larger improvements over STAMP in terms of MRR@10 than Recall@10, which is consistent with our findings in Table 2. For the Tmall dataset, the improvements are 9.03%, 4.54% and 1.83% in terms of MRR@10, for short, medium and long sessions, respectively, vs. improvements of 8.02%, 3.21% and 1.72% in terms of Recall@10. For Tianchi dataset, the improvements are 7.33%, 6.42% and 3.82% in terms of MRR@10, for short, medium and long sessions, respectively, vs. 5.02%, 3.69% and 1.74% for Recall@10.

The improvements of DCN-SR over STAMP are more obvious for short sessions than for long sessions. This may be because (1) we consider the users' historical interactions with the co-attention network, which can provide us with users' long-term preferences when making recommendations for short sessions; (2) the CGRU network incorporates users' action information, which supplies additional information on users' consumption motivations.

6.4 The length of historical interactions

To answer **RQ4**, we evaluate the sequential recommendation models that we consider with different volumes of users' historical interactions. This time, we group results by the length of users' historical interactions, which is denoted as H. That is, we use both datasets and partition the users into eight groups: $H < 100$, $H \in [100, 200)$, $H \in [200, 300)$, $H \in [300, 400)$, $H \in [400, 500)$, $H \in [500, 600)$, $H \in [600, 700]$, and $H > 700$. In order to see the impact of the length of a user's historical interactions on the recommendation performance, we compare the performance of DCN-SR with five baseline models except Item-pop and FPMC; see Fig. 5.

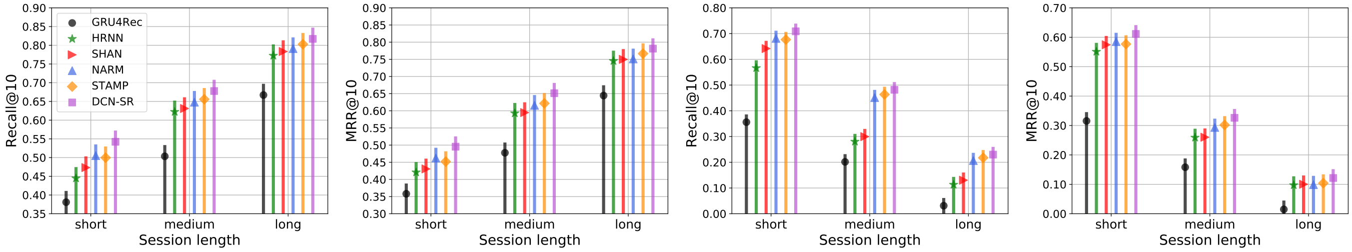
DCN-SR achieves the best performance for all of the eight groups on both datasets. For the Tmall dataset, when the number of users' historical interactions increases, the performance of all models begins to fluctuate at first but shows an upward trend overall. In particular, as the number of interactions increases, the performance of DCN-SR, SHAN and HRNN improves more noticeably than of STAMP and NARM. For example, SHAN shows better performance than STAMP and NARM in terms of Recall@10 and MRR@10 when the number of interactions is more than 700. The performance gap between DCN-SR and STAMP in terms of MRR@10 increases when the number of interactions increases from the seventh group ($[600, 700]$) to the eighth group (> 700).

For the Tianchi dataset, the performance of all models decreases in terms of both metrics as we consider longer histories. The results for DCN-SR, SHAN and HRNN decline more slowly than for the STAMP and NARM model, which is consistent with our findings in Fig. 5a and 5b. E.g., the improvements of DCN-SR over STAMP are 9.03%, 14.27%, 18.19% and 32.51% under the fifth ($H \in [400, 500)$), sixth ($H \in [500, 600)$), seventh ($H \in [600, 700]$), and eighth ($H > 700$) groups in terms of Recall@10. This shows the effectiveness of using personalization strategies, i.e., users' long-term preferences, to improve the recommendation performance.

6.5 Co-attention visualization

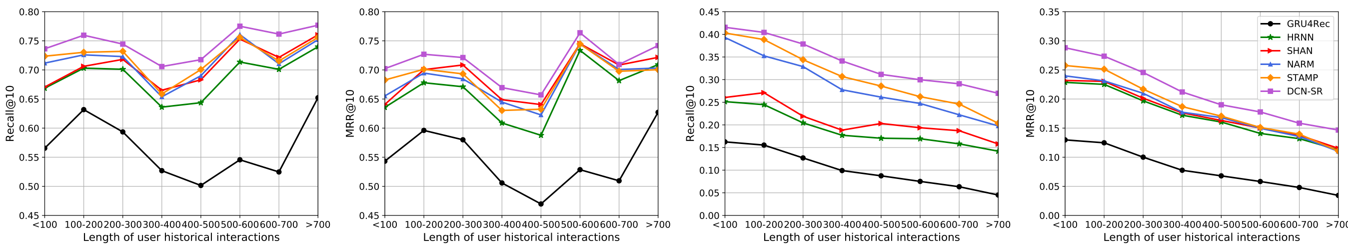
To illustrate the role of the co-attention mechanism, we present examples of two users in Fig. 6. For each, we randomly choose two sessions from the test set on the Tianchi dataset, as the Tianchi dataset contains category information for items, which helps us assess the association between interactions. In Fig. 6, the depth of the color indicates the importance of an event, the darker the color the more important an event is. The red numbers above the bar are the categories of the corresponding items.

DCN-SR is capable of highlighting a number of factors in predicting a user's next interaction as shown in Fig. 6. First, although the



(a) Performance in terms of Recall@10 on Tmall. **(b) Performance in terms of MRR@10 on Tmall.** **(c) Performance in terms of Recall@10 on Tianchi.** **(d) Performance in terms of MRR@10 on Tianchi.**

Figure 4: Effect on the performance of six models in terms of Recall@10 and MRR@10 of different session lengths.



(a) Performance in terms of Recall@10 on Tmall. **(b) Performance in terms of MRR@10 on Tmall.** **(c) Performance in terms of Recall@10 on Tianchi.** **(d) Performance in terms of MRR@10 on Tianchi.**

Figure 5: Effect on the performance of six models in terms of Recall@10 and MRR@10 with different numbers of historical interactions, tested on the Tmall and Tianchi datasets.

Figure 6: Co-attention visualization. The depth of color indicates the importance of an event. The red number above the bar is the category of the corresponding item.

two sessions of a single user share the same historical interactions, the weights of these historical interactions differ. For example, for User A, the first event in the historical interactions plays a more important role in Session1 than in Session2. Also, items that have the same category as the target item have larger attention weights than others. The category of an item can partially reflect the interest of the user, thus it indicates that the co-attention mechanism captures the user's dynamic interests to some extent.

Second, the interactions in a session also have different weights for predicting a user's preference, which proves that DCN-SR can select important events and ignore unintended interactions. In addition, interactions close to the end of the session often have larger importance, which is especially clear in Session1 for User B. This confirms our intuition that incorporating a user's last interaction in the co-attention mechanism can help to improve the performance.

Third, there are some important interactions in a session that are not near the user's last click. For example, in Session2 for User A, the sixth event is more important than the last event. This may be due to the user's interests drift. However, DCN-SR can also pick them up and give them high weights.

Therefore, based on the visualization results, we claim that the co-attention mechanism is able to capture important events both in users' historical interactions as well as their current interactions.

7 CONCLUSIONS AND FUTURE WORK

We propose a dynamic co-attention network for session-based recommendation, DCN-SR. DCN-SR applies a co-attention network to capture the dynamic relations between a user's long-term and short-term interactions and generate co-dependent representations of the user's long-term and short-term preferences. It not only exploits the combination of long-term and short-term knowledge, but also considers dynamic aspects of the relation between a user's long-term and short-term preferences. For modeling a user's short-term interests, we design a Contextual GRU network to take a user's actions into account, as different types of action, e.g., "click," "collect" and "buy," can help to reflect users' next consumption motivations.

Our experimental results confirm the effectiveness and robustness of DCN-SR with different session lengths and varying numbers of users' historical interactions. DCN-SR outperforms the best performing state-of-the-art model STAMP across different session lengths, especially for short sessions. As to users with different numbers of historical interactions, DCN-SR shows more competitive recommendation performance on all users than the state-of-the-art baseline model STAMP. In addition, the improvements of DCN-SR are higher on users with more historical interactions.

As to future work, on the one hand, we plan to investigate the use of information contained in different action sequences, e.g., click-click-buy, and click-click-collect, as sequential actions can provide more context information than single actions [2, 7, 31]. On the other hand, we plan to extend the DCN-SR model with more auxiliary information, such as content information, to generate more informative representations of items [9, 29, 39].

ACKNOWLEDGMENTS

This research was partially supported by Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), and the Innovation Center for Artificial Intelligence (ICAI), the National Natural Science Foundation of China under No. 61702526, the Defense Industrial Technology Development Program under No. JCKY2017204B064. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowledge and Data Engineering*. 17, 6 (2005), 734–749.

[2] Alexey Borisov, Martijn Wardenaar, Ilya Markov, and Maarten de Rijke. 2018. A Click Sequence Model for Web Search. In *SIGIR '18*. ACM, 45–54.

[3] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *SIGIR '18*. ACM, 1093–1096.

[4] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *WSDM '18*. ACM, 108–116.

[5] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *RecSys '17*. ACM, 152–160.

[6] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation. In *IJCAI'15*. AAAI Press, 2069–2075.

[7] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *CoRR* abs/1705.03122 (2017).

[8] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *International Conference on Data Mining*. IEEE, 191–200.

[9] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM '15*. ACM, 1661–1670.

[10] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *SIGIR '18*. ACM, 355–364.

[11] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yugang Jiang, and Tatseng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Trans. Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW '17*. ACM, 173–182.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR'16*. 1–10.

[14] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing Historical Interaction Data for Faster Online Learning to Rank for Information Retrieval. In *WSDM '13*. ACM, 183–192.

[15] Xue Hong-Jian, Dai Xinyu, Zhang Jianbing, Huang Shujian, and Chen Jiajun. 2017. Deep Matrix Factorization Models for Recommender Systems. In *IJCAI '17*. 3203–3209.

[16] Dietmar Jannach. 2018. Keynote: Session-Based Recommendation – Challenges and Recent Advances. In *KI 2018: Advances in Artificial Intelligence*. Springer International Publishing, Cham, 3–7.

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD '08*. ACM, 426–434.

[19] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM '17*. ACM, 1419–1428.

[20] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-Behavioral Sequential Prediction with Recurrent Log-Bilinear Model. *IEEE Trans. Knowledge and Data Engineering* 29, 6 (2017), 1254–1267.

[21] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *KDD '18*. ACM, 1831–1839.

[22] Chen Ma, Yingxue Zhang, Qinglong Wang, and Xue Liu. 2018. Point-of-Interest Recommendation: Exploiting Self-Attentive Autoencoders with Neighbor-Aware Influence. In *CIKM '18*. ACM, 697–706.

[23] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *RecSys '17*. ACM, 130–137.

[24] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-based Recommendation. In *AAAI '19*. AAAI.

[25] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW '10*. ACM, 811–820.

[26] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *DLRS 2016*. ACM, 17–22.

[27] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM '18*. ACM, 565–573.

[28] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In *WWW '18*. International World Wide Web Conferences Steering Committee, 729–739.

[29] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. *CoRR* abs/1801.09251 (2018).

[30] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2018. Mix 'n match: Integrating Text Matching and Product Substitutability within Product Search. In *CIKM '18*. ACM, 1373–1382.

[31] Mengting Wan and Julian McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *RecSys '18*. ACM.

[32] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *SIGIR '15*. ACM, 403–412.

[33] Shoujin Wang, Longbing Cao, and Yan Wang. 2019. A Survey on Session-based Recommender Systems. *CoRR* abs/1902.04864 (2019).

[34] Bo Wu, Wen Huang Cheng, Yongdong Zhang, Huang Qiushi, Li Jintao, and Tao Mei. 2017. Sequential Prediction of Social Media Popularity with Deep Temporal Context Networks. In *IJCAI '17*. 3062–3068.

[35] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI'18*. 3926–3932.

[36] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *SIGIR '16*. ACM, 729–732.

[37] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep Learning based Recommender System: A Survey and New Perspectives. *arXiv preprint arXiv:1707.07435* (2017).

[38] Shuai Zhang, Lina Yao, Aixin Sun, Sen Wang, Guodong Long, and Manqing Dong. 2018. NeuRec: On Nonlinear Transformation for Personalized Ranking. *arXiv preprint arXiv:1805.03002* (2018).

[39] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM '17*. ACM, 425–434.