Abstract: Session-based recommendation aims to model a user’s intent and predict an item that the user may interact with in the next step based on an ongoing session. Existing session-based recommender systems mainly aim to model the sequential signals based on Recurrent Neural Network (RNN) structures or the item transition relations between items with Graph Neural Network (GNN) based frameworks to identify a user’s intent for recommendation. However, in real scenarios, there may be strong sequential signals existing in users’ adjacent behaviors or multi-step transition relations among different items. Thus, either RNN- or GNN-based methods can only capture limited information for modeling complex user behavior patterns. RNNs pay attention to the sequential relations among consecutive items, while GNNs focus on structural information, i.e., how to enrich the item embedding with its adjacent items. In this paper, we propose a Collaborative Co-attention Network for Session-based Recommendation (CCN-SR) to incorporate both sequential and structural information, as well as capture the co-relations between them for obtaining an accurate session representation. To be specific, we first model the ongoing session with an RNN structure to capture the sequential information among items. Meanwhile, we also construct a session graph to learn the item representations with a GNN structure. Then, we design a co-attention network upon these two structures to capture the mutual information between them. The designed co-attention network can enrich the representation of each node in the session with both sequential and structural information, and thus generate a more comprehensive representation for each session. Extensive experiments are conducted on two public e-commerce datasets, and the results demonstrate that our proposed model outperforms state-of-the-art baseline model for session based recommendation in terms of both Recall and MRR. We also investigate different combination strategies and the experimental results verify the effectiveness of our proposed co-attention mechanism. Besides, our CCN-SR model achieves better performance than baseline models with different session lengths.

Keywords: session-based recommendation; recurrent neural network; graph neural network; co-attention mechanism

1. Introduction

With the information increasing at a rapid speed on the Internet, recommender systems have been proposed to provide users with their required information in an efficient way [1–4]. Many general recommendation approaches rely on users’ historical behaviors to make personalized recommendations. For example, collaborative filtering (CF) [5] builds the user-item interaction matrix and learns user as well as item representations so as to fill in the matrix and make recommendations. However, in some real scenarios, users’ personal information, e.g., user IDs, is not available. For instance, users may not log into the recommender system when using some online shopping service. It is a challenging task to recommend users satisfying items only based on limited behaviors in a session in those cases. Session-based recommendation (SBRS) is then proposed to deal with the task and make recommendations based on an ongoing session [3].
Since the items in a session may be connected due to sequential relations, early methods model the transition and co-occurrence relations between items with ItemKNN [6] and Markov Chain [7]. However, those models have a strong assumption of the independence of the past interactions and are mainly based on the last behavior to make recommendations, which confines the recommendation accuracy for SBRS. Recently, Recurrent Neural Network (RNN) has played an important role in session-based recommendation tasks due to its ability in modeling the sequential relations among items. Hidasi et al. [3] first use Gated Recurrent Unit (GRU) to model user behavior sequences in a session and proposed the GRU4Rec model. After that, attention mechanism has been adopted and helped to boost the performance of session-based recommendation. Li et al. [8] apply the attention mechanism to distinguish different item importances and then combine the weighted hidden states and the last hidden state to make final recommendations. However, when users click some unrelated items in a session, RNN-based approaches may get misled by those noisy interactions, which results in an inaccurate session representation and unsatisfied recommendations. As RNNs mainly focus on the sequential transitions among items in a single way, Graph Neural Network (GNN)-based approaches have been proposed to enrich the item representation with its neighbors through propagating information between adjacent items. For example, Wu et al. [2] propose to construct a session graph to represent a session and then learn the item embeddings with graph neural networks, which achieves satisfied results. However, GNN-based approaches often ignore the sequential information among user behaviors and cannot capture long-term context information for the next item recommendation.

Thus, in our paper, we propose a Collaborative Co-attention Network for Session-based Recommendation (CCN-SR), which takes the advantages from both RNN as well as GNN structures. More specifically, we first input the user behaviors in an ongoing session, i.e., user interacted items, into a GRU network to model the sequential relations among those behaviors. Meanwhile, we also construct a session graph for the ongoing session and use another GNN network to model the structural information in those behaviors. After that, we can get the hidden state for each item in the session from the GRU network, and the node embedding for each item in the session from the GNN network. Then, we propose a co-attention mechanism to incorporate the sequential as well as structural information to get an accurate representation for the session. The co-attention mechanism can capture the mutual relations between these two kinds of information, i.e., sequential as well as structural information, and thus generate a more comprehensive representation of each session. Specifically, we design two strategies to achieve our co-attention mechanism, i.e., parallel co-attention and alternating co-attention. We conduct experiments on two public e-commerce datasets to verify the effectiveness of our CCN-SR model and explore the differences between the performances of our proposed two kinds of co-attention mechanisms as well as the simple concatenation strategy. The results demonstrate that our CCN-SR model can achieve better performance than the state-of-the-art baseline model in terms of both Recall and MRR.

The main contributions in this paper can be summarized as follows:

- To the best of our knowledge, we are the first to incorporate both structural as well as sequential information and capture their co-dependent relations for session-based recommendation;
- We propose a Collaborative Co-attention Network for Session-based Recommendation (CCN-SR) model, which introduces the co-attention mechanism upon RNN and GNN networks to get the mutual information between them and enrich the session representations;
- We conduct comprehensive experiments on two publicly available datasets by comparing with state-of-the-art baselines to validate the effectiveness of our proposal. Experimental results show that CCN-SR can beat the baselines in terms of both Recall and MRR.
We summarize related literature in Section 2. The details of our proposed model are described in Section 3. The experimental setups and datasets are introduced in Section 4. Finally, we give our analysis of the experimental results in Section 5 and the conclusion in Section 6.

2. Related Works

In this section, we give a summarization of the related literature for our work. We mainly divide them into three aspects: general recommendation approaches, session-based recommendation approaches and attention based approaches.

2.1. General Recommendation Approaches

General recommendation approaches have been applied widely in recommender systems, which predict users’ general preference based on their historical interactions. Most general recommenders are based on Collaborative Filtering (CF), which aims to factorize the user-item interaction matrix into two low rank matrices containing user latent vectors as well as item latent vectors [9–13]. Traditional methods such as Singular Value Decomposition (SVD) [14] generate a user’s preference towards an item with a linear product of the user’s latent vector and the item’s latent vector. However, the linear kernel often cannot model the users’ preference accurately, and many researchers have pointed out that nonlinearity has potential advantages for improving the performance of recommender systems with extensive experiments [15–17]. Thus, the deep learning-based recommendation approaches have been proposed and boost the performance of general recommendations.

Restricted Boltzmann Machines (RBM) [18–20] was proposed as an early neural based recommender system. It applies a two-layer undirected graph to model tabular data, such as users’ explicit ratings of movies. For top-N recommendation, He et al. [5] propose to use multi-layer perceptrons to model the two-way interaction between users and items, which captures the non-linear relationship between users and items and achieves satisfied results. Some other recommendation models [21–23] use a convolutional neural network (CNN) to integrate external information, e.g., the review text or contextual information, which helps to improve the recommendation performance.

However, those general recommendation models often ignore the changes in users’ preferences and always generate the same recommendations to a user. Thus, they are not suitable for session-based recommendation, where the recommended items should be adopted to a user’s current interest.

2.2. Session-Based Recommendation Approaches

Session-based recommendation aims to capture users’ dynamic preferences in an ongoing session. In early stages, Markov Chains has been adopted to model the transition relations between adjacent items [7,24–26]. Recently, neural network based approaches, e.g., RNN-based models, have been widely adopted for session-based recommendation. Hidasi et al. [3] first introduce Gated Recurrent Unit (GRU) to model the current session and propose a GRU4Rec model as well as a session-parallel mini-batch training process. Following [3], an improved RNN-based approach has been proposed in [27], which applies a data augmentation strategy and solves the distribution shifts in the input data. Hidasi and Karatzoglou [28] propose an improved loss function to optimize the training process of the GRU4Rec model and achieve good performance. Bogina and Kuflik [29] incorporate the dwell time to the RNN structure and boost the performance for session-based recommendation. It also indicates that the sequential relation among items in a session is importance when making recommendations. There are also some memory-based approaches for session-based recommendation. For example, Chen et al. [30] propose a Recommendation with User Memory Network (RUM) model, which uses external memory to store and distinguish users’ interactions in a session.
RNNs and memory network cannot capture some complex relations among items in a session, e.g., some structural information between those items. Thus, GNN-based approaches have been proposed [2,31]. Wu et al. [2] is the first to introduce graph neural network into session-based recommendation and propose a SR-GNN model. They construct a session graph for each session and then apply the gated graph neural network (GGNN) to generate node representations in the graph, which finally help to make recommendations. Based on this work, some researches take the long-term dependencies among items into consideration and generate more accurate session representations [32,33]. Moreover, Yu et al. [34] propose a target-attention mechanism within a graph neural network, which also improves the recommendation performance over the SR-GNN model. In addition, Qiu et al. [35] adopt a weight graph neural network to distinguish different importance of the propagated information. However, those GNN-based approaches cannot model the sequential relations between user behaviors accurately and only propagation information between adjacent items, which limits its ability in capturing context information in a session.

2.3. Attention-Based Recommendation Approaches

The attention mechanism helps to distinguish the importance of different items in a session, which can boost the performance for session-based recommendation [36,37]. Li et al. [8] propose a Neural Attentive Recommendation Machine (NARM) model, which regards the last hidden state modeled by a session-based RNN as the global encoder, and uses other hidden states for calculating attention weights of different items to capture users’ current intents. As for memory-based models, Liu et al. [36] propose a short-term attention memory priority model, i.e., Short-Term Attention/Memory Priority Model (STAMP), where the attention weights for different interactions are calculated based on the session context and the final records in the current session. SR-GNN model [2] also applies the attention mechanism to distinguish importance of different items in the current session, which is the same as the way in NARM [8].

The aforementioned methods often adopt the attention mechanism that is enhanced by the last hidden state, which is not suitable for capturing the mutual information between different structures, i.e., RNNs and GNNs. In contrast, we propose to use co-attention mechanism to generate co-dependent representations of each item in a session and thus can make more accurate recommendations.

3. Methods

The Collaborative Co-attention Network for Session-based Recommendation (CCN-SR) model we propose in this paper mainly contains four components: an RNN-based session encoder, a GNN-based session encoder, a co-attention network and a prediction layer. We show the main framework of our model in Figure 1, in which these components can be trained and optimized in an end-to-end way. In the following sections, we first describe the problem formulation as well as notations, and then we give detailed descriptions of each components in CCN-SR.

3.1. Problem Formulation and Notation

Given a user and their sequential interactions in a session, we aim to recommend their next interaction based their short-term preferences learned from previous behaviors in the session.

We denote the current session as $S = \{v_1, v_2, \ldots, v_T\}$, where $v_i$ is the $i$-th item interacted by a user in the session; $T$ denotes the number of events in the current session. In Figure 1, an embedding layer is built at the bottom of the network which is used for generating the item embeddings shared by both the RNN network as well as the GNN network. We use $v_i$ to indicate the embedding of $v_i$. We summarize the notations we use in our paper in Table 1.
Figure 1. Structure of the CCN-SR model.

Table 1. Summary of the main notations used in the paper.

| Notation | Description |
|----------|-------------|
| $S$ | the current session |
| $v_i$ | the embedding of item $v_i$ |
| $h_t$ | hidden state at timestep $t$ in RNN-based session encoder |
| $z_t$ | the update gate in GRU |
| $r_t$ | the reset gate in GRU |
| $W_z, U_z, W_r, U_r, W_o, U_o$ | trainable parameters in RNN-based session encoder |
| $S_r$ | the output of the RNN-based session encoder |
| $V_S$ | the unique items set of session $S$ |
| $E_s$ | the edges in the session graph $G_s$ |
| $V_s$ | the nodes in the session graph $G_s$ |
| $v_{s,i}$ | item $v_i$ in the current session, where $v_{s,i} \in V_S$ |
| $A_{out}$ | the matrix containing the connection between nodes in the session graph with outgoing edges |
| $A_{in}$ | the matrix containing the connection between nodes in the session graph with incoming edges |
| $z^s t$ | the update gate in GNN |
| $r^s t$ | the reset gate in GNN |
| $W_{z}, U_{z}, W_{r}, U_{r}, W_{o}, U_{o}$ | trainable parameters in GNN-based session encoder |
| $v^g_t$ | the node vector generated by the GNN-based session encoder for item $v_i$ |
| $S^g$ | the output of the GNN-based session encoder |
| $C$ | the affinity matrix in the co-attention network |
| $W_{sr}, W_{sg}, W_{r}, W_{hr}, w_{hg}$ | the weight parameters in the parallel co-attention strategy |
| $B_r, B_p, B_{sp}$ | trainable parameters in the parallel co-attention strategy |
| $B_{cg}, m, A_1, A_2, B^e$ | trainable parameters in the alternating co-attention strategy |
| $U_S$ | the final session representation of $S$ generated by the co-attention network |
| $y_i$ | the prediction score of item $v_i$ |
3.2. RNN-Based Session Encoder

Recurrent Neural Network (RNN) has been widely used to model the sequential data. Given a sequence like \{v_1, v_2, \ldots, v_T\}, RNN calculates a hidden state \( h_t \) for each step \( t \) in the sequence, which mainly contains summative information of the sequence until the step \( t \). \( h_t \) is computed based on the hidden state of its former step \( h_{t-1} \) and its current input \( v_t \):

\[
h_t = f(h_{t-1}, v_t),
\]

where \( f \) is the main function in RNN. Different RNN architectures, e.g., Long Short-Term Memory unit (LSTM) \[38\] and Gated Recurrent Unit (GRU), have different functions. In our paper, we use GRU as the RNN-based session encoder since it shows better performance than simple RNNs and LSTM \[3\].

GRU contains the input gate, reset gate and update gate, which are used to control the information propagated from former steps to the current step. The hidden state \( h_t \) can be calculated by a linear combination of former hidden state \( h_{t-1} \) and the candidate hidden state \( \hat{h}_t \):

\[
h_t = z_t h_{t-1} + (1 - z_t) \hat{h}_t,
\]

where the update gate \( z_t \) is given by:

\[
z_t = \sigma(W_z v_t + U_z h_{t-1}),
\]

where \( W_z \) and \( U_z \) are update parameters for \( v_t \) and \( h_{t-1} \), respectively. The candidate hidden state can be computed as:

\[
\hat{h}_t = \tanh(W_o v_t + r_t \odot U_o h_{t-1}),
\]

where \( \odot \) denotes the Hadamard product, which is an element-wise product of matrices. The reset gate \( r_t \) can be calculated by:

\[
r_t = \sigma(W_r v_t + U_r h_{t-1}),
\]

where \( W_r \) and \( U_r \) are reset parameters for \( v_t \) and \( h_{t-1} \), respectively.

As the hidden state of each step contains sequential information among user previous behaviors until this step as well as the user’s current intent, we collect the hidden state of each node of the start node of the edge. We then build the adjacency matrices \( A_{out} \) and \( A_{in} \), which represents the connection between nodes in the session graph with outgoing edges and incoming edges, respectively. By concatenating these two matrices, we can get the matrix \( A_s \), which is then used in the learning process of the graph neural network.
The graph neural network can incorporate a node’s neighbor features and update the node representation of \( v_{s,j} \) as follows:

\[
\begin{align*}
    a_{s,j}^t &= A_{s,j} [v_{s}^{t-1}, \ldots, v_{T}^{t-1}]^T \mathbf{H} + \mathbf{b}, \\
    z_{s,j}^t &= \sigma(\mathbf{W}_{s}^a a_{s,j}^t + \mathbf{U}_{s}^a v_{s}^{t-1}), \\
    r_{s,j}^t &= \sigma(\mathbf{W}_{s}^r a_{s,j}^t + \mathbf{U}_{s}^r v_{s}^{t-1}), \\
    \hat{v}_{s,j}^t &= \tanh(\mathbf{W}_{s}^g a_{s,j}^t + \mathbf{U}_{s}^g (r_{s,j}^t \odot v_{s}^{t-1})), \\
    v_{s,j}^t &= z_{s,j}^t \odot \hat{v}_{s,j}^t + (1 - z_{s,j}^t)v_{s,j}^{t-1},
\end{align*}
\]

where \( \mathbf{H} \in \mathbb{R}^{D \times 2D} \) indicates the weights, \( A_{s,j} \) is the column in \( \mathbf{A} \) corresponding to the node \( v_{s,j} \). \( a_{s,j}^t \), \( z_{s,j}^t \), and \( r_{s,j}^t \) are the update gate and reset gate. \( \mathbf{W}_{s}^a, \mathbf{U}_{s}^a, \mathbf{W}_{s}^r, \mathbf{U}_{s}^r, \mathbf{W}_{s}^g \) and \( \mathbf{U}_{s}^g \) are trainable parameters. \( \sigma(\cdot) \) is the sigmoid function. The final state of node \( v_{s,j} \) is calculated by a combination of its previous state \( v_{s,j}^{t-1} \) and the candidate state \( \hat{v}_{s,j}^t \). With several steps of updating, we can get all node vectors in the session until convergence. We denote the node vectors of items in the session as \( S_s = \{v_1^s, v_2^s, \ldots, v_T^s\} \) and \( S_s \in \mathbb{R}^{D \times T} \), where \( D \) is the dimension of each hidden state in \( S_s \).

### 3.4. Co-Attention Network

As we can see from Equation (6), the GNN-based encoder mainly captures the transition relations between adjacent items and models the structural information in the session. Meanwhile, the output of the RNN-based encoder contains the sequential information in the session. It is beneficial to incorporate this two kinds of information so as to generate a comprehensive session representation for making recommendations. Intuitively, using concatenation of the outputs of the RNN- as well as GNN-based session encoder, i.e., \( S_s \) and \( S_g \), would be a choice. However, we argue that the two kinds of information can provide context for each other and thus help to distinguish the importance of different items while simple concatenation cannot capture this mutual relation.

Thus, in this section, we design a co-attention mechanism to explore the relations between \( S_s \) and \( S_g \) and make an accurate session representation. Specifically, we design two strategies to achieve the co-attention mechanism, i.e., parallel co-attention and alternating co-attention. We give detailed analysis in the following sections.

#### 3.4.1. Parallel Co-Attention

After modeling the sequential as well as structural information in the session, we use \( S_s \) and \( S_g \) as inputs of our parallel co-attention mechanism. We show the detailed calculation process of the parallel co-attention mechanism in Figure 2. We first calculate the affinity matrix \( C \):

\[
C = \tanh(S_s^T \mathbf{W}_c S_g),
\]

where \( \mathbf{W}_c \in \mathbb{R}^{D \times D} \) is a transformer matrix. \( C \) can be regarded as a co-relation matrix between \( S_s \) and \( S_g \). We then use it as a context information and calculate the attention scores for hidden state of each step in \( S_s \):

\[
\mathbf{H}_s = \tanh(\mathbf{W}_{sr} S_s + \mathbf{W}_s^h h_s^T + (\mathbf{W}_{sk} S_k + \mathbf{W}_s^g v_g^T) C^T)
\]

\[
a_s = \text{softmax}(\mathbf{w}_s^T \mathbf{H}_s)
\]

and the attention scores for each node vector in \( S_g \) with:

\[
\mathbf{H}_g = \tanh(\mathbf{W}_{sg} S_g + \mathbf{W}_g^h v_g^T + \mathbf{W}_{sr} S_r + \mathbf{W}_g^g h_g^T) C
\]

\[
a_g = \text{softmax}(\mathbf{w}_g^T \mathbf{H}_g),
\]
where $W_{sr}, W_{sg} \in \mathbb{R}^{K \times D}, w_{hr}, w_{hg} \in \mathbb{R}^{K}$ are weight parameters for $S_r$ and $S_g$, respectively. Here, $\alpha_r \in \mathbb{R}^T$ and $\alpha_g \in \mathbb{R}^T$ are the co-attention scores of items in $S_r$ and $S_g$.

Here, $\alpha_r \in \mathbb{R}^T$ and $\alpha_g \in \mathbb{R}^T$ are the co-attention scores of items in $S_r$ and $S_g$, respectively.

In both Equations (8) and (10), we emphasis the importance of the user last behavior, which is marked with red lines in Figure 2, i.e., $h_r^T$ and $v_g^T$. It is because that the final interaction often plays an important role in predicting the user’s next behavior in session-based recommendation especially in some e-commerce scenarios, which has been proved by many researches, e.g., NARM [8] and STAMP [36]. Here, we use $W_r, W_g \in \mathbb{R}^{K \times D}$ as the weight parameters for $h_r^T$ and $v_g^T$ so as to emphasis the last behavior adaptively.

After calculating the co-attention weights, we can generate co-dependent session representations modeled by the RNN-based session encoder and the GNN-based session encoder as follows:

$$U_{\text{co}-r} = \sum_{n=1}^{T} \alpha_{n}^{r} h_{n}^{r},$$

$$U_{\text{co}-g} = \sum_{n=1}^{T} \alpha_{n}^{g} v_{n}^{g}. \tag{12}$$

Combining $U_{\text{co}-r}$ with $h_r^T$ and $U_{\text{co}-g}$ with $v_g^T$, respectively, we can get:

$$U_r = B_r [h_r^T; U_{\text{co}-r}],$$

$$U_g = B_g [v_g^T; U_{\text{co}-g}], \tag{13}$$

where $B_r \in \mathbb{R}^{D \times 2D}$ and $B_g \in \mathbb{R}^{D \times 2D}$ are used to compress the two vectors to get the hybrid representations.

Finally, we use a concatenation of $U_r$ and $U_g$ to generate the final session representation of $S$ as:

$$U_S = B'[U_r; U_g], \tag{14}$$

where $B' \in \mathbb{R}^{D \times 2D}$.

3.4.2. Alternating Co-Attention

In the aforementioned parallel co-attention strategy, we calculate the co-dependent representations $U_{\text{co}-r}$ and $U_{\text{co}-g}$ in parallel for a time. In this section, we introduce another co-attention strategy, i.e., alternating co-attention, which can also capture the mutual information between $S_r$ and $S_g$, as well as integrate the sequential information and structural information for session-based recommendation. We show the details of the alternating co-
attention mechanism in Figure 3. As shown in Figure 3, we sequentially alternate between the initial outputs of the RNN-based session encoder and the GNN-based session encoder, as well as the attentive representations of them, which can thus take more information into consideration.

Figure 3. Details of the alternating co-attention mechanism.

First, we calculate the affinity matrix $C$ using Equation (7). Then we do normalization row-wise to produce the attention weights $A^g$ across the outputs of the RNN-based session encoder, i.e., $S_r$, for each node vector in the outputs of the GNN-based session encoder, i.e., $S_g$. At the same time, $C$ is also normalized column-wise to conduct the attention scores $A'$ across $S_g$ for each hidden state in the outputs of the RNN-based session encoder, i.e., $S_r$:

$$A^g = \text{softmax}(C),$$
$$A' = \text{softmax}(C^T).$$

Next, we generate attentive representation of $S_r$ as:

$$C' = S_r A'.$$
We can then incorporate $C'$ with the initial representation $S_r$ to generate the candidate attentive representation of $S_g$ as:

$$\hat{C}^g = [S_r; C']A,$$  \hfill (17)

where $\hat{C}^g \in \mathbb{R}^{2D \times T}$. In this way, we can preserve the initial sequential information containing in $S_r$.

We also keep the initial structural information and integrate $S_g$ into $\hat{C}^g$ and generate the final attentive representation of $S_g$ as:

$$C^g = B^g[S_g; \hat{C}^g] , \hfill (18)$$

where $B^g \in \mathbb{R}^{D \times 3D}$. We finally concatenate $C^g$ and $C'$ to generate the reformulated representations of behaviors in the session and adopt the same attention mechanism as NARM [8]:

$$C_i = [C'_i; C'_T] \quad \forall i = 1, 2, \ldots , T. \hfill (19)$$

The global and local representations of the session $S$ can be denoted as $U^S_{\text{global}}$ and $U^S_{\text{local}}$:

$$U^S_{\text{global}} = C_T,$$

$$U^S_{\text{local}} = \sum_{i=1}^{T} \alpha_i C_i , \hfill (20)$$

where $\alpha$ is the weighted factor calculated by:

$$\alpha_i = m^T \sigma (A_1 C_i + A_2 C_T) , \hfill (21)$$

where $\sigma$ is an activation function, and $m$, $A_1$ and $A_2$ are learnable parameters. Then, by concatenating the global as well as local representations, we can generate the final session representation as:

$$U_S = B^g[U^S_{\text{global}}; U^S_{\text{local}}], \hfill (22)$$

where $B^g \in \mathbb{R}^{D \times 2D}$.

3.5. Prediction and Optimization

With the co-attention mechanism, we can generate the final session representation $U_S$, which integrates the sequential as well as structural information in the ongoing session. Then, we make predictions by conducting the dot product of the session representation and the embedding of each candidate item:

$$y_i = U^T_S v_i \quad \forall i = 1, 2, \ldots , |V|. \hfill (23)$$

As $y_i$ is an unnormalized value, we then do softmax across all candidate items so as to get the prediction probability of item $i$. For training, we apply the widely used cross-entropy loss as our loss function:

$$L(p, q) = -\sum_{i=1}^{|V|} p_i \log(q_i) , \hfill (24)$$

where $p$ is the ground truth distribution while $q$ is the prediction probability distribution generated based on Equation (23). Then, we can optimize our model with Equation (24).

4. Experimental Setup

In order to investigate the effectiveness of our proposal, we compare the recommendation performance of CCN-SR and several baseline methods on two public e-commerce datasets. In this section, we introduce the baselines, datasets and experimental setups in detail.
4.1. Model Summary

We compare our model with several baselines including traditional methods, i.e., Item-pop and BPR-MF, and neural network based approaches, i.e., RNN-based and GNN-based models. Our baselines are as follows:

**Item-pop** It recommends items with high popularities, i.e., items with a large number of interactions [40].

**BPR-MF** Factorization-based methods Bayesian personalized ranking (BPR-MF) proposes to use a pairwise ranking loss to optimize the matrix factorization model and make recommendations [41].

**GRU4Rec** It proposes an RNN model for session-based recommendation, which utilizes session-parallel mini-batches as well as a pair-wise loss function for training [3].

**NARM** It is an RNN based model, which also applies the attention mechanism to emphasis the importance of the last item in the ongoing session [8].

**SR-GNN** It proposes to construct a session graph for each session and then adopt the Gated Graph Neural Network (GGNN) model as well as the attention mechanism to capture transition relations among items [2].

We also investigate the performance of different variants of our model proposed in this paper:

**CCN-SR** Our proposed CCN-SR model, which adopts the parallel co-attention strategy upon the RNN and GNN networks to get the mutual information between them.

**CCN-SR** Our proposed CCN-SR model, which adopts the alternating co-attention strategy upon the RNN and GNN networks to get the mutual information between them.

4.2. Datasets and Experimental Setup

4.2.1. Datasets

We evaluate our model as well as the baselines on two public e-commerce datasets:

**Tmall** Tmall is a dataset released by Taobao. It contains records of online transactions, with 884 users, 9531 brands and 182,880 interactions.

**Tianchi** Tianchi is a dataset provided by Alibaba. ([https://tianchi.aliyun.com/getStart/information.htm?spm=5176.100067.5678.2.3oa8bd933N6Rr&raceId=231522, accessed on 11 June 2021]), which is larger than the Tmall dataset. It is based on user-commodity behavior data of Alibaba’s Mobile-Commerce platforms. It contains 23,291,027 interactions of 20,000 customers on 4,758,484 items within a month.

Both of the two datasets contain several lines where each line records a user ID, an item ID, and a timestamp when the user interacts with the item. Following [42], we preprocess the datasets as follows. For the Tmall dataset, we filter out users with fewer than 5 interactions and items that appear less than 5 times. For the Tianchi dataset, we filter out users with fewer than 20 interactions and items with fewer than 50 interactions. The sessions with less than 3 items or more than 200 items are also filtered out. The characteristics of the datasets after preprocessing are summarized in Table 2.

Table 2. Dataset statistics.

| Dataset | Tmall | Tianchi |
|---------|-------|---------|
| # of items | 3751 | 36,369 |
| # of interactions | 131,857 | 3,438,390 |
| # interactions in training set | 99,057 | 2,632,480 |
| # interactions in test set | 32,800 | 805,910 |
4.2.2. Settings and Parameters

We divide two datasets into training set and test set for training and evaluation, respectively. For the Tmall dataset, we use the last 30 days of interactions, as the test set and the remaining days of interactions are regarded as the training set. For Tianchi, the training set consists of all but the last 7 days of interactions; the test set contains the remaining 7 days of interactions. We also remove items in the test sets that do not appear in the training set [2]. Following [2], we apply the data augmentation strategy for our model as well as all baseline models in our paper.

We adopt the widely used metrics, i.e., Recall and MRR, to evaluate the performance of all models [8,36]. Recall considers whether the ground truth item is contained in the recommendation list; MRR evaluates the ranking accuracy of the model, i.e., whether the ground truth item is at the top position of the recommendation list.

We use the Adam [43] as our optimizer when training the model; the learning rate is initialized as 0.001; the batch size is set as 100 and the dimension of the item embeddings is set to 100. We initialize all trainable parameters using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. Unless specified differently, we set the recommendation number $N$ as 20.

5. Results

In this section, we conduct several experiments to evaluate the performance of our model as well as the baselines. We first analyze the overall performance of all models in Section 5.1. We then investigate the impact of different combination strategies of the sequential and structural information on our model, i.e., the parallel co-attention strategy, the alternating co-attention strategy, and the concatenation strategy, in Section 5.2. Finally, we analyze the performance of our model as well as the baseline models on sessions with different lengths in Section 5.3.

5.1. Overall Performance

As shown in Table 3, as for baselines, it is obvious that neural-based approaches, i.e., GRU4Rec, NARM and SR-GNN, outperform traditional methods, i.e., Item-pop and BPR-MF. When comparing among the neural-based approaches, for the RNN-based models, i.e., GRU4Rec and NARM, we can see that NARM generally outperforms GRU4Rec in terms of Recall@20 and MRR@20 on both of the two datasets. It is because the attention mechanism in NARM can distinguish different item importances, which can help to filter out some noisy interactions and capture users’ main purposes in the current session. As for the GNN-based model, i.e., SR-GNN, it achieves the best performance among all of the baseline models, which indicates that modeling the transition relations between adjacent items can help to improve the recommendation accuracy.

Table 3. 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 best model vs. the best baseline is determined by a paired $t$-test ($\triangle$ for $p$-value $\leq 0.05$).

| Model  | Tmall Recall@20 | Tmall MRR@20 | Tianchi Recall@20 | Tianchi MRR@20 |
|--------|-----------------|--------------|-------------------|----------------|
| Item-pop| 0.1509          | 0.0432       | 0.0044            | 0.0011         |
| BPR-MF  | 0.1659          | 0.0537       | 0.0014            | 0.0017         |
| GRU4Rec | 0.6815          | 0.6234       | 0.4213            | 0.2263         |
| NARM    | 0.7314          | 0.6969       | 0.4393            | 0.2198         |
| SR-GNN  | 0.7370          | 0.7072       | 0.4576            | 0.2294         |
| CCN-SR$_{pu}$ | 0.7523     | 0.7226       | 0.4655            | 0.2361         |
| CCN-SR$_{al}$ | 0.7435     | 0.7254       | 0.4774$^\triangle$| 0.2376$^\triangle$ |
As to our proposed CCN-SR models, we can see that both CCN-SR_{pa} and CCN-SR_{al} outperform the best baseline model in terms of Recall@20 as well as MRR@20 on Tmall and Tianchi. It demonstrates the effectiveness of considering structural as well as sequential information simultaneously and capturing the co-attentive relations between them for session-based recommendation. Besides, the improvements of our model over the best baseline, i.e., SR-GNN, are significant in terms of Recall@20 and MRR@20 on Tianchi. It may be because Tianchi is sparser than Tmall and the recommendation task is more difficult on Tianchi. However, CCN-SR takes both sequential as well as structural information into consideration, and thus can generate accurate session representations even on a sparse dataset.

Comparing the two variants of CCN-SR, we can find that CCN-SR_{al} shows better performance than CCN-SR_{pa} in terms of all metrics on both datasets except Recall@20 on the Tmall dataset. It maybe because the alternating co-attention strategy incorporates both of the initial and attentive information with Equations (17) and (18), which helps to preserve enough information for recommendation.

5.2. Impact of Different Combination Strategies

In this section, we aim to explore the effect of different strategies combining the structural as well as sequential information on the performance of CCN-SR. In Section 3.4, we propose two co-attention strategies to capture the relation between this two kinds of information and combine them for session-based recommendation, i.e., parallel co-attention and alternating co-attention. Besides, simply using concatenation of $S_r$ and $S_g$ is also an intuitive way to integrate the two kinds of information for session-based recommendation. Thus, in this section, we test the performance of CCN-SR with three different combination strategies, i.e., parallel co-attention, alternating co-attention and concatenation, on the top N recommendation task, with $N = 10, 20, 30, 40, 50$. We denote the three models as CCN-SR_{pa}, CCN-SR_{al} and CCN-SR_{concat} correspondingly. The results of them on the Tmall and Tianchi datasets are presented in Figure 4.

On the Tmall dataset, as shown in Figure 4a,b, we can see that with the increases of the recommendation number $N$, i.e., from 10 to 50, the performance of all models improves in terms of Recall and MRR. Our proposed CCN-SR_{pa} and CCN-SR_{al} always shows better performance than CCN-SR_{concat}, which indicates that simply using concatenation cannot fully take advantages of the structural as well as sequential information. Specifically, our CCN-SR_{al} model shows more improvements over CCN-SR_{concat} in terms of MRR when there are less recommendations. For example, when the recommendation number $N = 10$ and $N = 50$, CCN-SR_{al} outperforms CCN-SR_{concat} by 2.4% and 1.3% in terms of MRR, respectively. It may be because our CCN-SR_{al} model can always rank the ground truth item at the top position of the list. The ranking accuracy of CCN-SR_{al} is not influenced much by the recommendation numbers. This also demonstrates that our model can be applied in some scenarios where the number of recommendations is limited, e.g., mobile recommendations.

For the Tianchi dataset shown in Figure 4c,d, the results are similar to that on the Tmall dataset, except that CCN-SR_{al} outperforms CCN-SR_{pa} in terms of both Recall and MRR on Tianchi while CCN-SR_{al} beats CCN-SR_{pa} only in terms of MRR on Tmall. It indicates that CCN-SR_{al} is better at improving the ranking accuracy than CCN-SR_{pa}, since CCN-SR_{al} takes more information (both initial and attentive information) into consideration through the alternating co-attention calculation process.
5.3. Impact of the Current Session Length

In this section, we evaluate the performance of our models as well as baselines with different lengths of sessions. We group the sessions on the test sets into short, medium and long sessions. For Tmall, we regard sessions with no more than five interactions as short sessions, sessions with more than 10 interactions as long sessions and others as medium sessions. For Tianchi, short sessions contain no more than 25 interactions, long sessions contain more than 50 interactions and others are medium. We report the results for short, medium and long sessions on the test set of Tmall and Tianchi in Figure 5.

Clearly, as shown in Figure 5a,b, with the increases of the session length, the performance of all models in terms of Recall@20 and MRR@20 improves, and our CCN-SR models always achieve better performance than those baseline models on the Tmall dataset. For baselines, we can find that SR-GNN and NARM shows better performance than GRU4Rec. However, the improvements of SR-GNN over GRU4Rec decrease with the increases of session length. For instance, SR-GNN beat GRU4Rec by 10.5% and 17.8% in terms of Recall@20 and MRR@20 with short sessions on Tmall while 5.7% and 9.0% with long sessions on Tmall. It indicates that the RNN model is better at dealing with long sessions than short sessions. As for our CCN-SR models, the improvements of our models over SR-GNN in terms of MRR@20 is more than that in terms of Recall@20. For instance, CCN-SR_{al} beats SR-GNN by 1.6% and 3.2% in terms of Recall@20 and MRR@20 with long sessions...
on Tmall. This demonstrates that our model can achieve better performance in terms of recommendation ranking accuracy.

![Recall@20 on Tmall](image1)

![MRR@20 on Tmall](image2)

![Recall@20 on Tianchi](image3)

![MRR@20 on Tianchi](image4)

**Figure 5.** Effect on performance of five models in terms of Recall@20 and MRR@20 with different session lengths, tested on the Tmall and Tianchi datasets.

On the Tianchi dataset reported in Figure 5c,d, the performance of all models decreases with the increases of the session length and our proposal can still achieve the best performance compared with the baseline models. Specifically, for Recall@20, SR-GNN outperforms GRU4Rec on short sessions. However, the gap between the performance of the two models becomes smaller when the lengths of sessions change from short to medium. To be more specific, on long sessions, GRU4Rec even shows better performance than SR-GNN. This phenomenon is more obvious for MRR@20. SR-GNN beats GRU4Rec by 9.5% and 3.7% in terms of Recall@20 and MRR@20 with short sessions while the improvements of GRU4Rec over SR-GNN are 2.8% and 19.3% for Recall@20 and MRR@20 with long sessions. This demonstrates that the RNN and GNN models are good at dealing with different lengths of sessions and thus integrating the information learned from them can help to boost the performance for session-based recommendation.

6. Conclusions

In this paper, we propose a collaborative co-attention network for session-based recommendation, i.e., CCN-SR. CCN-SR incorporates both sequential as well as structural information learned from an RNN-based session encoder and a GNN-based session en-
coder, and capture their co-dependent relations for session-based recommendation. Specifically, we propose two co-attention strategies, i.e., parallel co-attention and alternating co-attention, to generate co-dependent representations of the two kinds of information and then combine them for making recommendations. We conduct comprehensive experiments to verify the effectiveness of our model and explore the impact of different combination strategies. We prove that our proposed co-attention mechanism shows more competitive performance with different recommendation numbers than the simple concatenation strategy. As to sessions with different lengths, the experimental results demonstrate that our CCN-SR model outperforms state-of-the-art baseline model across all session lengths. Our proposal shows more competitive performance especially when the recommendation number is smaller. Thus, it can be applied in some mobile recommendation scenarios, where the length of recommendation list is limited.

As to limitations of this work, on the one hand, our proposal has higher computation complexity than the baseline models due to the co-attention network; on the other hand, the improvements of our model over the best baseline model are not significant in some cases, which may be due to the sparsity of different datasets. The improvements of our model over the baselines are more obvious on the dataset, which is sparser. Thus, we plan to explore our model with more datasets in the future.

As for future work, on the one hand, we plan to incorporate some external knowledge, e.g., category information and content information, to capture the item relations more accurately [44–48]. For example, some items are complement products to each other and should be recommended together, especially in some e-commerce scenarios. On the other hand, different user behaviors, e.g., “click”, “collect” and “buy”, can also provide important information for capturing the user’s preference. For instance, “collect” and “buy” show a stronger consumption motivation of a user for an item than “click”. Thus, we plan to extend our model with the behavior information to generate more informative representations of sessions [49–51].

Author Contributions: Conceptualization, W.C. and H.C.; methodology, W.C.; validation, W.C.; formal analysis, W.C.; investigation, W.C.; resources, W.C.; data curation, W.C.; writing—original draft preparation, W.C.; writing—review and editing, H.C.; visualization, W.C.; supervision, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Tianchi dataset can be found at https://tianchi.aliyun.com/getStart/information.htm?spm=sf176.100067.5678.2.30a8b6d933N6Re&raceld=231522 (accessed on 11 June 2021).

Acknowledgments: We would like to thank the editor-in-chief and reviewers for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Perez, H.; Tah, J.H.M. Improving the Accuracy of Convolutional Neural Networks by Identifying and Removing Outlier Images in Datasets Using t-SNE. Mathematics 2020, 8, 662. [CrossRef]
2. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-Based Recommendation with Graph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 346–353.
3. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based Recommendations with Recurrent Neural Networks. arXiv 2015, arXiv:1511.06939.
4. Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. Mathematics 2021, 9, 197. [CrossRef]
5. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
6. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, 1–5 May 2001; pp. 285–295.

7. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th international conference on World wide web 2010, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.

8. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural Attentive Session-based Recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1419–1428.

9. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 2019, 52, 1–38. [CrossRef]

10. Su, X.; Khoshgoftaar, T.M. A Survey of Collaborative Filtering Techniques. Adv. Artif. Intell. 2009, 4. [CrossRef]

11. Betru, B.T.; Onana, C.A.; Batchuki, B. Deep Learning Methods on Recommender System: A Survey of State-of-the-art. Int. J. Comput. Appl. 2017, 162, 17–22.

12. Liu, J.; Wu, C. Deep Learning Based Recommendation: A Survey. In Proceedings of the International Conference on Information Science and Applications; Springer: Singapore, 2017; pp. 451–458.

13. Sardinas, C.; Ballas Papadatos, G.; Varlamis, I. Optimizing Parallel Collaborative Filtering Approaches for Improving Recommendation Systems Performance. Information 2019, 10, 155. [CrossRef]

14. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. Computer 2009, 42, 30–37. [CrossRef]

15. Li, S.; Kawale, J.; Fu, Y. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, VIC, Australia, 19–23 October 2015; pp. 811–820.

16. Wu, Y.; DuBois, C.; Zheng, A.X.; Ester, M. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, 22–25 February 2016; pp. 153–162.

17. Sedhain, S.; Menon, A.; Sanner, S.; Xie, L. AutoRec: Autoencoders Meet Collaborative Filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.

18. Salakhutdinov, R.; Mnih, A.; Hinton, G. Restricted Boltzmann Machines for Collaborative Filtering. In Proceedings of the 24th International Conference on World Wide Web 2007, ICML ’07, Corvalis, OR, USA, 20–24 June 2007; pp. 791–798.

19. Truyen, T.T.; Phung, D.Q.; Venkatesh, S. Ordinal Boltzmann Machines for Collaborative Filtering. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI ’09, Montreal, Canada, 18–21 June 2009; pp. 548–556.

20. Liu, X.; Ouyang, Y.; Rong, W.; Xiong, Z. Item Category Aware Conditional Restricted Boltzmann Machine Based Recommendation. In Proceedings of the 22nd International Conference, ICONIP 2015, Istanbul, Turkey, 9–12 November 2015; pp. 609–616.

21. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems; Association for Computing Machinery: New York, NY, USA, 2016; pp. 233–240.

22. Wang, C.; Blei, D.M. Collaborative Topic Modeling for Recommending Scientific Articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Association for Computing Machinery: New York, NY, USA, 2011; pp. 448–456.

23. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep Content-based Music Recommendation. In Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS ’13), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2643–2651.

24. Shani, G.; Heckerman, D.; Brafman, R.I. An MDP-Based Recommender System. J. Mach. Learn. Res. 2005, 6, 1265–1295.

25. Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; Cheng, X. Learning Hierarchical Representation Model for NextBasket Recommendation. In Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval; Association for Computing Machinery: New York, NY, USA, 2015; pp. 403–412.

26. Wang, B.; Cai, W. Attention-Enhanced Graph Neural Networks for Session-Based Recommendation. Mathematics 2020, 8, 1607. [CrossRef]

27. Tan, Y.K.; Xu, X.; Liu, Y. Improved Recurrent Neural Networks for Session-based Recommendations. In Proceedings of the 1st Workshop on Mobile Medical Applications; Association for Computing Machinery (ACM): New York, NY, USA, 2016; pp. 17–22.

28. Hidasi, B.; Karatzoglou, A. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management; Association for Computing Machinery: New York, NY, USA, 2018; pp. 843–852.

29. Bogina, V.; Kuflik, T. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks. In Proceedings of the Eleventh ACM International Conference on Recommender Systems; Association for Computing Machinery: New York, NY, USA, 2017; pp. 108–116.

30. Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; Zha, H. Sequential Recommendation with User Memory Networks. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining; Association for Computing Machinery (ACM): New York, NY, USA, 2018; pp. 108–116.

31. Wang, B.; Cai, W. Knowledge-Enhanced Graph Neural Networks for Sequential Recommendation. Information 2020, 11, 388. [CrossRef]

32. Xu, C.; Zhao, P.; Liu, Y.; Sheng, V.S.; Xu, J.; Zhuang, F.; Fang, J.; Zhou, X. Graph Contextualized Self-Attention Network for Session-based Recommendation. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, 10–16 August 2019; pp. 3940–3946.
33. Pan, Z.; Cai, F.; Chen, W.; Chen, H.; de Rijke, M. Star Graph Neural Networks for Session-based Recommendation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management; Association for Computing Machinery (ACM): New York, NY, USA, 2020; pp. 1195–1204.

34. Yu, F.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi’an, China, 25–30 July 2020; pp. 1921–1924.

35. Qiu, R.; Li, J.; Huang, Z.; Yin, H. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. In Proceedings of the the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 579–588.

36. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In Proceedings of the KDD’18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 1831–1839.

37. Pan, Z.; Cai, F.; Ling, Y.; de Rijke, M. Rethinking Item Importance in Session-based Recommendation. In Proceedings of the SIGIR ’20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval Virtual Event China 25–30 July 2020; pp. 1837–1840.

38. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef] [PubMed]

39. Ren, P.; Chen, Z.; Li, J.; Ren, Z.; Ma, J.; de Rijke, M. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. Proc. Conf. AAAI Artif. Intell. 2019, 33, 4806–4813. [CrossRef]

40. Adomavicius, G.; Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 2005, 17, 734–749. [CrossRef]

41. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence; AUAI Press: Arlington, VA, USA, 2009; pp. 452–461.

42. Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. A Dynamic Recurrent Model for Next Basket Recommendation. In Proceedings of the SIGIR ’16: The 39th International ACM SIGIR conference on research and development in Information Retrieval Pisa, Italy, 17–21 July 2016; pp. 729–732.

43. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv 2014, arXiv:1412.6980.

44. Zheng, L.; Noroozi, V.; Yu, P.S. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining; ACM: New York, NY, USA, 2017; pp. 425–434.

45. Tay, Y.; Tuan, L.A.; Hui, S.C. Multi-Pointer Co-Attention Networks for Recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018.

46. Hao, J.; Zhao, T.; Li, J.; Dong, X.L.; Faloutsos, C.; Sun, Y.; Wang, W. P-Companion: A Principled Framework for Diversified Complementary Product Recommendation. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management; ACM: New York, NY, USA, 2020; pp. 2517–2524.

47. Abbas, S.M.; Alam, K.A.; Shamshirband, S. A Soft-Rough Set Based Approach for Handling Contextual Sparsity in Context-Aware Video Recommender Systems. Mathematics 2019, 7, 740. [CrossRef]

48. Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Comparative Analysis of New Personalized Recommender Algorithms with Specific Features for Large Scale Datasets. Mathematics 2020, 8, 1106. [CrossRef]

49. Wan, M.; McAuley, J. Item Recommendation on Monotonic Behavior Chains. In Proceedings of the 12th ACM Conference on Recommender Systems; ACM: New York, NY, USA, 2018.

50. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional Sequence to Sequence Learning. arXiv 2017, arXiv:1705.03122.

51. Borisov, A.; Wardenaar, M.; Markov, I.; de Rijke, M. A Click Sequence Model for Web Search. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval; ACM: New York, NY, USA, 2018; pp. 45–54.