Rethinking Adjacent Dependency in Session-based Recommendations

Qian Zhang[1], Shoujin Wang[2], Wenpeng Lu[1(✉)], Chong Feng[3], Xueping Peng[4], and Qingxiang Wang[1]

[1] School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
qianzhang9706@gmail.com, wenpeng.lu@qlu.edu.cn,wangqx@qlu.edu.cn
[2] The School of Computing Technologies, RMIT University, Melbourne, Australia
shoujin.wang@rmit.edu.au
[3] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
fengchong@bit.edu.cn
[4] Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia
xueping.peng@uts.edu.au

Abstract. Session-based recommendations (SBRs) recommend the next item for an anonymous user by modeling the dependencies between items in a session. Benefiting from the superiority of graph neural networks (GNN) in learning complex dependencies, GNN-based SBRs have become the main stream of SBRs in recent years. Most GNN-based SBRs are based on a strong assumption of *adjacent dependency*, which means any two adjacent items in a session are necessarily dependent here. However, based on our observation, the adjacency does not necessarily indicate dependency due to the uncertainty and complexity of user behaviours. Therefore, the aforementioned assumption does not always hold in the real-world cases and thus easily leads to two deficiencies: (1) the introduction of *false dependencies* between items which are adjacent in a session but are not really dependent, and (2) the missing of *true dependencies* between items which are not adjacent but are actually dependent. Such deficiencies significantly downgrade accurate dependency learning and thus reduce the recommendation performance. Aiming to address these deficiencies, we propose a novel review-refined inter-item graph neural network (RI-GNN), which utilizes the topic information extracted from items' reviews to refine dependencies between items. Experiments on two public real-world datasets demonstrate that RI-GNN outperforms the state-of-the-art methods[5].

Keywords: Recommender system · Session-based recommendation · Graph neural network · Adjacent dependency.

[5] The implementation is available at https://github.com/Nishikata97/RI-GNN.

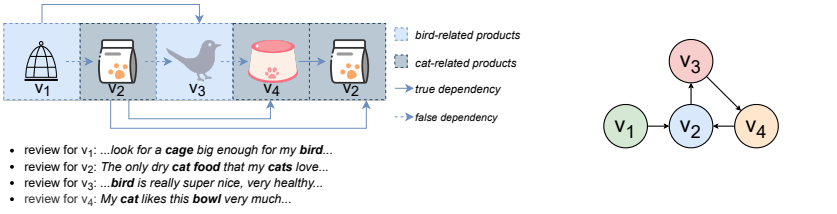
1 Introduction

In recent years, session-based recommendations (SBRs) have attracted extensive attention [4, 14] for its strong capability to capture users' dynamic and short term preference. SBRs recommend the next item to a user by modeling the sequential dependencies over items within sessions.

Driven by the development of deep learning, many neural network based SBRs have been developed. Among them, recurrent neural network (RNN) and graph neural network (GNN) [17] based approaches have shown good performance. RNN-based methods attempt to capture sequential dependencies between items within the sessions, which is based on the assumption that there is a strict chronological order inside the session [4]. However, this assumption does not always hold in the real-world scenarios since users' behaviours are usually uncertain and dynamic and thus not all interacted items in one session are sequentially dependent. Benefiting from the capability of GNN in learning complex dependencies, many GNN-based SBRs [3, 9, 18, 19] have been proposed. Leveraging the flexibility of graph structure used in GNN, the problem of strict chronological order confusing RNN-based methods is thus alleviated.

However, GNN-based SBRs often rely heavily on the strong assumption of adjacent dependency, namely, the adjacent items within one session are necessarily dependent. This is determined by its particular work mechanism. Specifically, most GNN-based SBRs first convert a given session consisting of a sequence of interacted items into a session graph by mapping each item to a node and the adjacency relation between any two items to an edge [19] to indicate the dependency between them, as shown in Fig. 1(b). This common practice of constructing session graphs often leads to two significant deficiencies: (1) the introduction of **false dependencies between adjacent but actually independent items** in a session, e.g., the item v_1 (i.e., a bird cage) and item v_2 (i.e., cat food) in the session described in Fig. 1(a), and (2) the missing of **true dependencies between items which are non-adjacent but actually dependent** in a session, e.g., the item v_1 and item v_3 (i.e., a bird) in the session described in Fig. 1(a). In practice, both false dependencies and true dependencies mentioned above are not uncommon in the real-world cases [16]. Obviously, these two deficiencies significantly downgrade the accurate learning of inter-item dependencies embedded in session data and thus reduce the performance of the downstream next-item recommendations. Therefore, it is critical to refine the dependencies between items by identifying and keeping all true dependencies while removing the false ones.

In practice, in addition to the session information, the review information associated with items can reveal dependencies between them to some degree. For example, the reviews associated with item v_1 and v_3 shown in Fig. 1 (a) are closely related and fall into the same topic. This actually provides extra information to enable the possibility to refine dependencies between items in sessions. To this end, we propose review-refined inter-item graph neural network (RI-GNN) to address the two deficiencies mentioned above in this paper. By leveraging the topic information from reviews written for items, RI-GNN can not only reduces



(a) A session $s_1 = [v_1, v_2, v_3, v_4, v_2]$ and the associated reviews.

(b) The corresponding adjacency-driven inter-item graph.

Fig. 1: A toy example for the construction of adjacency-driven inter-item graph.

the false dependencies between adjacent but actually independent items, but also well captures true dependencies between non-adjacent but dependent items which are usually ignored or weaken by existing GNN-based SBRs. The main contributions of this work are summarized below:

- We propose and discuss a novel and important research question: *does adjacency necessarily indicate dependency between items in sessions*? We perform a preliminary exploration with the hope of shedding some light in this area.
- We propose a novel review-refined inter-item graph neural network, called RI-GNN, for session-based recommendations. To the best of our knowledge, this is the first work for leveraging reviews to enhance the dependency learning for SBR on anonymous sessions.
- We propose a novel method for constructing a novel review-refined inter-item graph for each session. In the graph, reviews for items are employed to filter out the false dependencies between some adjacent items, and recall the true dependencies between non-adjacent items missed by existing methods.

2 Related Work

2.1 Session-based Recommendation

Existing methods for SBR can be summarized into: (1) Markov chain-based SBR; (2) RNN-based SBR; (3) Attention-based SBR; and (4) GNN-based SBR.

(1) *Markov chain-based SBR.* Early researches on SBR rely Markov chain to model the short-term dependencies to predict the next item. For example, FPMC [10] combined Markov chain and matrix factorization to model sequential behavior between two adjacent items and recommend next item. However, Markov chain-based methods only focus on first-order dependencies between adjacent items, while neglecting the high-order dependencies between long-distance items. (2) *RNN-based SBR.* Due to the powerful ability in modeling sequential data, RNN-based methods are applied widely to SBR [4, 7, 8]. GRU4Rec [4] first applied RNN to SBR, which adopted gated recurrent unit (GRU) to model the dependencies within sessions. However, RNN-based SBR also suffers the similar problem in Markov chain-based methods, which always biases to short-distance items while missing the information from the long-distance items in sessions. (3) *Attention-based SBR.* The attention mechanism is applied to further improve

SBR [13, 15] by identifying important items within sessions. NARM [7] first integrated the attention mechanism into SBR to extract the the user's main purpose in the current session. STAMP [8] proposed a short-term memory priority model based on multi-layer perceptron and attention mechanism, which captured the long-term and short-term interests of users. However, the attention mechanism only focuses on few important items that belong to the user's main purpose, while neglecting the other purposes indicated by few inferior items. (4) *GNN-based SBR.* Due to the superiority of GNN on modeling transition dependencies between items, GNN-based methods have been applied widely to SBR. SR-GNN [19] modeled all sessions via directed graphs, utilized GNN to capture the dependencies between items with sessions, and extracted the long-term and short-term interest of users to suggest the next item. FGNN [9] proposed to capture the sequence order and latent order in session graph, which devised the weighted attention graph layer to learn item embeddings and session embeddings for more accurate next item recommendation. GCE-GNN [18] proposed to learn the transitions between items from local and global perspectives simultaneously, so as to make better recommendations by leveraging the information from other sessions. Although GNN-based methods have achieved great success on SBR, all of these approaches construct the session graph according to the adjacent items, which ignore the dependencies from the non-adjacent items.

2.2 Review-based Recommendation

Considering the great value of user reviews on items, some works strive to model reviews to improve the performance of SR [6, 21]. DeepCoNN [21] employed two parallel cooperative neural networks to learn user behaviors by exploiting reviews written by the user and learn item properties from the reviews written for the item, then utilized factorization machine to predict item ratings. RNS [6] proposed a review-driven neural sequential recommendation, which learned user's long-term preference according to her historical reviews. Although the existing methods improve the recommendation performance, most of them are devised for the task of rating prediction instead of session-based recommendation. Although RNS is proposed for sequential recommendation, it requires to collect all reviews written by a user according to the user's explicit ID. This means that it is unable to work well on the anonymous session-based recommendation. Neither of the existing works really solve the problem of session-based recommendations based on review information.

3 Preliminary

3.1 Problem Statement

Let $V = \{v_1, v_2, \ldots, v_m\}$ represent the whole item set. Each anonymous session $s = [v_1, v_2, \ldots, v_n](v_i \in V)$ is an ordered list of items, where all the items in s are interacted by an anonymous user in a chronological order. We embed each

item into the same embedding space and let $\mathbf{h}_{v_i} \in \mathbb{R}^d$ denote the embedding of item v_i, where d is the dimensionality. To accurately identify item dependencies within the session, we utilize review information to enhance item representations. Given an item v_i, all of its reviews are collected to form the review document D_i, where each word is represented by the corresponding embedding with the dimension d_w. For the session-based recommendation problem, the goal is to predict the top-N items that the user is most likely to click in the next step.

3.2 Graph Construction

In this subsection, we first introduce the *adjaceny-driven inter-item graph* (AIG), which is widely adopted by existing GNN-based approaches [18, 19], and then we present a novel graph, i.e., *review-refined inter-item graph* (RIG). RIG is used as an additional graph to complement AIG rather than to replace it by enhancing the learning of true dependencies.

Adjaceny-driven inter-item graph (AIG). AIG captures important sequential patterns based on pair-wise adjacent items within the current session, which is first proposed by SR-GNN [19]. AIG converts each session s into a directed graph $\mathcal{G}_s^{adj} = (\mathcal{V}_s, \mathcal{E}_s^{adj})$, where $\mathcal{V}_s \subseteq V$ denotes the node set, $\mathcal{E}_s^{adj}$ denotes the edge set. The weight of each edge is set as the value of the occurrences of the edge divided by the outdegree of the edge's start node. This means that the more frequent occurrences of the edge, the stronger dependency between the items connected by it. The connection matrix $\mathbf{A}_s \in \mathbb{R}^{n \times 2n}$ describes how nodes are connected with each other in the graph, and $\mathbf{A}_{s,i:} \in \mathbb{R}^{1 \times 2n}$ are the two columns of blocks in $\mathbf{A}_s$, corresponding to node v_i. $\mathbf{A}_s = \mathbf{A}_s^{(\text{out})} \| \mathbf{A}_s^{(\text{in})}$, where $\mathbf{A}_s^{(\text{out})}$ and $\mathbf{A}_s^{(\text{in})}$ are the outgoing and incoming adjacency matrix respectively. $\|$ indicates the concatenation operation, and n is the length of session s.

Review-refined inter-item graph (RIG). The aforementioned AIG faces two deficiencies caused by the adjacent but independent items and the nonadjacent but dependent items. To address them, we utilize reviews to refine the session graph, and thus devise the review-refined inter-item graph (RIG).

We convert the session s into a directed graph $\mathcal{G}_s^{re} = (\mathcal{V}_s, \mathcal{E}_s^{re})$, where $\mathcal{V}_s \subseteq V$ indicates the node set, $\mathcal{E}_s^{re}$ denotes the edge set. As shown in Fig. 2, there are two steps for RIG to recognize the *cross-item dependencies* (i.e., the dependencies between non-adjacent items) to obtain the correct edges. *Firstly*, once AIG is generated, in order to filter out the noise edges and refine the session graph, we only reserve the edges between items sharing the same topic in the AIG, i.e., the same user purpose, and remove the other edges. For recognizing the

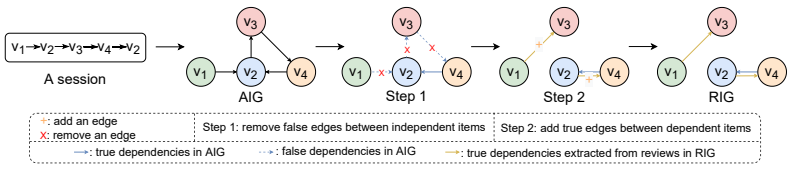


Fig. 2: Construction of AIG.

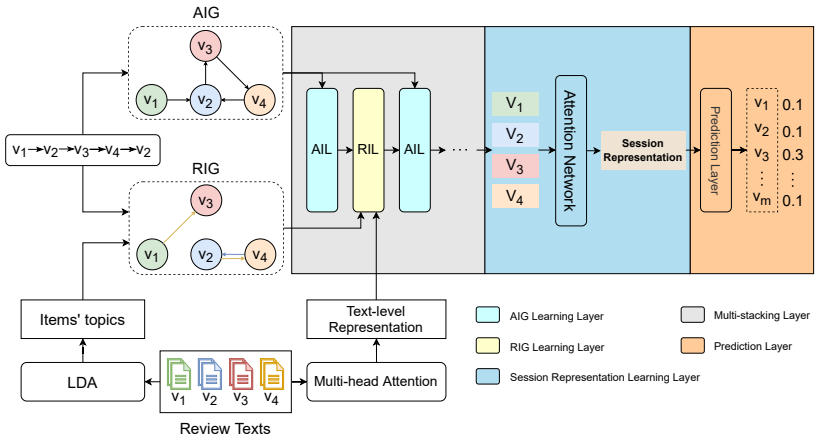


Fig. 3: Architecture of our proposed RI-GNN model.

topic information conveniently, we collect all reviews written for the item into the document set D together, and utilize LDA [1] to extract their topics. Once obtaining the topics of each item, we filter out the edges between items that do not belong to the same topic. *Secondly*, for each ordered pair of nodes (v_{t_1}, v_{t_2}) in the session sequence, we add the directed edge $(v_{t_1} \to v_{t_2})$ if item v_{t_1} shares the same topic with item v_{t_2} and $t_1 < t_2$. This makes the current item directly connect to all the following dependent items in the session for capturing the cross-item dependencies. Finally, we obtain the refined edge set $\mathcal{E}_s^{re}$. Similar to AIG, we calculate the connection matrix $\mathbf{B}_s \in \mathbb{R}^{n \times 2n}$ for RIG, where $\mathbf{B}_s = \mathbf{B}_s^{(\text{out})} \| \mathbf{B}_s^{(\text{in})}$.

4 Architecture of RI-GNN Model

The architecture of our proposed RI-GNN is shown in Fig. 3, which mainly consists of five components, i.e., *adjacency-driven inter-item graph (AIG) learning layer*, *review-refined inter-item graph (RIG) learning layer*, *multi-stacking layer*, *session representation learning layer* and *prediction layer*.

4.1 Adjaceny-driven Inter-item Graph Learning Layer (AIL)

AIL aims to capture the sequential dependencies between items based on AIG within the current session. Next, we will present how to learn the sequential dependencies between adjacent pair-wise items, as follows:

$$
\begin{aligned}
\mathbf{a}_{s,i}^t &= \mathbf{A}_{s,i:}^s \left[\mathbf{h}_{v_1}^{t-1}, \ldots, \mathbf{h}_{v_n}^{t-1}\right]^\top \mathbf{H} + \mathbf{b}_1, \\
\mathbf{z}_{s,i}^t &= \sigma\left(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{h}_{v_i}^{t-1}\right), \\
\mathbf{r}_{s,i}^t &= \sigma\left(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{h}_{v_i}^{t-1}\right), \\
\widetilde{\mathbf{h}}_{v_i}^t &= \tanh\left(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o \left(\mathbf{r}_{s,i}^t \odot \mathbf{h}_{v_i}^{t-1}\right)\right), \\
\mathbf{h}_{v_i}^t &= \left(1 - \mathbf{z}_{s,i}^t\right) \odot \mathbf{h}_{v_i}^{t-1} + \mathbf{z}_{s,i}^t \odot \widetilde{\mathbf{h}}_{v_i}^t,
\end{aligned}
\tag{1}
$$

where $\mathbf{a}_{s,i}^t \in \mathbb{R}^{2d}$ is the current state at time step t, which aggregates the adjacent items' embeddings for item v_i in AIG. $[\mathbf{h}_{v_1}^{t-1}, \ldots, \mathbf{h}_{v_n}^{t-1}]$ is the list of item embeddings in session s at previous time step $t-1$, $\mathbf{A}_{s,i:} \in \mathbb{R}^{1\times 2n}$ are the two columns of blocks in $\mathbf{A}_s$ corresponding to item v_i^s, $\mathbf{H} \in \mathbb{R}^{d\times 2d}$, $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_o \in \mathbb{R}^{d\times 2d}$, $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_o \in \mathbb{R}^{d\times d}$, $\mathbf{b_1} \in \mathbb{R}^{2d}$ are trainable parameters, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise multiplication operator, $\mathbf{z}_{s,i}^t \in \mathbb{R}^d$ and $\mathbf{r}_{s,i}^t \in \mathbb{R}^d$ are the update and reset gates respectively, $\mathbf{h}_{v_i}^t$ is the final state at the time step t. We mark the final representation of item v_i in AIL as $\mathbf{h}_{v_i}^{adj}$.

4.2 Review-refined Inter-item Graph Learning Layer (RIL)

In order to filter out the noise information and improve the representations of items, we next present how to propagate features on RIG to encode item dependencies from reviews. This layer is built based on the architecture of graph neural network, and we generate attention weights based on the similarity of reviews between items by exploiting the idea of graph attention network.

For each item's review document D_i obtained from *Section 3.2*, we first convert it into a representation vector $\mathbf{E}_i \in \mathbb{R}^{l\times d_w}$ through word embeddings. In order to better extract item features from the review representation $\mathbf{E}_i$, we utilize the self-attention method proposed by Transformer:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d}}\right)\mathbf{V}, \tag{2}$$

where $\mathbf{Q}$ is the queries, $\mathbf{K}$ is the keys, $\mathbf{V}$ is the values and $\sqrt{d}$ is the scale factor. We adopt multi-head attention to enable the model to jointly focus on information from different representation subspaces from different positions. For item v_i, the detailed operations are described as below:

$$\begin{aligned} \text{head}_k &= \text{Attention}\left(\mathbf{E}_i\mathbf{W}_k^{\mathbf{Q}}, \mathbf{E}_i\mathbf{W}_k^{\mathbf{K}}, \mathbf{E}_i\mathbf{W}_k^{\mathbf{V}}\right), \\ \mathbf{r}_i &= \text{MultiHead}(\mathbf{E_i}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\,\mathbf{W}_1, \end{aligned} \tag{3}$$

where $\mathbf{r}_i$ is the review representation of item v_i extracted by the multi-head attention, $\mathbf{W}^{\mathbf{Q}} \in \mathbb{R}^{d_w\times d_q}$, $\mathbf{W}^{\mathbf{K}} \in \mathbb{R}^{d_w\times d_k}$, $\mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{d_w\times d_v}$, and $\mathbf{W_1} \in \mathbb{R}^{hd_v\times d_w}$ are the learnable parameters. In our experiments, we set the number of parallel attention heads h to 3, set the dimensions of d_q, d_k, d_v, d_w to 100, 100, 100, 300.

In order to distinguish the importance of neighbor items for obtaining the representation of current item, we adopt attention mechanism and calculate the attention weight by cosine similarity:

$$\pi(v_i, v_j) = \begin{cases} \text{sim}(\mathbf{r}_i, \mathbf{r}_j), & \text{if } TP_i = TP_j \\ 0, & \text{if } TP_i \neq TP_j \end{cases}, \tag{4}$$

where $\pi(v_i, v_j)$ estimates the importance weight of different neighbor items, sim() is the cosine similarity function, $\mathbf{r}_i$ and $\mathbf{r}_j$ are multi-head review representation of item v_i and item v_j respectively, TP_i and TP_j are the topics of item v_i

and item v_j respectively. Next, we can obtain the final item representation by the linear combination of neighbor items:

$$\hat{\pi}(v_i, v_j) = \frac{\exp\left(\pi\left(v_i, v_j\right)\right)}{\sum_{v_k \in \mathcal{N}_{v_i}^{re}} \exp\left(\pi\left(v_i, v_k\right)\right)}, \quad \mathbf{h}_{v_i}^{re} = \sum_{v_j \in \mathcal{N}_{v_i}^{re}} \hat{\pi}\left(v_i, v_j\right) \mathbf{h}_{v_j}, \tag{5}$$

where $\hat{\pi}(v_i, v_j)$ is attention coefficient normalized by softmax, which means the different contribution of neighbor item v_j to the current item v_i. $\mathcal{N}_{v_i}^{re}$ is the neighbor set of item v_i in the *RIG*, $\mathbf{h}_{v_j}$ is the representation of the neighbor item v_j of item v_i, $\mathbf{h}_{v_i}^{re}$ is the final representation of item v_i in the RIL.

4.3 Multi-stacking Layer

In order to fully capture the deep dependencies between items, inspired by the work of Chen et al. [2], we stack multiple AIL and RIL layers, which can capture the complex dependencies (i.e., both adjacent-item and cross-item) within the session, described as below:

$$\mathbf{h}_{0,v_i}^{adj} \rightarrow \mathbf{h}_{1,v_i}^{re} \rightarrow \ldots \rightarrow \mathbf{h}_{l,v_i}^{*} \rightarrow \ldots \rightarrow \mathbf{h}_{k,v_i}^{*}, \tag{6}$$

where $\mathbf{h}_{l,v_i}^{*}$ denotes the representation of item v_i which is the output of layer l, $l \in (2, k)$, k is the hyper-parameter, and $*$ indicates either adj or re.

To fully utilize all features captured by all layers, we apply dense connections in our work. The input of each layer consists of the output representations of all previous layers. More specifically, the input of the l-th layer is $[\mathbf{h}_{0,v_i}^{adj} \| \mathbf{h}_{1,v_i}^{re} \| \cdots \| \mathbf{h}_{l-1,v_i}^{*}]$. This allows the higher layers to utilize not only the features through their previous layer, but also the low-level features at lower layers. For each item v_i, we obtain its representations $\mathbf{h}'_{v_i} \in \mathbb{R}^d$ by stacking multiple AIL and RIL.

4.4 Session Representation Learning Layer

Through the three layers mentioned above, given session $s = [v_1, v_2, \ldots, v_n]$, we can obtain all item representations in it, i.e., $\mathbf{H} = [\mathbf{h}'_{v_1}, \mathbf{h}'_{v_2}, \ldots, \mathbf{h}'_{v_n}]$. Then, we can generate session representation with the representations of items in it.

To reflect the different importance of different positions in the session sequence to the target item, we utilize a learnable position embedding matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]$, where $\mathbf{p}_i \in \mathbb{R}^d$ is a position vector for specific position i and n is the length of the session. We combine the position information with item representations through concatenation and non-linear transformation:

$$\mathrm{z}_i = \tanh\left(\mathbf{W}_2\left[\mathbf{h}'_{v_i} \| \mathbf{p}_{n-i+1}\right] + \mathbf{b}_2\right), \quad \mathbf{s}' = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}'_{v_i}, \tag{7}$$

where parameters $\mathbf{W}_2 \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_2 \in \mathbb{R}^d$ are trainable parameters, $\mathbf{s}'$ is the session information computed as the average of representations of items in the session.

Next, we adopt a soft-attention mechanism to learn the contribution of item v_i to the next prediction, and then we can obtain the session representation by linearly combining the item representations:

$$\beta_i = \mathbf{q}^\top \sigma\left(\mathbf{W}_3 \mathbf{z}_i + \mathbf{W}_4 \mathbf{s}' + \mathbf{b}_3\right), \quad \mathbf{s} = \sum_{i=1}^{l} \beta_i \mathbf{h}'_{v_i}, \tag{8}$$

where $\mathbf{W}_3, \mathbf{W}_4 \in \mathbb{R}^{d \times d}$ and $\mathbf{q}, \mathbf{b}_3 \in \mathbb{R}^d$ are learnable parameters.

4.5 Prediction Layer

We first utilize dot product and then apply softmax function to predict the click probability $\hat{\mathbf{y}}$ for the item v_i :

$$\hat{\mathbf{y}}_i = \text{Softmax}\left(\mathbf{s}^\top \mathbf{h}_{v_i}\right), \tag{9}$$

where $\hat{\mathbf{y}}_i \in \hat{\mathbf{y}}$ denotes the probability of item v_i to be the true next item. The loss function is defined as the cross-entropy of the prediction results.

5 Experiments and Analysis

5.1 Experimental Settings

Datasets. We select two datasets from the Amazon dataset[6] for our experiments: *Pet Supplies* and *Movies and TV*. The datasets contain purchase history of users and users' reviews for products. Following a common manner, we remove items appearing less than 5 times. Following [11], we split user's purchase behaviors into week-long sessions. To evaluate our method more comprehensively, we prepare two versions for each dataset. The first version (Case 1) keeps all sessions with more than 1 item [7, 18, 19] while the second version (Case 2) keeps sessions with more than 5 items [19]. Obviously, Case 2 is a subset of Case 1 which keeps long sessions only. We set the sessions of last year as the test data, and the remaining sessions for training. Then, we adopt sequence splitting preprocessing method which is commonly adopted in SBR. For an input session $[v_1^s, v_2^s, \ldots, v_n^s]$, we generate multiple input sequence-label pairs, i.e., $([v_1^s], v_2^s), ([v_1^s, v_2^s], v_3^s), \ldots, ([v_1^s, v_2^s, \ldots, v_{n-1}^s], v_n^s)$.

Evaluation Metrics and Baselines. Following [18, 19], we adopt P@K (Precision) and MRR@K (Mean Reciprocal Rank) as evaluation metrics. We compare RI-GNN with the following representative methods, including S-POP [5], S-KNN [5], GRU4Rec [4], NARM [7], STAMP [8], BERT4Rec [12], SR-GNN [19], GCE-GNN [18] and DHCN [20].

[6] https://nijianmo.github.io/amazon/index.html

Hyper-parameters Settings. Following previous studies [7, 18, 19], the dimension of node embedding is 100, the size of mini-batch is 100, and the L_2 regularization is 10^{-5} for all models. For RI-GNN, we use the Adam optimizer with the initial learning rate 0.001. The dropout ratio of session graph is 0.2. Moreover, the number of topics are empirically set to 24 and 20 on *Pet Supplies* and *Movies and TV* dataset, respectively.

Table 1: Experimental Results on Sessions with More than 1 Item.

Dataset	Pet Supplies				Movies and TV			
Metrics	P@10	P@20	MRR@10	MRR@20	P@10	P@20	MRR@10	MRR@20
S-POP	6.52	6.52	5.25	5.26	1.83	1.84	1.59	1.59
S-KNN	21.86	24.30	12.22	12.39	20.33	23.46	8.40	8.62
GRU4Rec	8.79	9.84	6.26	6.34	8.32	9.47	5.64	5.72
NARM	24.48	27.68	19.48	19.70	22.40	25.40	17.05	17.26
STAMP	21.50	24.66	16.41	16.60	20.59	23.96	14.69	14.92
BERT4Rec	24.18	27.40	19.29	19.51	23.21	26.46	17.75	17.97
SR-GNN	24.64	27.81	19.53	19.75	23.69	26.74	18.24	18.45
GCE-GNN	24.73	28.28	18.16	18.41	24.25	27.64	17.02	17.25
DHCN	24.23	27.45	17.80	18.02	23.33	26.49	15.94	16.16
RI-GNN	**25.43***	**28.88***	**19.93***	**20.15***	**24.69***	**27.94***	**18.93***	**19.14***
Improv.(%)	2.83	2.12	2.05	2.02	1.81	1.09	3.78	3.74

[1] The best results of each column are highlighted in boldface, the suboptimal one is underlined, the improvements are calculated by using the difference between the performance of our proposed RI-GNN and the best baseline, and * denotes the significant difference for t-test.

Table 2: Experimental Results on Sessions with More than 5 Items.

Dataset	Pet Supplies				Movies and TV			
Metrics	P@10	P@20	MRR@10	MRR@20	P@10	P@20	MRR@10	MRR@20
NARM	26.07	30.34	19.33	19.62	20.98	24.73	14.27	14.53
STAMP	24.44	28.06	18.28	18.53	21.27	25.40	14.49	14.77
BERT4Rec	25.85	30.04	19.49	19.77	22.24	26.43	15.87	16.15
SR-GNN	26.29	30.49	19.31	19.60	22.73	26.57	16.11	16.37
GCE-GNN	25.95	30.36	17.51	17.81	23.18	27.51	14.39	14.71
DHCN	24.96	29.20	17.25	17.54	21.83	25.71	13.01	13.27
RI-GNN	**27.72***	**32.11***	**20.26***	**20.57***	**24.15***	**28.30***	**16.99***	**17.27***
Improv.(%)	5.44	5.31	3.95	4.05	4.19	2.87	5.46	5.50

Overall Performance. The experimental results of overall performance on sessions of different lengths are reported in Table 1 and Table 2 respectively, according to the tables, we can draw the following conclusions:

Case 1 (Performance on sessions with more than 1 item): (1) Traditional methods (i.e., S-POP, S-KNN) show a significant inferiority to neural methods, except for GRU4Rec. This demonstrates that neural networks can learn more sophisticated features than the traditional methods. (2) Neural network based methods (i.e., GRU4Rec, NARM, STAMP, BERT4Rec) usually have better performance for SBR. GRU4Rec shows worse performance, which is probably because it strictly defines a session as a sequence. The other neural methods (i.e., NARM, STAMP, BERT4Rec) outperform GRU4Rec significantly. Among them, NARM combines RNN and attention mechanism, STAMP and BERT4Rec are completely based on attention mechanism. This result demonstrates that attention-based methods are also an effective way besides the RNN-based methods. (3) Among all the baseline methods, the GNN-based methods (i.e. SR-GNN,

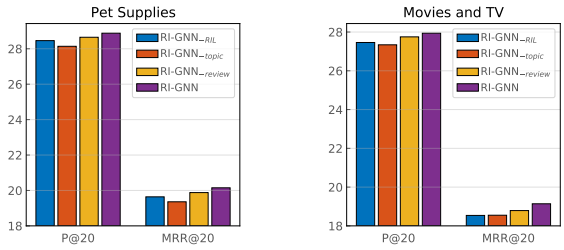


Fig. 4: Comparison of Ablation Variants.

GCE-GNN and DHCN) demonstrate the superiority over the others. This indicates that GNN-based models would be more effective than RNN-based and attention-based models when capturing the complex dependencies between items in SBR. (4) It is obvious that our proposed RI-GNN model outperforms all the baselines on all datasets. Compared with the GNN-based methods, the superiority of RI-GNN may due to that it is equipped with RIL and review information, which can model item dependencies more accurately.

Case 2 (Performance on sessions with more than 5 items): Table 2 shows that our proposed RI-GNN model can outperform the state-of-the-art SBR methods by 2.87% to 5.50% on different metrics. Compared with the result in Table 1, RI-GNN outperforms state-of-the-art SBR methods with a larger margin on long sessions. This demonstrates the superiority of RI-GNN is more outstanding for the long sessions. The reason may be that there exist more dependencies between non-adjacent items within the long sessions. This further verifies our hypothesis.

Ablation Study. To investigate the effectiveness of RIL component and review information for RI-GNN, we implement three variants of RI-GNN, denoted as RI-GNN$_{-RIL}$, RI-GNN$_{-topic}$, and RI-GNN$_{-review}$, by removing RIL, topic and review respectively. The comparison of them is shown in Fig. 4. It is clear that all variants are inferior to the standard RI-GNN, which demonstrates that RIL, topics and review information are critical and necessary for the success of RI-GNN model.

6 Conclusion

In this paper, we rethink adjacent dependencies in SBR, argue that adjacent items in a session are not always dependent and non-adjacent items are not necessarily independent. Accordingly, we propose a novel review-refined inter-item graph neural network (RI-GNN), which leverages reviews to reduce the false dependencies between adjacent but actually independent items and capture true dependencies between non-adjacent but dependent items.

Acknowledgment. Wenpeng Lu is the corresponding author. The research work is partly supported by National Natural Science Foundation of China under Grant No.11901325 and No.61502259, National Key R&D Program of China un-

der Grant No.2018YFC0831700, and Key Program of Science and Technology of Shandong Province under Grant No.2020CXGC010901 and No.2019JZZY020124.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**, 993–1022 (2003)
2. Chen, T., Wong, R.C.W.: Handling information loss of graph neural networks for session-based recommendation. In: SIGKDD. pp. 1172–1180 (2020)
3. Guo, W., Wang, S., Lu, W., et al.: Sequential dependency enhanced graph neural networks for session-based recommendations. In: DSAA. pp. 1–10 (2021)
4. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (2016)
5. Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: RecSys. pp. 306–310 (2017)
6. Li, C., Niu, X., Luo, X., et al.: A review-driven neural model for sequential recommendation. In: IJCAI. pp. 2866–2872 (2019)
7. Li, J., Ren, P., Chen, Z., et al.: Neural attentive session-based recommendation. In: CIKM. pp. 1419–1428 (2017)
8. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: STAMP: Short-term attention/memory priority model for session-based recommendation. In: KDD. pp. 1831–1839 (2018)
9. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: CIKM. pp. 579–588 (2019)
10. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: WWW. pp. 811–820 (2010)
11. Song, W., Xiao, Z., Wang, Y., et al.: Session-based social recommendation via dynamic graph attention networks. In: WSDM. pp. 555–563 (2019)
12. Sun, F., Liu, J., et al.: BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In: CIKM. pp. 1441–1450 (2019)
13. Wang, S., Cao, L., Hu, L., et al.: Hierarchical attentive transaction embedding with intra-and inter-transaction dependencies for next-item recommendation. IEEE Intelligent Systems **36**(04), 56–64 (2021)
14. Wang, S., Cao, L., Wang, Y., et al.: A survey on session-based recommender systems. ACM Computing Surveys **54**(7), 1–38 (2021)
15. Wang, S., Hu, L., Cao, L., et al.: Attention-based transactional context embedding for next-item recommendation. In: AAAI. pp. 2532–2539 (2018)
16. Wang, S., Hu, L., Wang, Y., et al.: Sequential recommender systems: challenges, progress and prospects. In: IJCAI. pp. 6332–6338 (2019)
17. Wang, S., Hu, L., Wang, Y., et al.: Graph learning based recommender systems: A review. In: IJCAI. pp. 4644–4652 (2021)
18. Wang, Z., Wei, W., Cong, G., et al.: Global context enhanced graph neural networks for session-based recommendation. In: SIGIR. pp. 169–178 (2020)
19. Wu, S., Tang, Y., Zhu, Y., et al.: Session-based recommendation with graph neural networks. In: AAAI. pp. 346–353 (2019)
20. Xia, X., Yin, H., Yu, J., et al.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: AAAI. pp. 4503–4511 (2021)
21. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: WSDM. pp. 425–434 (2017)