

Article

Attention-Enhanced Graph Neural Networks for Session-Based Recommendation

Baocheng Wang and Wentao Cai *

College of Information, North China University of Technology, Beijing 100144, China; wbaocheng@ncut.edu.cn

* Correspondence: 2018309040101@mail.ncut.edu.cn; Tel.: +86-155-7912-3964

Received: 12 August 2020; Accepted: 13 September 2020; Published: 18 September 2020



Abstract: Session-based recommendation, which aims to match user needs with rich resources based on anonymous sessions, nowadays plays a critical role in various online platforms (e.g., media streaming sites, search and e-commerce). Existing recommendation algorithms usually model a session as a sequence or a session graph to model transitions between items. Despite their effectiveness, we would argue that the performance of these methods is still flawed: (1) Using only fixed session item embedding without considering the diversity of users' interests and target items. (2) For user's long-term interest, the difficulty of capturing the different priorities for different items accurately. To tackle these defects, we propose a novel model which leverages both the target attentive network and self-attention network to improve the graph-neural-network (GNN)-based recommender. In our model, we first model user's interaction sequences as session graphs which serves as the input of the GNN, and each node vector involved in session graph can be obtained via the GNN. Next, target attentive network can activates different user interests corresponding to varied target items (i.e., the session embedding learned varies with different target items), which can reveal the relevance between users' interests and target items. At last, after applying the self-attention mechanism, the different priorities for different items can be captured to improve the precision of the long-term session representation. By using a hybrid of long-term and short-term session representation, we can capture users' comprehensive interests at multiple levels. Extensive experiments demonstrate the effectiveness of our algorithm on two real-world datasets for session-based recommendation.

Keywords: session-based recommendation; graph neural network; attention network

1. Introduction

In the scenario based on Internet e-commerce, streaming media and other recommendations, users interact with related items in a chronological order. Items that users will access in the near future are always based on items that they have interacted with in the past strongly. This property makes the practical application of the session-based recommendation method possible.

As there has been a revival of neural networks in recent years, session-based recommendation, an effective approach that matches user needs with rich resources based on anonymous sessions to alleviate information overload effectively, has attracted attention from industry and academia, which can help satisfy diverse service demands and reduce information overload. Due to the highly practical value of session-based recommendation methods, lots of methods for session-based recommendation have been proposed [1].

Previous work highlights that modeling only a session as a session sequence or a session graph to capture transitions between items in a session. Although effective, session-based recommendation is still in its infancy [2]. For example, as a classic example, Markov chain (MC) is based on an assumption that users' next behaviors are strongly based on previous ones [3]. However, an independent combination of the past interactions may limit the accuracy of recommendation. With the rapid

development of machine learning, artificial neural networks have made remarkable progress in session-based recommendation tasks. For example, Hidasi, Balázs et al. [4] propose a novel method to model interaction sequences via gated recurrent unit (GRU). In this method, the sequences of click behaviors are fed into the embedding layer as input data at first, and then into n-layer GRU. Finally, the probabilities for each item clicked next time are obtained via a full connection layer. Although effective to some extent, most of existing recurrent-neural-network-based (RNN) models have limited representation power in capturing complex transitions between items. Recently, Wu, Shu et al. [5] introduced the graph neural network into the session-based recommendation task. In this method, session sequences are constructed into a session graph at first, and the node vectors in the session graph are obtained via the graph neural network, and finally, the session representation for users is obtained and integrates long-term and short-term interest preference. Although this method expresses the complex transitions between items, it still ignores specific user interests related to a target item and accurate user long-term preference. With the increasing popularity of the deep neural network, attention network has been widely developed in many fields (e.g., recommender system, computer vision and natural language processing). Li et al. [6] firstly introduce standard vanilla attention mechanism into recommender system. Recently, Xu et al. [7] proposed a novel model based on self-attention networks to capture accurate long-term preference, which performs better than RNN-based recommendation models. More recently, Yu et al. [8] proposed a novel target attentive graph neural network to implement a session-based recommendation task which is unable to accurately capture users' long-term preferences.

To overcome the limitations mentioned above, we argue that incorporating specific user interests related to a target item and accurately capturing the different priorities for different items can improve the user's session representation effectively. In this work, we propose a novel model using target attentive network and self-attention network to improve the graph-neural-network (GNN)-based recommender. At first, we construct users' interaction sequences into session graphs, which serves as the input of the graph neural network, thus, we can obtain item embeddings which correspond to nodes in the session graph. Secondly, user-specific interests which correspond to a target item in the current session are activated by a target attentive unit. Then, to better capture user's long-term interests, each session learns the different priorities of different items by leveraging the self-attention mechanism. Finally, we combine long and short-term preferences to express users' interests more comprehensively.

To summarize, the main contributions of our work are summarized as follows:

- To improve the representation of session sequences, we propose a novel model combining self-attention network with target attentive network, which can capture specific user interests related to a target item and the accurate priorities for different items.
- To model the representation of session sequences, we combine short-term user preferences with long-term user preferences based on attention-enhanced GNN-based recommender.

2. Related Work

2.1. Classical Recommendation Methods

Markov-chain-based models are widely used for session-based recommendation tasks, which can infer users' behavior in the near future after estimating the transition matrix. For example, Zimdars, A et al. [9] model the click sequences via using Markov chains which applying an instance expression method containing item information in interaction sequences, and the performance of this method is significantly better than the item-based method. Inspired by the merits of matrix factorization and Markov chains, Rendle, Steffen et al. [3] proposed a novel model, namely the factorizing personalized Markov chain model (FPMC), leveraging the combination of matrix factorization and Markov chains, which infers the probability of items and generates an item rating list to implement session-based recommendation tasks.

2.2. Deep-Neural-Network-Based Recommendation Methods

Recurrent neural network is a classic model that can capture sequence information, and widely develops in many fields (e.g., natural language processing and recommender system). For instance, Hidasi, Balázs et al. [4] first applied RNN to session-based recommendation tasks which attained remarkable achievements.

Recently, with the remarkable development of attention networks in the field of computer vision, researchers turn to apply attention networks to other fields, such as natural language processing and recommender systems. For instance, Li, Jing et al. [6] proposed a novel model incorporating attention network based on RNN to capture different weights for different items in interaction sequences, namely neural attentive recommender model (NARM). Liu, Qiao et al. [10] proposed an improved recommendation model based on attention mechanism which can effectively capture users' long and current-term preferences, namely short-term attention/memory priority model (STAMP).

Graph is a mathematical concept that contains a number of edges and nodes. More recently, graph neural network is a very popular model which denotes graph-structured data as input, widely develops in many fields (e.g., recommender system) [11]. In the session-based recommendation task, interaction sequences containing many items and orders between items correspond to nodes and edges in a graph. Thus, interaction sequences can be constructed as graph-structured data and the graph neural network can be used to implement session-based recommendation tasks. For example, Wu, Shu et al. [5] construct session sequences into session graphs, and obtain the node vectors in the session graph via graph neural network to implement a session-based recommendation task, which only capture transitions between items simply. Xu et al. [7] proposed a novel model fusing GNN-based recommender with self-attention networks, which can capture transitions between items, but ignores specific user interests related to a target item. Yu et al. [8] proposed a novel model using graph neural network and target-attentive network, incorporating the classic attention networks, which, though difficult, captures the different priorities accurately. Wang et al. [12] introduce the external knowledge base into the recommender system and combine the memory network with the graph neural network, but this model does not capture specific user interests related to a target item effectively, and the introduction of external knowledge reduces the efficiency of the model.

Compared with the above methods, our method integrates self-attention network and target attention network on the basis of the GNN-based model, which can activate different user interests corresponding to varied target items and capture the different priorities of the user's different interests accurately. Of course, our approach can also capture the complex transitions between items, similar to the classic GNN-based model.

3. Research Methodology

3.1. Preliminaries

In the session-based recommendation task, the unique items involved in interaction sequences are denoted as $V = \{v_i\}_{i=1}^m$ and users' interaction sequences can be ordered by timestamps (i.e., $s = [v_{s,i}]_{i=1}^{s_n}$), where an interaction event involved in the interaction sequence s is denoted as $v_{s,i} \in V$. Based on the above symbols and descriptions, the problem to be solved for session-based recommendation is: Given a historical interaction sequence s , to predict user's interaction event v_{s,s_n+1} in the near future. The model architecture of our approach is shown in Figure 1.

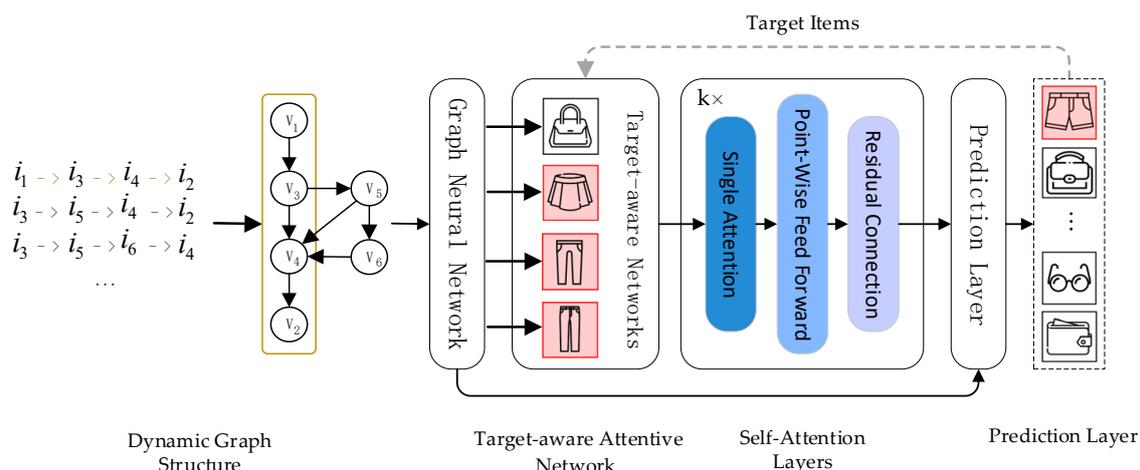


Figure 1. The model architecture of our approach. We first model user’s interaction sequences as session graphs which serves as the input of the graph neural network, and we can obtain each node vector involved in session graph via graph neural network. Next, target attentive network activates different user interests corresponding to varied target items adaptively (i.e., the learned interest representation vector varies with different target items). Then, we leverage the self-attention mechanism to accurately capture users’ long-term interests. Finally, we combine long-term and short-term interests to infer the probabilities for each candidate item for recommendation task.

3.2. Dynamic Graph Structure

Traditional methods the using neural network model have achieved great success for Euclidean data. Although it works to a certain extent, it learns only simple sequence relations from interaction sequences, but ignores complex transitions (i.e., graph-structured data contains a large number of complex transitions). Recently, many researchers turn to employ neural-network-based method to graph-based research. For example, the graph neural network was proposed in the literature [13] in 2009, which is a novel neural-network-based model taking graph-structured data as input.

In this part, we firstly construct the session sequence as session graphs, and feed session graphs into the graph neural network. Finally, we can obtain node vectors involved in session graphs via the process of graph neural network.

3.2.1. Construction for Session Graphs and Session Matrix

Each session sequence s can be constructed as a session graph $\zeta_s = (v_s, \varepsilon_s)$ in which each node $v_{s,i} \in v_s$ corresponds to an item in interaction sequences and each edge $(v_{s,i-1}, v_{s,i}) \in \varepsilon_s$ corresponds to user’s two successive clicks in sequence s . Since edges in session graphs may be repeated, we have given each edge a normalized weight which is calculated as the occurrence of the edge divided by the outdegree of that edge’s start node. A_s is defined as the combination matrix of matrices $A_s^{(out)}$ and $A_s^{(in)}$, which represents the weights of the outgoing and incoming edges in the session graph respectively [5]. For example, for a sequence $s = [v_1, v_2, v_3, v_1, v_4]$, the corresponding session graph and session matrix are shown as Figure 2:

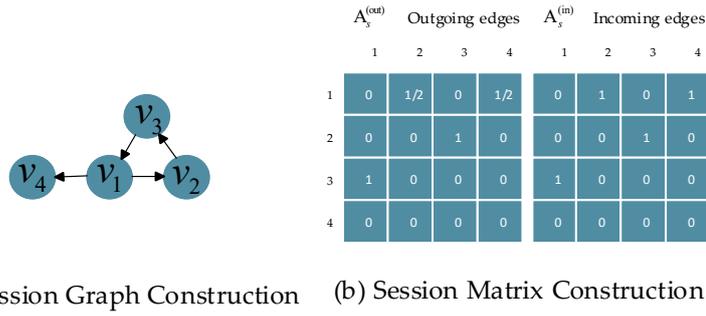


Figure 2. An example of constructing a session graph and session matrix. (a) shows the construction of the session graph, and (b) shows the construction of the session matrix.

3.2.2. Learning Node Vectors on Session Graphs

Next, we introduce how to obtain node vectors using a graph neural network. Session graphs constructed are used as the input data to the graph neural network. As the GNN can automatically extract the rich node transition relationships in the session graphs, it is suitable for the sequential recommendation task [5]. The learning process of node vectors in the session graph is given as follows:

$$\mathbf{a}_{s,i}^t = \mathbf{A}_{s,i} \cdot [\mathbf{v}_1^{t-1}, \dots, \mathbf{v}_n^{t-1}]^T \mathbf{H} + \mathbf{b} \tag{1}$$

$$\mathbf{z}_{s,i}^t = \sigma(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}) \tag{2}$$

$$\mathbf{r}_{s,i}^t = \sigma(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}) \tag{3}$$

$$\tilde{\mathbf{v}}_i^t = \tanh(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o (\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1})) \tag{4}$$

$$\mathbf{v}_i^t = (1 - \mathbf{z}_{s,i}^t) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \tilde{\mathbf{v}}_i^t \tag{5}$$

where $\mathbf{v}_i \in \mathbb{R}^d$ is node vector to $\mathbf{v}_{s,i}$, $\mathbf{A}_{s,i} \in \mathbb{R}^{1 \times 2n}$ are the two columns of blocks in \mathbf{A}_s corresponding to node $\mathbf{v}_{s,i}$, $\mathbf{H} \in \mathbb{R}^{d \times 2d}$ is the weight, $[\mathbf{v}_1^{t-1}, \dots, \mathbf{v}_n^{t-1}]$ is the list of item vectors in session s , \mathbf{b} denotes bias vector, $\mathbf{a}_{s,i}^t$ represents the information interaction between different nodes based on neighborhoods. $\mathbf{W}_s, \mathbf{W}_r, \mathbf{W}_o, \mathbf{U}_s, \mathbf{U}_r, \mathbf{U}_o$ are learnable parameters, $\sigma(\cdot)$ represents the sigmoid function, $\mathbf{z}_{s,i}, \mathbf{r}_{s,i}$ is update and reset gate respectively. \odot represents the element-wise multiplication operator, $\tilde{\mathbf{v}}_i^t$ denotes the process for candidate state construction based on reset gate, current state and previous state. \mathbf{v}_i^t denotes the process of final state construction based on candidate state, update and previous state. After that, we can obtain all latent item embeddings involved in session graphs [5,14].

3.3. Target-Aware Attentive Network Construction

Traditional recommendation methods use only intra-session item representations to obtain users' interest and ignore the relevance of historical behaviors concerning target items. To alleviate this defect, we construct a target-aware attentive network module. Once we captured the node embeddings in session graph, we proceed to activate user-specific interests with respect to a target item, via employing a target-aware attentive network [8].

After that, we further exploit the process of calculating attention scores between all items v_i in session s and each target item $v_i \in V$ via a local target attentive module. We firstly apply a shared non-linear transformation to every node-target pair, which is parameterized by a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$. After that, we use the softmax function to normalize the self-attention scores.

$$\beta_{i,t} = \text{softmax}(e_{i,t}) = \frac{\exp(v_t^T \mathbf{W} v_i)}{\sum_{j=1}^m \exp(v_t^T \mathbf{W} v_j)} \tag{6}$$

At last, we represent the users' interests towards a target item v_i as $s_{\text{target}}^t \in \mathbb{R}^d$, which can be defined as:

$$s_{\text{target}}^t = \sum_{i=1}^{S_n} \beta_{i,t} v_i \quad (7)$$

With different target items, the obtained target embedding for representing users' interests varies respectively [8].

3.4. Self-Attention Layers Construction

To capture global dependencies between input and output, and item-item transitions across the entire input and output sequence itself without regard for their distances, we employ self-attention, which is a unique network of attention and develop rapidly in many fields (e.g., natural language processing [15] and question answering system [16]).

3.4.1. Self-Attention Layer

After the processing of the graph neural network and target-aware attentive network, we are able to obtain the latent embedding for all nodes in the session graph (i.e., $\mathbf{H} = [\beta_1 v_1, \beta_2 v_2, \dots, \beta_n v_n]$). Then, we define them as inputs to the self-attention layer to capture users' global interests more accurately [7].

$$\mathbf{E} = \text{softmax}\left(\frac{(\mathbf{H}\mathbf{W}^Q)(\mathbf{H}\mathbf{W}^K)^T}{\sqrt{d}}\right)(\mathbf{H}\mathbf{W}^V) \quad (8)$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{2d \times d}$ denote the projection matrices.

3.4.2. Point-Wise Feed-Forward Network

We further endow the model with nonlinearity, and consider interactions between different latent dimensions via apply two linear transformations with a ReLU activation function. However, self-attention operations may cause transmission loss, thus, we add a residual connection after the feed-forward network, which is very easy to use low-layer information inspired by:

$$\mathbf{F} = \text{ReLU}(\mathbf{E}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 + \mathbf{E} \quad (9)$$

where \mathbf{b}_1 and \mathbf{b}_2 denote d -dimensional vectors, \mathbf{W}_1 and \mathbf{W}_2 denote $d \times d$ matrices. Specifically, to alleviate overfitting problems, we employ dropout regularization during training [7]. To simplify the above method, we define the above whole self-attention mechanism as:

$$\mathbf{F} = \text{SAN}(\mathbf{H}) \quad (10)$$

3.4.3. Multi-layer Self-Attention

In recent years, many studies have shown that different layers capture different types of features. Inspired by this phenomenon, we further explore the influence of levels of layers in features modeling. We define 1-st layer as $\mathbf{F}^{(1)} = \mathbf{F}$, and the k -th ($k > 1$) self-attention layer can be defined as:

$$\mathbf{F}^{(k)} = \text{SAN}(\mathbf{F}^{(k-1)}) \quad (11)$$

where $\mathbf{F}^{(k)} \in \mathbb{R}^{n \times d}$ denotes the final output of the multi-layer self-attention network [7].

3.5. Hybrid Session Embeddings Construction

To better predict the users' next clicks, we plan to combine long-term preference and short-term preference, which is denoted as the final session representation. We take the last dimensions of $\mathbf{F}^{(k)}$

generated from self-attention layers as the global embedding [17], and the last clicked-item vector in a session $S = \{s_1, s_2, \dots, s_n\}$ as the local embedding.

$$\mathbf{S}_f = \omega \mathbf{F}_n^{(k)} + (1 - \omega) \mathbf{h}_n \quad (12)$$

where $\mathbf{F}_n^{(k)} \in \mathbf{R}^d$ denotes n-th row of the matrix.

3.6. Prediction Layer

After that, we predict the user's behavior in the near future given session embedding, as shown below:

$$\hat{y}_i = \text{softmax}(\mathbf{S}_f^T \mathbf{v}_i) \quad (13)$$

where \hat{y}_i represents the possibilities for each item v_i . Finally, the objective function for training our model is given as the following:

$$J = - \sum_{i=1}^n y_i \log(\hat{y}_i) + (1 - \hat{y}_i) \log(1 - \hat{y}_i) + \lambda \|\theta\|^2 \quad (14)$$

where θ denotes the set of all learnable parameters y represents the one-hot encoding vector of the ground truth item.

4. Experiments

In this section, we first introduce an experimental setup containing four parts, namely datasets, baseline algorithms, parameter setting and evaluation metrics. Next, we compare the proposed model with baseline algorithms. Finally, we make an ablation analysis of our model.

4.1. Experimental Setup

4.1.1. Datasets

In this work, we use two public real-world datasets, including Diginetica and Yoochoose, to evaluate our model comprehensively. To make a fair and effective comparison, following [5,6,10,18], we preprocessed the datasets. To be exact, active items appearing more than five times and sessions consisting of more than two items are retained. Sessions of last days and sessions of last weeks are denoted as test set for Yoochoose and Diginetica, respectively.

4.1.2. Baseline Algorithms

In order to verify the effectiveness of the proposed model, we compare it with following methods. The baseline method is described below:

- (1) POP is a frequency-based recommendation model.
- (2) Item-k-nearest neighbor (Item-KNN) [19] is a similarity-based recommendation model.
- (3) Factorization-based methods Bayesian personalized ranking (BPR-MF) [20] is factorization-based methods Bayesian personalized ranking.
- (4) FPMC [3] is an effective session-based recommendation based on factorizing personalized Markov chain.
- (5) Gated recurrent unit for recommendation (GRU4REC) [4] is an RNN-based recommendation model for the session-based recommendation.
- (6) STAMP [10] is a short-term attention/memory priority model for current interests.
- (7) Session-based recommendation with graph neural networks (SR-GNN) [5] is a GNN-based model capturing users' long-term interests and current interests by using graph neural network.

- (8) Target attentive graph neural networks (TAGNN) [8] are a GNN-based model introducing target attentive units for session-based recommendation tasks.

4.1.3. Parameter Setting

In this work, the hidden dimensionality is set to 100 all experiments, and other hyperparameters are tuned based on a random 10% validation set. We set the initial learning rate for Adam to 0.001 which decayed by 0.1 after every 3 epochs, batch size to 100 for both Diginetica and Yoochoose [5,10].

4.1.4. Evaluation Metrics

We use two evaluation metrics (i.e., Precision@20 (P@20) and MRR@20) to evaluate our method with the baseline algorithms, which can describe the proportion of successful recommendations in an unranked list, and the position of successful items recommended in a ranked list, respectively.

4.2. Results and Analysis

The experimental results of our model and algorithms are shown as Table 1:

Table 1. The experimental results of our model and baseline algorithms.

Methods	Diginetica		Yoochoose 1/64	
	P@20	MRR@20	P@20	MRR@20
POP	0.87	0.19	6.74	1.63
Item-KNN	35.69	11.54	51.59	21.79
BPR-MF	5.21	1.89	31.28	12.07
FPMC	26.52	6.94	45.61	15.01
GRU4REC	29.40	8.31	60.67	22.90
STAMP	45.59	14.29	68.69	29.64
SR-GNN	50.72	17.56	70.56	30.92
TAGNN	51.29	17.93	70.98	31.03
OUR MODEL	52.02	18.58	71.45	31.29

4.2.1. Observations about Our Model

First, our model achieves almost the best performances on the two metrics, which revealed the superiority of our model (i.e., the combination of target attentive network and self-attention network). Second, our model performs better than TAGNN. The major reason may be that it ignores the accuracy of user's long-term preferences. Third, our model outperforms SR-GNN. Although SR-GNN employs graph neural networks to obtain the complex transitions, it fails to obtain the variabilities between session item vectors with different target items. Instead, our model effectively captures those variabilities via a target attentive network. Fourth, our model achieves better results compared with the STAMP and GRU4REC. One possible reason is that STAMP and GRU4REC are both based on the recurrent neural networks, which are unable to capture complex transitions in historical sessions. Fifth, our model is superior to FPMC, BPR-MF, Item-KNN and POP. One key reason is that these methods do not incorporate the advanced deep neural network.

4.2.2. Other Observations

First, TAGNN achieves better experimental results than SR-GNN on the two metrics. The main reason is that TAGNN combine target attentive network with GNN. Second, SR-GNN outperforms STAMP and GRU4REC in both of these datasets, which confirms the superiority of the graph neural network (i.e., capturing complex transitions). Third, STAMP has better experimental performance than GRU4REC, which verifies the superiority of the short-term attention/memory priority method. Fourth, FPMC performs better than BPR-MF, Item-KNN and BPR-MF, which reveals the advantages of combining matrix factorization and Markov chain. Fifth, all non-artificial-neural-network-based

methods lag behind those methods based on the artificial neural network. This shows that the artificial-neural-network-based method is a revolutionary machine learning method compared with the classical machine learning methods.

4.3. Ablation Analysis of Our Model

The above encompasses the main result comparisons on two evaluation metrics for our model and baselines based on two datasets. Next, to better understand the architecture of our model, we further analyze our model in detail.

4.3.1. Impact of the Number of Self-attention Blocks

Since different levels of self-attention layers capture different types of features, we explore the relationship between levels of self-attention layers with performance in the experiments. As is shown in Figure 3, the blocks in self-attention layer are varied from 1 to 6. The performance of our model boosts as increasing k before a proper value, and then gets worse after this value, which illustrates that our model will have difficulty in capturing low-layer features using more blocks effectively.

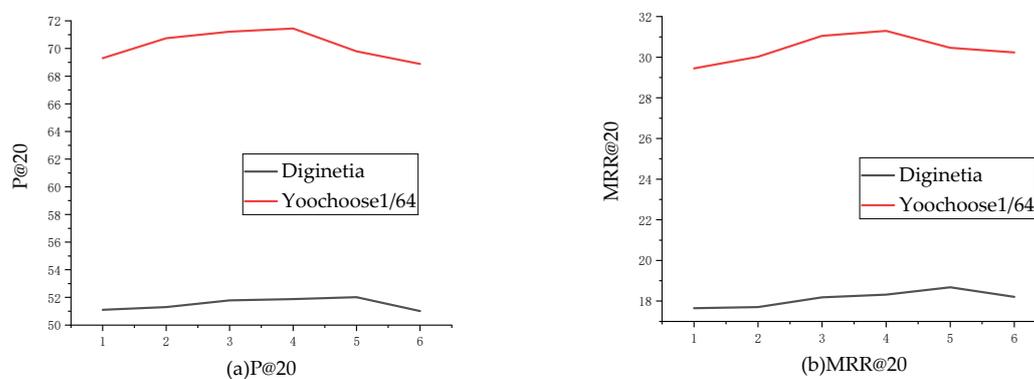


Figure 3. Performance under different number of blocks. (a) shows the experimental results on P@20, and (b) shows the experimental results on MRR@20.

4.3.2. Impact of Varying Session Representations

After that, we further discuss the impact of different session embedding scheme to the performance of our model. We use different schemes for session embedding to evaluate the relationships between session embedding with the performance in experiments, including (1) using attentive-based long-term session embedding only. (2) Using short-term session embedding only. The experimental results are shown in Figure 4; we can observe that our original scheme achieves the best results compared to other incomplete scheme variants, which confirms the superiority of using target-aware attention. The scheme (2), which only using short-term session embedding, performs effectively, which verifies the power of the last item in the experiments. Moreover, the performance of scheme (1) is better than scheme (2), which confirms that the effectiveness of attention-based global preference is greater than last-item-based local preference.

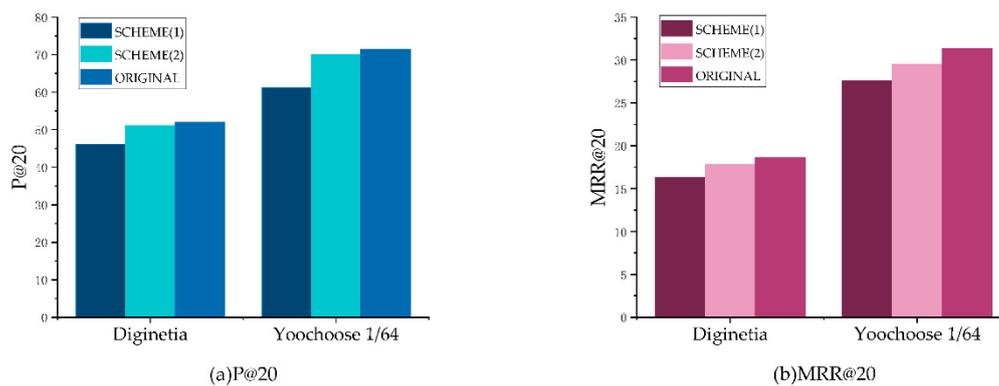


Figure 4. Performance under different session embedding scheme. (a) shows the experimental results on P@20, and (b) shows the experimental results on MRR@20.

5. Conclusions and Future Work

Session-based recommendation is indispensable to practical application scenarios, such as e-commerce and entertainment industries. In this paper, we present a novel model incorporating graph neural network, self-attention network and target attention network for session-based recommendation task, which can capture the user-specific interests corresponding to a target item, and accurate users' long-term preferences. Of course, our model can also capture complex transitions between items. Extensive experimental analysis confirms the superiority of our model.

However, we only use a classic scheme to construct session sequences as session graphs, which does not depict the transitions between items well. On the other hand, we use only the user's behavior sequences and ignore the external knowledge (e.g., user social networks, items knowledge base), which can complement the reasons behind the user's behavior to a certain extent. Therefore, in the future work, we will focus on optimizing the construction scheme of the session graph and exploring the way to integrate external knowledge base.

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

Funding: This research was funded by The Fundamental Research Funds for Beijing Universities, grant number 110052971921/021.

Acknowledgments: The authors thank the editor, and the anonymous reviewers for their valuable suggestions that have significantly improved this study.

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

References

1. Hidasi, B.; Karatzoglou, A. Recurrent neural networks with top-k gains for session-based recommendations. In Proceedings of the Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 843–852.
2. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 20–27 April 2018; pp. 1835–1844.
3. 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, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
4. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.

5. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based Recommendation with Graph Neural Network. In Proceedings of the National Conference on Artificial Intelligence, Hilton Hawaiian Village, Honolulu, HI, USA, 27 January–1 February 2019; pp. 346–353.
6. 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.
7. 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 International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 3940–3946.
8. 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, Virtual Event, Xi'an, China, 25–30 July 2020; pp. 1921–1924.
9. Zimdars, A.; Chickering, D.M.; Meek, C. Using temporal data for making recommendations. In Proceedings of the Uncertainty In Artificial Intelligence, Seattle, WA, USA, 2–5 August 2001; pp. 580–588.
10. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: Short-term attention/memory priority model for session-based recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1831–1839.
11. Huang, J.; Zhao, W.X.; Dou, H.; Wen, J.; Chang, E.Y. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In Proceedings of the International ACM Sigir Conference on Research and Development In Information Retrieval, Ann Arbor Michigan, MI, USA, 8–12 July 2018; pp. 505–514.
12. Wang, B.; Cai, W. Knowledge-enhanced graph neural networks for sequential recommendation. *Information* **2020**, *11*, 388. [[CrossRef](#)]
13. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [[CrossRef](#)] [[PubMed](#)]
14. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R.J.A.P.A. Gated graph sequence neural networks. In Proceedings of the International Conference on Learning Representations Caribe Hilton, San Juan, Puerto Rico, 2–4 May 2016.
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 3–9 December 2017; pp. 5998–6008.
16. Li, X.; Song, J.; Gao, L.; Liu, X.; Huang, W.; Gan, C.; He, X. Beyond RNNs: Positional self-attention with co-attention for video question answering. In Proceedings of the National Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 8658–8665.
17. Kang, W.; Mcauley, J. Self-attentive sequential recommendation. In Proceedings of the International Conference on Data Mining, Singapore, 17–19 November 2018; pp. 197–206.
18. 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](#)]
19. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the Web Conference, Hong Kong, China, 1–5 May 2001; pp. 285–295.
20. Miller, A.H.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; Weston, J. Key-value memory networks for directly reading documents. In Proceedings of the EMNLP16, Austin, TX, USA, 1–5 November 2016.

