BILINGUAL PUBLISHING GROUP

Journal of Computer Science Research

https://journals.bilpubgroup.com/index.php/jcsr

ARTICLE

SGT: Session-based Recommendation with GRU and Transformer

Lingmei Wu, Liqiang Zhang*, Xing Zhang, Linli Jiang, Chunmei Wu

School of Mathematics & Computer Science, Guangxi Science & Technology Normal University, Laibin, Guangxi, 546199, China

ABSTRACT

Session-based recommendation aims to predict user preferences based on anonymous behavior sequences. Recent research on session-based recommendation systems has mainly focused on utilizing attention mechanisms on sequential patterns, which has achieved significant results. However, most existing studies only consider individual items in a session and do not extract information from continuous items, which can easily lead to the loss of information on item transition relationships. Therefore, this paper proposes a session-based recommendation algorithm (SGT) based on Gated Recurrent Unit (GRU) and Transformer, which captures user interests by learning continuous items in the current session and utilizes all item transitions on sessions in a more refined way. By combining short-term sessions and long-term behavior, user dynamic preferences are captured. Extensive experiments were conducted on three session-based recommendation datasets, and compared to the baseline methods, both the recall rate Recall@20 and the mean reciprocal rank MRR@20 of the SGT algorithm were improved, demonstrating the effectiveness of the SGT method.

Keywords: Recommender system; Gated recurrent unit; Transformer; Session-based recommendation; Graph neural networks

1. Introduction

Recommendation systems have become increasingly important in filtering and recommending potentially interesting items to target users, as well as promoting product marketing and generating significant commercial benefits, particularly in multimedia websites and e-commerce. Traditional recommenda-

*CORRESPONDING AUTHOR:
Liqiang Zhang, School of Mathematics & Computer Science, Guangxi Science & Technology Normal University, Laibin, Guangxi, 546199, China; Email: cute_2023@163.com

ARTICLE INFO
Received: 30 March 2023 | Revised: 5 April 2023 | Accepted: 11 April 2023 | Published Online: 20 April 2023
DOI: https://doi.org/10.30564/jcsr.v5i2.5610

CITATION
Wu, L.M., Zhang, L.Q., Zhang, X., et al., 2023. SGT: Session-based Recommendation with GRU and Transformer. Journal of Computer Science Research. 5(2): 37-51. DOI: https://doi.org/10.30564/jcsr.v5i2.5610

tion systems are based on users' historical interaction information, which is not always suitable for many applications.

For instance, in cases where a user browses a set of items without logging in, their identity might be anonymous, and within the current session, only the user's historical actions are available. In addition, traditional recommendation systems mostly focus on static settings, which are also unsuitable in real life, as user preferences are often dynamic. Therefore, short-term histories can capture more accurate user intentions. To address this issue, a recommendation system based on sessions has been proposed, which predicts the likelihood of the next item being clicked based on the current session's sequence [1].

The recommendation system can be broadly divided into feature-based recommendation [2], social recommendation [3,4], and sequential recommendation [5]. The feature-based method combines user information and product features to model the user-product interaction behavior and predicts the probability of users selecting products [2]. Although it is effective in learning the embedding vectors of products in the user-product interaction network, it requires significant computational resources to obtain user preferences. Social recommendation [3,6], which is based on social information [7] can alleviate data sparsity and cold-start problems,but these methods assume that users with all social links have similar preferences. As users' interests are dynamic and subject to change, they depend not only on user preferences but also on understanding the change of user interests [8].

Session-based recommendation is mainly relies on user behavior logs within a session to predict the next item of interest. Previous research on session recommendation has mainly focused on the sequential features of sessions. Based on Markov chain methods, sequential behavior data is used to predict the next behavior of users based on their prior behavior [9-11]. Recently, deep learning methods have been introduced to session-based recommendation scenarios. Recurrent Neural Networks (RNNs) have achieved significant results due to their exceptional sequence modeling ability [12-15]. However, RNN-based models often only simulate the transitions between continuously interacting items, ignoring the rich information between contexts. Graph Neural Network (GNN) methods convert session sequences into graph structures and utilize them as input to learn the complex transformation dependencies between item nodes to explore complex item transitions [16-20].

Despite the promising performance and potential of GNN-based methods in session-based recommendation, there are still limitations that need to be addressed, such as the challenges of effectively modeling long-range dependencies and the risk of over-smoothing [21]. Over-smoothing refers to the convergence of all node representations to a constant after a sufficient number of layers. Therefore, designing new architectures is crucial for addressing these issues.

In recent years, transformers have been shown to be successful in natural language understanding [22], computer vision [23], and biological sequence modeling [24]. They can capture the interaction information between nodes through self-attention layers, rather than just aggregating local neighbor information in the message passing mechanism.

However, most of the current approaches only consider a single item as the basic unit for extracting user preferences, ignoring the user intent implied by a set of contiguous and adjacent items. The user intent may change over time, and the items that have been clicked, saved, or purchased in the past may affect the subsequent items. Different numbers or levels of continuous items contain different user intentions, which can aid in providing multiple candidate recommendation items and accurate session intent information. In this paper, we extract user intent from both single items and combinations of contiguous items. Firstly, we use a gated graph neural network to model the session sequence and obtain aggregated embedding representations of the items in the session, followed by self-attention mechanism

to obtain the global embedding representation of the session, and finally, the recommendation decision is made.

2. Related work

The most basic approach based on Markov chains is to estimate the transition matrix heuristically by using the frequency of transitions in the training set. However, this method is not able to deal with unobserved transitions. For example, Rendle et al. [9] proposed the personalized Markov chain (FPMC) which combines matrix factorization with a first-order Markov chain to capture continuous user behavior and short-term interests. Wang et al. [25] proposed a hierarchical representation model (HRM) that improves FPMC with a hierarchical structure. Nevertheless, Markov chain-based methods typically cannot capture more complex higher-order sequence patterns. As considering more previous items quickly makes the state size difficult to manage, most Markov chain-based models only use first-order transitions to construct the transition matrix, resulting in their inability to capture more complex higher-order sequential patterns.

With the great success of deep learning in various fields, more and more neural network-based methods have been applied to session-based recommendation tasks. Hidasi et al. [12] modeled session data using multiple gated recurrent units (GRUs) [26] layers. Tan et al. [14] then further improved its performance by using data augmentation, pre-training, and privileged information. Li et al. [13] proposed the NARM model, combines the attention mechanism with GRU to encode the user's behavioral sequence and emphasize its main intention in the current session. Liu et al. [27] created STAMP, a short-term memory priority model based on multi-layer perceptron and attention mechanism, which effectively captures users' global preferences and local interests. Wu et al. [28] converted contextual information into low-dimensional real vector features, and subsequently integrated them into a session-based recursive neural network recommendation model using three merging methods: Add, Stack, and Multilayer Perceptron.

Given the impressive results of deep learning in various domains, an increasing number of neural network-based techniques have been applied to session-based recommendation tasks. However, Wu et al. [16] argued that RNN-based models can only simulate one-way transitions between adjacent items, failing to capture context transitions between entire session sequences. They proposed an SR-GNN model that introduced graph neural networks to achieve stronger performance. In addition, Xu et al. [17] combined GNNs and self-attention networks (SANs) to capture long-range dependencies within sessions. Qiu et al. [18] used a weighted graph attention network to obtain item representations and then used a readout function to generate recommended session representations. Yu et al. [29] proposed a novel target attention graph neural network that considers candidate items when generating session representations. As most of the aforementioned works rely only on anonymous sessions with a lack of user long-term profiles, Zhang et al. [30] proposed a user behavior graph construction method based on long-term and short-term user interactions. Chen et al. [19] proposed a LESSR model to tackle the problem of inadequate long-term dependency capture and lossy session encoding in prior GNN-based approaches.

Most of the above approaches mainly focus on the item transition information within the ongoing session or directly employ all sessions to construct the model, neglecting the influence of the sequential items at distinct levels on the recommendation performance.

3. The proposed method

In this section, we first introduce the formal definition of the general session-based recommendation problem (Section 3.1). Then we explain the session graph construction (Section 3.2). Afterwards, we elaborate the proposed model i.e. session representation layer (Section 3.3), and prediction layer (Section 3.4).

In this paper we propose the SGT model, which

utilizes both GRU and Transformer, for session based recommendation. Which consists of four main components: Input layer, embedding layer, session representation layer, and prediction layer. The structure of the SGT model is shown in **Figure 1**, and the structure of the Transformer layer is shown in **Figure 2**. In **Figure 1**, first construct a heterogeneous session graph of continuous intent units for the user. Then these input features are embedded into low-dimensional vectors. The Gated Recurrent Unit is applied to obtain all node vectors involved in the session graph. Next, the Transformer layer is used to capture the long-range dependencies between items in the session and assign different weights to the different items. The session representation layer integrates the user's long-term and short-term interests using a long and short interest gate fusion module. In the prediction layer, the score of each candidate item is calculated by multiplying its embedding with the session representation linearly transformed, and

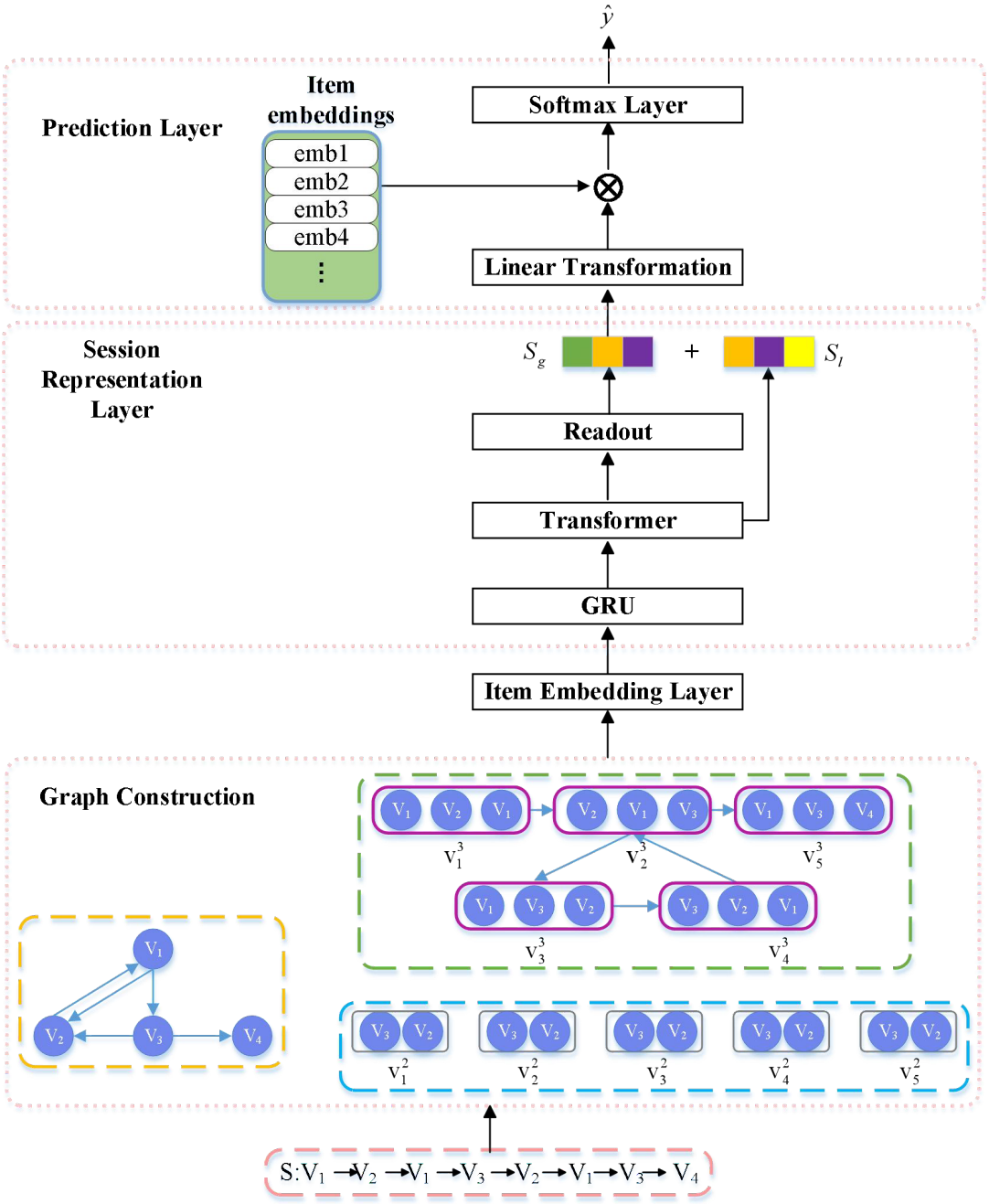

Figure 1. The overall framework of the proposed model.

recommends the top-ranked items.

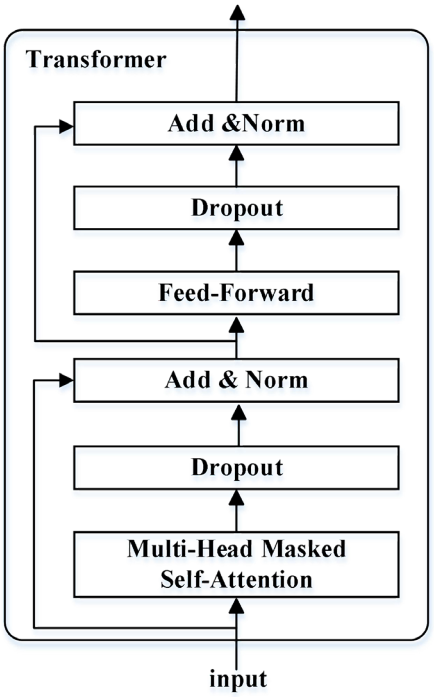

Figure 2. Transformer layer.

3.1 Problem definition

The task of session-based recommendation is to predict the user's next action based on their behavior within the current session. Here we present a formal definition of the session-based recommendation problem.

Let $S = \{S_i\}_{|S|}$ be a set of sessions and $V = \{v_1, v_2, ...v_m\}$ be a set of candidate items that appear in all sessions, m indicates the number of items. Each session, ordered by timestamps, can be represented as $S_i = \{v_1^s, v_2^s, ...v_l^s\}$, where l is the session length, v_i^s indicates the item that the user clicked at the location i in the session S_i.

The goal of session-based recommendation is to predict the next click, i.e. the sequence label, v_{l+1} for a session S_i. Given a session S_i, a session-based recommendation model outputs probabilities for all possible items, where the element value of the vector $\hat{y}$ represents the recommendation score of the corresponding item. The top-K values in the vector are considered as the recommended candidate items.

3.2 Session graph construction

The current session-based recommendation mainly focuses on individual items. This paper predicts users' interests by combining continuous sequences of different granularities to provide better recommendations. $v_j^k = (v_j, ...v_{j+k-1})$ is defined as a continuous segment with a length of k starting from j, $k$represents the granularity level of continuous projects. Taking the session $s = \{v_1, v_2, v_1, v_3, v_2, v_1, v_3, v_4\}$ as an example, $v_1^1, ..., v_4^1$ represent the first-level continuous intent unit, $(v_1, v_2), ..., (v_3, v_4)$ represent the second-level continuous intent unit, denoted as $v_1^2, ...v_5^2$, $(v_1, v_2, v_1), ..., (v_1, v_3, v_4)$ represent the third-level continuous intent unit, denoted as $v_1^3, ...v_5^3$.

Each input item $v_i \in S$ is transformed into a dense vector $e_i \in \mathrm{R}^d$ through the embedding layer, which allows them to be directly inputted into the deep neural network. d is dimension of the representation e_i. For 1-level continuous items, they are initialized to generate learnable embedding vectors e_j^1. For higher-level continuous items, the initialization uses GRU to extract sequence-sensitive intents. The initialization of k-level continuous items is represented as e_j^k, which can be defined as:

$$e_j^k = \delta(\{e_j^1, \cdots, e_{j+k-1}^1\}) \tag{1}$$

$$e_j^k = e_j^{k,set} + e_j^{k,seq} \tag{2}$$

The session graph is mainly composed of multiple subgraphs of different granularities, using different levels of subgraphs to capture the relationships between items. For example, for a session $s = \{v_1, v_2, v_1, v_3, v_2, v_1, v_3, v_4\}$, 5 groups of level-2 continuous intent units can be constructed to model the relationships between items at the level-2. In order to combine subgraphs of different granularities into a complete session graph, special edges are used to connect the high-order session graph and the first-order session graph. Intra-granular edges $(v^k, \text{intra}-k, v^k)$ are used for items at the same level, while inter-granular edges (v^1, intra, v^k) and (v^k, intra, v^1) are used for higher-order and first-order items.

3.3 Session representation layer

Gated recurrent unit layer

Capturing sequence information is a critical aspect of session-based recommendation. Wu et al. [7] have demonstrated that RNN is effective in this regard. Among RNN variants, GRU mitigate the vanishing gradient problem that plagues RNNs, and has fewer parameters and faster training speed than LSTM, another variant of RNN. Hence, in this paper, we employ GRU to capture sequence information.

We use GRU to model item embedding representations, and update the current node's feature representation as follows:

$$r_t = \sigma(W_r e_t + U_r h_{t-1}) \tag{3}$$

The update gate determines whether to update the hidden state, and the formula is as follows:

$$z_t = \sigma(W_z e_t + U_z h_{t-1}) \tag{4}$$

The calculation formula for the hidden state $\hat{h}_t$ based on the reset gate is as follows:

$$\hat{h}_t = \tanh[W_h e_t + U_h (r_t \odot h_{t-1})] \tag{5}$$

The equation for updating the hidden state using the update gate can be expressed as:

$$h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t \tag{6}$$

In the above equations, r_t and z_t represent reset and update gates, respectively. $\sigma(\cdot)$ represents the sigmoid function. e_t represents the input at time step t. h_{t-1} represents the previous hidden state of the GRU. These two parts explore the correlation between e_t and the current state of the GRU. W_r, W_z, W_h, U_r, U_z, U_h are parameter matrices and $\odot$ represents matrix dot products. When all nodes in the graph are updated and converge, the final representation of each node can be obtained.

Then using the last hidden state of the GRU layer represents the sequential behavior of the user in the current session.

$$c_t^g = h_t \tag{7}$$

Transformer layer

The core of the transformer model lies in the design of the self-attention layer, which uses multi-head attention to map the sequence to different semantic subspaces and internally extract sequence features for each subspace. This ultimately completes the feature extraction of the original sequence information, as shown in **Figure 2**.

Users often have multiple interests,and a single attention network may not be enough to capture all the relevant information. For example, when browsing for a new smartphone, the user may consider aspects such as camera quality, battery life, and screen size. Multi-head attention is a technique that allows the model to attend to multiple aspects of the input simultaneously, by constructing several parallel attention modules [22]. This technique can effectively capture the user's interests and preferences from their session click sequence, enabling better recommendations.

To predict the next item that a user may click in a session, it is necessary to model the user's interests from the user's session click sequence and capture the user's main intent. In this paper, multi-head attention is used to learn the representation of each continuous intent unit by constructing multiple parallel attention modules. It learns a deeper representation of each item by capturing its relationship with other items in the behavior sequence, thus improving the recommendation effectiveness of the model.

The multi-head attention layer aggregates the self-attention output vectors $\mathrm{H} = [h_1, ..., h_t]$ from the previous hidden outputs of the GRU. By constructing multiple parallel attention modules, the model can learn user interests from different semantic subspaces, thus modeling the user session sequence and learning a session feature vector that can express user intent. The calculation formula is as follows:

$$\mathrm{S} = \mathrm{MultiHead}(\mathrm{H}) = \mathrm{Concat}(head_1, head_2, ...head_h) W^O \tag{8}$$

Define $Q = \mathrm{H}W_i^Q, K = \mathrm{H}W_i^K, V = \mathrm{H}W_i^V$,

$$head_i = \mathrm{Attention}(Q\mathrm{W}_i^Q, K\mathrm{W}_i^K, V\mathrm{W}_i^V) \tag{9}$$

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{10}$$

Here, Q,K,V represent the query, key, and value matrices, respectively, and W_i^Q, W_i^K, W_i^V and W^O are learnable parameter matrices. h is the number of heads. The function $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ is the scaled dot-product attention with softmax activation. $\sqrt{d_k}$ is the dimension of the key vector. The multi-head attention layer produces an output by concatenating the outputs from all attention heads and applying a linear transformation with a weight matrix.

The self-attention mechanism is enhanced by the residual normalization layer and the point-wise feed-forward network (FFN). The former employs the idea of residual networks by adding the original input and output before normalization, thereby enhancing the memory capacity of the original sequence information. The latter performs a corresponding linear transformation on the output of the multi-head attention. The calculation can be expressed as:

$$\mathrm{F} = FFN(\mathrm{S}) = \max(0, \mathrm{S}W_1 + b_1)W_2 + b_2 \tag{11}$$

where W_1 and W_2 are parameter matrices, b_1 and b_2 are multi-dimensional bias vectors, and F is the output of multi-head attention. Different layers capture different types of features. After the first self-attention network module, it aggregates all the previous item embeddings. To further simulate the complex relationships behind the item sequence, self-attention network modules are stacked together. The $m(m>1)$ layer is defined as:

$$\mathbf{S}^m = SA(F^{(m-1)}) \tag{12}$$

$$\mathbf{F}^m = FFN(\mathrm{S}^m), \forall i \in 1,2\cdots,n \tag{13}$$

$\mathbf{F}^m \in R^{n\times d}$ is the final output of the multi-layer attention.

Generating session embedding vectors

For each level of continuous intent units, a local representation z_l^k is generated, as well as a global representation z_g^k to capture user preferences. As shown in **Figure 2**, given a session s_i and corresponding embeddings of continuous intent units $h_i^k \in \boldsymbol{R}^d, i = 1,...,n_k, k = 1,...,K$, n_k is the number of intent units per k level, and K is the number of levels of intent. The last intent unit $h_{n_k}^k$ is taken as the local representation z_l^k, and a soft attention mechanism is used to obtain the global representation z_g^k. The calculation is as follows:

$$z_g^k = \sum_{c=1}^{|C|} \text{Softmax}_c(\gamma_c^k) h_c \tag{14}$$

$$\gamma_c^k = W_0^{k^T} \sigma(W_1^k h_c + W_2^k z_l^k + b^k) \tag{15}$$

Aggregating all the embedded intent units to generate the context representation, i.e., $C = \{h_i^k | i = 1,...,n_k, k = 1,...,K\}$, where $h_c \in C$ as one of the context embeddings. $W_0^k \in \boldsymbol{R}^d, W_1^k \in \boldsymbol{R}^{d\times d}, W_2^k \in \boldsymbol{R}^{d\times d}$ are trainable parameters, $b^k \in \mathrm{R}^d$ is bias. $\sigma(\cdot)$ is sigmoid function. Finally, we compute the hybrid embedding z_s^k by taking transformation over the concatenation of the local and global embedding vectors:

$$z_s^k = W_3^k [z_g^k; z_l^k] \tag{16}$$

where $[;]$ is concatenation operation and matrix $W_3^k \in \mathrm{R}^{2d}$ compresses two combined embedding vectors into the latent space.

3.4 Prediction layer

After obtained the embedding of each session from different levels of granularity, we further calculate the recommendation score y_i^k for each candidate item over the whole item set V by multiplying their initial embeddings e_i, which can be defined as:

$$y_i^k = z_s^k e_i \tag{17}$$

Then, we apply a softmax function over y_i^k to transform it into probability distribution form $\hat{y}$:

$$\hat{y} = soft\max(y_i^k) \tag{18}$$

Finally, we select the K items with the highest recommendation scores based on $\hat{y}$ for top-K recommendation.

To optimize the model, backpropagation is used for neural network by minimizing the cross-entropy

loss between the predictions and the ground truth. The loss function is defined as follows:

$$L(\hat{y}) = -\sum_{i=1}^{|I|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (19)$$

where y represents the one-hot encoded vector of the ground truth item.

4. Experiments and analysis

In this section, we provide an overview of the experimental setup. Firstly, we introduce the Datasets, evaluation metrics and compared methods used in our experiments. Next, we compare the performance of our proposed SGT with other state-of-the-art methods. Finally, we conduct a comprehensive analysis of SGT under different experimental settings to provide insights into its effectiveness.

4.1 Datasets

We evaluate the effectiveness of our proposed method on three widely used real-world datasets, i.e. *Diginetica*[①]*,Gowalla*[②] *and Last.fm*[③].

- *Diginetica* is an anonymous user browsing and transaction record dataset provided in CIKM Cup 2016, which includes transaction logs and user browsing histories suitable for session-based recommendation.
- *Gowalla* is a check-in behavior dataset widely used for interest recommendation. In this experiment, the top 30,000 popular locations are retained, and the user's check-in records are grouped into unrelated time periods by splitting intervals exceeding 1 day between adjacent records. The last 20% of the sessions are used as the test set.
- *Last.fm* is a music dataset that includes a list of the user's most popular artists, album and track names as features, as well as timestamps and play counts, and user application tags that can be used to build content vectors. In this experiment, the top 40,000 popular artists are retained, and the interval is set to 8 hours for segmentation. The most recent 20% of sessions are used as the test set.

Following other works [13,16,19,26,28], we applied filtering to remove sessions with a length of 1 and items that appeared less than 5 times. Additionally, same as the studies [13,26], we utilized the data augmentation techniques to process the datasets. Furthermore, for session-based recommendation, we designated the sessions from the last week as the test data and used the remaining data for training. The resulting statistics of the datasets are presented in **Table 1**.

Table 1. Statistics of datasets used in the experiments.

Datasets	Diginetica	Gowalla	Last.fm
# of clicks	982961	1122788	3835706
# of training sessions	719470	675561	2837644
# of test sessions	60858	155332	672519
# of items	43097	29510	38615
#length<=5	537546	627100	1136909
#length>5	239483	203793	2373254
Average length	5.12	4.32	9.16

4.2 Evaluation metrics

To assess the recommendation performance of all models, we utilize the following two commonly used metrics.

Recall@*K* (Recall calculated over top-K items) is commonly used to measure predictive accuracy. It represents the proportion of correctly recommended items among the top-K items. It is calculated as:

$$\text{Recall@}K = \frac{n_{hit}}{N} \quad (20)$$

where n_{hit} represents the number of sessions with desired items in top-K recommended items and N denotes the number of test data. The Recall measure is order-insensitive in the recommendation list, where large Recall value indicates better recommendation performance of the model.

MRR@K (Mean Reciprocal Rank calculated over top-K items) is the average of reciprocal ranks

① http://cikm2016.cs.iupui.edu/cikm-cup
② https://snap.stanford.edu/data/loc-gowalla.html
③ http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html

of the correctly-recommended items. $\frac{1}{Rank(i)}$ is set to zero when the rank is large than K. The MRR measure considers the order of recommendation ranking and higher value indicates that correct recommendations are at the top of the ranking list. It is calculated as:

$$MRR@K = \frac{1}{Q}\sum_{i}^{|Q|}\frac{1}{Rank(i)} \tag{21}$$

$|Q|$ denotes the number of users, $Rank(i)$ represents the position of the first correct recommendation in the item list recommended by the model for the $i-\text{th}$ user.

In our experiment, we consider Top-K (K = 20) for recommendation.

4.3 Baselines

To evaluate the effectiveness of our proposed method, we compare it with the following representative baselines:

- **Item-KNN** [31] uses the nearest neighbor idea to recommend items similar to the last clicked item in the session.
- **FPMC** model [9] combines Markov chain and matrix factorization models, using a pairwise interaction model to perform matrix factorization on the personalized transition matrix of items, thus solving the Next Basket recommendation problem.
- **GRU4Rec** [12] is an RNN-based model that uses gated recurrent units (GRU) to model user sequences.
- **NARM** [13] uses GRU to extract sequence information and improves recommendation performance by adding attention mechanisms.
- **SR-GNN** [16] uses graph neural networks to model the order relationship between items, learns user interests in the session using attention mechanisms, and self-attends to the last item to predict the next item that the user is likely to click on.
- **GC-SAN** [17] is an improvement on SR-GNN, capturing local dependencies through graph neural networks and applying self-attention mechanisms to learn long-range dependencies.
- **LESSR** [19] introduces two session graphs to solve the problem of lost order information and long-term dependency.

4.4 Comparison with baseline methods

To evaluate the overall performance of the proposed model, we compare it with other state-of-art session-based recommendation methods. We randomly split 10% of the samples from the training set as the validation set, and the intention unit granularity level was set to {1, 2, 3, 4, 5, 6} to obtain the optimal value using the Adam optimizer. The batch size was set to 512, the embedding dimension was 256, and the number of heads in the multi-head attention was set to 2. The learning rate was set to 0.001, and the model's learning rate decayed to 0.1 times the previous value every 3 iterations. The overall performance in terms of Recall@20 and MRR@20 is shown in **Table 2**, with the best results highlighted in boldface.

The performance of the traditional Item-KNN and FPMC methods is relatively poor on the datasets used in the experiment, as these methods cannot well capture the complex temporal relationships between items in the session sequence.

All neural network algorithms have better performance in Recall@20 and MRR@20. The experimental results demonstrate the powerful ability of these algorithms to extract features, including sequence features. The NARM model uses attention mechanisms and the collaborative effect of user long-term and short-term features, which performs better than GRU4Rec in terms of indicators. This indicates that different items in a user session have different effects on user interests. The SR-GNN uses graph neural networks to model the complex dependency relationships between items and extracts node features using attention mechanisms, improving the recommendation performance. The GC-SAN improves upon SR-GNN by using self-attention networks to capture the global dependency relationships between items. The LESSR performs better than SR-GNN by solving

the problem of losing sequence information when converting session sequences into graph networks, indicating the effectiveness of retaining sequence information in sessions.

The proposed SGT model performs well on all datasets, which suggests that intent can be utilized at various granularity levels for modeling intricate transitions between user intents. The multi-head attention layer can effectively extract deep features of user sessions, capturing more comprehensive and precise user preferences, thereby predicting the next item that the user is likely to click on with greater accuracy.

4.5 Comparison with different connection schemes

In this section, we propose a set of comparative models to validate the effectiveness of incorporating last-click information into session context for session-based recommendations:

SGT-L: Local embedding only.

SGT-G: Global embedding with the attention mechanism.

The results of methods with two different embedding strategies are given in **Table 3**.

According to the table, it can be seen that SGT model with hybrid embedding method achieves the best results on all three datasets, indicating the significance of explicitly integrating current session interests with long-term preferences. For example, taking the Diginetica dataset as an example, the SGT model improved the performance of hit rate evaluation metric by 0.35% and 3.13% compared to SGT-L and SGT-G, respectively, while the performance improvement of mean reciprocal rank evaluation metric was 1.28% and 3.08%, respectively. These results indicate that the SR-BE model with both local and global encoders provides a more accurate and comprehensive recommendation system, effectively capturing the relevant features of current and nearby items. Furthermore, the **Table 3** shows that SGT-L performs better than SGT-G on three datasets. This indicates that focusing on the item features in the current session is more important than focusing on the items in its neighborhood.

In conclusion, the ablation experiments and analyses presented in this paper demonstrate the effectiveness of different modules in the proposed SGT model, and the performance of the model can be further improved when multiple modules work together. The results of this study provide insights into the importance of incorporating local and global encoders for achieving optimal performance in session-based recommendation systems.

Table 2. The performance of SR-GNN with other baseline methods over three datasets.

Algorithm	Diginetica		Gowalla		Last.fm	
	Recall@20	MRR@20	Recall@20	MRR@20	Recall@20	MRR@20
Item-KNN	39.51	11.22	38.60	16.66	14.90	4.04
FPMC	28.50	7.67	29.91	11.45	12.86	3.78
GRU4REC	42.55	12.67	39.55	16.99	22.13	7.15
NARM	52.89	16.84	52.24	25.13	23.09	7.90
SRGNN	53.44	17.31	53.24	26.03	23.85	8.23
GC-SAN	54.78	18.57	53.66	25.69	22.64	8.42
LESSR	51.71	18.15	51.34	25.49	23.37	9.01
SGT	56.95	19.74	56.59	28.03	27.92	9.70

Table 3. The performance of different session representations.

Algorithm	Diginetica		Gowalla		Last.fm	
	Recall@20	MRR@20	Recall@20	MRR@20	Recall@20	MRR@20
SGT-L	56.75	19.49	54.17	26.02	28.35	9.45
SGT-G	55.22	19.15	53.13	25.21	15.19	8.63
SGT	56.95	19.74	56.59	28.03	27.92	9.70

4.6 Model analysis and discussion

Impact of the dimension size

From **Figures 3 and 4**, these results are evident that an appropriate increase in the dimension of embedding vectors results in a significant improvement in the model's recommendation performance. This is because a higher embedding dimension can accommodate more latent information, thereby enhancing the model's expression ability. Specifically, for the Diginetica dataset, the model's recommendation performance is optimal when using embedding vectors of around 250 dimensions, and any further increase in dimensionality would lead to a decrease in performance. For the Gowalla dataset, the model's recommendation performance is relatively better at an embedding dimension of 200. Finally, for the Last.fm dataset, the model's recommendation performance is relatively better at an embedding dimension of 150. It is crucial to note that excessively high dimensions can cause the model to have too many parameters, leading to overfitting.

Results on different intent unit granularity

We evaluate the impact of intent granularity level on the performance of our proposed model on three datasets. The corresponding results are illustrated in **Figure 5** and **Figure 6**. These results indicate that for the Diginetica dataset, performance initially improved with increasing granularity, decreased at granularity 5, increased again at granularity 6, and then stabilized with further granularity increase. For the Gowalla dataset, performance decreased at granularity 4, increased at granularity 5, and then decreased again with further granularity increase. Last.fm showed better performance at granularity 3, with a decrease in performance at higher granularity levels. The study suggests that incorporating higher-level granularity is useful for datasets with long session lengths, but performance will become stable with coarser granularity, as longer sessions may not necessarily provide more useful information.

Impact of self-attention layer

Figures 7 and 8 depict the impact of the number of self-attention layers on evaluation metrics. The results indicate that increasing the number of layers does not always lead to better performance for the Diginetica and Gowalla datasets. The optimal number of layers for these datasets is 1, and when the number of layers exceeds this value, the model tends to overfit, resulting in a rapid decline in performance. In contrast, for the Last.fm dataset, performance gradually improves with an increase in the number of layers, but then decreases at layer 4. This is because the model's learning ability increases with more layers, but having too many layers can lead to over-smoothing even when the model is not overfitting. Therefore, adding more layers is not an effective approach for capturing long-range dependencies.

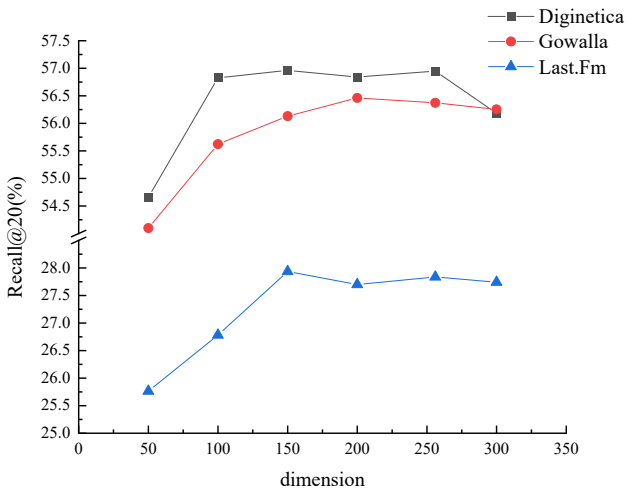

Figure 3. Effects of different embedding dimension on recall.

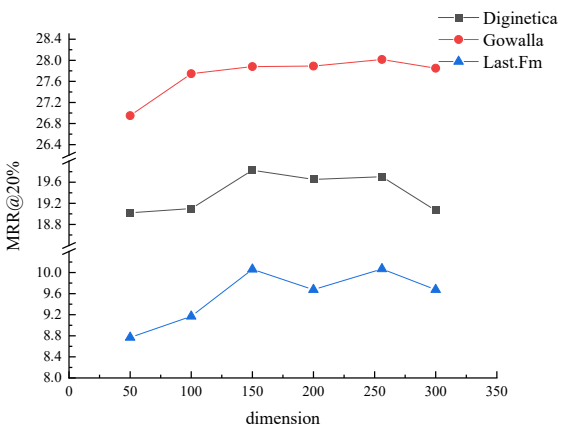

Figure 4. Effects of different embedding dimension on MRR.

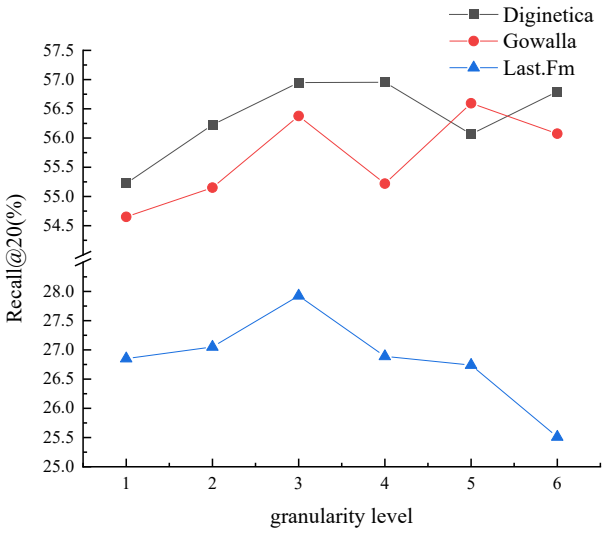

Figure 5. Impact of intent unit granularity levels on recall.

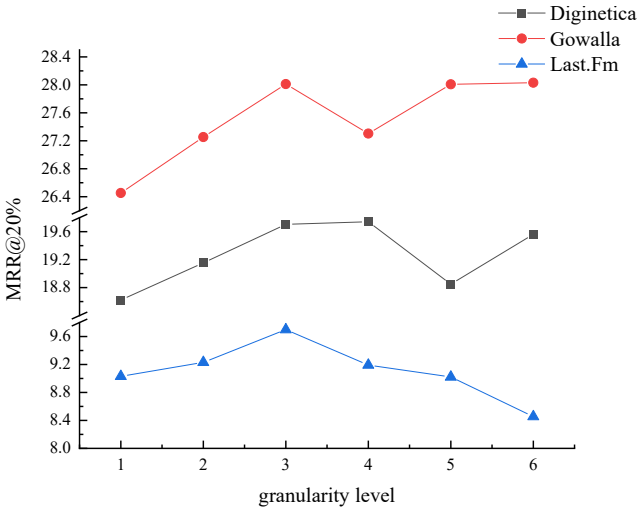

Figure 6. Impact of intent unit granularity levels on MRR.

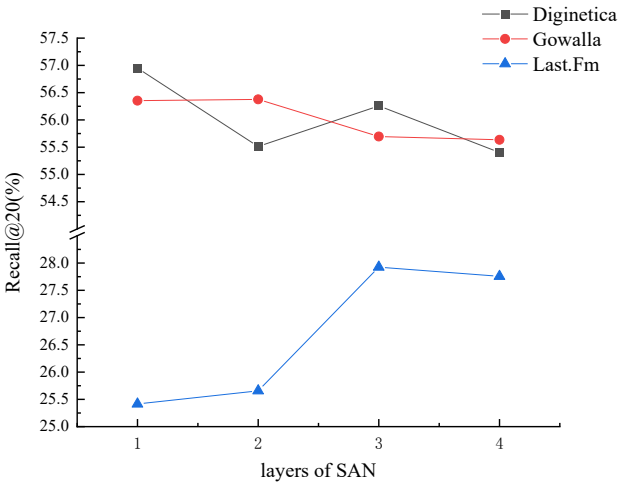

Figure 7. Impact of self-attention layer on recall.

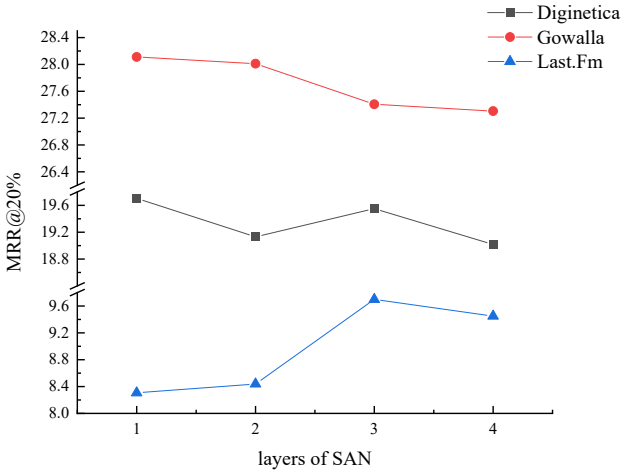

Figure 8. Impact of self-attention layer on MRR.

5. Conclusions

This paper proposes the SGT model based on GRU and Transformer for session-based recommendation. Specifically. We first construct session graphs from anonymous session records by establishing intra-granular and inter-granular edges to represent continuous item units at the same and different levels, respectively. This allows us to capture the complex preference transition relationships and long-term dependencies among multi-level continuous intent units. We then apply GRU to generate new latent vectors for all items, followed by employing transformer to capture multiple interests and assign different weights to different items. The attention network is used to capture global dependencies. Finally, we combine the local short-term dynamics and global dependencies to represent session sequences. Our experiments on three real-world datasets demonstrate that SGT outperforms other baseline methods. In future work, we plan to integrate some available auxiliary information, such as item attributes, to obtain more informative item representations, and explore various types of user behaviors to improve the accuracy of our recommendations.

Author Contributions

All the authors have made significant contributions to the work of the report. Lingmei Wu is mainly responsible for the construction of the idea of this article, the simulation experiment and the writing

of the paper. Liqiang Zhang is mainly responsible for controlling the full text. Xing Zhang is mainly responsible for providing ideas. Linli Jiang is mainly responsible for simulation experiments, and Chunmei Wu is mainly responsible for obtaining data.

Conflict of Interest

There is no conflict of interest.

Acknowledgement

This work was supported by the Scientific Research Basic Ability Enhancement Program for Young and Middle-aged Teachers of Guangxi Higher Education Institutions, "Research on Deep Learning-based Recommendation Model and its Application" (Project No. 2019KY0867), Guangxi Innovation-driven Development Special Project (Science and Technology Major Special Project), "Key Technology of Human-Machine Intelligent Interactive Touch Terminal Manufacturing and Industrial Cluster Application" (Project No. Guike AA21077018), Sub-project: "Touch display integrated intelligent touch system and industrial cluster application" (Project No.: Guike AA21077018-2).National Natural Science Foundation of China (Project No.: 42065004).

References

[1] Wang, H., Zeng, Y., Chen, J., et al., 2022. A spatiotemporal graph neural network for session-based recommendation. Expert Systems with Applications. 202, 117114.

[2] Yang, C., Bai, L., Zhang, C., et al. (editors), 2017. Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2017 Aug 13-17; Halifax. New York: Association for Computing Machinery. p. 1245-1254.

[3] Shu, K., Wang, S., Tang, J., et al. (editors), 2018. Crossfire: Cross media joint friend and item recommendations. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining; 2018 Feb 5-9; Los Angeles. New York: Association for Computing Machinery. p. 522-530.

[4] Corò, F., D'Angelo, G., Velaj, Y. (editors), 2019. Recommending links to maximize the influence in social networks. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019); 2019 Aug 10-16; Macao. International Joint Conferences on Artificial Intelligence. p. 2195-2201.

[5] Kim, Y., Kim, K., Park, C., et al. (editors), 2019. Sequential and diverse recommendation with long tail. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019); 2019 Aug 10-16; Macao. International Joint Conferences on Artificial Intelligence. p. 2740-2746.

[6] Fan, W., Derr, T., Ma, Y., et al. (editors), 2019. Deep adversarial social recommendation. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19); 2019 Aug 10-16; Macao. International Joint Conferences on Artificial Intelligence. p. 1351-1357.

[7] Al Ridhawi, I., Otoum, S., Aloqaily, M., et al., 2020. Providing secure and reliable communication for next generation networks in smart cities. Sustainable Cities and Society. 56, 102080.

[8] Song, W., Xiao, Z., Wang, Y., et al. (editors), 2019. Session-based social recommendation via dynamic graph attention networks. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining; 2019 Feb 11-15; Melbourne. New York: Association for Computing Machinery. p. 555-563.

[9] Rendle, S., Freudenthaler, C, Schmidt-Thieme, L. (editors), 2010. Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web; 2010 Apr 26-30; Raleigh. New York: Association for Computing Machinery. p. 811-820.

[10] Kang, W.C., McAuley, J. (editors), 2018. Self-attentive sequential recommendation. 2018 IEEE International Conference on Data Mining (ICDM); 2018 Nov 17-20; Singapore. New York: IEEE. p. 197-206.

[11] He, R., McAuley, J. (editors), 2016. Fusing similarity models with Markov chains for sparse sequential recommendation. 2016 IEEE 16th International Conference on Data Mining (ICDM); 2016 Dec 12-15; Barcelona. New York: IEEE. p. 191-200.

[12] Hidasi, B., Karatzoglou, A., Baltrunas, L., et al. (editors), 2016. Session-based recommendations with recurrent neural networks. 4th International Conference on Learning Representations; 2016 May 2-4; San Juan. International Conference on Learning Representations. p. 289.

[13] Li, J., Ren, P., Chen, Z., et al. (editors), 2017. Neural attentive session-based recommendation. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management; 2017 Nov 6-10; Singapore. New York: Association for Computing Machinery. p. 1419-1428.

[14] Tan, Y.K., Xu, X., Liu, Y. (editors), 2016. Improved recurrent neural networks for session-based recommendations. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems; 2016 Sep 15; Boston. New York: Association for Computing Machinery. p. 17-22.

[15] Song, K., Ji, M., Park, S., et al., 2019. Hierarchical context enabled recurrent neural network for recommendation. Proceedings of the AAAI Conference on Artificial Intelligence. 33(1), 4983-4991.

[16] Wu, S., Tang, Y., Zhu, Y., et al., 2019. Session-based recommendation with graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence. 33(1), 346-353.

[17] Xu, C., Zhao, P., Liu, Y., et al. (editors), 2019. Graph contextualized self-attention network for session-based recommendation. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19); 2019 Aug 10-16; Macao. International Joint Conferences on Artificial Intelligence. p. 3940-3946.

[18] Qiu, R., Li, J., Huang, Z. (editors), et al., 2019. Rethinking the item order in session-based recommendation with graph neural networks. Proceedings of the 28th ACM International Conference on Information and Knowledge Management; 2019 Nov 3-7; Beijing. New York: Association for Computing Machinery. p. 579-588.

[19] Chen, T., Wong, R.C.W. (editors), 2020. Handling information loss of graph neural networks for session-based recommendation. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020 Jul 6-10; New York. New York: Association for Computing Machinery. p. 1172-1180.

[20] Pan, Z., Cai, F., Chen, W., et al. (editors), 2020. Star graph neural networks for session-based recommendation. Proceedings of the 29th ACM International Conference on Information & Knowledge Management; 2020 Oct 19-23; New York. New York: Association for Computing Machinery. p. 1195-1204.

[21] Xu, K., Hu, W., Leskovec, J., et al., 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

[22] Vaswani, A., Shazeer, N., Parmar, N., et al. (editors), 2017. Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017); Long Beach. New York: Association for Computing Machinery. p. 5998-6008.

[23] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv: 2010.11929.

[24] Rives, A., Meier, J., Sercu, T., et al., 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences. 118(15), e2016239118.

[25] Wang, P., Guo, J., Lan, Y., et al. (editors), 2015.

Learning hierarchical representation model for nextbasket recommendation. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015 Aug 9-13; Santiago. New York: Association for Computing Machinery. p. 403-412.

[26] Cho, K., Van Merriënboer, B., Gulcehre, C., et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[27] Liu, Q., Zeng, Y., Mokhosi, R., et al. (editors), 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018 Aug 19-23; London. New York: Association for Computing Machinery. p. 1831-1839.

[28] Wu, T., Sun, F., Dong, J., et al., 2022. Context-aware session recommendation based on recurrent neural networks. Computers and Electrical Engineering. 100, 107916.

[29] Yu, F., Zhu, Y., Liu, Q., et al. (editors), 2020. TAGNN: target attentive graph neural networks for session-based recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020 Jul 25-30; Xi'an. New York: Association for Computing Machinery. p. 1921-1924.

[30] Zhang, M., Wu, S., Gao, M., et al., 2020. Personalized graph neural networks with attention mechanism for session-aware recommendation. IEEE Transactions on Knowledge and Data Engineering. 34(8), 3946-3957.

[31] Sarwar, B., Karypis, G., Konstan, J., et al. (editors), 2001. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web; 2001 May 1-5; Hong Kong.New York: Association for Computing Machinery. p. 285-295.