SEGAR: Knowledge Graph Augmented Session-Based Recommendation

Xinyi Xu, Yan Tang(✉), and Zhuoming Xu

College of Computer and Information, Hohai University, Nanjing 210036, China
{xy.xu.hhu,tangyan,zmxu}@hhu.edu.cn

Abstract. Predicting the next interaction item in the session-based recommendation system is an emerging and challenging research task. Existing studies model a session as a sequence or graph of items for predicting the next-click item. However, these approaches ignore the global graph-based relations between the session items and the local neighborhood-based item relevance to external knowledge bases, thus fail to encode rich semantic knowledge between items for achieving comprehensive and accurate recommendations. To overcome the current shortcomings, we proposed a novel knowledge graph augmented model called SEGAR (Knowledge Graph Augmented Session-based Recommendation) by leveraging graph convolutional network and knowledge graph attention network. When integrating the static local attributes and the knowledge about all the last session items encoded in their k-hop neighborhoods in the knowledge graph, SEGAR models all sessions as a session graph and captures the dynamic global temporal and popularity-aware information from the session context. The model encodes a comprehensive semantic knowledge between items for achieving more accurate recommendation. Extensive experiments on two benchmark datasets show that SEGAR outperforms four state-of-the-art models on the session-based recommendation task.

Keywords: Session-based recommendation · Attention network · Graph convolutional network · Knowledge graph

1 Introduction

In recent years, more and more researches are paying attention to the new research field of session-based recommender systems [2,3,6,15]. A session-based recommender system provides recommendations solely based on a user's interactions in an ongoing session, and which does not require the existence of user-profiles or their entire historical preferences. Such system aim to capture short-term but dynamic user preferences to provide more timely and accurate recommendations sensitive to the evolution of users' session contexts.

Numerous studies focus on modeling the user sessions and on leveraging deep learning models such as Recurrent Neural Networks (RNNs) [13] to capture users' general interests from the session of user actions. [6] uses the encoder-decoder

H. Qiu et al. (Eds.): KSEM 2021, LNAI 12815, pp. 229–241, 2021.
https://doi.org/10.1007/978-3-030-82136-4_19

structure and the attention mechanism for the recommendation, which cannot model complex session patterns. Current RNN-based methods only focus on the current session and the transition to the last-clicked item and ignore the dynamic information contained in the item's neighborhood in the session.

A Knowledge Graph (KG) is the source of factual knowledge, depicted in the form of a graph of entities and their relationships. KG is a semantic-rich computational model that maps entities and relations to low-dimensional representation vectors through KG embedding methods to obtain a semantic-rich item presentation [4]. The KG domain can associate item entities, and uses local semantic neighborhood knowledge to improve the precision of recommendation predictions. The representation of the item entity in the knowledge graph contains the structural information and feature information on its $k-hop$ neighbors, which is useful for the recommendation. However, there are few studies that take advantage of KG to assist the session-based recommendation task.

In this paper, we proposed the SEGAR (Knowledge Graph Augmented Session-based Recommendation) to overcome the current research limitations mentioned above. As shown in Fig. 1. Firstly, SEGAR models all sessions as a session graph to obtain the embedding of every session node through a Graph Convolutional Network that integrates both time and popularity weights into the session node global embedding. The embedding of the session is then obtained by means of an attention mechanism. Secondly, SEGAR makes use of the one-to-one mapping between the item entity in the external knowledge graph to the last item in the session to be predicted. It uses the Knowledge Graph Attention Network [12] to aggregate the $k - hop$ neighborhood knowledge of the item entity to obtain the local embedding for the last session item. Lastly, SEGAR jointly combines the global embedding of the session and the local embedding of last session item to generate the final session embedding for predicting the next-step clicking likelihood of each candidate item. Extensive experiments and a case study on two benchmark datasets show that SEGAR outperforms four state-of-the-art session-based models. As far as we know, this is one of the first research attempts that aims to simultaneously taking into account the dynamic global relationship and static knowledge of local neighborhoods for session-based recommendations. To summarize, the major contributions to this study are as follows:

- We propose a graph-based session node embedding method based on graph convolutional neural network that considers the weights of item's temporal and popularity factors to jointly obtain the session item's dynamic global embedding. This method helps to improve the recommendation accuracy.

- We leverage the state-of-the-art knowledge graph attention network on external KG by associating item entity in KG with the last session item to obtain $k - hop$ neighborhood semantic knowledge. This encodes the static local neighborhood-based item relevancy for further improving the accuracy of session-based recommendation.

- Extensive experiments are conducted on two real-world data sets for our proposed model, the results show that the performance of SEGAR is better than the four SOTA models.

2 Related Work

Common session-based recommendation methods include conventional methods, latent representation methods, and deep neural networks. FPMC [9] is a session prediction method based on Markov chains. It also has the problem that it can only deal with uncomplicated datasets. Nowadays, there are more researches using deep neural networks to solve the session-based recommendation problem. The model STAMP [8] obtains the users' general interest in the long-term memory of the session context, while taking into account that the users' current interest comes from the short-term memory of the last click. NARM [6] only captures general interest, and combines the main purpose and sequential behavior to get a session representation. However, our model clearly emphasizes the mixture feature of current interest and general interest in the last click. The SR-GNN [14] transforms sessions into a graph structure and constructs a directed graph based on historical sessions using graph neural network but does not encode the local relations between items in external knowledge bases and does not explicitly consider the impact on time and item popularity. RippleNet [10] uses KG and takes user's favorite items as a starting point, and distributes it on KG through multiple layers such as ripples to achieve user feature extraction but the importance of item relations to RippleNet is sparse and difficult to converge. In short, SEGAR can combine dynamic global knowledge and static local knowledge and learn at the same time in a unified model to achieve more accurate recommendations.

3 Problem Definition

In this section, we give the following problem definition. Let $V = \{v_1, v_2 \dots v_n\}$ represents the total set of items contained in all sessions. The session S can be sorted by time stamp as the list $S = [v_{s,1}, v_{s,2} \dots v_{s,n-1}]$, where $v_{s,i} \in V$ means an item is interacted by the user for the i-th time in the session S. $k_{s,i}$ represents the total number of occurrences of the item node $v_{s,i}$ in all sessions, and $t_{s,i}$ represents the time stamp of the item node $v_{s,i}$ in all sessions. Our task of the session-based recommendation is to predict the next clicked item of session S, denoted as $v_{s,n+1}$. Given a session S, the SEGAR outputs the probabilities $\hat{Y}$ of all candidate items where the value of $\hat{Y}$ is the score that the corresponding item may be recommended. We select the item with the highest score of the recommendation.

Session Graph: Each session S can be modeled as a directed graph $G = (V, E)$, where $v_{s,i} \in V$ represents an item node. Edge $(v_{s,i-1}, v_{s,i}) \in E$ represents that the user clicks on the item $v_{s,i}$ after $v_{s,i-1}$ in the session S. Then, we can get the adjacency matrix A of the session graph, which can be used as the input of GCN when generating the embedding of the session node. Each session S can be represented as a vector $\mathbf{S}$ composed of the node vectors.

Knowledge Graph: The knowledge graph is composed of a large number of entities and the relationships between entities. In this task, the entities we filter out are mainly items that appear in the sessions and extract the triples associated with the item entities, represented as (h, r, t), where h and t represent the head entity and the tail entity, respectively, and r represents the relationship between the head entity and the tail entity, such as item attributes.

4 The Proposed Model

The schematic illustration of the SEGAR model is presented in Fig. 1. Firstly, we model all sessions which composed of items as session graphs and use the GCN to obtain the dynamic global session embedding $\mathbf{S_g}$ that integrates the temporal and popularity weights of each session node. Secondly, we use the Knowledge Graph Attention Network [12] to aggregate the $k-hop$ neighborhood knowledge of the item entity to obtain the local embedding $\mathbf{S_l}$ for the last session item. Lastly, SEGAR jointly combines the dynamic global embedding $\mathbf{S_g}$ of the session and the local embedding $\mathbf{S_l}$ of last session item to generate the final session embedding $\mathbf{S}$ for predicting the next-step clicking likelihood of each candidate item. In this section, we describe the SEGAR model in three steps: Global Embedding, Local Embedding, and Recommendation Generation.

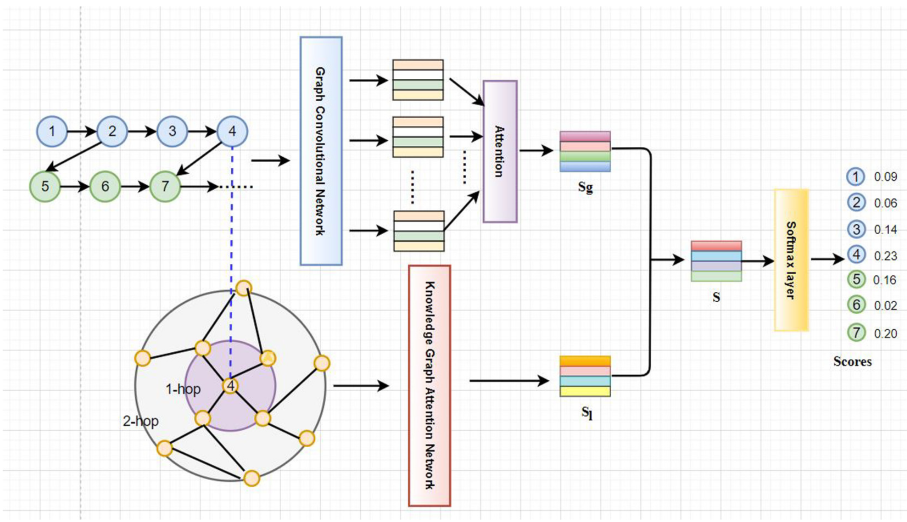


Fig. 1. The schematic illustration of our proposed model

4.1 Global Embedding

After establishing the session graph, SEGAR obtains the embedding of the session node. The structure of the graph is usually irregular. The ingenious design of GCN provides feature extraction and calculation capabilities from graphic data, and finally obtains a vectorized representation of the graph. There are n nodes in the session graph. We randomly initialize the characteristics of these nodes to form a matrix X. Then the relationship between each node forms an adjacency matrix A. The matrix X and A are inputs to the GCN model. We first show the learning process of node embeddings in the session graph. Formally, the nodes in the graph G update function is defined as follows:

$$v_{s,i}^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}v_{s,i}^{(l)}W^{(l)} + b^{(l)}\right) \tag{1}$$

$v_{s,i}^{(l+1)}$ represents the node at the layer of $l+1$ of the GCN. For the input layer, $v_{s,i}$ is represented as X and $\hat{A}$ represents $A+I$, where I is the identity matrix. $\hat{D}$ represents the corresponding degree matrix of $\hat{A}$. σ is a non-linear activation function. This operation obtains the initial embedding vector of the item node.

The popularity weight of our item is the number of times each item is clicked divided by the total number of all candidate nodes. The item popularity weight of a session node is defined as Eq. (2):

$$W_{p,i} = \frac{k_{s,i}}{\sum_{i=1}^{n-1} k_{s,i}} \tag{2}$$

In addition, the time span between the nodes in the session indicates the temporal distance from two items in the session S. Common sense tells us items with long temporal distance from the current session node tend to have a weaker impact on the interaction of the next item. Therefore, SEGAR uses the time stamp of each node in the session graph to calculate the session node item temporal weight. The equation for calculating the temporal weight of a session node is as defined as Eq. (3):

$$W_{t,i} = \frac{|t_{s,i} - t_{s,n-1}|}{t_{s,n-1}} \tag{3}$$

As a result, the weight of a session node combining both popularity and temporal factors is defined as Eq. (4):

$$W_{s,i} = W_{p,i} \cdot W_{t,i} \tag{4}$$

Lastly, the global embedding of the session node in the session graph $\mathbf{V_{s,i}}$ combined both popularity and temporal factors is defined as Eq. (5):

$$\mathbf{V_{s,i}} = v_i^{(l+1)} \cdot W_{s,i} \tag{5}$$

Finally, we use the attention mechanism to obtain the global embedding $\mathbf{S}_{\mathrm{g}}$.

$$\alpha_i = \mathbf{q}^\top \sigma\left(W_1 \mathbf{V}_{\mathrm{s},\mathbf{n}-1} + W_2 \mathbf{V}_{\mathbf{s},\mathbf{i}} + c\right) \tag{6}$$

$$\mathbf{S}_{\mathrm{g}} = \sum_{i=1}^{n-1} \alpha_i \mathbf{V}_i \tag{7}$$

where parameters $\mathbf{q} \in \mathbb{R}^d$,$W_1, W_2 \in \mathbb{R}^{d\times d}$ control the embedding of session node and $\mathbf{S}_{\mathrm{g}} \in \mathbb{R}^d$ represents the global embedding vectors.

4.2 Local Embedding in SEGAR

After getting the session nodes' dynamic global embedding, SEGAR leverages external knowledge graph to obtain the session node's static local embedding that encodes the neighborhood knowledge. First, SEGAR forms an one-to-one mapping relationship between the entities in the item KG and the last session node in the session. Then, KGAT(Knowledge Graph Attention Network) [12] is used to aggregate the attribute and neighborhood knowledge of the item entity in the KG to obtain the local embedding of the last session item in the session.

The initial embedding layer of the KG is a vector representation of the structural nodes and edges of the KG using TransR [7]. The practice of multi-layered attention embedding propagation is the same as that of the first layer, except that the operation is repeated multiple times. The first layer of propagation is divided into three steps: message propagation, knowledge-based attention, and information aggregation [12]. The whole process recursively propagates the node neighborhood embeddings to update their representations and uses the attention mechanism to learn the weights of each neighbor in the propagation process.

Message Propagation: For a node n, record the set of triples with its head node as N_h represented as (h, r, t), then the message from the neighborhood is $e_t \in \mathbb{R}^d$. Here, we use TransR [7] to get tail entity embedding denoted as e_t.

$$e_{N_h} = \sum_{(h,r,t)\in N_h} \pi(h, r, t) e_t \tag{8}$$

$\pi(h, r, t)$ represents the weight of the neighborhood and controls the decay factor on each propagation on edge r, indicating how much information being propagated from t to h conditioned to relation r.

Knowledge-Based Attention: The weight of the neighborhood is obtained through the relational attention mechanism, which is defined as Eq. (9):

$$\pi(h, r, t) = (W_r e_t)^\top \tanh\left((W_r e_h + e_r)\right) \tag{9}$$

We choose tanh as the non-linear activation function.

Attention score depends on the distance between e_h and e_r under relation r. Finally, SEGAR uses softmax normalization on the weight of the neighborhood, which is defined as Eq. (10):

$$\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{(h,r',t') \in \mathcal{N}_h} \exp\left(\pi\left(h, r', t'\right)\right)} \tag{10}$$

Information Aggregation: Then, an Information aggregator is used to aggregate the entity representation. According to KGAT [12], there exists three types of aggregator functions:*GCN Aggregator, GraphSage Aggregator* and *Bi-Interaction aggregator*. Among them, we choose the Bi-Interaction aggregators since it works best for session-based recommendation tasks by considering both interaction directions. The Bi-Interaction aggregator is defined in Eq. (11):

$$f = LeakyReLU\left(W\left(e_h + e_{N_h}\right)\right) + LeakyReLU\left(W\left(e_h \odot e_{N_h}\right)\right) \tag{11}$$

We choose the LeakyReLU activation function, which introduces a small slope to keep the updates process alive with the ability to retain some degree of the negative values that flow into it.

The advantage of the embedding propagation layer lies in explicitly exploiting the first-order connectivity information. The equation for the k_{th} hops Information propagation is defined in Eq. 12:

$$e_{s,n-1}^{(k)} = e_h^{(k)} = f\left(e_h^{(k-1)}, e_{N_h}^{(k-1)}\right) \tag{12}$$

After the information aggregation, SEGAR iteratively aggregates multi-hop neighborhood information to obtain a richer and more complete representation of the last session node. Finally, the local embedding vector $\mathbf{S_l}$ of the last session node is represented in Eq. 13:

$$\mathbf{S_l} = \mathbf{V_{s,n-1}} = e_{s,n-1}^{(0)} \| \ldots \| e_{s,n-1}^{(k)} \tag{13}$$

where $\|$ represents the concatenation operation.

4.3 Recommendation Generation

After generating both global and local session node embeddings, then we consider the embedding $\mathbf{S}$ of the session graph G_s by combining all session node embeddings. The overall session representation $\mathbf{S}$ is then obtained by jointly combing the local and global session embeddings:

$$\mathbf{S} = W_3\left[\mathbf{S_l}; \mathbf{S_g}\right] \tag{14}$$

where $\mathbf{S_l} \in \mathbb{R}^d$ represents the local embedding in our model,$W_3 \in \mathbb{R}^{d \times 2d}$ compresses the local embedding and the global embedding.

After obtaining the embedding for the entire session, we calculate the score $\hat{\mathbf{y}}_i$ for each candidate item:

$$\hat{\mathbf{y}}_i = \mathbf{S}^\top \mathbf{V}_{s,i} \tag{15}$$

Then, we apply the softmax function to obtain the prediction output vector $\hat{\mathbf{Y}}$ of the model:

$$\hat{\mathbf{Y}} = softmax(\hat{\mathbf{y}}) \tag{16}$$

where $\hat{\mathbf{Y}}$ represents a probability distribution over the items $v_{s,i} \in V$, each element $\hat{y_i} \in \hat{\mathbf{Y}}$ denotes the probability of item $v_{s,i}$ appearing as the next-click in the session.

For each session graph, the loss function is defined as the cross entropy of $\hat{\mathbf{Y}}$:

$$\mathcal{L}(\hat{\mathbf{Y}}) = -\sum_{i=1}^{m} \mathbf{Y}_i \log\left(\hat{\mathbf{Y}}_i\right) + (1 - \mathbf{Y}_i) \log\left(1 - \hat{\mathbf{Y}}_i\right) \tag{17}$$

5 Experiments

In this section, we first introduce the experimental data set, then we describe the experimental setup, and finally analyze and compare the performance of the SEGAR model with four SOTA baselines.

5.1 Experiments Setup

Datasets and Data Preparation: We evaluate the proposed models on two datasets. There are no publicly available session recommendation datasets that also have corresponded session item KG. Therefore, we use benchmark datasets in the field of music and movie recommendation, namely Last.fm [11], MovieLens 1M[1] We then linked the item in the datasets with the Freebase [1] entity preserving the user interactions with the linked items in the experimental dataset. We group interaction records by users and sort them according to time stamps and form a session. We also filter out rarely visited items with less than two clicks, and we filter out clicks (items) that did not appear in training set.

Same as [8], we use a session splitting preprocess to generate sessions and corresponding labels by segment the input session for training and testing for both datasets. For example, for an input session,$s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$, we generate a series of sessions and labels $([v_{s,1}, v_{s,2}, \ldots, v_{s,n-1}], v_{s,n})$. In this session, $[v_{s,1}, \ldots, v_{s,n-1}]$ is the generated session, and $v_{s,n}$, represents the next clicked item as the label of the session. The statistics of two benchmark datasets, namely Last.fm and MovieLens 1M, and their corresponding KGs built from Freebase are shown in Table 1 and Table 2.

Evaluation Metrics: We use the following two metrics for evaluation of the performance of the Session-based Recommendation model, which is also widely used in related studies.R@20 is the proportion of correctly recommended items in the test cases amongst the top-20 items.MRR@20 is the average of the reciprocal rank of the recommended items.

[1] https://grouplens.org/datasets/movielens/1m/.

Table 1. Statistics of datasets

Statistics	Last.fm	MovieLens 1M
#clicks	21,074	916,714
#items	3,846	3,952
#sessions	3,009	13,633
#average length	5.67	19.06

Table 2. Statistics of KGs

Statistics	Last.fm	MovieLens 1M
#interactions	42,346	756,684
#data density	3.704%	0.268%
#entities	9,366	18,920
#relations	60	81
#KG Triples	15,518	968,038

Parameters: We construct training set and test set by randomly splitting the dataset with the ratio of 7 : 3. In the SEGAR model, we set the dimension of the latent vector d as 128 for both datasets. The mini-batch Adam [5] optimizer is exerted to optimize these parameters, where the initial learning rate is set to 0.001. The initial learning rate is in $\{0.001, 0.0005, 0.0001\}$, and it will decay by 0.1 every three cycles. The batch size is 128, and the epoch is 30. The embedding size is 128. All the local items embeddings are initialized using TransR [7].

Baselines: The following models, including the state-of-art and closely related works, namely:SR-GNN, NARM, FPMC and RippleNet. They are used as baselines to evaluate the performance of SEGAR:

5.2 Comparison with Baseline Models

In this study, we compared SEGAR with four SOTA baselines mentioned above. The overall performance of R@20 and MRR@20 is shown in Table 3. From the experiment results, it is observed that SEGAR consistently achieves the best performance on both benchmark datasets in terms of R@20 and MRR@20, expect for obtaining a bit lower the MRR values on Last.fm dataset compared to NARM. These results verify the effectiveness of the proposed model.

Table 3. The comparison of SEGAR with four SOTA baselines over two datasets

Method	Last.fm		MovieLens 1M	
	R@20(%)	MRR@20(%)	R@20(%)	MRR@20(%)
SR-GNN	12.007	2.844	25.000	6.232
NARM	11.990	**3.370**	21.137	5.370
FPMC	7.437	0.801	16.742	3.676
RIPPLENET	7.910	1.903	23.560	5.138
SEGAR(1hop)	**14.490**	2.899	**34.375**	**9.983**

The neural network-based method such as NARM and SR-GNN achieves decent performance, proving the effectiveness of deep learning in the session-based recommendation. For the FPMC method, the main disadvantage of the Markov chain-based model is that the independence assumption is too strong, which limits the accuracy of the prediction. Short-term/long-term memory model such as NARM uses recurring units to capture the user's overall interest. This method has the disadvantage of ignoring the transition to distant objects. That is why NARM has a strong performance on short sessions as for Last.fm datasets, but performs not so well on datasets with long sessions such as Movielens. In contrast, the local and global embedding of SEGAR ensures a steadily accurate performance on both short and long sessions-based recommendation. For RippleNet, which like SEGAR that also uses KGs to make recommendations, the importance of the relationship is weak, because the embedded matrix of the relationship is sparse and slow to converge. Besides, as the size of KG increases, the size of the fluctuation set may grow unexpectedly, which not only leads to calculation and storage overhead but also reduces the accuracy.

5.3 Ablation Study

To evaluate how much can the introduction of knowledge graph enhance the session-based recommendation, and to study how many hops of the node neighborhood aggregation are sufficient for achieving accurate results, we perform ablation study and present the results and findings in this section. The following methods are used in this study:

Table 4. Results of the ablation experiment

Method	Last.fm		MovieLens 1M	
	R@20(%)	MRR@20(%)	R@20(%)	MRR@20(%)
SRGAR(w/o KG)	12.145	2.807	28.125	6.812
SRGAR(KG-1hop)	**14.490**	**2.899**	**34.375**	**9.983**
SRGAR(KG-2hop)	14.445	2.844	33.082	7.737
SRGAR(KG-3hop)	14.253	2.849	31.250	7.406

SEGAR-without KG: This method removes the part of the model that uses KGs to generate local session embeddings, and only uses graph convolutional neural networks and temporal and popularity weights to update session node embeddings.

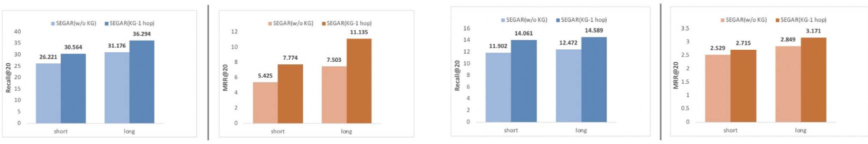
SEGAR-KG (k hop): This method is controlled by the number of hops of node neighborhoods aggregated into the static local item embedding.

Combining Table 3 and Table 4, we find that the overall effect of SEGAR with KG is better, which proves that the GCN, the temporal weight, and popularity weights indeed help update session nodes with dynamic global knowledge and assist in achieving more accurate prediction. Moreover, the popularity weight has a greater influence on the musical dataset. So, it will have a slightly better result. In another dataset, the variance of the length of the sessions in the MovieLens 1M dataset is large, so this has a certain impact on the model results which are not as close as the Last.fm dataset. Because of the large number of sessions in the MovieLens 1M dataset, the Recall and MRR values are higher.

Furthermore, the model with the KG allows an additional improvement of the results. It shows that 1-hop neighborhood aggregation performs better than 2-hop and 3-hop neighborhood aggregation. This demonstrates that information directly related to the node has more significant impact in the session-based recommendation. The 1-hop neighborhood is a key entity related to items. More neighborhood may reduce and dilute the distinguishing features of the local integration vector, thus reducing the prediction accuracy. In summary, this study show that the SEGAR with KG performs better than the model without KG. The knowledge graph plays an important role in helping the session based recommendation.

5.4 Comparison Between Long and Short Sessions

The last comparative experiment aims to verify the effectiveness of the model over long and short sessions. On the Last.fm dataset, we represent short sessions with a session length less than or equal to 6, and those with a session length of larger than 6 are regarded as long sessions. Their proportions are 55.44% and 44.56%, respectively. The division is based on the average length. Similarly, on the MovieLens 1M dataset, we represent those with a session length of less than or equal to 19 as short sessions and those with a session length of more than 19 are regarded as long sessions. Their proportions are 37.43% and 62.57%, respectively. The experimental results are shown in Fig. 2.



(a) Last.fm (b) MovieLens 1M

Fig. 2. The results of comparison between long and short sessions

We find that the model has a better recommendation effect on long sessions. This is because, as the length of the session grows, the model can capture more complete user interaction information, generating better the recommendation results. In addition, the overall result of the Last.fm dataset is not as good as MovieLens 1M, because the average session length of the Last.fm is much shorter. There is also a large gap in the ratio of session length distribution.

6 Conclusions

Current session-based recommendation models ignore the dynamic global connection between the items and the static local neighborhood-based item relevancy from external knowledge bases, which encode a complete semantic knowledge between items for a more comprehensive and accurate recommendation. To address these issues, we proposed a novel model called SEGAR by leveraging the convolutional graph network and knowledge graph attention network. SEGAR has a unique collaborative knowledge integration mechanism that incorporates static, local and dynamic representations of global session entities with time and popularity factors. Extensive experiments on two benchmark datasets show that SEGAR outperforms four state-of-the-art methods in the accuracy for session-based recommendations. We plan to reduce the time complexity of this model for broader applications in the future.

Acknowledgments. The work was supported by Key Technologies Research and Development Program of China (2017YFC0405805-04) and Basal Research Fund of China (2018B57614).

References

1. Chen, D., Zhao, H.: Research on the method of extracting domain knowledge from the freebase RDF dumps. IEEE Access **6**, 50306–50322 (2018)
2. Chen, M., Zhang, Y., Qiu, M., Guizani, N., Hao, Y.: SPHA: smart personal health advisor based on deep analytics. IEEE Commun. Mag. **56**(3), 164–169 (2018)
3. Gu, P., Han, Y., Gao, W., Xu, G., Wu, J.: Enhancing session-based social recommendation through item graph embedding and contextual friendship modeling. Neurocomputing **419**, 190–202 (2021)
4. Huang, J., Ren, Z., Zhao, W.X., He, G., Wen, J.R., Dong, D.: Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 573–581 (2019)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
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, pp. 1419–1428 (2017)
7. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 25–30 January 2015, pp. 2181–2187 (2015)
8. 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, pp. 1831–1839 (2018)
9. 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, pp. 811–820 (2010)

10. Wang, H., et al.: Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426 (2018)
11. Wang, H., Zhao, M., et al.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019, pp. 3307–3313. ACM (2019)
12. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, 4–8 August 2019, pp. 950–958. ACM (2019)
13. Wu, C., Wang, J., Liu, J., Liu, W.: Recurrent neural network based recommendation for time heterogeneous feedback. Knowl.-Based Syst. **109**, 90–103 (2016)
14. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. Proc. AAAI Conf. Artif. Intell. **33**, 346–353 (2019)
15. Zhang, C., Nie, J.: Spatio-temporal attentive network for session-based recommendation. In: Li, G., Shen, H.T., Yuan, Y., Wang, X., Liu, H., Zhao, X. (eds.) KSEM 2020. LNCS (LNAI), vol. 12275, pp. 131–139. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55393-7_13