Unsupervised Graph Embeddings for Session-based Recommendation with Item Features

ANDREAS PEINTNER, University of Innsbruck, Austria
MARTA MOSCATI, Johannes Kepler University Linz, Austria
EMILIA PARADA-CABALEIRO, Johannes Kepler University Linz and Linz Institute of Technology, Austria
MARKUS SCHEDL, Johannes Kepler University Linz and Linz Institute of Technology, Austria
EVA ZANGERLE, University of Innsbruck, Austria

In session-based recommender systems, predictions are based on the user's preceding behavior in the session. State-of-the-art sequential recommendation algorithms either use graph neural networks to model sessions in a graph or leverage the similarity of sessions by exploiting item features. In this paper, we combine these two approaches and propose a novel method, ***G**raph **C**onvolutional **N**etwork **Ext**ension (GCNext)*, which incorporates item features directly into the graph representation via graph convolutional networks. *GCNext* creates a feature-rich item co-occurrence graph and learns the corresponding item embeddings in an unsupervised manner. We show on three datasets that integrating GCNext into sequential recommendation algorithms significantly boosts the performance of nearest–neighbor methods as well as neural network models. Our flexible extension is easy to incorporate in state-of-the-art methods and increases the *MRR@20* by up to 12.79%.

ACM Reference Format:
Andreas Peintner, Marta Moscati, Emilia Parada-Cabaleiro, Markus Schedl, and Eva Zangerle. 2022. Unsupervised Graph Embeddings for Session-based Recommendation with Item Features. In *CARS: Workshop on Context-Aware Recommender Systems (RecSys '22), September 23rd, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems (RecSys) traditionally leverage the users' rich interaction data with the system. However, in some cases, such data are not available. Session-based recommender systems, in contrast, aim to predict the next item the user will interact with (e. g., click on, purchase, or listen to) only based on the preceding interactions in the current session. The task of session-based recommendation can be defined as follows. Consider the set X of all items in the catalog, $x \in X$ being an individual item, and $m = |X|$ being the number of items in the catalog. Given an interaction session $[x_1, x_2, ..., x_n]$ (ordered by timestamp), the goal is to predict a ranked list $[y_1, y_2, ..., y_m]$ of items with corresponding relevance scores to continue the session. The top-k values of the ranked list are chosen as recommendation candidates. As opposed to session-aware or sequential recommendations, the inputs to session-based RecSys are only items of the current session and their features; users are anonymous and no inter-session data is available.

Current approaches for session-based recommendation leverage Recurrent Neural Networks (RNNs) [14, 31, 34], attention networks [20, 21], Graph Neural Networks (GNNs) [27, 41, 42], or transformer architectures [3, 17, 33]. Also, classical nearest–neighbor methods have been used [6, 16, 24, 25]. Most current methods focus on the sequential nature of sessions; RNNs and nearest–neighbor methods have dominated research in the past few years. Extensions to these models use additional item features to enrich the item representations. Item features capture contextual information (e. g., item category) which is relevant to the task of session-based recommendation, which itself can be considered a special case of context-aware recommender systems [30]. However, recently, GNN models have been shown to outperform RNN- and nearest-neighbor–based methods [9, 25, 41]. Yet, to the best of our knowledge, no approach combines auxiliary item features and GNNs to learn informative embeddings for sequential models. In this paper, we therefore propose ***G**raph **C**onvolutional **N**etwork **Ext**ension (GCNext)*, which extracts node embeddings from a feature-rich item co-occurrence graph via unsupervised learning with Graph Convolutional Networks (GCNs). We then use these pre-trained item embeddings as auxiliary features describing items and their structural dependence. One major advantage and novelty of GCNext is that it can flexibly be plugged into any current sequential recommendation method. Particularly, we (1) use the computed item embeddings to initialize sequential neural network models, and (2) extend (non-neural) nearest–neighborhood methods with pre-trained item graph embeddings to refine the search of candidate sessions for recommendation.

Our main contributions are as follows: (1) We present GCNext, a novel method for session-based recommendation based on a item co-occurence graph for sessions. GCNext combines the topological representation power of GCNs with the session representation generated by neural network sequential models without modifying their architecture; (2) GCNext can easily extend nearest-neighbor methods as well as neural network models in a plug-in fashion to further enhance the performance of these models; (3) We perform a large-scale evaluation of graph-based item embeddings and their impact on a diverse set of sequential models. We find that adding GCNext is not only able to boost the performance of current methods, but also shows significant performance improvements over current state-of-the-art sequential models.

2 RELATED WORK

In the following, we briefly present related works in the field of graph and node embedding. We subsequently discuss approaches for sequential and session-based recommendation, which incorporate side information or GNNs.

2.1 Graph and Node Embeddings

Graph embedding aims to generate low-dimensional vector representations of the graph's nodes which preserve topology and leverage node features. Non-deep learning methods are mainly based on random walks to explore node neighborhoods [8, 28, 35]. With GCNs [19, 38], more sophisticated graph embedding methods were introduced: To scale GCNs to large graphs, the layer sampling algorithm [10] generates embeddings from a fixed node neighborhood. Current state-of-the-art methods in unsupervised learning of representations rely on contrastive methods which base their loss on the difference between positive and negative samples. Deep Graph Infomax (DGI) [39] contrasts node and graph encodings by maximizing the mutual information between them. Hassani and Khasahmadi [11] propose multi-view representation learning by contrasting first-order neighbor encodings with a general graph diffusion. Contrastive learning methods usually require a large number of negative examples and are, therefore, not scalable for large graphs. The approach by Thakoor et al. [37] learns by predicting substitute augmentations of the input and circumventing the need of contrasting with negative samples.

2.2 Sequential Recommendation

Non-neural sequential recommendation approaches focus on the similarity of sessions to extract potential next items. Several works extend the session-based nearest-neighbors method with additional factors such as positions, recency, and popularity [6, 16, 24, 25]. Other works [17, 20, 21, 34] model item-to-item transitions using neural networks, possibly incorporating item features [3, 15, 43].

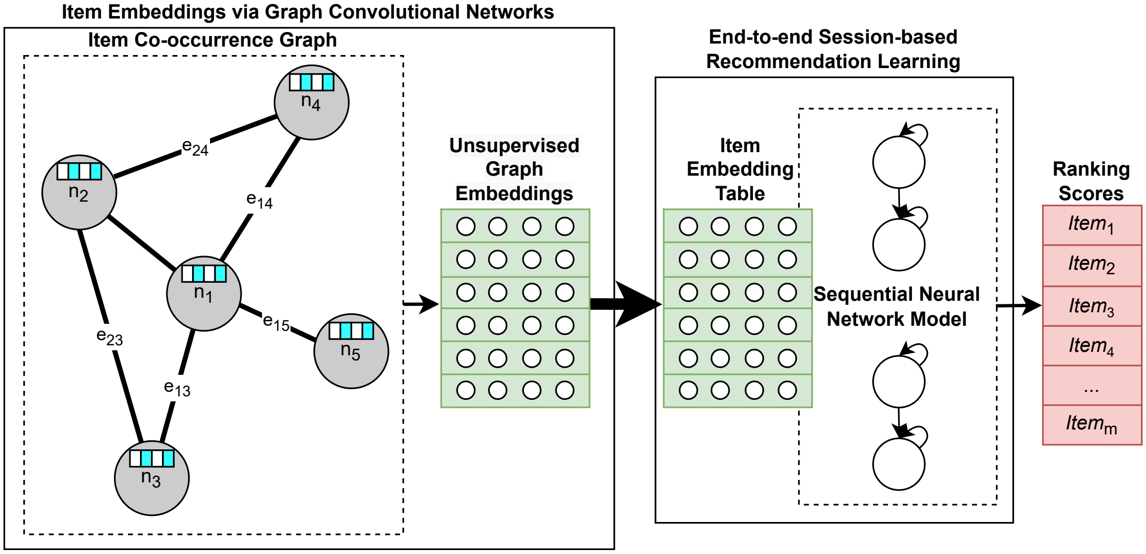
Recent works exploit the graph-based representation of sessions for improved recommendations. Current state-of-the-art use GNNs—in combination with attention or self-attention modules—to capture complex transitions and rich local dependencies [41, 42]. Further approaches enrich the graph topology with knowledge base entities [1, 40]. Gwadabe and Liu [9] use an item co-occurrence graph to generate session co-occurrence representations which are combined with the local and global preferences of users.

In contrast to models that integrate item features by extending the network with additional paths, GCNext extracts item embeddings from the item co-occurrence graph, in which content-based features are attached to each node. GCNext can be added to different sequential models without modifying their architecture, essentially using it in a plug-in fashion. Compared to already existing graph-based pretraining schemes [22, 26] for general recommendation, our approach specifically tackles the task of sequential and session-based recommendations.

3 GRAPH CONVOLUTIONAL NETWORK EXTENSION (GCNEXT)

In this section, we present the proposed GCNext approach. An overview of GCNext applied to sequential neural network models is shown in Figure 1. The first component represents the item co-occurrence graph from which we extract corresponding node embeddings by applying an unsupervised learning method. We subsequently use these embeddings to initialize the item embedding table of the underlying end-to-end sequential model, which learns session-based recommendations. Furthermore, we show how GCNext can also be employed for nearest–neighbor methods.

Fig. 1. Overview of the graph-based generation of item embeddings and its application in sequential neural network models.



3.1 Unsupervised Graph Embeddings

In contrast to other graph-based models [41, 42], we generate the item-item graph by extracting item co-occurrences in sessions. Each item is modeled as a node with the item's features as descriptors. For two nodes n_1 and n_2, an edge between the nodes represents the co-occurrence of two items i_1 and i_2 in a session. Edges are undirected, which prevents sequential information from being modeled in the graph. Each edge is weighted by $\mathbf{e}_{ij}$ which denotes the normalized number of co-occurrences (in all sessions) of the two items it connects. The learning of item-item transitions is solely the task of the underlying sequential model and the item-item graph embedding only incorporates the similarity to other items. We strictly rely on inductive representations for new items to obviate any data leakage into the embeddings.

The item catalog can contain millions of unique items which strongly impacts the scalability of the generated item-item graph. Therefore, we apply Bootstrapped Graph Latents (BGRL) [37] that can be scaled up to graphs with hundreds of millions of nodes and reduce the memory requirement substantially by enriching the original graph with simple augmentations to produce two different, but semantically similar views. Two encoders then generate online and target embeddings. The online embedding is used as input to a predictor which forms a prediction for the target embedding. The cosine similarity of the predictor output and the generated target embedding by the encoder is the final objective. Our graph embedding in GCNext is based on the BGRL learning method with our custom encoder architecture. We use the attentional convolution as introduced in [38] and optimized in [2] as *GATv2*. The node-wise formulation of this graph operation is defined as:

$$\mathbf{h}_i^{\prime} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \cdot \mathbf{W}\mathbf{h}_j\right), \tag{1}$$

where $\mathbf{h}$ denotes the node features, $\mathbf{W}$ the linear transformation's weight matrix and the average over the neighbor nodes features is weighted by the normalized attention weights α_{ij}. The computation of the normalized attention coefficients with softmax including edge weights can be seen in:

$$\alpha_{ij} = \frac{\exp\left(\mathbf{a}^{\top}\text{LeakyReLU}\left(\mathbf{W} \cdot [\mathbf{h}_i \,\|\, \mathbf{h}_j \,\|\, \mathbf{e}_{ij}]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\mathbf{a}^{\top}\text{LeakyReLU}\left(\mathbf{W} \cdot [\mathbf{h}_i \,\|\, \mathbf{h}_k \,\|\, \mathbf{e}_{ik}]\right)\right)}, \tag{2}$$

where $\mathbf{W}$ is a learnable weight matrix, $\mathbf{h}$ again denotes the corresponding node features, $\mathbf{e}_{ij}$ is the edge weight from node n_i to neighboring node n_j and $\|$ denotes vector concatenation.

Along the lines of other encoder architectures [37, 39], we stack multiple graph convolutional layers using skip connections and PReLU [12] as an activation function. The second layer of the encoder performs the computation:

$$\mathbf{H_2} = \sigma(\text{GATv2Conv}(\mathbf{H_1} + \mathbf{X}\mathbf{W}_{skip}, \mathbf{A})), \tag{3}$$

where $\mathbf{H_1}$ is the output of the previous layer, $\mathbf{X}$ are the input node features, $\mathbf{W}_{skip}$ is a learnable projection matrix for the skip connection, and $\mathbf{A}$ denotes the adjacency matrix.

Since large graphs, i. e., our proposed item co-occurrence graph, generally do not fit into GPU memory, we rely on a sub-graph sampling approach based on neighborhood batch sampling in [10], subsampling a fine-tuned, fixed-sized neighborhood per node.

3.2 Extension of Sequential Models

In sequential neural network models, next items are predicted by multiplying the candidate item embeddings with the learned session representation and applying the softmax operation to obtain the corresponding item probabilities.

To combine the advantage of graph-based item embeddings and current state-of-the-art models in session-based recommendation, we propose the following approach: Instead of initializing the sequential model's item embedding table with weights based on sophisticated initialization methods (for instance, the widely used Xavier initialization [7]), we directly adopt the graph-generated item embeddings. Thereby, GCNext improves the learning process of the underlying sequential model as these embeddings capture topological knowledge about the item-item relations and contain additional item feature information.

We train the graph-based item embeddings in an unsupervised manner. Compared to end-to-end embedding learning, this has two advantages: First, GCNext is a modular method that can easily be applied to different sequential models without altering their architectures. Second, splitting up the training process into two stages supports the usage in production, since the training of sequential models with the pre-trained item embeddings converges faster.

In contrast to neural network models, nearest–neighbor methods do not make use of an item embedding table; they find similar sessions based on the input sequence to predict the next candidate item. To use GCNext in nearest–neighbor approaches, we propose using item graph embeddings to find similar session neighbors. To integrate item graph embeddings in nearest–neighbor methods, we compute the similarity of sessions based on the cosine distance of the embedding of each item in the input session to every item in the candidate session. A threshold value on the distance is adapted to find similar embeddings. Candidate sessions are then scored by the corresponding position mapping of the similar items based on the nearest–neighbor method or again by their cosine similarity; as defined by the r-score in Equation 4 [25].

$$\mathrm{r}(S^{(\mathrm{i})}, S^{(\mathrm{c})}) = \frac{|T_{\mathrm{i,c}}|}{\sqrt{|S^{(\mathrm{i})}|}\sqrt{|S^{(\mathrm{c})}|}}, \tag{4}$$

where $S^{(\mathrm{i})}$ and $S^{(\mathrm{c})}$ refer to the sets of items in input and candidate session, respectively, and $T_{\mathrm{i,c}}$ represents the set of (pairs of) items in the input and candidate session with embedding similarity below the threshold.

4 EXPERIMENTAL SETUP

4.1 Datasets and Preprocessing

To evaluate GCNext, we conduct experiments on three widely used datasets with different characteristics from the e-commerce and music domains. The **Diginetica**[1] dataset (CIKM Cup 2016) provides different item features; we use the category and price of each item as features (side information) [20, 21]. The **Tmall**[2] dataset as part of the IJCAI-15 competition contains users' shopping logs along with the category, brand, and seller as additional item features [36, 43]. Furthermore, we evaluate GCNext on the **Music4All+** dataset, a version of the Music4All[3] dataset [32] with 11 item features. We enrich this dataset with i-vectors [4] of dimensionality 100 based on the 13-dimensional Mel-Frequency Cepstral Coefficients of the songs and a Gaussian Mixture Model with 256 components [5], using the `kaldi` toolkit [29]. Similar to [32], we consider listening events to belong to the same session if there are no gaps of more than 30 minutes between them.

Following previous works [21, 41], we filter out items appearing less than 5 times and ignore sessions consisting of a single interaction. Additionally, training sequences are generated by splitting the input sequence into smaller sub-sequences. Consider, for example, the input sequence $s = [i_1, i_2, ..., i_n]$, then the generated sequences and corresponding next items are $([i_1], i_2), ([i_1, i_2], i_3), ..., ([i_1, i_2, ...i_{n-1}], i_n)$. The maximum sequence length is set to 50. Each dataset is

[1] https://cikm2016.cs.iupui.edu/cikm-cup/
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[3] https://sites.google.com/view/contact4music4all

sorted by its timestamps and temporally split into training, validation, and test set (80%, 10%, and 10%). Table 1 provides an overview of the datasets.

Table 1. Dataset Statistics: Number of items, features, sessions, and average session length.

Dataset	Items	Feat.	Sessions	Avg. Length
Diginetica	44,527	2	205,698	4.85
Tmall	97,259	3	188,113	8.11
Music4All+	80,471	12	601,858	7.70

4.2 Base Algorithms and Implementation

One of the main advantages of GCNext is that it can be added to sequential models in a plug-in fashion to boost their performance. Therefore, we evaluate the following base algorithms for session-based recommendation with and without adding GCNext: SKNN [16], STAN [6], V–SKNN [24], and VSTAN [25] as representative nearest–neighbor methods; GRU4Rec+ [34], Caser [36], NARM [20], STAMP [21], and SASRec [17] as state-of-the-art neural network models. In addition, we compare GCNext to current graph-based approaches (SR–GNN [41], GCSAN ,[42] and LightGCN [13]), and models including additional item features: GRU4RecF [15] and FDSA [43]. We additionally implement the graph-based approaches to incorporate the original item features in their initial embedding tables with naive sum-pooling, which are referred to as SR–GNNF, GCSANF, and LightGCNF.

All base models use the implementation in *RecBole* [44] for neural network methods and *session-rec* [25] for nearest-neighbor methods. For the BGRL implementation, we rely on the code given in [37]. We use AdamW [23] to optimize item graph embeddings and Adam [18] in the sequential model training. The embedding size is fixed to 128 for comparability and all models performed best with this configuration according to preliminary experiments. We conduct hyperparameter optimization via grid search including the learning rate, number of layers and heads, layer sizes, and dropout rates for augmentation. Each experiment is repeated five times and the average results are reported. We provide our implementation on Github[4].

We adopt two widely used evaluation metrics to assess the quality of the recommendation lists: *HR@k* (Hit Rate) and *MRR@k* (Mean Reciprocal Rank). Similar to previous works [3, 21, 31], each metric is computed with k set to 10 and 20.

5 RESULTS AND ANALYSIS

Table 2 presents the experimental results of all base methods and their performance when extended with the proposed GCNext approach (column GCNext). When integrating GCNext into the nearest–neighbor methods (top part of Table 2), we observe a significant performance increase for some base models, and no significant decrease for any of the base models. In particular, V-SKNN's performance improves by 0.88% to 6.33% over all datasets on the *HR@10* score. Although the metrics show that nearest–neighbor methods are able to keep up with certain neural network approaches, they lack in state-of-the art performance across all three datasets.

The effectiveness of our approach is distinctly indicated by the results of neural network approaches in session-based recommendation. Compared to all neural network methods, Caser, which uses convolution-based sequence embeddings, and STAMP, a complete attention-based method, benefit the most by incorporating GCNext across all three datasets.

[4] https://github.com/dbis-uibk/gcnext

On the Diginetica dataset GCNext combined with SASRec, a transformer-based model, achieves the highest score on each metric throughout. This effect can also be seen with the Tmall dataset, where SASRec with GCNext increases the state-of-the-art *HR@10* score by 1.12% and significantly outperforms feature-based methods such as GRU4RecF and FDSA. STAMP extended with GCNext even achieves an increase in performance of 12.79% on the *MRR@20.* Interestingly, this effect becomes less impactful for the Music4All+ dataset which provides a large set of additional item features. On this dataset the graph-based self-attention network GCSAN extended with item features achieves the highest *MRR* scores. Nonetheless, the evaluation shows significantly improved scores for each of the neural network-based models on the Music4all+ dataset.

It is also noteworthy that neural network models trained with pre-trained item graph embeddings converge faster than their corresponding standard initialized counterparts. We assume this is due to the enriched information contained in the item embeddings.

Table 2. Model performances on all datasets; column GCNext indicates the use of GCNext. Significant improvements over the underlying sequential models (paired t-test, $p < .05$) are marked with †; best results in bold, second-best results underlined.

Model	GCNext	Diginetica				Tmall				Music4All+			
		MRR		HR		MRR		HR		MRR		HR	
		@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
Nearest–neighbor Methods													
SKNN	✗	17.98	18.62	36.67	45.94	22.13	22.41	32.32	36.30	19.49	19.89	42.83	48.45
STAN	✗	15.43	16.12	28.95	38.97	20.13	20.41	26.15	30.20	18.48	18.85	37.56	42.78
V-SKNN	✗	17.99	18.63	36.59	45.84	21.84	22.09	30.95	34.55	19.58	19.99	43.06	48.68
VSTAN	✗	15.27	15.97	28.64	38.73	20.04	20.32	26.01	30.04	18.41	18.77	36.99	42.18
SKNN	✓	18.07†	18.71†	37.11†	46.40†	22.14	22.40	32.32	36.31	19.43	19.85	43.10†	48.90†
STAN	✓	15.43	16.12	28.95	38.97	20.14	20.42	26.16	30.21	18.49	18.85	37.60†	42.80†
V-SKNN	✓	18.18†	18.83†	37.23†	46.47†	22.38†	22.67†	32.91†	36.95†	19.66†	20.07†	43.44†	49.20†
VSTAN	✓	15.27	15.97	28.64	38.73	20.04	20.32	26.01	30.05	18.56†	18.92†	37.41†	42.57†
Neural Network Methods													
GRU4Rec+	✗	17.09	17.94	38.19	50.45	28.62	29.00	45.71	51.02	25.18	25.62	41.35	47.53
Caser	✗	14.29	14.82	26.54	34.20	24.30	24.54	35.61	39.04	19.26	19.62	31.51	36.63
STAMP	✗	16.38	17.17	35.62	47.04	21.63	21.88	31.62	35.21	28.12	28.45	41.96	46.60
NARM	✗	17.35	18.18	38.54	50.57	28.11	28.46	44.58	49.71	28.82	29.20	42.64	48.00
SASRec	✗	19.88	20.73	43.09	55.24	29.46	29.84	**47.72**	53.15	28.83	29.23	45.25	51.01
GRU4Rec+	✓	17.43†	18.27†	38.59†	50.78†	28.74†	29.12†	46.11†	51.60†	28.78†	29.18†	43.33†	48.99†
Caser	✓	15.63†	16.28†	30.60†	39.99†	26.33†	26.68†	41.15†	46.12†	20.56†	20.92†	33.35†	38.54†
STAMP	✓	17.43†	18.23†	37.50†	49.08†	24.33†	24.68†	37.75†	42.82†	28.60†	28.95†	42.81†	47.87†
NARM	✓	17.89†	18.73†	39.51†	51.73†	28.99†	29.39†	46.48†	52.19†	28.89†	29.28†	43.11†	48.67†
SASRec	✓	**19.97**†	**20.80**†	**43.10**	**55.41**†	**29.51**†	**29.93**†	47.70	**53.75**†	29.84†	30.15†	**45.43**†	**51.13**†
Feature & Graph-based Methods													
GRU4RecF	✗	16.04	16.91	36.35	48.77	25.25	25.66	42.04	47.97	28.67	29.04	42.52	47.80
FDSA	✗	18.92	19.79	41.29	53.73	28.76	29.15	46.48	52.01	30.04	30.43	45.20	50.66
SR-GNN	✗	17.75	18.58	39.18	51.23	27.47	27.84	44.67	49.89	28.90	29.27	42.62	47.91
SR-GNNF	✗	17.49	18.33	38.45	50.56	25.55	25.98	41.52	47.70	28.72	29.08	41.98	47.14
GCSAN	✗	19.20	20.03	41.00	53.01	29.01	29.41	47.37	53.04	29.68	30.05	43.79	49.09
GCSANF	✗	17.21	18.04	37.60	49.57	25.16	25.58	40.60	46.67	**30.13**	**30.49**	43.74	48.84
LightGCN	✗	15.90	16.67	35.11	46.29	26.03	26.47	45.92	52.16	8.43	9.01	21.98	30.18
LightGCNF	✗	15.90	16.66	34.96	45.92	25.80	26.27	45.49	52.25	10.95	11.59	28.49	37.45

6 CONCLUSION AND FUTURE WORK

We proposed GCNext, an extension to sequential recommendation models based on GCNs. In the first phase, we generate an item co-occurrence graph with nodes/items enriched with item descriptors to learn its node representations in an unsupervised manner. In the second phase, we use the learned item embedding weights to initialize the item embedding table of the underlying sequential base model. Our experimental results on three different datasets show the effectiveness of GCNext.

For future work, we plan to further investigate the potential improvements of graph-based item embeddings in cold-start scenarios for session-based recommendation. Additionally, future experiments will compare our approach with different graph embeddings methods and investigate the diverging impact on baseline models.

ACKNOWLEDGMENTS

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) [P33526]. For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] Mehrnaz Amjadi, Seyed Danial Mohseni Taheri, and Theja Tulabandhula. 2021. KATRec: Knowledge Aware aTtentive Sequential Recommendations. In *Discovery Science - 24th International Conference, DS 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12986).* Springer, 305–320.

[2] Shaked Brody, Uri Alon, and Eran Yahav. 2021. How Attentive are Graph Attention Networks? *CoRR* abs/2105.14491 (2021). arXiv:2105.14491 https://arxiv.org/abs/2105.14491

[3] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems.* ACM, 143–153.

[4] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 4 (2010), 788–798.

[5] Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, and Gerhard Widmer. 2015. I-Vectors for Timbre-Based Music Similarity and Music Artist Classification. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015.* 554–560.

[6] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and Time Aware Neighborhood for Session-based Recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019.* ACM, 1069–1072.

[7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010 (JMLR Proceedings, Vol. 9).* JMLR.org, 249–256.

[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016.* ACM, 855–864.

[9] Tajuddeen Rabiu Gwadabe and Ying Liu. 2022. IC-GAR: item co-occurrence graph augmented session-based recommendation. *Neural Computing and Applications* (2022), 1–16.

[10] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017.* 1024–1034.

[11] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020 (Proceedings of Machine Learning Research, Vol. 119).* PMLR, 4116–4126.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015.* IEEE Computer Society, 1026–1034.

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings.*

[15] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems 2016.* ACM, 241–248.

[16] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*. ACM, 306–310.

[17] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018*. IEEE Computer Society, 197–206.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

[19] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

[20] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017*. ACM, 1419–1428.

[21] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. 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, KDD 2018*. ACM, 1831–1839.

[22] Zhuang Liu, Yunpu Ma, Matthias Schubert, Yuanxin Ouyang, and Zhang Xiong. 2022. Multi-Modal Contrastive Pre-training for Recommendation. In *ICMR '22: International Conference on Multimedia Retrieval*. ACM, 99–108.

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019*.

[24] Malte Ludewig, Iman Kamehkhosh, Nick Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018*. ACM, 3:1–3:6.

[25] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical analysis of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 31, 1 (2021), 149–181.

[26] Zaiqiao Meng, Siwei Liu, Craig Macdonald, and Iadh Ounis. 2021. Graph Neural Pre-training for Enhancing Recommendations using Side Information. *CoRR* abs/2107.03936 (2021).

[27] Zhiqiang Pan, Wanyu Chen, and Honghui Chen. 2021. Dynamic Graph Learning for Session-Based Recommendation. *Mathematics* 9, 12 (2021).

[28] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*. ACM, 701–710.

[29] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

[30] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36.

[31] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*. AAAI Press, 4806–4813.

[32] Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Gabiato Catharin, Rafael Biazus Mangolin, Yandre Maldonado e Gomes da Costa, Valéria Delisandra Feltrim, and Marcos Aurélio Domingues. 2020. Music4All: A New Music Database and Its Applications. In *2020 International Conference on Systems, Signals and Image Processing, IWSSIP 2020*. IEEE, 399–404.

[33] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*. ACM, 1441–1450.

[34] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016*. ACM, 17–22.

[35] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*. ACM, 1067–1077.

[36] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 565–573.

[37] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Remi Munos, Petar Veličković, and Michal Valko. 2021. Bootstrapped Representation Learning on Graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*.

[38] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*. OpenReview.net.

[39] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

[40] Baocheng Wang and Wentao Cai. 2020. Knowledge-Enhanced Graph Neural Networks for Sequential Recommendation. *Inf.* 11, 8 (2020), 388.

[41] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01, 346–353.

[42] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence,*

IJCAI 2019. ijcai.org, 3940–3946.

[43] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, 4320–4326.

[44] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*. ACM, 4653–4664.