

# GenRec: Generative Sequential Recommendation with Large Language Models

Panfeng Cao  
 panfengc@umich.edu  
 University of Michigan  
 Ann Arbor, USA

Pietro Liò  
 pl219@cam.ac.uk

## ABSTRACT

Sequential recommendation is a task to capture hidden user preferences from historical user item interaction data and recommend next items for the user. Significant progress has been made in this domain by leveraging classification based learning methods. Inspired by the recent paradigm of “pretrain, prompt and predict” in NLP, we consider sequential recommendation as a sequence to sequence generation task and propose a novel model named Generative Recommendation (GenRec). Unlike classification based models that learn explicit user and item representations, GenRec utilizes the sequence modeling capability of Transformer and adopts the masked item prediction objective to effectively learn the hidden bidirectional sequential patterns. Different from existing generative sequential recommendation models, GenRec does not rely on manually designed hard prompts. The input to GenRec is textual user item sequence and the output is top ranked next items. Moreover, GenRec is lightweight and requires only a few hours to train effectively in low-resource settings, making it highly applicable to real-world scenarios and helping to democratize large language models in the sequential recommendation domain. Our extensive experiments have demonstrated that GenRec generalizes on various public real-world datasets and achieves state-of-the-art results. Our experiments also validate the effectiveness of the proposed masked item prediction objective that improves the model performance by a large margin.

## CCS CONCEPTS

- **Information systems** → **Recommender systems; Users and interactive retrieval;**
- **Computing methodologies** → **Natural language generation.**

## KEYWORDS

Sequential Recommendation, Generative Recommender Systems, Large Language Models

## ACM Reference Format:

Panfeng Cao and Pietro Liò. 2018. GenRec: Generative Sequential Recommendation with Large Language Models. In *Proceedings of Make sure to*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

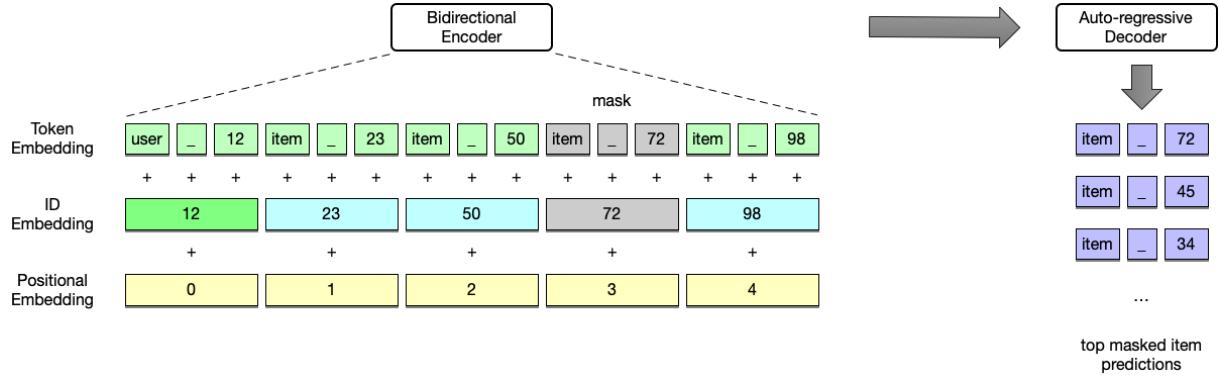
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/XXXXXX.XXXXXXX>

enter the correct conference title from your rights confirmation email (Conference acronym 'XX). ACM, New York, NY, USA, 6 pages. <https://doi.org/XXXXXX.XXXXXXX>

## 1 INTRODUCTION

Recent years have witnessed the great success of recommender systems in online platforms such as Amazon and Yelp, which help people make micro decisions and fulfil their demands in daily life. Within online platforms, historical sequential user interaction data is utilized to capture the evolved and dynamic user behaviors and make appropriate recommendations of next items for the user. Different from traditional recommender systems that treat each user behavior as a sample and directly model the user preference on a single item, sequential recommendation learns timestamp-aware sequential patterns to recommend the next item [7]. For example, one might purchase peripheral accessories after buying a desktop. Various methods have been proposed to accurately model sequential user behaviors. [6, 10, 11, 22, 23] employ Recurrent Neural Networks (RNN) to encode the left-to-right context. [20] utilizes Convolutional Neural Networks (CNN) to learn sequential patterns as local features. [2] leverages Graph Neural Networks (GNN) to construct the item-item interest graphs and user interest sequences. [27] designs contrastive loss to maximize the mutual information between different views of the sequential data. Contextual information such as item attributes is incorporated as self-supervised signals to learn user and item representations. The recent success of Transformer [21] based large language models (LLMs) in various NLP tasks has inspired the research of utilizing LLMs in recommender systems [9, 13, 19]. [13] utilizes the left-to-right self-attention mechanism to identify next relevant items from the sequential interaction history and [19] further improves the performance by employing the bidirectional self-attention mechanism. Although both methods are effective, they do not consider personalization and the model performance is potentially limited. Generative LLMs based recommender system is considered a promising approach [15]. [9] is the first model to unify different recommendation tasks in the same framework to facilitate the knowledge transfer and it is pretrained across task-specific datasets to capture the deep semantics for personalization and recommendation. However, different modality information is encoded in the textual format of natural language, which might cause suboptimal modality representation. Moreover, [9] relies on manually designed task-specific hard prompts to formulate the problem as question answering, which requires additional data processing and prompt search.

To address the mentioned limitations, we propose GenRec, a novel generative framework for personalized sequential recommendation. GenRec utilizes the encoder-decoder Transformer as the



**Figure 1: An illustration of the architecture of GenRec.** The input textual user item interaction sequence is first tokenized into a sequence of tokens. Token embedding, ID embedding and positional embedding are summed up to produce the bidirectional encoder input. In pretraining and finetuning, a random item is masked and the auto-regressive decoder generates the masked item. In inference, the decoder generates top 20 masked item predictions to calculate the evaluation metrics.

backbone and formulates sequential recommendation as a sequence-to-sequence generation task. We utilize the cloze task [5] as the training objective and pretrain the model on the corpus to learn the bidirectional sequential patterns. Inspired by [19], to make the downstream sequential recommendation task consistent with the pretraining cloze task, we append the [MASK] token at the end of the input sequence to predict the next item. Extensive experiments on public real-world datasets demonstrate that GenRec generalizes effectively and achieves state-of-the-art performance across all datasets. In conclusion, our contributions include:

- We propose GenRec, a novel generative model for personalized sequential recommendation that can generate next items auto-regressively without prompts.
- As far as our knowledge, it is the first generative sequential recommendation method that adopts the cloze objectives in both pretraining and finetuning. Our method only utilizes task specific datasets and does not rely on additional large pretraining corpus<sup>1</sup>.
- We conduct experiments on three public real-world datasets, demonstrating consistent improvements on multiple evaluation metrics compared with baseline methods.

## 2 RELATED WORK

Transformer based LLMs have achieved remarkable success on various research fields such as text summarization, question answering, document understanding, etc. Typically, LLMs are trained on a vast amount of textual corpus from diverse sources including wikipedia, news, articles and books. LLMs can have emergent zero-shot learning capability as the model parameter size scales up with large training datasets [26], making LLMs better generalize to unseen domains. Recent efforts have been made to explore the potential of LLMs as recommender systems. [9] employs T5 as the model backbone and achieves competitive performance on a variety of recommendation tasks, demonstrating the generalization capability

of LLMs based recommender systems. [4] develops a unified foundation model based on M6 and reduces the model size and training time by utilizing prompt tuning. Similar to [9], user interaction data is represented as plain texts and recommendation tasks are formulated as either natural language understanding or generation. [8] leverages ChatGPT to build a conversational recommender system to improve the interactivity and explainability of the recommendation process. [24] finetunes T5 with user-personalized instruction data, which is generated from manually designed templates, enabling users to communicate with the system with natural language instructions. [1] proposes an efficient framework to finetune LLMs for recommendation tasks and demonstrates significant improvements in the domains of movie and book recommendations. [3] employs a funneling approach where it first retrieves the candidates utilizing the user item interactions and then generates recommended items from candidates with a LLMs based framework.

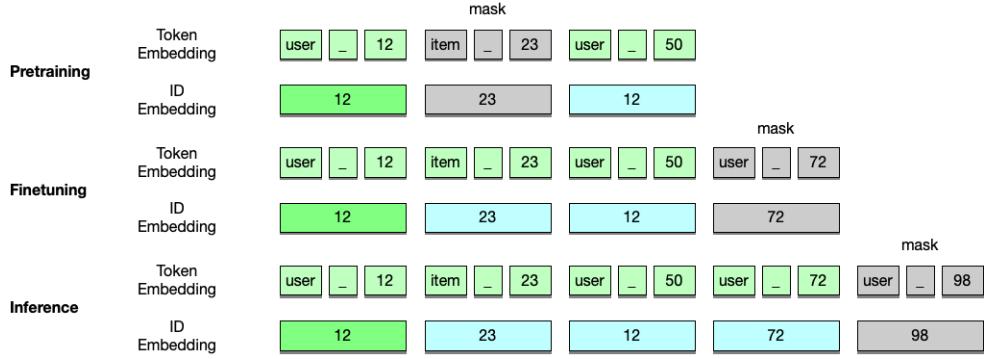
## 3 TASK FORMULATION

The goal of sequential recommendation is to predict the next item that the user is most likely to interact with given the historical item interaction sequence of the user. Formally,  $i_{th}$  user  $u_i$  has item interaction sequence  $t_i = \{t_1^i, \dots, t_{n_i}^i\}$  ordered chronologically.  $t_j^i$  and  $n_i$  denotes the  $j_{th}$  item in the sequence and the length of  $t_i$  respectively.  $t_{j:k}^i$  denotes the subsequence, i.e.,  $t_{j:k}^i = \{t_j^i, t_{j+1}^i, \dots, t_k^i\}$ , where  $1 \leq j < k \leq n_i$ . Given the user item sequence  $\{u_i, t_{1:n_i}^i\}$ , the task of sequential recommendation is to predict the next item that  $u_i$  is most likely to interact with at the  $n_i + 1$  timestep.

## 4 METHODOLOGY

The overall architecture of GenRec is shown in Figure 1. Our GenRec is established upon a sequence-to-sequence Transformer encoder-decoder framework. The *encoder* of GenRec embeds cross modal user and item features from the input sequence and the *decoder* generates next items auto-regressively. In the following sections,

<sup>1</sup>Our source code, datasets and pretrained models are publicly available at <https://github.com/caop-kie/GenRec>.



**Figure 2: An illustration of different masking mechanisms in pretraining, finetuning and inference. In pretraining, a random item in the sequence is masked while in finetuning and inference, masked items are appended to the end of the sequence. Note, the last two items in the user item interaction sequence are excluded in pretraining to avoid data leakage. Similarly, the last one item in the sequence is excluded in finetuning.**

we elaborate the feature embedding and learning objectives of pretraining and finetuning.

## 4.1 Model

The feature embedding consists of token embedding, positional embedding, user ID embedding and item ID embedding. The input sequence to the model  $\{u_i, t_{1:n_i}^i\}$  is tokenized into a sequence of tokens, which is wrapped around with the start indicator token [BEG] and the end indicator token [END]. Extra [PAD] tokens are appended to the end to unify the sequence length in the batch. The token sequence  $S$  is represented as:

$$S = [\text{BEG}], \text{Tok}(u_i), \text{Tok}(t_1^i), \dots, \text{Tok}(t_{n_i}^i), [\text{END}], \dots, [\text{PAD}], \quad (1)$$

where  $\text{Tok}$  is the tokenizer function. For the  $j$ th token in  $S$ , we apply textual token embedding, positional embedding and user or item ID embedding. The textual token embedding is formulated as  $\text{EMB}(S_j)$ , where  $\text{EMB}$  is the token embedding function. The positional information of tokens in the sequence is captured by the positional embedding  $\text{PosEMB}(j)$ , where  $\text{PosEMB}$  is the 1D positional embedding function. And we utilize the user ID and item ID embedding to add personalized information of users and items to the sequence. Personalized information helps reveal the sequential patterns of user behaviors. For example, a user that previously viewed a Samsung tablet is more likely to continue viewing electronic products from Samsung or other brands. Different from P5 [9], which apply whole-word embeddings shared by both user and item tokens. Our ID embedding layers are not shared to better capture modality specific features. The user ID embedding is represented by  $\text{UIDEMB}(\text{UID}(j))$  and the item ID embedding is represented by  $\text{TIDEMB}(\text{TID}(j))$ .  $\text{UIDEMB}$  and  $\text{TIDEMB}$  are the ID embedding functions for users and items respectively. Since a word in the sequence can be split into multiple tokens after tokenization, we utilize  $\text{UID}(j)$  or  $\text{TID}(j)$  to retrieve the user ID or item ID for the  $j$ th token. For example, **item\_1234** is split into 4 tokens i.e., **item**, **\_**, **12** and **34**. Those tokens share the same item ID **1234** and embedding. Note only user tokens have user ID embeddings and only item tokens have item ID embeddings. All the aforementioned cross modal embeddings are added element-wisely to produce the final

embedding  $X$ . Specifically the  $j$ th token cross modal embedding is formulated as:

$$\begin{aligned} X_j &= \text{EMB}(S_j) + \text{PosEMB}(j) + \text{IDEMB}(\text{ID}(j)), \quad j \in [0, n_i], \\ \text{IDEMB} &\in \{\text{UIDEMB}, \text{TIDEMB}\}, \\ \text{ID} &\in \{\text{UID}, \text{TID}\} \end{aligned} \quad (2)$$

The decoder input is the next item that serves as the learning target. The same tokenizer and textual embedding in the encoder are utilized. Following baseline methods, the model generates top 20 items with beam search during inference to calculate the evaluation metrics.

## 4.2 Pretraining

We pretrain GenRec with the masked sequence modeling task to deeply fuse the personalized sequential patterns. Given a user item sequence, we randomly sample an item, which is not necessarily the last item, from the sequence and replace it with the [MASK] token. The model is trained to generate the masked item in the decoder output. Cross-entropy loss is computed between the generated item and the masked item and minimized in pretraining. The masking process is also illustrated in Figure 2. Different from [19], where a number of items are masked, we only mask a single item in the sequence to unify the learning objectives of pretraining and finetuning. Our bidirectional masked sequence modeling task is based on the observation that the item sequence is not strictly ordered. For example, given similar items item\_1 and item\_2, user\_1 might interact with item\_1 and then item\_2 while user\_2 interacts with item\_2 and then item\_1. It is crucial to incorporate both left and right contexts to encode the user behavior [19]. To avoid the data leakage, we pretrain the model on the training split of datasets (See §6 for details about the training split).

## 4.3 Finetuning

In the finetuning stage, the learnt user and item sequential patterns in pretraining are utilized for the next item prediction task. As mentioned in section 4.2, finetuning has the same learning objective with pretraining for knowledge transfer. Following the masking

Models	Sports				Beauty				Yelp			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0151	0.0096	0.0253	0.0129
HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0186	0.0115	0.0326	0.0159
FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0158	0.0098	0.0276	0.0136
GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0152	0.0099	0.0263	0.0134
BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0051	0.0033	0.0090	0.0045
P5-S	0.0272	0.0169	0.0361	0.0198	0.0503	0.0370	0.0659	0.0421	0.0568	0.0402	0.0707	0.0447
P5-B	<u>0.0387</u>	<u>0.0312</u>	<u>0.0460</u>	<u>0.0336</u>	<u>0.0508</u>	<u>0.0379</u>	<b>0.0664</b>	<u>0.0429</u>	-	-	-	-
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0162	0.0100	0.0274	0.0136
S3-Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0201	0.0123	0.0341	0.0168
GenRec	<b>0.0397</b>	<b>0.0332</b>	<b>0.0462</b>	<b>0.0353</b>	<b>0.0515</b>	<b>0.0397</b>	0.0641	<b>0.0439</b>	<b>0.0627</b>	<b>0.0475</b>	<b>0.0724</b>	<b>0.0507</b>

Table 1: Performance comparison between GenRec and baselines on Sports, Beauty and Yelp datasets.

Dataset	Sports	Beauty	Yelp
# Users	35,598	22,363	30,431
# Items	18,357	12,101	20,033
Avg. Items / User	8.3	8.9	10.4
Avg. Users / Item	16.1	16.4	15.8

Table 2: Statistics of the datasets.

mechanisms in Figure 2, we prepare the dataset in the same format as in pretraining by appending the [MASK] token to the end of the token sequence. The input to the model is the masked user item sequence and the output is the predicted next item.

## 5 EXPERIMENTS

### 6 DATASETS

Amazon Sports and Amazon Beauty datasets are obtained from Amazon review datasets [18], which are collected from Amazon.com online platform. Yelp dataset is a large business recommendation dataset. Following [9, 27], we use transactions between January 1, 2019 and December 31, 2019. For all datasets, item sequences are extracted per user and sorted ascendingly by the interaction timestamps. Following the same data preprocessing in the baseline methods, unpopular items and inactive users with less than five interaction records are filtered. The statistics of the datasets after preprocessing are summarized in Table 2.

To generate the training, validation and test splits of datasets, we apply the leave-one-out strategy following [9, 13, 19, 27]. In each user item sequence, we use the last item as the test data, the item before the last item as the validation data, and the remaining item sequence as the training data.

### 7 BASELINES

We compare the performance of GenRec with several competitive sequential recommendation baselines.

Caser [20] is a CNN based method, which utilizes both horizontal and vertical convolutional filters to capture high-order Markov chains for sequential recommendation.

FDSA [25] models the feature sequence with the self attention module and captures the feature transition patterns.

GRU4Rec [12] applies GRU to model the user click history sequence for session based recommendation.

HGN [17] adopts hierarchical gating networks to learn both long-term and short-term user behaviors.

Bert4Rec [19] learns the bidirectional item representation by masked language modeling for sequential recommendation.

SASRec [13] applies the self attention mechanism to model the user sequential behaviors.

S3Rec [27] designs self-supervised learning objectives to capture the correlations of items and attributes for sequential recommendation.

P5 [9] is a pretrained language model for recommendation and unifies different recommendation tasks by formulating them in natural language with prompts.

All the baselines except P5 are classification based.

### 7.1 Implementation Details

The model weights of GenRec is initialized from the pretrained BART [14] base model, which consists of a 6-layer Transformer encoder and a 6-layer Transformer decoder. There are 12 attention heads for both encoder and decoder and the model dimension is 768. The total number of model parameters is 184 million. We use the pretrained BART tokenizer for tokenization. GenRec is trained with an AdamW optimizer [16], in which 1e-5 is set as the learning rate and weight decay. The maximum length of the input tokens is set to 512. First 5% of iterations are used as the warmup stage. The beam size is set to 20 in inference. GenRec is pretrained for about an hour and finetuned for 25 epochs on one NVIDIA RTX 3090 GPU.

### 7.2 Evaluation Metrics

To evaluate the model performance, we employ top-k Hit Ratio (HR@k) and Normalized Discounted Cumulative Gain (NDCG@k) and report the the results of HR@{1, 5, 10} and NDCG@{5, 10} under the all-item setting, i.e. all items are possible candidates to be recommended as the next item. Our model is evaluated on Amazon Sports, Amazon Beauty and Yelp datasets. In the tables, **bold** numbers refer to the best scores, while underlined numbers refer to the second best scores.

Dataset	HR@5	NDCG@5	HR@10	NDCG@10
Sports	<b>0.0397</b>	<b>0.0332</b>	<b>0.0462</b>	<b>0.0353</b>
Sports w/o pretraining	0.0360	0.0286	0.0431	0.0310
Beauty	<b>0.0515</b>	<b>0.0397</b>	<b>0.0641</b>	<b>0.0439</b>
Beauty w/o pretraining	0.0422	0.0313	0.0548	0.0354
Yelp	<b>0.0627</b>	<b>0.0475</b>	<b>0.0724</b>	<b>0.0507</b>
Yelp w/o pretraining	0.0626	0.0469	0.0716	0.0499

**Table 3: Ablation study of our method on four datasets. Pre-training effectively improves the model performance across all datasets.**

### 7.3 Results and Analysis

This section provides experimental results and analyses of our model’s effectiveness. The performance metrics of the baselines are obtained from the public results in [9]. See Appendix 7 for more details about baselines. As is presented in Table 1, GenRec outperforms other baselines on all metrics on Sports and Yelp datasets. Although it is outperformed by P5 on the HR@10 metrics of the Beauty dataset, GenRec still achieves comparable performance with other baselines.

We speculate due to the smaller size of Beauty dataset, the sequential patterns are not learnt thoroughly in pretraining. Increasing the number of pretraining epochs could potentially improve the performance. Since we focus on low-resource efficient training in this work, we leave the improvement for future work. Since P5 is a unified model and trained using different types of recommendation tasks, it learns better user item representation with the knowledge transfer between tasks on smaller datasets. The advantage of our method compared with P5 is that our model is lightweight and the training only takes a few hours to complete. We do not leverage prompt engineering and no prompt search is required to find the best prompt.

## 8 ABLATION STUDY

To verify the effectiveness of the proposed masked sequence modeling task, we conduct the ablation study on Amazon Sports, Amazon Beauty and Yelp datasets. As is shown in Table 3, the model performance drops across all metrics without the masked sequence modeling task. It indicates the pretraining objective helps the model effectively capture the user behavior patterns and learn the user item representations. It is worth mentioning even without pretraining, GenRec still achieves comparable performance with P5 and outperforms other baselines, which proves the effectiveness of our method.

## 9 CONCLUSION

In this work, we propose GenRec, a novel sequence to sequence framework that generates personalized sequential recommendation. Compared with existing generative models, GenRec is lightweight and efficient and does not rely on prompt engineering to find the best prompt. By leveraging the Transformer architecture as the backbone, GenRec effectively learns the bidirectional sequential patterns with the attention mechanism and achieves state-of-the-art performance on various public datasets. Besides, our proposed

method is also flexible to integrate with other sequence to sequence language models and can be applied in other recommendation domains such as direct recommendation, which we leave for future work.

## REFERENCES

- [1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems* (Singapore, Singapore) (*RecSys '23*). Association for Computing Machinery, New York, NY, USA, 1007–1014. <https://doi.org/10.1145/3604915.3608857>
- [2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) (*SIGIR '21*). Association for Computing Machinery, New York, NY, USA, 378–387. <https://doi.org/10.1145/3404835.3462968>
- [3] Zheng Chen. 2023. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622* (2023).
- [4] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [6] Tim Donkers, Benedikt Loepf, and Jürgen Ziegler. 2017. Sequential User-Based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) (*RecSys '17*). Association for Computing Machinery, New York, NY, USA, 152–160. <https://doi.org/10.1145/3109859.3109877>
- [7] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhuan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. 2023. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Trans. Recomm. Syst.* 1, 1, Article 3 (mar 2023), 51 pages. <https://doi.org/10.1145/3568022>
- [8] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofei Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).
- [9] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.
- [10] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-Based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) (*CIKM '18*). Association for Computing Machinery, New York, NY, USA, 843–852. <https://doi.org/10.1145/3269206.3271761>
- [11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. *arXiv:1511.06939 [cs.LG]*
- [13] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206. <https://doi.org/10.1109/ICDM.2018.00035>
- [14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880. <https://doi.org/10.18653/v1/2020.acl-main.703>
- [15] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems. *arXiv preprint arXiv:2302.03735* (2023). <https://arxiv.org/abs/2302.03735>
- [16] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).
- [17] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 825–833.

[18] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) (SIGIR '15). Association for Computing Machinery, New York, NY, USA, 43–52. <https://doi.org/10.1145/2766462.2767755>

[19] 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* (Beijing, China) (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1441–1450. <https://doi.org/10.1145/3357384.3357895>

[20] 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* (Marina Del Rey, CA, USA) (WSDM '18). Association for Computing Machinery, New York, NY, USA, 565–573. <https://doi.org/10.1145/3159652.3159656>

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf)

[22] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) (WSDM '17). Association for Computing Machinery, New York, NY, USA, 495–503. <https://doi.org/10.1145/3018661.3018689>

[23] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Pisa, Italy) (SIGIR '16). Association for Computing Machinery, New York, NY, USA, 729–732. <https://doi.org/10.1145/2911451.2914683>

[24] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[25] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI* 4320–4326.

[26] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[27] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.

Received 10 May 2024