Exploiting Global Behavior Contextual Correlation in Sequential Recommendation Augmentation

Qian Yu[1], Xiangdong Wu[1], Chen Yang[1], Zihao Zhao[1], Haoxin Liu[2], Chaosheng Fan[1], Changping Peng[1], Zhangang Lin[1], Jinghe Hu[1] and Jingping Shao[1]

[1]*Marketing & Commercialization Center, JD.com*

[2]*Tsinghua University*

Abstract

The recently proposed Sequential Recommendation Augmentation (SRA) paradigm has shown valuable potential in sequential recommendation, especially for handling long-tail problem via extending short behavior sequences. However, the self-supervised SRA adopts autoregressive learning with fixed forward or backward direction, which cannot make full use of the contextual correlation information in the training behavior sequences. Due to the direction difference, discrepancy problem exists in the two training stages of SRA, i.e., pretraining and finetuning. In order to overcome the restriction of specific sequential learning direction, we propose to equip SRA with permutation autoregressive learning to extract global contextual correlation information from the behavior sequences in both directions. The adapted SRA method is implemented with two-stream self-attention. Empirical evaluations on multiple sequential recommendation benchmark datasets demonstrate the effectiveness of our proposed model, and the augmented data can significantly reduce the convergence rate.

Keywords

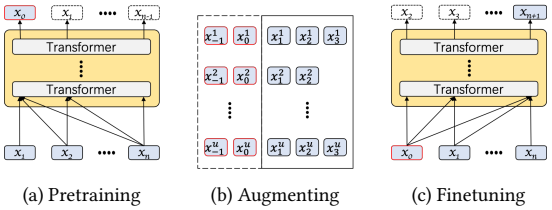
Sequential Recommendation, Data Augmentation, Permutation Autoregressive Learning

1. Introduction

Sequential recommendation aims to find the behavior pattern or item transition from the user behavior sequences. Variant architectures are developed, including Markov Chains [1], RNN [2], attention-based sequence models [3] and graph models [4], etc.

Data sparsity severely defects the performance of sequential recommendation. Data augmentation is a straightforward solution for handling short behavior sequences in sequential recommendation [5]. There are mainly two kinds of data augmentation methods, namely the heuristic augmentation [6] and generative augmentation [7]. Recently, Sequential Recommendation Augmentation (SRA) is proposed as an augmentation paradigm in sequential recommendation [7]. Consisting of two training stages, namely pretraining and finetuning, SRA is a verified effective solution for handling short sequences in sequential recommendation, namely long-tail problem.

However, the current learning procedure of sequential recommendation cannot completely extract the contextual correlation in the given sequential training instances. In any learning stage, the item is predicted given the subsequence located at the single side of it. Specifically, SRA pretrains the sequence model with reversed training sequence, in order to generate the pseudo-prior items, while the finetuning stage adopts the normal autoregressive objective. Therefore, the trained model is never aware of the bidirectional context behaviors around the current position, namely the learning of the framework is insufficient. Besides, the two stages update to the same set of parameters but with different learning directions. Similar discrepancy problems are commonly seen in this kind of pretrain-finetune methods, and it remains a constraint for further performance improvement.



(a) Pretraining (b) Augmenting (c) Finetuning

Figure 1: Stages in SRA. (a) Pretraining with inversed sequence. (b) Pseudo-prior item augmentation. (c) Finetuning.

To addressing the abovementioned problems, we propose to exploit global contextual correlation information with Permutation Autoregressive Learning (PAL) for SRA. Specifically, we unify the learning objectives with permutation language model objective and implement it on sequential recommendation with two-stream self-attention mechanism. PAL helps the model to exploit different permutations of the input in order to exploit global contextual information without restrictions of learning direction. For the inference stage, we try to adopt the

DL4SR'22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

yuqian81@jd.com (Q. Yu)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

beam search for generating more suitable subsequence as the augmented data. A latest revision named BiCAT [8] comes with a similar motivation by implemented via an additional loss regularization, but it is not designed for extracting contextual correlation information around the predicted position, and we will empirically compare them.

Our contributions can be summarized as following: (**a**) Global contextual correlation information is explored in Sequential Recommendation Augmentation (SRA). (**b**) Equipped with Permutation Autoregressive Learning and beam search method, an adapted SRA framework is designed and evaluated. (**c**) The proposed framework outperforms the state-of-the-art methods for sequential recommendation augmentation without extra information or heuristic rules.

2. Sequential Recommendation Augmentation

The sequential recommendation task can be regarded as the next-item prediction given the historical behavior sequence. We denote the user set as $\mathcal{U}$ and the item set as $\mathcal{X}$. The interaction behavior of the given user $u \in \mathcal{U}$ is denoted as $S^u = \{x_1^u, x_2^u, ..., x_n^u\}$. The sequential recommendation task can be formulated as:

$$x_{n+1}^u = \arg\max_x p(x|S^u) \tag{1}$$

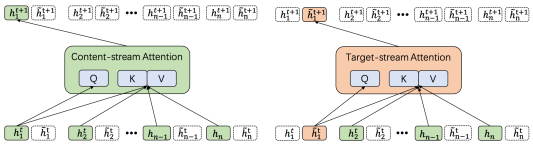
which means finding the next item x_{n+1}^u with the largest probability given the user behavior sequence S^u.

Recently, a Sequential Recommendation Augmentation (SRA) paradigm is proposed [7], and the basic SRA method is also known as ASPeP. As illustrated in Fig 1, ASPeP utilizes reverse pretraining for data augmentation. We take Transformer as an example backbone for describing this learning paradigm. The key component in Transformer, i.e., multi-head self-attention is constructed with linear transformation and the scaled dot-product attention [9]. There are two stages in the training procedure updating the same set of model parameter θ, namely reverse pretraining and left-to-right finetuning. The reverse pretraining intents to learn the inverse sequence generation via the autoregressive learning objective:

$$\max_\theta \quad p_\theta(x_i^u|x_{i+1}^u, x_{i+2}^u, ..., x_n^u) \tag{2}$$

With the pretrained model, pseudo-prior items can be recursively generated for short sequences, in order to eliminate the data sparsity in recommendation and further improve the quality of the whole training set. The pretrained model can be further finetuned for next-item prediction, and the learning objective is the forward autoregressive learning objective:

$$\max_\theta \quad p_\theta(x_i^u|x_1^u, x_2^u, ..., x_{i-1}^u) \tag{3}$$



(a) Content-stream Attention (b) Target-stream Attention

Figure 2: Different Forms of Attention.

For details of SRA paradigm, please refer to [7].

3. Methodology

3.1. Permutation Autoregressive Learning

Now our intention is to help the SRA framework making use of the global contextual correlation in the behavior sequences. The idea of "mask and reconstruct" is a commonly used method for helping the sequence model to learn from contextual information in arbitrary position, but the incorporation of [MASK] token in behavior sequence will bring more severe discrepancy problem as in BERT [10], especially considering that the trained model will be used for recursively generating behavior sequences. Our solution to exploit context information is adopting the permutation modeling objective [11], which gather the information in bidirectional context while remaining the autoregressive learning paradigm.

3.1.1. Permutations with Original Position Encoding

In order to exploit bidirectional item correlation in bidirectional context, we propose to train the sequence model with different permutations of each training sequence in an autoregressive way.

Assume that the length of the behavior sequence is T, we denote the set of all possible permutation of index as Z^T. For example, if $T = 4$, then the original permutation is $[1, 2, 3, 4]$, and the number of permutation in Z^T is $T! = 24$. For each permutation $\mathbf{z}$ in Z^T, $z_{<t}$ stands for indices of all the element before t-th element z_t, and z_t is for the current elements. Similar to autoregressive learning, we try to predict z_t given $z_{<t}$. In this way, the learning objective is rewritten according to the permutation $\mathbf{z}$ instead of the original order of original sequence $\mathbf{x}$ as follows.

$$\max_\theta \mathbb{E}_{\mathbf{z}\sim Z^T} \log p_\theta(x_{z_t}|x_{z_{<t}}) \tag{4}$$

where $x_{z_{<t}}$ stand for the items in $\mathbf{x}$ whose index is in $z_{<t}$. This new objective calculates the probability of an item conditioned on all possible permutations of items in an autoregressive way, as opposed to just those to the left side or right side of the target item in the existing

Algorithm 1 Permutation Autoregressive Learning for SRA

Input: A set of behavior sequence $\{S^u\}$
Output: A sequential recommendation model
procedure PAL($\{S^u\}$)
    for each epoch **do**
        for each instance in batch **do**
            Sample n permutations with length L.
            Pretrain with Eqn 4.
    Save the result pretrained model $\mathcal{M}_0$ for generation.
    Select behavior sequences shorter than m.
    Generate the pseudo-prior items with beam width k.
    for each epoch **do**
        for each batch **do**
            Finetune $\mathcal{M}_0$ with Eqn 3.
    Save the result finetuned model for sequential recommendation.

methods for sequential recommendation augmentation. It should be emphasised that only the indices, namely which elements are used for prediction, are changed, while the position of each item in the original sequence is retrained.

The above learning objective can help each position to learn information from bidirectional context, but it brings a new issue about the position information. In prediction given several known items, the predicted position or index is not fixed as in original autoregressive learning. So we need to learn the target-aware representation which can tell the position that the current predicted item located in. Therefore, the $p_\theta(x_{z_t}|x_{z_{<t}})$ is formulated as:

$$p_\theta(x_{z_t}|x_{z_{<t}}) = \frac{\exp[\mathbf{e}(x_{z_t})\ \widetilde{\mathbf{h}}_\theta(x_{z_{<t}, z_t})]}{\sum_{x^*}\exp[\mathbf{e}(x^*)\ \widetilde{\mathbf{h}}_\theta(x_{z_{<t}, z_t})]} \quad (5)$$

where $\widetilde{\mathbf{h}}_\theta(x_{z_{<t}, z_t})$ is the learned target-aware representation for the item with the z_t-th index.

3.1.2. Two-stream Attention for Contextual Representation

As aforementioned, the self-attention module in the sequence model need to be modified for obtaining target-aware representations. The two-stream attention structure was used in language modeling to provide target position information without leaking the content information of the target.

Specifically, two separated streams of attention vectors are maintained to store content information and position information. For each position z_t in the factorization $\mathbf{z}$, we keep updating the intermediate vectors h_{z_t} and $\widetilde{h}_{z_t}$, representing content stream and target stream respectively. Each stream is learned by a designated attention mechanism. The detailed formulations of the two-stream attention structure is described as follows.

dataset	Beauty	Phones	Sports	Tools	Baby	Office
#user	22363	27879	35598	16638	19445	4905
#item	12101	10429	18357	10217	7050	2420
#instance	198502	138681	296337	134476	160792	53258
avg. length	6.88	4.97	6.32	6.08	6.27	8.86

Table 1
Statistics of the Datasets.

The first one is the content-stream representation which is exactly the same as the hidden state in the standard self-attention. This corresponding attention is named **Content-stream attention**:

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = h_{z_{\le t}}^{(m-1)}; \theta) \quad (6)$$

where $h_{z_t}^{(m)}$ is the output of the m-th block Transformer. The second representation is target representation, which contains only the position information of the target in order to avoid content information leaking. The **Target-stream attention** is:

$$\widetilde{h}_{z_t}^{(m)} \leftarrow \text{Attention}(Q = \widetilde{h}_{z_t}^{(m-1)}, KV = h_{z_{<t}}^{(m-1)}; \theta) \quad (7)$$

The content representation $h_{z_t}^{(0)}$ is initialized by the item embedding $\mathbf{e}(x_{z_t})$ which is added with the positional encoding as in normal Transformer, and all the target representation $\widetilde{h}_{z_t}^{(0)}$ is initialized by an identical trainable vector $\mathbf{w}$. The output $\widetilde{h}_{z_t}$ of the last layer Transformer is used as $\widetilde{\mathbf{h}}_\theta(x_{z_{<t}, z_t})$ in Eqn 5 for prediction. In this way, equipped with this two-stream attention, we can force each position in the sequence to learn bidirectional information while maintaining the normal behavior order.

3.1.3. Optimization of Sequence Augmentation

We propose beam-search for obtain the optimal sequence as the pseudo-prior items. Instead of recursively predict the next item in a greedy way, Beam Search method maintains a buffer of candidate subsequences and selects the best one with the largest joint probability. Beam width value is denoted as k. More details can be found in [12].

3.2. PAL Algorithm for SRA

Considering the computing complexity of the permutation autoregressive learning, we need to sample from the set of permutation and predict partial of the sequence. For each sampled permutation, we train the model via maximizing the probability of last item. The detailed learning procedure is described in Algorithm 1.

4. Experiments

4.1. Datasets and Baseline Models

Following the setting in [7] and [8], 6 datasets are adopted which are collected from $Amazon.com$[1]. For the behavior sequence construction, we regard the presence of review as an interaction between a user and an item, and construct the behavior sequence according to the timestamp. Following the preprocession in [7], we use the last item in a sequence for test. The statistics of datasets is shown in Table 1.

We compare our proposed method with the following methods including the state-of-the-art BiCAT [8] method in sequential recommendation augmentation. **SASRec** [13] utilize the transformer to extract the correlation from the training sequences and predict the next item. **BERT4Rec** [14] exploit the training method in BERT to learn transformer for SR. **ASReP** [7] reversely pretrain the transformer to generate pseudo-prior items for short sequence and then finetune the transformer for SR. **BiCAT** [8] is the latest model for sequential recommendation augmentation. It incorporates an additional objective in pretraining. **PAL** is our proposed learning method. **PAL++** equips PAL with beam search.

4.2. Implementation Details

We select the Transformer as the backbone to verify our SRA solution. The block number is fixed to 2. The hidden length is selected in {32, 64, 128}. The head number in attention is selected in {2, 4}. The learning rate is fixed at 0.001 since the results are similar with other settings. The dropout rate is fixed to 0.5. The short sequence length threshold m is set to 18, and each short sequence is augmented with 15 pseudo-prior items. For the number of sampled permutations (n in Algorithm 1), we select it in {2, 4, 6} with model selection. The epoch number is fixed to 200 which is sufficient for all the models to converge. We conduct model selection via grid search. For each behavior sequence, we randomly sample 100 negative items for ranking with the last item, which is the ground-truth. Recall@n, NDCG@n, and Mean Reciprocal Rank (MRR) are employed for as the evaluation metrics, and n is selected in $\{5, 10\}$.

4.3. Performance of PAL

We performance sequential recommendation on all the 6 datasets with the above mentioned baseline models to demonstrate the effectiveness of **PAL**. Previous SRA work has shown the advantage of self-attention sequence model for recommendation, so we use the **SASRec** and **BERT4Rec** as two baselines without augmentation. The performance results are presented in Table 2. All the SRA methods achieve better performance than the others, which verified the effectiveness of augmentation. Compared with the strongest sequential recommendation augmentation baseline **BiCAT**, the proposed PAL can provide around 2% to 5% improvement on NDCG10, which is significant in these sparse datasets. The beam search method (**PAL++**) consistently shows effectiveness on all the datasets, and the further performance improvement on the "Tools and Home Improvement" and "Baby" are more significant than other datasets. The explanation for this improvement difference is that the behavior diversity varies in different datasets.

	Model	R@5	R@10	NDCG@5	NDCG@10	MRR
Beauty	SASRec	0.3849	0.4863	0.2884	0.3212	0.2870
	BERT4Rec	0.4243	0.5371	0.3075	0.3598	0.3021
	ASReP	0.4583	0.5743	0.3465	0.4042	0.3540
	BiCAT	0.4901	0.5892	0.3704 +6.8%	0.4289 +6.1%	0.3712 +4.8%
	PAL	0.4934	**0.6048**	0.3873 +11.7%	0.4400 +8.8%	0.3803 +7.4%
	PAL++	**0.4936**	0.6036	**0.3879** +11.9%	**0.4415** +9.2%	**0.3821** +7.9%
Phones	SASRec	0.3517	0.4706	0.2475	0.2859	0.2470
	BERT4Rec	0.3732	0.4942	0.2687	0.3006	0.2684
	ASReP	0.5489	0.6758	0.4107	0.4518	0.3946
	BiCAT	0.5663	0.7032	0.4274 +4.0%	0.4729 +4.6%	0.3990 +1.1%
	PAL	0.5736	0.7178	0.4432 +7.9%	0.4798 +6.2%	0.4100 +3.9%
	PAL++	**0.5745**	**0.7239**	**0.4436** +8.0%	**0.4809** +6.4%	**0.4113** +4.2%
Sports	SASRec	0.3847	0.5051	0.2732	0.3122	0.2699
	BERT4Rec	0.4136	0.5325	0.3014	0.3561	0.2988
	ASReP	0.4734	0.6011	0.3470	0.3884	0.3370
	BiCAT	0.4842	0.6246	0.3649 +5.1%	0.4003 +3.1%	0.3562 +5.7%
	PAL	0.4936	0.6385	0.3784 +9.0%	0.4112 +5.9%	0.3712 +10.1%
	PAL++	**0.4940**	**0.6398**	**0.3796** +9.4%	**0.4174** +7.5%	**0.3730** +10.7%
Tools	SASRec	0.2853	0.3903	0.1987	0.2325	0.2037
	BERT4Rec	0.3613	0.5600	0.3190	0.3574	0.3011
	ASReP	0.4133	0.5347	0.3014	0.3406	0.2976
	BiCAT	0.4287	0.5509	0.3279 +8.8%	0.3571 +4.8%	0.3100 +4.2%
	PAL	0.4327	0.5624	0.3404 +12.9%	0.3767 +10.6%	0.3231 +8.6%
	PAL++	**0.4374**	**0.5681**	**0.3421** +13.5%	**0.3805** +11.7%	**0.3273** +10.0%
Baby	SASRec	0.3076	0.4358	0.2094	0.2509	0.2144
	BERT4Rec	0.3295	0.4701	0.2212	0.2758	0.2338
	ASReP	0.3581	0.4885	0.2499	0.2920	0.2508
	BiCAT	0.3682	0.4972	0.2603 +4.2%	0.3007 +3.0%	0.2587 +3.1%
	PAL	0.3759	**0.5123**	0.2741 +9.7%	0.3178 +8.8%	0.2704 +7.8%
	PAL++	**0.3785**	**0.5123**	**0.2804** +12.2%	**0.3200** +9.6%	**0.2724** +8.6%
Office	SASRec	0.4053	0.5098	0.2994	0.3335	0.2947
	BERT4Rec	0.4400	0.5682	0.3149	0.3589	0.3024
	ASReP	0.4689	0.6101	0.3303	0.3764	0.3186
	BiCAT	0.4801	0.6221	0.3462 +4.8%	0.3894 +3.5%	0.3326 +4.4%
	PAL	0.4982	0.6353	0.3572 +8.1%	0.3997 +6.2%	0.3486 +9.4%
	PAL++	**0.4994**	**0.6363**	**0.3602** +9.1%	**0.4011** +6.6%	**0.3497** +9.8%

Table 2
Performance of Different Methods on Sequential Recommendation. Relative changes are based on **ASReP**.

[1] http://jmcauley.ucsd.edu/data/amazon/

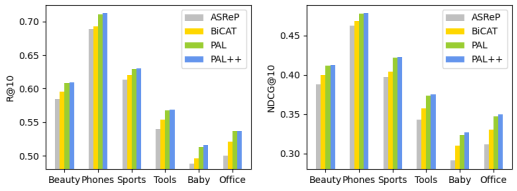


Figure 3: Performance on Short Sequence Instances.

4.3.1. Effectiveness of PAL for Short Sequences

Performance improvement on short behavior sequence is critical for an augmentation paradigm. To further analyze the advantages of **PAL** for short sequence, we reconstruct the test set with all the behavior sequence shorter than 3 and evaluate all the baseline methods and our **PAL** and **PAL++**. The results is presented in Figure 3. Improvement on short sequence is a critical for an augmentation paradigm. We can find that the **PAL** and **PAL++** method can significantly outperform the other sequential recommendation augmentation methods on all the datasets. This result illustrate that the proposed learning method can incorporate more contextual correlation information into the short sequence augmentation.

4.4. Analysis on Backbone Model

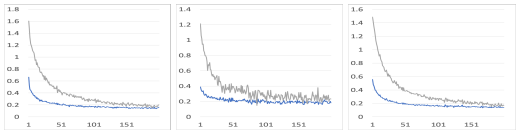
The default backbone model of the SRA methods, i.e., ASReP, BiCAT, PAL, PAL++ in Section 4.3 is the basic Transformer which is the same as in SASRec. All the SRA methods can also be applied to all the Transformer-based SR methods, such as SASRec, BERT4Rec and TiSASRec. For TiSASRec, we ignore the temporal information in the pretraining and sequence generating stages, and assign the smallest timestamp in the original sequence for the generated items. Here we report part of performance (NDCG@5) comparison on "Beauty" dataset in Table 3. According to the results, equipped with SRA methods, the performance of all the backbone models are improved, and the proposed PAL / PAL++ achieve the best results. Please note the **TiSASRec** is outperformed by **SASRec** as our current augmentation methods has not incorporated the temporal information, which is a future work for SRA.

Backbone	**SASRec**	**BERT4Rec**	**TiSASRec**
Base	0.2884	0.3075	0.3076
ASReP	0.3465	0.3562	0.3427
BiCAT	0.3704	0.3746	0.3625
PAL	0.3873	0.3886	0.3771
PAL++	0.3879	0.3886	0.3791

Table 3
NDCG@5 with Different Backbone Model

4.5. Analysis on Convergence Rate

One interesting finding is that the pseudo sequence generated by PAL can significantly improve the converge rate of the finetuning stage in sequential recommendation augmentation. We depict the loss value during the finetuning stage in Fig 4, where we can observe that the **PAL** method can converge earlier than **ASReP** to achieve a stable loss value. Similar results can be found in other datasets. Due to the permutation learning objective, the **PAL** is of advantage in the generated data and pretrained model, which lead to the improvement of the convergence rate in the final finetuning stage.



(a) Beauty (b) Cell Phones (c) Sports

Figure 4: Illustration of Convergence Rate in Finetuning. The x-axis is the epoch, and the y-axis is the loss value. The grey line is for ASReP method, and the blue one is for PAL.

References

[1] C. Cai, R. He, J. McAuley, Spmc: socially-aware personalized markov chains for sparse sequential recommendation, in: IJCAI, 2017, pp. 1476–1482.

[2] K. Song, M. Ji, S. Park, I.-C. Moon, Hierarchical context enabled recurrent neural network for recommendation, in: AAAI, volume 33, 2019, pp. 4983–4991.

[3] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, Y. Liu, V. S. Sheng, Long-and short-term self-attention network for sequential recommendation, Neurocomputing 423 (2021) 580–589.

[4] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: SIGIR, 2021, pp. 726–735.

[5] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, M. de Rijke, A collaborative session-based recommendation approach with parallel memory modules, in: SIGIR, 2019, pp. 345–354.

[6] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, C. Xiong, Contrastive self-supervised sequential recommendation with robust augmentation, arXiv preprint arXiv:2108.06479 (2021).

[7] Z. Liu, Z. Fan, Y. Wang, P. S. Yu, Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer, in: SIGIR, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1608–1612. URL: https://doi.org/10.1145/3404835.3463036. doi:10.1145/3404835.3463036.

[8] J. Jiang, Y. Luo, J. B. Kim, K. Zhang, S. Kim, Sequential recommendation with bidirectional chronological augmentation of transformer, arXiv preprint arXiv:2112.06460 (2021).
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, NeurIPS 30 (2017).
[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT (1), 2019.
[11] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, NeurIPS 32 (2019).
[12] S. Wiseman, A. M. Rush, Sequence-to-sequence learning as beam-search optimization, in: EMNLP, 2016, pp. 1296–1306.
[13] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: ICDM, IEEE, 2018, pp. 197–206.
[14] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: CIKM, 2019, pp. 1441–1450.