The Elephant in the Room: Rethinking the Usage of Pre-trained Language Model in Sequential Recommendation

Zekai Qu*
China University of Geosciences Beijing
Beijing, China
zekai_qu@163.com

Ruobing Xie*
Tencent Inc.
Beijing, China
xrbsnowing@163.com

Chaojun Xiao
Tsinghua University
Beijing, China
xcjthu@gmail.com

Xingwu Sun
Tencent Inc.
Beijing, China
sammsun@tencent.com

Zhanhui Kang
Tencent Inc.
Shenzhen, China
kegokang@tencent.com

Abstract

Sequential recommendation (SR) has seen significant advancements with the help of Pre-trained Language Models (PLMs). Some PLM-based SR models directly use PLM to encode user historical behavior's text sequences to learn user representations, while there is seldom an in-depth exploration of the capability and suitability of PLM in behavior sequence modeling. In this work, we first conduct extensive model analyses between PLMs and PLM-based SR models, discovering great underutilization and parameter redundancy of PLMs in behavior sequence modeling. Inspired by this, we explore different lightweight usages of PLMs in SR, aiming to maximally stimulate the ability of PLMs for SR while satisfying the efficiency and usability demands of practical systems. We discover that adopting behavior-tuned PLMs for item initializations of conventional ID-based SR models is the most economical framework of PLM-based SR, which would not bring in any additional inference cost but could achieve a dramatic performance boost compared with the original version. Extensive experiments on five datasets show that our simple and universal framework leads to significant improvement compared to classical SR and SOTA PLM-based SR models without additional inference costs. Our code can be found in https://github.com/777pomingzi/Rethinking-PLM-in-RS.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommendation, language model, pre-training

*Both authors have equal contributions. Ruobing Xie is the corresponding author.

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688107

ACM Reference Format:
Zekai Qu, Ruobing Xie, Chaojun Xiao, Xingwu Sun, and Zhanhui Kang. 2024. The Elephant in the Room: Rethinking the Usage of Pre-trained Language Model in Sequential Recommendation. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3640457.3688107

1 Introduction

Recommender system is an information processing system that is deployed to forecast user preferences and provide suitable items or information. Its application spans various domains such as e-commerce, advertising, and streaming services. Intensive studies in various real-world scenarios have found that historical interactions are important signals to model user preferences. Hence, sequential recommendation (SR) is a promising topic in the recommender systems community, aiming to capture the users' dynamic preferences and predict the next item they like based on their historical behaviors. Early studies mainly adopt Markov chains to model the transition patterns of users [36]. In recent years, due to the rapid development of deep learning, different kinds of neural networks have been introduced into the SR, which resulted in notable enhancements. The most representative works include GRU4Rec [21], Caser [38], SR-GNN [44], SASRec [24], and BERT4Rec [37].

With the thriving of PLMs, researchers begin to explore their application in SR. Some studies utilize PLMs as item text encoders, aiming to enhance item representations with the rich prior knowledge embedded within PLMs [22, 30, 40, 48, 50]. These methods always leverage a PLM to encode item text (e.g., title, brand) into text embeddings, which are then utilized to substitute or augment the original ID embeddings. Recently, propelled by the significant reasoning and long text modeling abilities demonstrated by Large Language Models (LLMs), some pioneering researchers have sought to harness the advanced sequence modeling capabilities of PLMs to enhance the performance of SR. One line of work uses fixed LLMs for recommendations via prompt or in-context learning, which achieve promising performance under the few-shot or zero-shot settings [14, 28]. However, due to the significant disparity between language and behavior modeling, these methods often perform poorly when interactions are sufficient. In that case, recent works choose to adopt recommendation objectives to fine-tune PLMs for adaptation, which achieve promising results [10, 15, 27, 34, 49].

Despite the efficacy of PLM-enhanced methods in existing literature, a critical question remains largely unexplored: do PLMs effectively and efficiently boost sequential recommendation by providing accurate prior knowledge and superior sequence modeling capabilities as theoretically expected? While some research has investigated their effect in item representation learning [40, 43, 48], their impact on behavior sequence modeling has yet to be thoroughly examined. Consequently, our study aims to figure out two questions: (1) whether the powerful sequence modeling capabilities of PLMs have been fully and economically utilized in the *behavior sequence modeling* of SR. (2) If not, is there a more *effective, universal, and economical* way for us to employ PLMs for SR?

For question one, we conduct two motivating experiments (in Sec. 3) using a representative SOTA PLM-enhanced multi-domain SR model, RECFORMER [27]. The findings highlight two critical aspects: (1) RECFORMER functions differently from its base PLM backbone in behavior modeling. Its attention maps exhibit a clear functional stratification, with higher layers (more crucial for SR adaptation) more assimilated to conventional ID-based SR models. (2) There appears to be significant parameter redundancy when employing PLMs for behavior modeling. The global attentions of the sequence display considerable resemblance among different heads and layers in the same functional stratification. Only tuning 1/4 selected layers of RECFORMER can yield performance comparable to or even surpassing the original RECFORMER.

Based on the above observation, we implement several enhanced variants of RECFORMER with simplified sequence modeling methods borrowed from classical ID-based SR models such as SASRec and BERT4Rec. We discover the key factor of improvement borrowed from PLM to SR: **item representations built by behavior-tuned PLM**. Experimental results indicate that PLM-involved SR models with simpler sequence models can achieve comparable results to RECFORMER with much better efficiency, suggesting that PLMs are not that suitable for behavior sequence modeling, for their superior sequence modeling and reasoning capacities in NLP are not fully leveraged and transferred into the task of SR. Consequently, we conclude a simple, effective, and universal framework to efficiently take advantage of PLM for SR: (a) **adopting simplified (ID-based) sequential models for behavior sequence modeling** (for question one), and (b) **using behavior-tuned PLMs (rather than vanilla PLMs) for item embedding initialization** (for question two). Further experiments demonstrate that our behavior-tuned PLMs initialized item embeddings are universal with different (ID-based or PLM-based) sequential models and settings. The contributions are concluded as follows:

- To the best of our knowledge, we are the first work to conduct a thorough analysis of the mechanism and effectiveness of PLMs in behavior sequence modeling of SR.
- We confirm the existence of functional stratification and parameter redundancy in the behavior sequence modeling processes of the current PLM-based SR model.
- We introduce a novel framework that uses behavior-tuned PLMs for item initialization and simplified methods for behavior sequence modeling. Our framework is simple, effective, and universal, leading to significant improvements over PLM-based and classical ID-based SR baselines on different settings without additional inference costs.

2 Related Works

2.1 Sequential Recommendation

Sequential recommendation (SR) aims to model the users' dynamic behavior patterns based on their historical interactions. The very early works usually adopt Markov chains (MCs) to model the behavioral patterns of users. Rendle et al. [36] integrates MF and MCs to simultaneously model users' overarching preferences and their historical interactions. Beyond first-order MCs, several approaches have embraced higher-order MCs, incorporating more preceding items for sequential modeling [19, 20]. Later, with the development of deep learning methods, various deep neural networks were introduced into SR. RNNs such as Gated Recurrent Unit (GRU) [9] are introduced to model sequential behaviors, including session-based GRU (e.g., GRU4Rec [21]) and user-based GRU [13]). Moreover, CNNs have also been demonstrated to effectively capture short-term sequential dynamics using both horizontal and vertical convolutional filters [38]. Many Graph Neural Network (GNN) based models [7, 16, 42, 44], were also implemented by researchers to seek better performance. Recently, due to the notable achievements in sequence modeling made by Transformers [39] in natural language processing, many attention-based models have also been intensively explored. SASRec [24] and BERT4Rec [37] employ unidirectional and bidirectional Transformers separately for learning dynamic user preferences. LSSA [47] combines long-term and short-term self-attention mechanisms, designed to model users' enduring preferences alongside their immediate needs. Self-supervised learning (SSL) is also introduced by recent works for improving the performance and training efficiency of SR models [8, 33, 45, 46, 50]. Besides, some recent efforts explore the cross-domain sequential recommendation [6, 17, 29], while most of them face challenges in smoothly transferring the generalized information to new domains. To address this, some works [22, 40, 48] further use PLMs and vision encoders to fuse modality information into item representations for a more comprehensive understanding of item contents, functioning as the bridge across different datasets.

2.2 PLM for Recommendation

With the thriving of foundation models, many applications of PLMs for SR have been researched, hoping to leverage the rich knowledge and powerful sequence modeling capabilities of PLMs to enhance SR. Some works propose to use them as text encoders (e.g., build item representations from their textual representations based on PLMs rather than merely using the conventional random initialized ID embeddings). For example, IDA-SR [30] and UniSRec [22] apply BERT to obtain item representations from their corresponding texts. LLM2BERT4Rec [18] utilizes an embedding model with the titles of the items to retrieve their embeddings. Besides, some researchers try to convert users' historical interactions and items into plain text and utilize them to fine-tune the PLMs for SR [11, 15, 27, 34].

Recently, the emergence of Large Language Models (LLMs) [1, 5, 32] has revolutionized the field of natural language processing with their powerful reasoning abilities and rich world knowledge. Therefore, some pioneer works explore the possibility of directly using

powerful LLMs for recommendation via prompt or in-context learning [14, 23, 25, 28, 41]. Although these methods achieve impressive results in cold-start scenarios, they still perform poorly in SR when interactions are sufficient since behavioral information is essential for making precise recommendations. Another line of work considers introducing collaborative signals into LLMs to improve their recommendation performance. TallRec [3] and InstructRec [49] perform instruct-tuning on LLMs to adapt them for understanding behavioral patterns. RLMRec [35] aligns the semantic space of LLMs with collaborative relational signals through cross-view alignment to improve the quality of representations. BIGRec [2] treats SR as a text generation task, generating item titles by a two-step approach with the LLM fine-tuned on user behavioral sequences. Nevertheless, these methods lack a thorough examination of whether PLMs' capabilities are being effectively and efficiently harnessed. In our paper, we deeply explore the involvement of PLMs in SR tasks. We intend to scrutinize whether their formidable abilities are truly being utilized to their full potential and offer insights on how to make the most out of PLMs in the context of SR.

3 Analyses on Behavior Sequence Modeling of PLM-based SR Models

In this section, we comprehensively analyze the capability and suitability of PLM-based SR models on behavior sequence modeling. Without loss of generality, we employ the SOTA PLM-based SR model RECFORMER [27] for the following explorations.

3.1 Background of RECFORMER

RECFORMER is a PLM-based SR model that comprises a classical PLM Longformer [4] and two additional embeddings that indicate the type of token (determining the token belongs to the special token [CLS], textual attribute key or textual attribute value) and its associated item (which item the token belongs to). It transforms items and behavioral sequences into text sequences and then utilizes the behavior-tuned model to obtain their corresponding representations for providing recommendations. RECFORMER's training process has three steps, including pre-training and two-stage fine-tuning. To simplify, we designate these stages as *stage-PT*, *stage-FT1*, and *stage-FT2* respectively. (1) In stage-PT, RECFORMER adopts Masked Language Modeling (MLM) and item-item contrastive (IIC) tasks to pre-train the model. Considering the computational costs, RECFORMER treats in-batch next items as negative instances instead of negative sampling or fully softmax for the IIC task. (2) In stage-FT1, RECFORMER encodes all items before each epoch starts, obtaining their corresponding item embeddings, then fine-tuning all parameters with these frozen embeddings (for efficiency considering the large size of item candidates). (3) In stage-FT2, the RECFORMER selects the optimal model from stage-FT1 to acquire its item embeddings and fix them. Stage-FT2 only tunes the parameters of the behavior sequence modeling part (similar to stage-FT1, but with the item embeddings remaining fixed and not updated each epoch). This strategy of maintaining fixed item embeddings is primarily motivated by the significant computational demands associated with generating all item representations during the cross-entropy loss calculation. Therefore, the fine-tuning process of RECFORMER could be simplified as: (1) obtaining the item embeddings from behavior-tuned Longformer, and (2) tuning PLM solely for behavior sequence modeling with fixed item embeddings. The latter step is our focus.

3.2 Analysis on Attention Discrepancy

Intuitively, the sequential patterns of items in SR and tokens in LM significantly differ. SR tends to prioritize the most recent interactions of users, focusing more on their immediate past behavior to make predictions [24, 37]. On the other hand, LM concentrates more on maintaining linguistic consistency and adhering to the syntactic structure of the text, ensuring coherence and grammatical accuracy throughout the content. To verify this assumption, we visualize the attention distributions of the special token [CLS] that is used to get the user representation for recommendation in RECFORMER (and its backbone Longformer [4] for comparisons).

Specifically, we follow the setting in [27] to pre-train and fine-tune RECFORMER, and stack the attention scores from different heads in the same layer on the vertical axis. As for Longformer, we directly utilize its original checkpoint, drawing the attention maps in the same way as RECFORMER. It's important to highlight that RECFORMER reverses the user's interaction sequence. Consequently, within the attention maps, items positioned to the left symbolize those with which the user has most recently interacted. In Fig. 1, we provide the attention maps of two instances from Pantry and Instruments [31] with its length of historical sequence as three (for better visualization). Upon examining various instances from multiple domains, we have some universal findings:

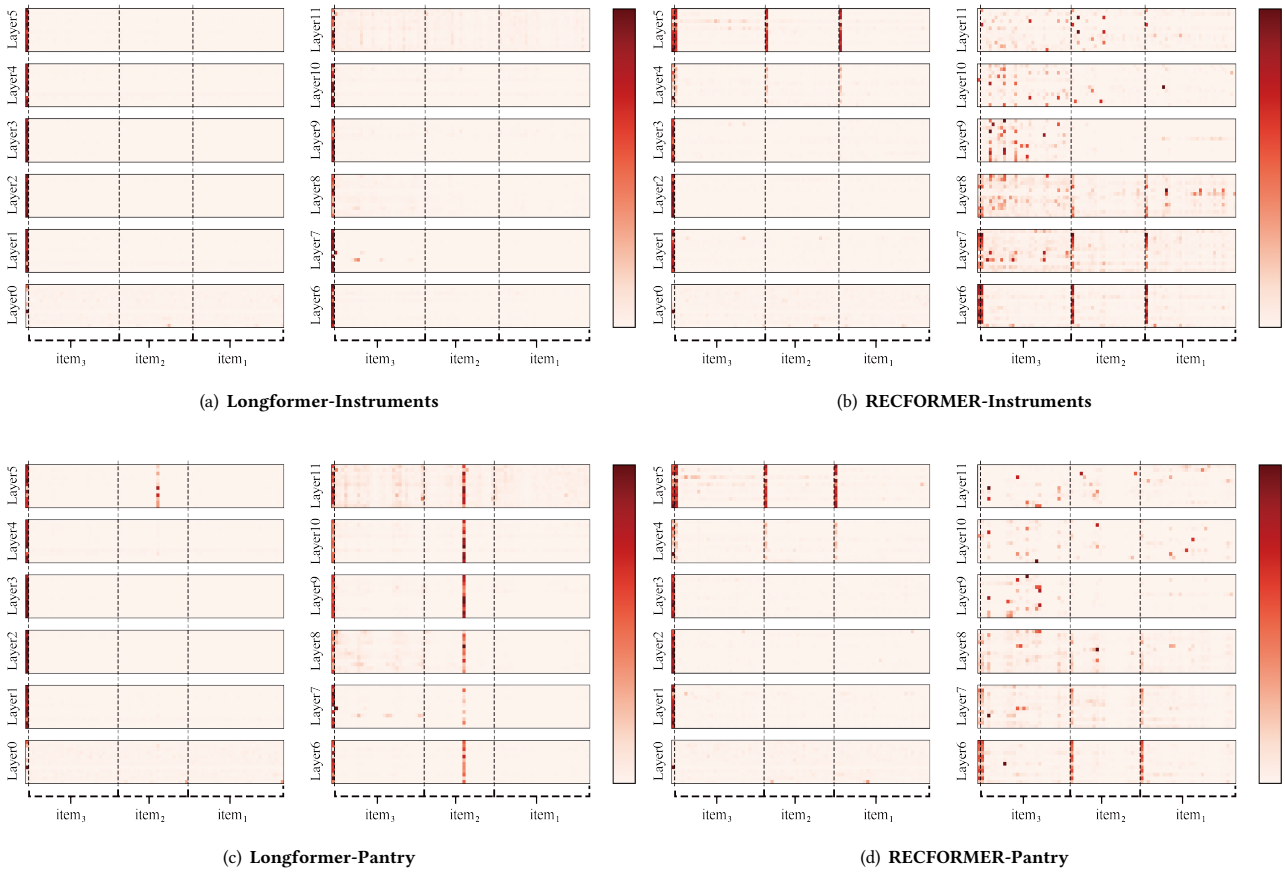
(1) There is a ***distinct stratification*** in RECFORMER's attention distributions (i.e., the attention distributions are similar within layers 0-3/4-7/8-11) across different instances and domains.

(2) The attention of RECFORMER's Layers 4-7 significantly focuses on the ***first token of each item***, revealing its function as modeling the boundaries between different items. Besides, we also find a universal phenomenon that these first tokens of each item mainly ***focus on the tokens inside each item at shallow layers***. These are obvious SR patterns that resemble the process of generating the corresponding item embeddings of each item in the behavior sequence, which cannot be found in Longformer.

(3) At Layers 8-11, RECFORMER's [CLS] begins to find fine-grained highlight tokens in each item. We find that the highlights of RECFORMER on items are ***very similar to those of SASRec*** (i.e., more focused on recent items). Therefore, Layers 8-11 might take a role similar to the sequential behavior modeling in classical ID-based SR models.

(4) There is a notable presence of almost the same attention distributions across certain layers and attention heads, indicating ***significant parameter redundancy*** within PLM-based SR models.

These phenomena are widely-existed in different instances and datasets. In conclusion, we observe that RECFORMER's sequential modeling pattern is totally different from its backbone Longformer's, but similar to conventional ID-based SR models (e.g., SASRec) to some extent. And significant degree of similarity in attention patterns is observed across various attention heads and adjacent layers within the same functional stratification. These observations imply the **underutilization and parameter redundancy** of the current usage of PLMs in behavior sequence modeling.



(a) **Longformer-Instruments** (b) **RECFORMER-Instruments**

(c) **Longformer-Pantry** (d) **RECFORMER-Pantry**

Figure 1: Attention maps of [CLS] in Longformer and RECFORMER. The horizontal axis represents the tokens of items, and the vertical axis represents the attention heads. They have different sequence modeling patterns.

3.3 Analysis on Model Redundancy

Conventional ID-based SR models typically feature a simplified architecture with fewer layers and heads (for example, SASRec employs 2 layers and 1 head). As reported in their respective studies [24, 37], augmenting the number of attention heads and layers tends to result in a marginal performance improvement and, in some cases, can even cause a reduction in effectiveness. However, PLMs usually have several times more layers and heads (e.g., Longformer has 12 layers and 12 heads), aiming to capture the complex grammar and semantic knowledge within natural language. This level of complexity might be over-qualified for the current requirements of behavior sequence modeling in SR. The observed attention homogenization on different layers and heads also indicates that the current sequential modeling of PLM-based SR models is redundant and could be compressed. Hence, we conduct an intuitive experiment that only tunes several layers of RECFORMER on *Amazon Pantry* as shown in Table 1. We have some findings:

(1) None (i.e., the original Longformer) performs worse than other settings, indicating that behavior-based tuning is essential for obtaining a satisfactory performance. There exists a giant gap between the sequence modeling of PLM and SR.

(2) When the number of tuned layers is fixed, tuning deeper layers always achieves better results in different settings, which may be due to their better task adaption to SR.

(3) By ***selectively tuning one layer of each stratification*** (e.g., tuning layers 3,7,11), we achieve astonishing results that are comparable to or even exceed those of full fine-tuning in the original RECFORMER. Additionally, we also evaluate on *Arts* and *Instruments*, where merely tuning Layers 3,7,11 could match or closely approach the performance of full fine-tuning. Specifically, we observed changes of HR@5 -0.07% and NDCG@5 +0.90% on Instruments, and HR@5 -0.62%, NDCG@5 -0.18% on Arts. It not only confirms the functional segmentation but also highlights the existence of substantial parameter redundancy in PLM for SR. These results imply that the power of PLM in modeling token sequences is not fully activated in existing PLM-based SR models (or is not perfectly suitable for the behavior sequence modeling in SR).

Based on the above findings, we begin to rethink the current usage of PLMs in SR, ambitiously aiming to investigate the following two problems: (1) *Can we use more simplified sequential models (e.g., classical ID-based SR models such as SASRec and BERT4Rec) to replace the cumbersome PLMs for more economical behavior sequence*

modeling? (2) *How to effectively and efficiently activate the magic power of PLMs for SR without much inference cost*?

Table 1: Results of different tuned layers in RECFORMER.

Metric	Tuned Layers								
	None	All	0,4,8	1,5,9	2,6,10	3,7,11	3	7	11
H@5	23.61	31.37	29.90	31.53	31.20	**32.55**	28.97	29.76	30.76
H@10	30.97	41.08	39.41	41.37	40.74	**42.32**	38.42	39.68	40.50
N@5	18.56	24.41	23.27	24.32	24.26	**25.27**	22.60	23.06	23.94
N@10	20.92	27.54	26.33	27.49	27.33	**28.42**	25.64	26.26	27.07

4 EXPERIMENTS

In this section, we aim to answer the following research questions: (RQ1): Does the powerful sequence modeling capability of PLMs effectively benefit SR for better modeling the behavior sequence? If not, could we replace it with a simpler sequence model? (RQ2): What is the optimal approach to leverage the formidable capabilities of PLMs in SR, especially considering their limitations in modeling sequences? (RQ3): What does an ideal framework for integrating PLMs into SR look like, one that harnesses the powerful capabilities of PLMs while maintaining relatively low training and inference costs? To investigate the above problems, we implement several simplified but effective PLM-based SR variants for exploration, regarding *behavior-tuned PLMs as item initializer* with *simplified classical ID-based behavior sequence modeling*.

4.1 Behavior-tuned PLMs as Item Initializer

We conduct extensive experiments on the SOTA PLM-based SR model RECFORMER [27], and implement our enhanced variants with simplified sequence modeling based on it. We adopt two classical SR models SASRec [24] and BERT4Rec [37] as the backbone sequence modeling method. Our PLM-based SR variants can be divided into three groups, namely embedding-init (freeze), embedding-init (trainable), and further-train. Note that we use the behavior-tuned PLM (i.e., Longformer tuned in RECFORMER's certain stages in Sec. 3.1) to initialize item embeddings in our methods.

(a) For **embedding-init (freeze)**, we experiment with three versions of RECFORMER: the original Longformer [4] (LF), RECFORMER trained on all pre-training datasets (PT), and RECFORMER after stage-FT1 on the target domain (FT). We only use the item representations obtained by these models as the fixed item ID embeddings of ID-based SR models (e.g., SASRec), and fine-tune the sequence modeling part of ID-based SR models via the target domain's training data. This setting could be viewed as an SR model with fixed item embeddings given by Longformer/RECFORMER, aiming to explore whether a simple sequence model from conventional ID-based SR is sufficient for behavior modeling of SR.

(b) For **embedding-init (trainable)**, we allow the initialized item representations above to be trainable along with the sequence modeling part (i.e., SASRec/BERT4Rec with trainable item embeddings that Longformer/RECFORMER has initialized). It provides insights into the impact of concurrent learning on well-initialized ID embeddings via PLM and simplified sequence modeling.

(c) For **further-train**, we start from the FT version of embedding-init (freeze) and proceed to optimize (i) all model parameters (All) or (ii) just the item embeddings (Emb). In this setting, our well-initialized item ID embeddings obtained from behavior-tuned PLM are further tuned under a matched and warmed-up sequence modeling component of ID-based SR models.

We should highlight that these 8 variants are all equipped with (a) **simplified sequence modeling of ID-based SR models**, and (b) **item initialization based on behavior-tuned PLM**. The PLM part is **NOT** tuned/used in the following tuning and serving.

4.2 Experimental Settings

Datasets. We conduct experiments on seven real-world datasets from the Amazon review datasets [31]. We use *Books*, *Movies and TV*, *Sports and Outdoors*, *Clothing Shoes and Jewelry* for pre-training, and choose *Arts, Crafts and Sewing*, *Musical Instruments*, and *Pantry* as the downstream domains for evaluation. In addition, we also select Movies and TV and Sports and Outdoors from the pre-training dataset for evaluation to test the impact of pre-training on the source domain. Following [24, 37], for pre-training datasets, we filter out users and items with less 5 interactions. But for new domain datasets, we just filter out the users with less than 4 interaction records to ensure adequate instances. We arrange each user's historical behavior sequence by timestamps and adopt leave-one-out setting for building validation and test sets. For each user historical sequence, the last item is used as the test data, the item before the last one is used as the validation data, and the remaining interaction sequences are used for training.

As our research is centered on assessing the efficacy of PLMs in behavior sequence modeling, we are particularly mindful of the potential influence of pre-existing knowledge within PLMs on zero-shot capabilities in conventional SR models. To address this concern and ensure a more accurate evaluation, we have strategically ***excluded all validation and test instances that contain items not previously trained in the model***. This step is crucial as it helps to isolate the effects of PLMs on zero-shot scenarios in our analysis and provides a clearer understanding of their true performance in SR tasks, free from the confounding effects of prior knowledge on cold-start items. The detailed statistics are in Table 2.

Evaluation Metrics. To evaluate the performance of the Sequential Recommendation task, we follow the classical setting [24, 37] and select 100 randomly sampled negative items for evaluation. We adopt two widely-used metrics Hit rate (HR@k) and Normalized Discounted Cumulative Gain (NDCG@k) with $k = 5, 10$.

Implementation Details. We implement RECFORMER with its provided source code, pre-training and fine-tuning it with the same hyper-parameters mentioned in the original paper. For simplified sequence models, the batch size is set to 256 and 128 for BERT4Rec and SASRec separately, and we carefully tune the learning rate in {0.0003, 0.001, 0.003, 0.01}. Additionally, to directly utilize the representations obtained by the RECFORMER, we set the embedding dimension as 768 for SASRec and BERT4Rec, which is identical to the RECFORMER's. To ensure a fair comparison, we optimize all the models with Adam optimizer. Early stopping is adopted with the patience of 10 epochs to prevent overfitting, and NDCG@10 is set as the indicator.

Table 2: Statistics of seven pre-training domain and new domain datasets.

Dataset	# User	# Item	# Interactions	Avg.len
Pre-training	402,979	930,518	3,547,017	8.8
- Books	197,891	504,085	1,990,164	10.1
- Clothing	135,041	294,788	1,004,679	7.4
- Sports	39,477	87,235	262,998	6.7
- Movies	30,570	44,410	289,176	9.5
Arts	131,149	138,116	718,628	5.5
Instruments	62,691	53,899	403,135	6.4
Pantry	22,601	8,249	179,735	8.0

4.3 Main Results

Table 3 shows the overall results on five datasets and two simplified SR models, we have the following astonishing findings:

(1) ***PLM's powerful sequence modeling capability is not fully activated or overqualified when performing behavior modeling in SR. The simple sequence modeling of ID-based SR performs well enough with better efficiency.*** In general, the FT embedding-init (freeze) has comparable or even better performance across most datasets compared to RECFORMER. It indicates that a simple sequence model is capable of behavior modeling in SR, while the sequence modeling ability of PLMs may be redundant.

(2) ***Item initialization based on behavior-tuned PLM could substantially boost the performance, while vanilla PLMs' initializations have no effort.*** Models with LF-initialized embeddings exhibit inferior performance compared to the original random initialization. However, both PT and FT initializations consistently lead to significant improvements. This holds true regardless of whether RECFORMER's original performance surpasses that of SASRec and BERT4Rec. It implies that the powerful semantic knowledge in vanilla PLMs has a huge gap with the personalized user preference in SR. Directly integrating PLMs into SR might inadvertently introduce excessive noise, potentially undermining the performance of the model. On the contrary, behavior-tuned PLMs could manage and utilize the knowledge within it effectively when facilitating SR.

(3) ***Further tuning on item representations given by behavior-tuned PLM could further improve the results.*** Further-train variants achieve the overall best performance, and embedding-init (trainable) outperforms embedding-init (freeze). These two settings conduct further tuning on item representations, which indicates that the item initialization of behavior-tuned PLM requires further adaptation to obtain sufficient behavior information. Moreover, we notice that solely further training item embeddings (i.e., with fixed sequence modeling part) in further-train has the best performance. It indicates that further training embedding might be a more effective and stable way to improve model performance further.

(4) ***The power of behavior-tuned PLMs as item initializers is potentially transferable.*** Comparing LF, PT, and FT settings, we observe that even PLMs tuned on pre-training datasets (PT) could also bring in essential user preference knowledge and largely benefit SR in the downstream new domains compared to the original ID-based SR models and vanilla PLM initialization. Future research could focus on the novel usage of item initialization from multi-domain behavior-tuned PLMs for various new domains. Furthermore, developing a more efficient framework for pre-training behavior-tuned PLMs is also worthy for researchers to investigate.

(5) ***Sequence modeling methods that possess analogous architectures and training objectives to those of behavior-tuned PLMs benefit more from our PLM-based initialization.*** Our item initialization is universal for different ID-based sequence modeling. Comparing the results based on SASRec and BERT4Rec, we find that our initialization facilitates more for BERT4Rec under the freeze setting. Besides, its FT (freezing) performs comparably to FT (trainable). We hypothesize that it is attributed to the similarities in sequence modeling architectures and training losses between RECFORMER and BERT4Rec. It also supports our hypothesis that PLMs may be excessively complex for behavior sequence modeling and could be substituted with simpler sequence models.

4.4 Results of Full-ranking Setting

The findings in Sec. 4.3 demonstrate the effectiveness of our proposed behavior-tuned PLMs initialized item embeddings under the random negative sampling setting. To further assess the robustness of our method, we conduct experiments with the full-ranking setting [12, 26] (i.e., ranking the positive item with the whole item set) on the same datasets. From Table 4, we have:

(1) Adopting our behavior-tuned PLMs initialized item embeddings could still greatly improve the performance of original SASRec under the full-ranking setting, while more improvements are made in metrics related to coarse-grained accuracy (HR@k). It indicates that our proposed method has more advantages in improving the model's generalization ability (i.e., tends to push positive samples to a relatively higher rank among all candidates, rather than the specific top 1 position).

(2) Our method still consistently outperforms RECFORMER on all datasets. Compared with the random sampling setting in Table 3, the improvement in the full-ranking setting is not that impressive. This phenomenon is consistent with previous works [27, 34] that PLMs-based SR models perform well at making precise recommendations (i.e., pushing positive samples to the very top part of all candidates). It should be highlighted that our method greatly reduces the training and inference costs compared with RECFORMER. We only use PLM for item initialization, and the subsequent tuning and inference are only based on simplified sequence models without PLM. Considering our simple structure in serving (the same as SASRec), the current improvement is acceptable. The advantage of our method might be more activated in the matching stage (selecting the top hundreds of items from millions of candidates).

4.5 Analysis on Further Cooperations

To further explore the effectiveness and robustness of our proposed behavior-tuned PLMs initialized embeddings for SR, we conduct two variants of our item initialization for evaluation.

Cooperating with trainable ID embeddings. Some previous works enhance item representation by adding the textual embeddings to its trainable item ID embeddings [22, 30]. Following this, we regard our behavior-tuned PLM initialization as the extra textual embeddings in SASRec and fix it during training (i.e., initialized embeddings as additional training signals). The results in Table

Table 3: Overall performance comparisons. SAS, BERT, and REC represent the original SASRec, BERT4Rec, and RECFORMER. LF, PT, FT represent using Longformer, pre-trained RECFORMER, and fine-tuned RECFORMER for embedding initialization. Improv. gives the relative improvements compared to the original base SR models (SASRec/BERT4Rec), which are significant.

Dataset	Metric	SAS	REC	Embedding-init (freeze)			Embedding-init (trainable)			Further-train		Improv.
				LF	PT	FT	LF	PT	FT	All	Emb	
Arts	H@5	58.51	59.60	44.23	56.61	59.66	58.13	67.75	70.85	**71.29**	71.01	**+21.84%**
	H@10	68.54	68.72	56.57	67.76	70.40	69.00	76.78	80.70	80.62	**81.05**	**+18.25%**
	N@5	48.35	49.88	33.33	45.16	48.63	47.46	55.27	58.86	**59.72**	59.13	**+23.52%**
	N@10	51.59	52.48	37.32	48.74	52.10	50.97	58.31	62.01	**62.78**	62.27	**+21.69%**
Instruments	H@5	58.42	53.18	43.84	54.22	54.49	58.73	66.06	67.28	68.28	**68.37**	**+17.03%**
	H@10	68.10	63.70	56.19	66.37	65.97	69.05	75.41	77.61	**78.55**	78.06	**+15.35%**
	N@5	48.66	44.06	33.16	43.10	43.63	48.64	55.27	55.14	**56.35**	56.34	**+15.80%**
	N@10	51.74	47.46	37.15	47.02	47.48	52.02	58.31	58.50	**59.68**	59.51	**+15.35%**
Pantry	H@5	31.80	31.37	20.50	31.51	30.42	31.78	**37.85**	33.03	36.39	35.92	**+19.03%**
	H@10	43.48	41.08	30.93	42.43	40.91	44.12	**49.81**	45.25	48.28	48.20	**+14.56%**
	N@5	23.62	24.41	13.73	23.78	23.00	22.47	**28.71**	23.90	27.48	27.02	**+21.55%**
	N@10	27.40	27.54	17.08	27.29	26.35	26.34	**32.56**	27.83	31.31	31.01	**+18.83%**
Sports	H@5	41.76	45.89	37.73	47.03	46.28	39.90	48.88	54.64	54.68	**56.08**	**+34.29%**
	H@10	52.16	57.26	49.01	59.42	58.21	51.29	59.69	66.36	66.53	**67.84**	**+30.06%**
	N@5	33.42	36.86	28.55	36.59	36.31	31.42	39.05	43.55	43.51	**44.70**	**+33.75%**
	N@10	36.84	40.53	32.17	40.59	40.03	35.09	42.54	47.33	47.27	**48.50**	**+31.65%**
Movies	H@5	61.58	53.90	39.87	56.30	55.46	61.14	68.86	68.36	68.19	**69.48**	**+12.83%**
	H@10	70.52	63.67	52.04	67.26	66.12	71.11	77.96	78.07	77.30	**79.02**	**+12.05%**
	N@5	51.42	45.27	28.78	45.40	45.29	50.55	57.50	56.38	56.86	**57.71**	**+12.23%**
	N@5	54.30	48.42	32.71	48.95	48.73	53.77	60.45	59.53	59.81	**60.81**	**+11.99%**

Dataset	Metric	BERT	REC	Embedding-init (freeze)			Embedding-init (trainable)			Further-train		Improv.
				LF	PT	FT	LF	PT	FT	All	Emb	
Arts	H@5	52.41	59.60	48.76	61.81	64.17	54.91	62.38	64.26	62.35	**64.64**	**+23.34%**
	H@10	62.34	68.72	60.66	71.33	73.24	65.75	71.99	73.29	70.95	**73.50**	**+17.90%**
	N@5	43.22	49.88	37.36	51.24	54.12	44.30	51.86	54.23	52.85	**54.66**	**+26.47%**
	N@10	46.43	52.48	41.21	54.33	57.07	47.81	55.00	57.16	55.64	**57.53**	**+23.91%**
Instruments	H@5	54.20	53.18	49.16	61.17	63.61	54.95	61.75	63.51	62.23	**64.23**	**+18.51%**
	H@10	63.51	63.70	59.12	71.04	72.86	65.74	71.50	72.88	71.07	**73.44**	**+15.64%**
	N@5	45.53	44.06	39.52	50.83	53.65	44.70	51.69	53.68	52.64	**54.38**	**+19.44%**
	N@10	48.54	47.46	42.74	54.02	56.66	48.09	54.85	56.71	55.50	**57.36**	**+18.17%**
Pantry	H@5	28.93	31.37	24.49	30.71	33.57	27.23	32.77	34.43	33.50	**34.75**	**+20.12%**
	H@10	39.91	41.08	36.48	42.36	45.01	39.61	44.62	45.59	44.66	**46.11**	**+15.53%**
	N@5	21.51	24.41	16.76	22.58	25.26	19.55	24.24	25.87	25.09	**26.24**	**+21.99%**
	N@10	25.03	27.54	20.57	26.32	28.94	23.52	28.05	29.48	28.69	**29.89**	**+19.42%**
Sports	H@5	31.12	**45.89**	30.87	43.68	43.23	33.19	43.83	43.11	42.98	43.47	**+40.84%**
	H@10	41.28	**57.26**	41.08	54.91	53.95	43.55	55.02	54.10	53.28	54.10	**+33.28%**
	N@5	24.51	**36.86**	22.75	34.29	34.05	25.24	34.59	34.15	34.10	34.51	**+41.13%**
	N@10	27.78	**40.53**	26.04	37.91	37.51	28.58	38.20	37.69	37.42	37.94	**+37.51%**
Movies	H@5	56.67	53.90	46.00	62.68	64.71	56.44	63.74	65.12	63.04	**65.83**	**+16.16%**
	H@10	66.08	63.67	56.79	72.58	74.26	67.33	73.48	74.53	71.97	**74.99**	**+13.48%**
	N@5	47.32	45.27	33.77	51.76	53.06	44.65	53.12	54.46	52.71	**55.33**	**+16.93%**
	N@10	50.35	48.42	37.77	54.97	57.19	48.18	56.23	57.50	55.60	**58.29**	**+15.77%**

Table 4: Results of RECFORMER, original SASRec, and FT Embedding-init (trainable) SASRec under the full-ranking setting (%). Improv. (REC) and Improv. (SAS) represent the relative improvements compared with the RECFORMER and original SASRec respectively.

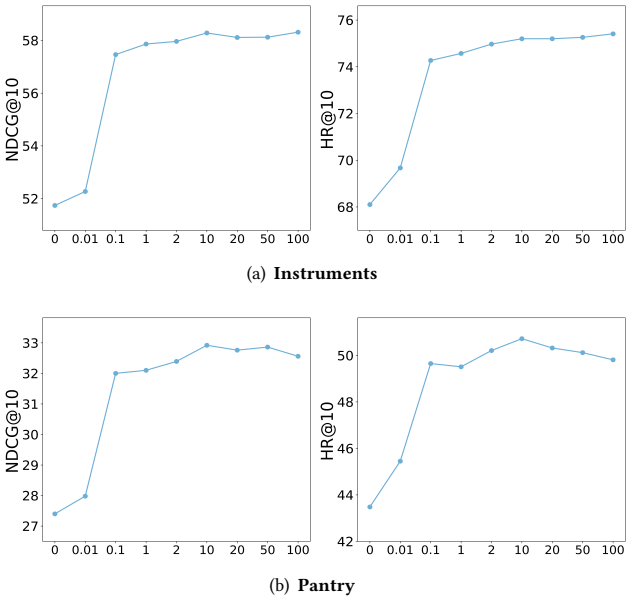
Dataset	Metric	REC	SAS	FT-init	Improv.	
					REC	SAS
Arts	H@5	13.83	13.62	14.46	+4.56%	+6.17%
	H@10	15.61	15.21	16.94	+8.52%	+11.37%
	H@50	21.02	20.46	24.69	+17.46%	+20.67%
	N@5	12.09	11.84	12.29	+1.65%	+3.80%
	N@10	12.66	12.35	13.09	+3.40%	+6.00%
	N@50	13.86	13.50	14.73	+6.28%	+9.11%
Instruments	H@5	20.00	20.04	20.71	+3.55%	+3.34%
	H@10	20.99	21.75	22.68	+8.05%	+4.28%
	H@50	24.49	27.71	29.50	+20.46%	+6.46%
	N@5	18.83	18.28	18.92	+0.48%	+3.50%
	N@10	19.15	18.83	19.54	+2.04%	+3.77%
	N@50	20.18	20.13	20.92	+3.67%	+3.92%
Pantry	H@5	7.75	7.04	9.10	+17.42%	+29.26%
	H@10	9.97	8.37	11.45	+14.84%	+36.80%
	H@50	15.76	13.25	18.31	+16.18%	+38.19%
	N@5	6.13	5.87	6.69	+9.14%	+13.97%
	N@10	6.85	6.30	7.45	+8.76%	+18.25%
	N@50	8.12	7.35	8.96	+10.34%	+21.90%
Sports	H@5	12.49	11.00	12.99	+4.00%	+18.09%
	H@10	13.47	11.65	14.07	+4.45%	+20.77%
	H@50	16.19	13.27	17.36	+7.23%	+30.82%
	N@5	11.10	9.88	11.12	+0.18%	+12.55%
	N@10	11.43	10.09	11.46	+0.26%	+13.58%
	N@50	12.09	10.44	12.18	+0.74%	+16.67%
Movies	H@5	21.48	21.43	21.97	+2.28%	+2.52%
	H@10	23.10	23.48	23.84	+3.20%	+1.53%
	H@50	27.50	30.33	29.95	+8.91%	-
	N@5	19.36	19.05	19.56	+1.03%	+2.68%
	N@10	19.89	19.71	20.15	+1.31%	+2.23%
	N@50	20.85	21.20	21.47	+2.97%	+1.27%

5 demonstrate that employing behavior-tuned PLMs initialized embeddings (i.e., PT and FT) leads to substantial performance enhancements, while embeddings from original PLMs (LF) do not. The results reconfirm that it is the behavior-aware semantics, not the original semantic information, that improves the SR performance.

Cooperating with RECFORMER's sequence modeling. In this evaluation, we set the item embeddings of RECFORMER in its stage-FT2 to be trainable, with other settings unchanged. This cooperation significantly improves over the original RECFORMER on all datasets, indicating that the behavior-tuned PLMs initialized embeddings are beneficial even with PLM-based sequence modeling. However, its performance is still slightly inferior with using SASRec's behavior sequence modeling. It validates the effectiveness of our PLM-based item initialization and further highlights the redundancy of PLMs in behavior sequence modeling. Besides, these results also reveal insufficient behavior information within item representations that are directly obtained from behavior-tuned PLMs. We believe that due to the huge conflicts between the semantic and behavioral representations of items, the superior strategy of representing items with PLM is: ***do not strongly coupled with texts, but using behavior-tuned PLMs to build item initialization***, which highlights behavioral information while also adopting relevant semantic information as an important supplement.

Table 5: Results of different variants for using our proposed behavior-tuned PLMs initialized embeddings (%).

Dataset	Metric	SASRec	with trainable ID			RECFORMER	
			LF	PT	FT	original	trainable
Arts	H@5	58.51	59.04	67.81	71.55	59.60	68.70
	H@10	68.54	69.79	77.74	80.91	68.72	78.01
	N@5	48.35	48.30	55.58	59.83	49.88	58.36
	N@10	51.59	51.77	58.80	62.87	52.48	61.37
Pantry	H@5	31.80	31.71	38.47	38.32	31.37	36.71
	H@10	43.48	43.75	50.61	50.38	41.08	47.82
	N@5	23.62	22.46	29.17	28.88	24.41	28.08
	N@10	27.40	26.43	33.10	32.78	27.54	31.66
Sports	H@5	41.76	39.83	53.95	50.32	45.89	49.84
	H@10	52.16	51.72	65.74	62.10	57.26	61.34
	N@5	33.42	29.16	42.58	38.78	36.86	40.34
	N@10	36.84	32.99	46.40	42.59	40.53	44.06



(a) **Instruments**

(b) **Pantry**

Figure 2: Results of different pre-training dataset sizes. The horizontal axis represents the proportion of the original pre-training dataset (%).

4.6 Analysis on Pre-training Sizes.

The Pre-training has been verified to benefit RECFORMER in SR on new domains. In Table 3, our PT setting also achieves satisfactory

results. In this section, we explore the influence of pre-training dataset size on the effectiveness of our initialization by conducting experiments with PLMs pre-trained on randomly selected subsets of 0.01%, 0.1%, 1%, 2%, 10%, 20%, and 50% from the original pre-training dataset. From Fig. 2, we can observe that leveraging pre-training on behavioral datasets could significantly improve our performance. It is also astonishing that even a small amount of behavior-based pre-training (e.g., involving several thousand users) could successfully add the behavior-level preference to PLMs and achieve large improvement compared to the original Longformer. This observation reinforces the conclusion that the effectiveness of our method is primarily attributed to behavior-aware PLMs rather than the original PLMs. It also suggests that a minimal amount of pre-training behavioral data is possibly sufficient and transferable for new-domain SR tasks, implying that our method is practical and efficient based on a unified behavior-tuned PLM-based model.

5 CONCLUSION and FUTURE WORK

In this work, we deeply explore the effect of PLMs in SR, identifying significant model underutilization and redundancy of PLM in behavior sequence modeling by attention analyses and freezing/simplifying the sequence modeling of PLM. Next, we conduct extensive experiments and analyses on different PLM-based SR variants, discovering that the simplified sequence modeling of conventional ID-based SR models enhanced with the proposed behavior-tuned PLMs initialized item embeddings achieves a giant performance boost, while vanilla PLMs initializations are unable to obtain any improvements. In the future, we will explore the scalability of both SR datasets (including the size and domain diversity) and PLMs, determining whether any scaling laws apply to our proposed initialization. Additionally, identifying more appropriate training objectives to align PLMs with behavioral knowledge for initialization presents a promising avenue for research.

Acknowledgments

This work is supported by the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001)

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnaan He, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434* (2023).

[3] 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. *arXiv preprint arXiv:2305.00447* (2023).

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 138–147.

[7] 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*. 378–387.

[8] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[10] 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).

[11] 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).

[12] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. 2021. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 505–514.

[13] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*. 152–160.

[14] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[15] 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.

[16] Jiayan Guo, Peiyan Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. 2022. Evolutionary preference learning via graph nested gru ode for session-based recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 624–634.

[17] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial feature translation for multi-domain recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2964–2973.

[18] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1096–1102.

[19] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[20] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[21] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[22] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[23] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845* (2023).

[24] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[25] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. *arXiv preprint arXiv:2305.06474* (2023).

[26] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1748–1757.

[27] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731* (2023).

[28] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *arXiv preprint arXiv:2304.10149* (2023).

[29] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 685–694.

[30] Shanlei Mu, Yupeng Hou, Wayne Xin Zhao, Yaliang Li, and Bolin Ding. 2023. ID-Agnostic User Behavior Pre-training for Sequential Recommendation. In *Information Retrieval: 28th China Conference, CCIR 2022, Chongqing, China, September 16–18, 2022, Revised Selected Papers*. Springer, 16–27.

[31] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 188–197.

[32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[33] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.

[34] Zekai Qu, Ruobing Xie, Chaojun Xiao, Yuan Yao, Zhiyuan Liu, Fengzong Lian, Zhanhui Kang, and Jie Zhou. 2023. Thoroughly Modeling Multi-domain Pre-trained Recommendation as Language. *arXiv preprint arXiv:2310.13540* (2023).

[35] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950* (2023).

[36] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[37] 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*. 1441–1450.

[38] 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*. 565–573.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[40] Jie Wang, Fajie Yuan, Mingyue Cheng, Joemon M Jose, Chenyun Yu, Beibei Kong, Zhijin Wang, Bo Hu, and Zang Li. 2022. TransRec: Learning Transferable Recommendation from Mixture-of-Modality Feedback. *arXiv preprint* (2022).

[41] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot Next-Item Recommendation using Large Pretrained Language Models. *arXiv preprint arXiv:2304.03153* (2023).

[42] Nan Wang, Shoujin Wang, Yan Wang, Quan Z Sheng, and Mehmet A Orgun. 2022. Exploiting intra-and inter-session dependencies for session-based recommendations. *World Wide Web* 25, 1 (2022), 425–443.

[43] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Llmrec: Large language models with graph augmentation for recommendation. *arXiv preprint arXiv:2311.00423* (2023).

[44] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[45] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Ao Xiang, Xu Zhang, Leyu Lin, and Qing He. 2022. Selective fairness in recommendation via prompts. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2657–2662.

[46] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[47] Chengfeng Xu, Jian Feng, Pengpeng Zhao, Fuzhen Zhuang, Deqing Wang, Yanchi Liu, and Victor S Sheng. 2021. Long-and short-term self-attention network for sequential recommendation. *Neurocomputing* 423 (2021), 580–589.

[48] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835* (2023).

[49] 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).

[50] 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.