Text Matching Improves Sequential Recommendation by **Reducing Popularity Biases**

Zhenghao Liu*† Northeastern University Shenyang, China liuzhenghao@mail.neu.edu.cn

Xiaohua Li Northeastern University Shenyang, China lixiaohua@mail.neu.edu.cn

Sen Mei* Northeastern University Shenyang, China meisen@stumail.neu.edu.cn

Shi Yu Tsinghua University Beijing, China yus21@mails.tsinghua.edu.cn

Chenyan Xiong Carnegie Mellon University Pittsburgh, United States cx@cs.cmu.edu

Zhiyuan Liu Tsinghua University Beijing, China liuzy@tsinghua.edu.cn

Yu Gu

Northeastern University Shenyang, China guyu@mail.neu.edu.cn

ABSTRACT

This paper proposes Text mAtching based SequenTial rEcommendation model (TASTE), which maps items and users in an embedding space and recommends items by matching their text representations. TASTE verbalizes items and user-item interactions using identifiers and attributes of items. To better characterize user behaviors, TASTE additionally proposes an attention sparsity method, which enables TASTE to model longer user-item interactions by reducing the self-attention computations during encoding. Our experiments show that TASTE outperforms the state-of-the-art methods on widely used sequential recommendation datasets. TASTE alleviates the cold start problem by representing long-tail items using full-text modeling and bringing the benefits of pretrained language models to recommendation systems. Our further analyses illustrate that TASTE significantly improves the recommendation accuracy by reducing the popularity bias of previous item id based recommendation models and returning more appropriate and text-relevant items to satisfy users. All codes are available at https://github.com/OpenMatch/TASTE.

CCS CONCEPTS

Information systems → Recommender systems.

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.

CIKM '23, October 21-25, 2023, Birmingham, United Kingdom

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

Ge Yu

Northeastern University Shenyang, China yuge@mail.neu.edu.cn

KEYWORDS

Sequential Recommendation, Text Matching, Popularity Bias, Long User-Item Interaction Modeling

ACM Reference Format:

Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. 2023. Text Matching Improves Sequential Recommendation by Reducing Popularity Biases. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21-25, 2023, Birmingham, United Kingdom. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnnnnnnnnn

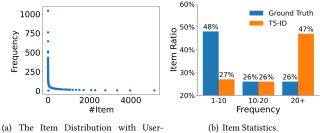
1 INTRODUCTION

Sequential recommendation systems [7, 52, 71] dynamically recommend the next items for users, which play a crucial role in lots of web applications, such as Yelp, TikTok, Amazon, etc. These recommendation systems model user behaviors by employing different neural architectures to learn the dependency among items in the user-item interactions [6, 24, 28, 53]. They usually represent items using ids, randomly initialize item id embeddings, and optimize these embeddings using the signals from user-item interactions.

Existing item id based recommendation systems usually face the popularity bias problem [1, 2, 73]. As shown in Figure 1(a), the distribution of items in recommendation benchmarks is long-tailed, making the item id based recommendation model (T5-ID) usually face the "Cold Start" problem [33, 51]. In Figure 1(b), about 74% golden items are interacted less than 20 times with users, but T5-ID returns more popular items in the recommendation results. It results in the popularity bias-just follow the crowd and return popular items as results. Furthermore, we visualize the embedding distribution of T5-ID in Figure 2(a). T5-ID learns a non-smooth anisotropic embedding space [19, 34, 47], which makes the popular items and other items distinguished. As shown in Figure 2(b), TASTE which represents items with full texts provides some opportunities to alleviate the popularity bias by mixing up the embeddings of popular items and others. It helps the recommendation system return

^{*}indicates equal contribution

[†]indicates corresponding author



Interacted Frequency

Figure 1: Item Distributions with Different User-Item Interaction Frequencies. We sort items according to the user-item interaction frequencies and plot the user-interacted frequency distribution of items in Figure 1(a). Then we conduct item statistics of different user-item interaction frequencies. The ground truth and top-5 items that are predicted by T5-ID are shown in Figure 1(b). T5-ID is an item id based recommendation model, which predicts the id of the next item. The items are divided into different groups according to the frequency.

more text-relevant but long-tailed items for users by matching text representations of users and items.

In this paper, we propose Text mAtching based SequenTial rEcommendation (TASTE), which represents items and users with texts, establishes relevance between them by matching their text representations and alleviates the popularity bias of previous item id based recommendation models. TASTE verbalizes users and items by designing some prompts [39] and filling up the templates with item ids and item attributes. The item information provides textmatching clues to model the dependency and relevance among users and items. TASTE also proposes an attention sparsity encoding technology to break the max length boundary of language models [3] and encode long text representations of user-item interactions. It separates the user-item interaction into different sessions, independently encodes the text representation of each session, and reduces the attention computation.

Our experiments demonstrate that TASTE surpasses previous sequential recommendation baselines [52, 58, 71] with over 18% improvements on Yelp and Amazon Product datasets. Our analyses show that, compared with item id based recommendation models, TASTE shows its ability in alleviating the popularity bias problem and recommends more than 18% long-tail items for users compared to T5-ID. Our text based recommendation modeling enables TASTE to model the relevance between users and items by capturing the text matching signals. It helps to learn more effective representations for these long-tail items, alleviates the "cold start" problem, and makes TASTE return more text-relevant items as the recommendation result. Thrives on our attention sparsity method, TASTE has the ability to reduce the GPU memory and achieves more than 2% improvements by modeling longer user-item interaction history. All experimental results show that the behavior modeling of user purchases and visits starts to be benefited from pretrained language models.

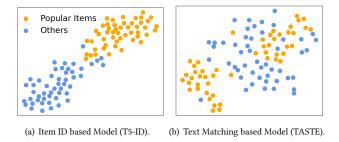


Figure 2: Embedding Visualization of Items in Beauty. We randomly select 50 items from the popular item set and select 50 items from the rest. We use t-SNE to visualize the item embedding spaces of the item id based recommendation model and text matching based recommendation model in Figure 2(a) and Figure 2(b), respectively.

RELATED WORK

Sequential recommendation systems attempt to model user behaviors according to the user-item interaction history. Early work [22, 49] adopts Markov Chain assumption and Matrix Factorization Methods to recommend items. Existing sequential recommendation systems employ Convolutional Neural Networks (CNNs) [53, 60, 63], Recurrent Neural Networks (RNNs) [16, 24, 40], Graph Neural Networks (GNNs) [4, 6, 67] or the self-attention architecture [17, 28, 52] to capture the dependencies among items in the user-item interaction sequences and predict the next item.

These recommendation systems represent items with only their ids and learn relevance among items using user-item interactions, even with Transformers. Bert4Rec [52] represents items with randomly initialized embeddings and pretrains self-attention heads [54] and item embeddings by mask language modeling [13]. P5 [20] represents items using utterances of their ids. It converts multiple recommendation tasks as seq2seq tasks and continuously trains T5 model [48] to generate the text representations of item ids that will be interacted with users in the next step.

In addition to representing items with identifiers, using item attributes shows convincing recommendation results by modeling item dependency and user-item relevance through the side information [38, 58, 71]. Existing work incorporates item attributes and focuses more on fully using additional information to identify the user intentions from the attributes of user-interacted items [68]. Moreover, some work [38, 58] further builds sophisticated attention mechanisms to denoise item attributes and focuses more on fusing side information to enhance the representations of items. Zhou et al. [71] also design an additional pretraining task to learn the embeddings for item attributes and identifiers, aiming to more effectively capture the correlation among item identifiers, attributes, and users. Nevertheless, these sequential recommendation systems often rely on randomly initialized embeddings to represent item attributes and identifiers, making it challenging to leverage the advantages of pretrained language models for enhancing sequential recommendation tasks.

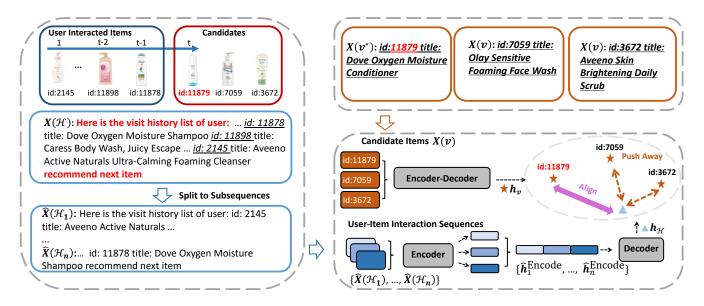


Figure 3: The Framework of Text mAtching based SequenTial rEcommendation (TASTE).

Instead of using randomly initialized item embeddings, fully text-based item modeling recently emerges in recommendation systems [15, 25, 37, 65]. It has proven that these randomly initialized item embeddings heavily depend on training with sufficient useritem interactions and usually face the cold start problem during representing long-tail items [33, 44, 51, 69]. Li et al. [37] explore the potential of ID-free style recommendation modeling. They propose a full text matching based method to alleviate the cold start problem and benefit the cross-domain recommendations. They employ pretrained language models to provide the token embeddings of the text sequences of items for downstream recommendation models. Yuan et al. [65] further discuss the advantages of id-based and modality-based recommendations, which represent items by randomly initialized embeddings and encoding multi-modal side information with pretrained language models, respectively. They confirm that id-based recommendation is usually effective when the user-item training signal is sufficient and modality-based recommendation can alleviate the cold start problem. However, Yuan et al. [65] do not fully integrate both id-based and modality-based item representations, which could potentially contribute to further enhancing recommendation performance.

Modeling long-term user-item interaction can better characterize the user behaviors and improve the recommendation accuracy [45]. However, using side information of items to verbalize user-item interactions makes the text sequences long and challenges existing language models [13, 48, 54]. The work [5, 46] tries to filter out unrelated items from the user-item interaction history to reduce the user-item interaction length. Nevertheless, it is inevitable to lose some user preference information during modeling user intentions. To model long text sequences, the work in other research directions shows some successful attempts. Some of them divide the long sequences into segments and separately encode them using pretrained language models and then fuse them [21, 27, 35]. The others make the attention sparse [3, 9, 31, 66] or adopt low rank

approximate [10, 30, 55] to reduce the self-attention computations. Different from the work, TASTE separates the user-item interaction into different sessions and employs the fusion-in-decoder architecture [27] to model the long text sequences.

3 METHODOLOGY

This section introduces our Text mAtching based SequenTial rEcommendation (TASTE) model, which is illustrated in Figure 3. We first describe how to verbalize users and items and then model the sequential recommendation using text matching (Sec. 3.1). Finally, TASTE proposes an attention sparsity method to encode long text sequences of user-item interactions (Sec. 3.2).

3.1 Text Matching based Sequential Recommendation Modeling

Given a user-item interaction history $\mathcal{H} = \{v_1, v_2, \dots, v_{t-1}\}$, the sequential recommendation task aims to recommend the item v_t to satisfy the user's needs at t-th time. TASTE encodes \mathcal{H} and item v using pretrained language model T5 [48] and models the relevance between users and items by matching their text representations.

Text Representation. For each item v, we can verbalize it using item ids and k item attributes <Attr> using the following template:

$$X(v) = id:v(id)..._k : v(_k),$$
(1)

where ">k is the name of attribute. v(id) and v(">k) are the text descriptions of item identifier and <math>i-th attribute of item v. In this case, one item in Yelp can be verbalized: "id: 5908 title: DoMazing address: 6659 S Las Vegas Blvd, Ste B-101 Las Vegas NV". Here v(id) is a kind of id prompt to help language models to capture matching signals among users and items, which is beyond the descriptions of item attributes.

Then we verbalize the user-item interaction history \mathcal{H} with the template: "Here is the visit history list of user: $X(\mathcal{H})$ recommend

next item". It demonstrates the definition of sequential recommendation and helps pretrained language models better characterize model behaviors [11, 20]. $X(\mathcal{H})$ is the concatenation of verbalized items $\{v_1, v_2, \ldots, v_{t-1}\}$ that are in \mathcal{H} :

$$X(\mathcal{H}) = X(v_{t-1}); ...; X(v_1),$$
 (2)

where; denotes the concatenation operation. We reverse the item sequences for truncating more previously interacted items.

Encoding. We use T5 [48] to encode the user-item interaction history \mathcal{H} and item v as embeddings $h_{\mathcal{H}}$ and h_v , using the representation of the first input token from the T5 decoder [43]:

$$h_{\mathcal{H}} = \text{T5}(X(\mathcal{H})); h_v = \text{T5}(X(v)). \tag{3}$$

The relevance between user-item interaction history \mathcal{H} and item v can be calculated using the ranking probability $P(v|\mathcal{H})$ between the encoded representations $h_{\mathcal{H}}$ and h_v :

$$P(v|\mathcal{H}) = \text{Softmax}_v(h_{\mathcal{H}} \cdot h_v),$$
 (4)

where \cdot is the dot product operation.

Training. We optimize the model parameters using the following loss function:

$$\mathcal{L} = \text{CrossEntropy}(P(v|\mathcal{H}), v^*), \tag{5}$$

where v^* denotes the ground truth item that is interacted by the user at the t-th time. We use in-batch negatives and randomly sampled negatives [14, 23, 50] to contrastively train models.

3.2 Long Text Encoding for User-Item Interactions with Attention Sparsity

In real-world scenarios, purchase or visit histories typically involve long-term interactions. The longer interaction usually contains more information to better model user behaviors and achieve more accurate recommendation results [45]. Instead of only using item ids to model user behavior [20], TASTE verbalizes the items in useritem interactions (Eq. 2) and makes the text utterance $X(\mathcal{H})$ longer, which challenges the long text processing ability of pretrained language models due to the max length boundary [13, 48, 54].

As shown in the bottom part of Figure 3, our attention sparsity mechanism learns session-based user behaviors by employing a fusion-in-decoder architecture [27]. We first split the text sequence $X(\mathcal{H})$ of user-item interaction history into n sub-sequences $\hat{X}(\mathcal{H}) = \{\hat{X}(\mathcal{H})_1,...,\hat{X}(\mathcal{H})_n\}$. $\hat{X}(\mathcal{H})_i$ contains m tokens, which is regarded as a user-item interaction session. It reflects the user preferences of a specific period and can be characterized using a set of items that are interacted with users in the session [18, 36, 57]. We use T5-Encoder to independently encode these subsequences:

$$\hat{h}_{i}^{\text{Encode}} = \text{T5-Encoder}(\hat{X}(\mathcal{H})_{i}).$$
 (6)

Then we concatenate $\hat{h}^{\text{Encode}} = \{\hat{h}_1^{\text{Encode}}; \hat{h}_2^{\text{Encode}}; ...; \hat{h}_n^{\text{Encode}}\}$ as the hidden state of the user-item interaction history \mathcal{H} . It eliminates attention computations among different sessions within Transformer encoder modules, thereby reducing the encoding computational complexity from $O(n^2)$ to O(n). This approach also facilitates the encoding of longer text sequences of user-item interactions.

Finally, we can get the representation of user-item interaction by feeding the sparsely encoded user-item interaction sequences to the decoder module of T5:

$$h_{\mathcal{H}} = \text{T5-Decoder}(\hat{h}^{\text{Encode}}, h_0^{\text{Decode}}),$$
 (7)

Table 1: Statistics of Preprocessed Datasets.

Dataset	Da	ta Inform	ation	Split		
Dataset	#Users	#Items	#Actions	Train	Dev	Test
Beauty	22,363	12,101	198,502	131,413	22,363	22,363
Yelp	30,499	20,068	317,182	225,685	30,499	30,499
Sports	35,598	18,357	296,337	189,543	35,598	35,598
Toys	19,412	11,924	167,597	109,361	19,412	19,412

where $h_0^{\rm Decoder}$ is the token embedding of the first input token of the decoder, which is the same as Eq. 3. T5-Decoder uses the cross-attention mechanism to reweight different user-item interaction sessions and model the user behaviors by capturing text-matching signals from all tokens in the verbalized user-item interactions.

4 EXPERIMENTAL METHODOLOGY

This section describes datasets, evaluation metrics, baselines, and implementation details in our experiments.

Dataset. Four sequential recommendation datasets are used in our experiments, including Yelp¹, Amazon Beauty, Sports and Toys [42], which aim to recommend locations and Amazon products² to satisfy users. All data statistics are shown in Table 1.

We use Recbole [70] to process all datasets and keep all experiment settings the same as previous work [58, 71], which allows us to directly compare TASTE with baseline models [58]. For all datasets, we filter out the items and users that have less than five user-item interaction times in the whole datasets and treat all user-item interactions as implicit feedback [7, 52, 58, 71]. After data processing, each example can be a user-item interaction sequence $\mathcal{H} = \{v_1, v_2, \dots, v_T\}$. Then we use the leave-one-out evaluation strategy [7, 52, 58, 71], and separate the processed datasets into training, development, and testing sets. We construct the testing and development sets by using $v_{1,\dots,T-1}$ to predict v_T and using $v_{1,\dots,T-2}$ to predict v_{T-1} , respectively. For the training set, we follow Zhao et al. [70] and Xie et al. [58] to use interaction history $v_{1,\dots,i-1}$ to predict v_i , where 1 < i < T-1.

Evaluation Metrics. We utilize the same evaluation metrics as DIF-SR [58] and use Recall@10/20 and NDCG@10/20 to evaluate the recommendation performance of different models. Statistic significances are tested by permutation test with P< 0.05.

During evaluating models, we follow DIF-SR [58] and employ a full ranking testing scenario [12, 32]. Some work evaluates model performance on a small item subset by randomly sampling or sampling items according to item popularity, making the evaluation results inconsistent of the same model. Instead of reranking items in the sampled subset, some work [12, 32] builds a more realistic recommendation evaluation setting by ranking all items and choosing the top-ranked items as recommendation results.

Baselines. Following our main baseline model [58], we compare TASTE with several widely used sequential recommendation models. GRU4Rec [24] uses RNN to model user-item interaction sequences for recommendation. SASRec [28] and Bert4Rec [52] employ the self-attention mechanism to capture user preferences from user-item interaction sequences. To better capture relevance

¹https://www.yelp.com/dataset

²http://jmcauley.ucsd.edu/data/amazon/

Table 2: Overall Performance. We keep the same experimental settings and report the scores of baselines from previous work [58]. Underlined scores are the highest results of baselines. \dagger , \S , \ddagger indicate statistically significant improvements over DIF-SR † , T5-DPR § and T5-ID ‡ , respectively. We also show relative improvements over DIF-SR.

Dataset	Metrics	GRU4Rec	Bert4Rec	SASRec	S ³ Rec	NOVA	ICAI-SR	DIF-SR	T5-DPR	T5-ID	TAST	ΓE
	Recall@10	0.0530	0.0529	0.0828	0.0868	0.0887	0.0879	0.0908	0.0716	0.0785 [§]	0.1030†‡§	13.44%
Beauty	Recall@20	0.0839	0.0815	0.1197	0.1236	0.1237	0.1231	0.1284	0.1082	0.1138^{\S}	0.1550 ^{†‡§}	20.72%
Beauty	NDCG@10	0.0266	0.0237	0.0371	0.0439	0.0439	0.0439	0.0446	0.0345	0.0440^{\S}	0.0517 ^{†‡§}	15.92%
	NDCG@20	0.0344	0.0309	0.0464	0.0531	0.0527	0.0528	0.0541	0.0437	0.0529^{\S}	0.0649 ^{†‡§}	19.96%
	Recall@10	0.0312	0.0295	0.0526	0.0517	0.0534	0.0527	0.0556	0.0329	0.0464 [§]	0.0633†‡§	13.85%
Cnarta	Recall@20	0.0482	0.0465	0.0773	0.0758	0.0759	0.0762	0.0800	0.0554	0.0689^{\S}	0.0964 ^{†‡§}	20.50%
Sports	NDCG@10	0.0157	0.0130	0.0233	0.0249	0.0250	0.0243	0.0264	0.0159	0.0252^{\S}	0.0338†‡§	28.03%
	NDCG@20	0.0200	0.0173	0.0295	0.0310	0.0307	0.0302	0.0325	0.0216	0.0308^{\S}	0.0421†‡§	29.54%
	Recall@10	0.0370	0.0533	0.0831	0.0967	0.0978	0.0972	0.1013	0.0805 [‡]	0.0754	0.1232†‡§	21.62%
Toys	Recall@20	0.0588	0.0787	0.1168	0.1349	0.1322	0.1303	0.1382	0.1243^{\ddagger}	0.1065	0.1789 ^{†‡§}	29.45%
10ys	NDCG@10	0.0184	0.0234	0.0375	0.0475	0.0480	0.0478	0.0504	0.0375	0.0421^{\S}	0.0640†‡§	26.98%
	NDCG@20	0.0239	0.0297	0.0460	0.0571	0.0567	0.0561	0.0597	0.0485	0.0499	0.0780 ^{†‡§}	30.65%
	Recall@10	0.0361	0.0524	0.0650	0.0589	0.0681	0.0663	0.0698	0.0460	0.0450	0.0738 ^{†‡§}	5.73%
Voln	Recall@20	0.0592	0.0756	0.0928	0.0902	0.0964	0.0940	0.1003	0.0740	0.0745	0.1156 ^{†‡§}	15.25%
Yelp	NDCG@10	0.0184	0.0327	0.0401	0.0338	0.0412	0.0400	0.0419	0.0250^{\ddagger}	0.0235	$0.0397^{‡\$}$	-5.25%
	NDCG@20	0.0243	0.0385	0.0471	0.0416	0.0483	0.0470	0.0496	0.0320	0.0309	0.0502 ^{‡§}	1.21%

among attributes, items, and users, S³Rec [71] comes up with four self-supervised methods to pretrain self-attention modules. ICAI-SR [64] is compared, which proposes a heterogeneous graph to represent the relations between items and attributes to model item relevance. Besides, NOVA [38] is also compared, which builds attention modules to incorporate item attributes as the side information. DIF-SR [58] is the previous state-of-the-art, which builds a non-invasive attention mechanism to fuse information from item attributes during modeling user behaviors.

Besides, we implement a T5-ID model, which encodes user-item interaction sequences to generate the identifier of the next item [20]. We follow the previous dense retrieval model, DPR [29], to implement T5-DPR, which regards the text sequences of user-item interaction history and items as queries and documents, encodes them with T5 and is trained with in-batch sampled negatives.

Implementation Details. Different from previous work [28, 52, 58], we represent items using full-text sequences (Eq. 1) instead of randomly initialized item embeddings. In our experiments, we use names and addresses to represent locations in the Yelp dataset. And we only use product names to represent shopping products in Amazon Product datasets, because the product name usually contains the item attributes, such as "WAWO 15 Color Professional Makeup Eyeshadow Camouflage Facial Concealer Neutral Palette" contains both category and brand.

Our models are implemented with OpenMatch [41, 62]. In our experiments, we truncate the text representations of items by 32 tokens and set the max length of the text representation of user-item interactions to 512. The user-item interaction sequence is split into two subsequences to make the attention sparse in T5-encoder (Eq. 6). TASTE is initialized with the T5-base checkpoint from huggingface transformers [56]. During training, we use Adam optimizer and set learning rate=1e-4, warm up proportion=0.1, and batch size=8. Besides, we use in-batch negatives and randomly sampled negatives to optimize TASTE. We randomly sample 9 negative items for each training example and in-batch train TASTE model.

5 EVALUATION RESULT

In this section, we first evaluate the recommendation performance of TASTE and conduct ablation studies. Then we study the effectiveness of different item verbalization methods, the ability of TASTE in reducing the popularity bias, the advantages of TASTE in representing long-tail items, and modeling user behaviors using longer user-item interactions. Finally, we present several case studies.

5.1 Overall Performance

The recommendation performance of TASTE is shown in Table 2. Overall, TASTE significantly outperforms baseline models on all datasets by achieving 18% improvements.

Compared with item id embedding based recommendation models, e.g. Bert4Rec, TASTE almost doubles its recommendation performance, thriving on modeling relevance between users and items through text-matching signals. TASTE also outperforms the previous state-of-the-art recommendation model, DIF-SR, which leverages item attributes as side information to help better learn user behaviors from user-item interactions beyond item identifiers. It represents item attributes as embeddings and focuses more on decoupling and fusing the side information in item representations. Instead of designing sophisticated architectures for fusing side information, TASTE employs a general template to verbalize items and users, leveraging pretrained attention heads of T5 to match the textual representations of users and items. It demonstrates the direct advantages of utilizing pretrained language models for recommendation systems.

5.2 Ablation Study

In this experiment, we further explore the effectiveness of prompt modeling, different negative sampling strategies, and our long useritem interaction modeling method.

We first conduct several experiments by in-batch training TASTE with 9 additional negatives that are from popular sampling, random

Table 3: Ablation Study. We show the effectiveness of different negative sampling strategies during training TASTE, including Popular Negs, Hard Negs, and Random Negs. The popular negatives are selected from the top-500 items that are more frequently interacted with users. The Longer History uses our attention sparsity strategy for modeling longer user-item interactions.

D.44	Matrica	TASTE w/o Prompt		TAS	TE		TASTE (Rand Negs)
Dataset	Metrics	(Inbatch)	Inbatch	Popular Negs	ANCE	Random Negs	w/ Longer History
	Recall@10	0.0441	0.0460	0.0351	0.0324	0.0726	0.0733
Yelp	Recall@20	0.0714	0.0740	0.0550	0.0519	0.1131	0.1150
reip	NDCG@10	0.0235	0.0250	0.0186	0.0173	0.0388	0.0393
	NDCG@20	0.0304	0.0320	0.0236	0.0222	0.0489	0.0498
	Recall@10	0.0327	0.0329	0.0284	0.0354	0.0510	0.0545
Recall@20	Recall@20	0.0550	0.0554	0.0476	0.0550	0.0812	0.0851
Sports	NDCG@10	0.0155	0.0159	0.0135	0.0182	0.0249	0.0270
ND	NDCG@20	0.0211	0.0216	0.0184	0.0232	0.0325	0.0346
	Recall@10	0.0688	0.0716	0.0601	0.0750	0.0921	0.0935
Beauty	Recall@20	0.1085	0.1082	0.0950	0.1147	0.1401	0.1441
Deauty	NDCG@10	0.0335	0.0345	0.0294	0.0368	0.0444	0.0445
	NDCG@20	0.0435	0.0437	0.0382	0.0468	0.0565	0.0573
	Recall@10	0.0777	0.0805	0.0678	0.0878	0.1032	0.1056
Т	Recall@20	0.1222	0.1243	0.1065	0.1297	0.1577	0.1594
Toys	NDCG@10	0.0369	0.0375	0.0326	0.0426	0.0488	0.0508
	NDCG@20	0.0480	0.0485	0.0423	0.0532	0.0625	0.0643

Table 4: Performance of the Attention Sparsity Module of TASTE. We evaluate TASTE on Yelp by splitting user-item interactions into different sequences.

Model	Seq Length	Recall@20	NDCG@20	Memory
	256	0.1076	0.0468	9.52GB
	128×2 seqs	0.1056	0.0453	9.18GB
TASTE	64×4 seqs	0.1097	0.0472	9.05GB
TASTE	512	0.1142	0.0499	13.09GB
	256×2 seqs	0.1156	0.0502	11.84GB
	128×4 seqs	0.1090	0.0470	11.16GB

sampling, and hard negative sampling. The popular negatives are sampled according to the item appearance frequencies in all useritem interaction sequences. The items that are higher-frequently interacted with users means more "popular" items. During sampling popular negatives, we follow Sun et al. [52] to build the popular item set, which consists of 500 items that are more frequently interacted with users. The hard negatives are sampled from TASTE (Inbatch), which are more informative to avoid vanishing gradients [59]. For each user, all user-interacted items are filtered out from the negative item set. As shown in Table 3, using additional negatives that are sampled from popular items slightly decreases the recommendation performance of TASTE (Inbatch). The reasons may lie in that high-frequently interacted items indicate the general interests of users and can be easily interacted with different users. TASTE (ANCE) outperforms TASTE (Inbatch) while showing less effectiveness than TASTE (Rand Negs). This phenomenon illustrates that

modeling longer-term user-item interactions.

Finally, we show the effectiveness of our attention sparsity method in Table 4. In our experiments, we maintain maximum text sequence

these hard negatives are also high-potentially interacted with users

and are usually not the real negatives. Besides, benefiting from our attention sparsity method, TASTE achieves further improvements,

demonstrating its effectiveness in characterizing user behaviors by

Table 5: Effectiveness of Attribute Information. We conduct several experiments on Yelp by filling in the templates using names, addresses, and categories to represent items.

Attribute Info	Recall@20	NDCG@20
T5-DPR w/ Item ID	0.0687	0.0313
+ Name	0.0915	0.0415
+ Name & Category	0.0909	0.0410
+ Name & Address	0.1057	0.0451
+ Name & Address & Category	0.1107	0.0475

lengths of 256 and 512 for user-item interaction history, splitting them into 4 or 2 subsequences. Then we evaluate the memory usage and recommendation performance of TASTE. Overall, by adding more subsequences or extending their lengths to model additional user-interacted items, TASTE achieves more accurate recommendation results. It demonstrates that modeling longer user-item interaction sequences can help to better characterize user behaviors [45]. When the sequences are separated into different numbers of subsequences, TASTE reduces the GPU memory usage and achieves even slightly better recommendation performance. Our attention sparsity method has the ability to reduce the self-attention computations and potentially break the boundary of existing pretrained language models to model long user-item interactions, which is important in more realistic scenarios that have sufficient product purchase history and restaurant visiting history.

5.3 Effectiveness of Item Verbalization Methods

In this experiment, we show the recommendation effectiveness of TASTE by using different item verbalization methods. We first study the effectiveness of item attributes and then model user/item ids using different methods. Finally, the recommendation behaviors of different item modeling methods are further explored.

Table 6: Recommendation Performance of Different Identifier Modeling Strategies. The evaluation results of two identifier modeling methods, Embed and Prompt, are shown. Embed randomly initializes the embeddings of item/user, while Prompt verbalizes the identifiers in the text space. TASTE only models the item identifiers using the Prompt method.

Dataset	Metrics	TASTE	w/ U:	ser ID	w/ Ite	em ID
Dataset	Metrics	w/o ID	Embed	Prompt	Embed	Prompt
	Recall@10	0.0935	0.0600	0.0910	0.0743	0.1030
Doorster	Recall@20	0.1441	0.0982	0.1399	0.1138	0.1550
Beauty	NDCG@10	0.0445	0.0289	0.0437	0.0374	0.0517
	NDCG@20	0.0573	0.0385	0.0560	0.0474	0.0649
	Recall@10	0.0545	0.0358	0.0522	0.0427	0.0633
Sports	Recall@20	0.0851	0.0581	0.0817	0.0661	0.0964
	NDCG@10	0.0270	0.0184	0.0258	0.0218	0.0338
	NDCG@20	0.0346	0.0240	0.0332	0.0277	0.0421
	Recall@10	0.1056	0.0657	0.1021	0.0750	0.1232
Torre	Recall@20	0.1594	0.1068	0.1552	0.1169	0.1789
Toys	NDCG@10	0.0508	0.0318	0.0486	0.0382	0.0640
	NDCG@20	0.0643	0.0422	0.0620	0.0488	0.0780
V-1	Recall@10	0.0733	0.0503	0.0731	0.0492	0.0738
	Recall@20	0.1150	0.0842	0.1150	0.0817	0.1156
Yelp	NDCG@10	0.0393	0.0265	0.0401	0.0241	0.0397
	NDCG@20	0.0498	0.0350	0.0506	0.0322	0.0502

Side Information Modeling. In Table 5, we illustrate the effectiveness of different item attributes in representing items in the Yelp dataset. We do not conduct experiments on Amazon Products, because the names of these products usually contain the information of other attributes. Our experimental results show distinct effects on the two item attributes. The reason might be that location information potentially indicates user staying areas, offering more text-matching signals for user-item relevance modeling. However, the category information consists of discrete words like "Greek," "Mediterranean," and "Restaurants," rather than sentences that the model finds easier to understand. Additionally, some of these words are partially included within the name attributes. Therefore, modeling category information has an adverse effect. TASTE can be easily improved by filling attributes in the prompt template, demonstrating its expandability in modeling side information.

User/Item Identifier Modeling. Then we further explore the recommendation effectiveness of TASTE using different user/item id modeling methods. The user/item ids are crucial to modeling the relevance and dependency among users and items [26, 61]. Here we explore two methods to model user/item ids to boost TASTE, including Embed and Prompt. We follow Yuan et al. [65] to implement the Embed method. We first randomly initialize the id embeddings of users/items, encode the text utterances of item attributes and sum the id embedding and attribute embedding as the representations of users/items. Different from the Embed method, Prompt follows previous work [20], verbalizes the item ids using texts (Eq. 1), and modifies the template by adding user ids, such as "Here is the visit history list of user_1401:". The Prompt method regards user/item ids as a part of the templates in prompt learning.

The experimental results are shown in Table 6. The recommendation performance of TASTE is indeed increased with the help of item ids. It shows that the item ids probably provide additional

Table 7: Recommendation Behaviors with Different Item Verbalization Methods. Top-5 recommended items are used for evaluations. Bleu (↑) and Dist (↑) are used for evaluating the relevance and diversity of the text representations of items in recommendation results, respectively. Popular calculates the ratios of items, which are top-500 items that are more frequently interacted with users.

Dataset	Metrics	T5-ID	TASTE w/o ID	TASTE
	Dist-1	0.1456	0.1385	0.1398
	Dist-2	0.5003	0.4452	0.4641
Beauty	Bleu-4	0.0267	0.0321	0.0322
	Recall-5	0.0509	0.0540	0.0639
	Popular	52%	29%	31%
	Dist-1	0.1740	0.1571	0.1623
Consulta	Dist-2	0.5668	0.4917	0.5193
Sports	Bleu-4	0.0130	0.0153	0.0163
	Recall-5	0.0304	0.0326	0.0403
	Popular	62%	36%	33%
	Dist-1	0.1728	0.1564	0.1601
Т	Dist-2	0.5268	0.4493	0.4697
Toys	Bleu-4	0.0467	0.0566	0.0588
	Recall-5	0.0507	0.0636	0.0793
	Popular	38%	29%	29%
	Dist-1	0.1402	0.1363	0.1351
Val.	Dist-2	0.4367	0.4184	0.4177
Yelp	Bleu-4	0.0479	0.0542	0.0540
	Recall-5	0.0276	0.0465	0.0469
	Popular	46%	30%	30%

text-matching signals to model dependency and relevance among items and users, making TASTE return more appropriate items as the recommendation results. On the other hand, user ids play different roles in product recommendation (Beauty, Sports, and Toys) and restaurant recommendation (Yelp). The main reason may lie in that shopping behaviors are less personalized and can be usually described by the shopping history. While modeling restaurant visiting behaviors needs more to memorize some characteristics of users, such as the taste, preferred cuisines, and active area of users. Besides, the prompt-based modeling method (Prompt) outperforms the embedding-based method (Embed), which illustrates that pretrained language models have the ability to understand the user/item identifiers and establish relevance between users and items via identifiers. It further supports the motivation of TASTE, which fully uses the learned knowledge from pretrained language models to build sequential recommendation systems.

Evaluation on Recommendation Behaviors. Finally, we explore the recommendation behaviors of TASTE using different item modeling methods. As shown in Table 7, three models, T5-ID, TASTE w/o ID, and TASTE, are compared. T5-ID randomly initializes item embeddings and directly predicts the item ids. TASTE w/o ID and TASTE employ a two-tower architecture [29] and encode items using attributes and identifiers & attributes, respectively. As shown in our evaluation results, T5-ID returns an average of 49.5% of popular products in the recommendation results of all datasets, showing that it faces the popularity bias problem during recommending items. TASTE alleviates the popularity bias by reducing

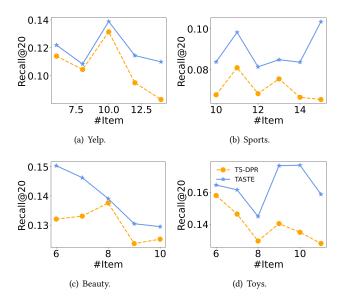


Figure 4: Effectiveness of TASTE with Different Numbers of User-Interacted Items. Recall@20 scores are plotted along different numbers of user-interacted items.

on average 18.75% popular items in its recommendation results. It represents items using full texts and utilizes text matching to calibrate the popularity-oriented recommendation behavior of T5-ID. TASTE demonstrates its effectiveness in recommending more appropriate and text-relevant items by achieving higher Bleu scores and Recall scores. Besides, compared with TASTE w/o ID, TASTE achieves higher Bleu and Dist scores with the help of item ids. It shows that the item ids can serve as a kind of prompt to provide additional matching signals beyond item attributes to better model the relevance between users and items.

5.4 Effectiveness of TASTE on Long-Tail Items and Long-Term Interactions

This experiment evaluates the effectiveness of TASTE on modeling long-tail items and long-term user-item interactions.

As shown in Figure 4, we first evaluate the recommendation performance of T5-DPR and TASTE with different numbers of interacted items. Overall, TASTE shows its advantages in modeling user-item interactions by outperforming T5-DPR with different numbers of user-interacted items. For Amazon Product datasets, TASTE achieves more improvements over T5-DPR when modeling longer user-item interactions. It thrives on our attention sparsity method and showcases the ability to accurately capture text-matching signals from longer-term user-item interactions. Different from the recommendation performance on Amazon Products, TASTE shows less effectiveness on the Yelp dataset (Figure 4(a)) with longer user-item history. It shows that the restaurant visiting behaviors are hard to be characterized with only interacted items and should be specifically modeled for each person, such as modeling user characteristics using user identifier embeddings (Table 6).

Then we evaluate the recommendation performance of TASTE on these items with different user-interacted frequencies in Figure 5.

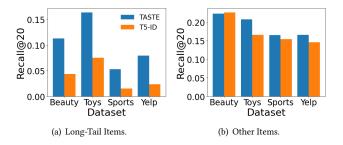


Figure 5: Recommendation Effectiveness on Items with Different User-Interacted Frequencies. We follow Zhu et al. [72], and split the testing sets into two groups according to the user-interacted frequencies of the items of test labels. The items are sorted according to user-item interaction frequencies and are grouped into long-tail items and others by setting the item number ratio to 2:8 [8].

The items are divided into two groups according to user-interacted frequencies, including long-tail items and others that are more frequently interacted with users. TASTE shows more significant improvements over T5-ID on these long-tail items, which illustrates its few-shot effectiveness in learning representations of items using their identifiers and attributes to alleviate the "Cold Start" [33, 51] problem in recommendation systems. Such an encouraging phenomenon demonstrates that TASTE can broaden the advantages of pretrained language models to these long-tail items and shed some lights to build a self-contained recommendation system by directly modeling user-item relevance by text matching.

5.5 Case Studies

In Table 8, we present two cases from Amazon Products and Yelp to study the recommendation effectiveness of TASTE. The top-5 predicted items of T5-ID, TASTE w/o ID, and TASTE are shown.

As shown in the first case, the user usually buys some "Shampoo" and "Body Wash" products for men. During predicting the next item, T5-ID, TASTE w/o ID, and TASTE show distinct recommendation behaviors. T5-ID prefers to predict items that are more related with the last user-interacted item and returns a series of products of the same brand "Axe". Compared with T5-ID, TASTE w/o ID has the ability to recommend more text-relevant items by matching the text representations of users and items. It makes TASTE w/o ID prefer to predict the historically bought items, leading to conventional recommendation results. After adding item ids as prompts, TASTE has the ability to better model item dependencies by recognizing that the user has bought a "Shower Tool" as the last step and the next item should be related to "Body Wash" instead of "Shampoo". Besides, compared with T5-ID, TASTE can better pick up the text clues to better model the relevance between items and shopping behavior, such as "men", "3 in 1", and "Pack of 3".

In the second case, we can find that the user usually visits the coffee bar and dessert bar, such as "Starbucks", "DoMazing" and "Meet Fresh". Unfortunately, the T5-ID model seems to fail to learn such user characteristics by returning the restaurants of "Steakhouse", "Hot Dog" and "Pizza". Thanks to our text matching based

Table 8: Case Studies. We present two cases from Beauty and Yelp and show top-5 retrieved items from TASTE, T5-ID, and TASTE w/o ID. The matched text contents are emphasized.

Case #1 in Beau	ıty
History	id: 10694 title: Axe 3 in 1 Shampoo Plus Conditioner Plus Bodywash Total Fresh, 12 Ounce, id: 4271 title: Neutrogena Ageless
	Essentials Continuous Hydration, Night, 1.7 Ounce, id: 3336 title: Zeno Mini Acne Clearing Device, White, id: 2778 title: Nivea
	For Men Energy Hair and Body Wash, 16.9-Ounce Bottle (Pack of 3), id: 3182 title: Axe Detailer Shower Tool-Colors May Vary
Label	id: 10918 title: Nivea 3-in-1 Pure Impact Body Wash for Men, 16.9 Ounce (Pack of 3)
	1st: id: 10844 title: Axe Shower Gel Deep Space, 16 Ounce
	2nd: id: 10811 title: Axe Shower Gel Apollo, 16 Ounce
T5-ID	3rd: 10538 title: Dove Men+Care Sensitive + Face Lotion 1.69 FL.Oz.
	4th: 10701 title: Dove Men+Care Anti Dandruff Fortifying Shampoo, 12-Ounces
	5th: 10570 title: Axe Energizing Face Wash, Boost, 5 Ounce
	1st: id: 10694 title: Axe 3 in 1 Shampoo Plus Conditioner Plus Bodywash Total Fresh, 12 Ounce
	2nd: id: 10693 title: Axe 2 in 1 Shampoo Plus Conditioner Apollo, 12 Ounce
TASTE w/o ID	3rd: id: 10811 title: Axe Shower Gel Apollo, 16 Ounce
	4th: id: 10701 title: Dove Men+Care Anti Dandruff Fortifying Shampoo, 12-Ounces
	5th: id: 10698 title: Axe Anti-dandruff Styling Cream, 3.2 Ounce
	1st: id: 10918 title: Nivea 3-in-1 Pure Impact Body Wash for Men, 16.9 Ounce (Pack of 3)
	2nd: id: 5293 title: Dove Men + Care Body and Face Bar, Extra Fresh, 4 Ounce, 8 Count
TASTE	3rd: id: 2778 title: Nivea For Men Energy Hair and Body Wash, 16.9-Ounce Bottle (Pack of 3)
	4th: id: 4205 title: Nivea For Men Active3 Body Wash for Body, Hair Shave
	5th: id: 2776 title: Nivea For Men Sensitive Body Wash 3-in-1 Body, Hair Face
Case #2 in Yelp	
History	id: 5908 title: DoMazing address: 6659 S Las Vegas Blvd, Ste B-101 Las Vegas NV, id: 2053 title: Village Pub & Poker address:
	7575 S Rainbow Blvd Las Vegas NV, id: 15510 title: Starbucks address: 7855 Blue Diamond Rd, 101 Las Vegas NV, id: 8414 title:
	Bowlology address: 4680 Maryland Pkwy, Ste 102 Las Vegas NV, id: 1477 title: IKEA - Cafe address: 6500 Ikea Way Las Vegas
	NV, , id: 16133 title: Meet Fresh address: 3930 Spring Mountain Rd Las Vegas NV, id: 12190 title: Yanni's Greek Grill address:
	9620 S Las Vegas Blvd, Ste E-7 Las Vegas NV
Label	id: 15108 title: Black Rock Coffee Bar address: 7565 S Rainbow Blvd Las Vegas NV
	1st: id: 195 title: Outback Steakhouse address: 7380 Las Vegas Blvd S Las Vegas NV
	2nd: id: 17172 title: Myungrang Hot Dog address: 4284 Spring Mountain Rd, D101 Las Vegas NV
T5-ID	3rd: id: 6201 title: Dutch Bros Coffee address: 4585 Blue Diamond Rd Las Vegas NV
	4th: id: 12468 title: Brother's Pizza address: 7575 S Rainbow Blvd, Ste 104 Las Vegas NV
	5th: id: 13175 title: Graffiti Bao address: 7355 S Buffalo Dr, Ste 1 Las Vegas NV
	1st: id: 15108 title: Black Rock Coffee Bar address: 7565 S Rainbow Blvd Las Vegas NV
	2nd: id: 15011 title: La Luna Tea and Dessert Bar address: Ste 8000F Las Vegas NV
TASTE w/o ID	3rd: id: 2836 title: Mazzoa Donuts address: 5180 Blue Diamond Rd, Ste 110 Las Vegas NV
	4th: id: 373 title: Blaze Fast-Fire'd Pizza address: 8229 Dean Martin Dr, Ste 115 Las Vegas NV
	5th: id: 8355 title: Starbucks address: 8263 Dean Martin Drive Las Vegas NV
	1st: id: 15108 title: Black Rock Coffee Bar address: 7565 S Rainbow Blvd Las Vegas NV
	2nd: id: 5908 title: DoMazing address: 6659 S Las Vegas Blvd, Ste B-101 Las Vegas NV
TASTE	3rd: id: 4838 title: Babystacks Cafe address: 8090 Blue Diamond Rd, Ste 170 Las Vegas NV
	4th: id: 17094 title: Black Rock Coffee Bar address: 4835 Blue Diamond Rd Las Vegas NV
	5th: id: 2836 title: Mazzoa Donuts address: 5180 Blue Diamond Rd, Ste 110 Las Vegas NV

recommendation modeling method, TASTE can accurately capture such user preferences and assign higher recommendation ranks to coffee bars and dessert shops. It further confirms that the item dependency and user-item relevance can be fully modeled by TASTE, which generalizes items in the text space and employs pretrained language models to model text relevance between items and users.

6 CONCLUSION

In this paper, we propose a Text mAtching based SequenTial rEcommendation (TASTE) model, which represents users and items with text utterances and leans text matching signals to model the relevance between them. TASTE achieves the state-of-the-art on widely used sequential recommendation datasets. It outperforms

previous item id base methods by alleviating the popularity bias and better representing the long-tail items using their ids and attributes. Notably, TASTE has the ability to better model user behaviors according to long-term user-item interactions and return more text-relevant and diverse items to satisfy user needs.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 62206042, No. 62137001, No. 61991404, and No. 62272093, the Fundamental Research Funds for the Central Universities under Grant No. N2216013, China Postdoctoral Science Foundation under Grant No. 2022M710022, and National Science and Technology Major Project (J2019-IV-0002-0069).

REFERENCES

- [1] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *ArXiv preprint* (2020).
- [2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. ArXiv preprint (2019).
- [3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. CoRR (2020). arXiv:2004.05150
- [4] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In Proceedings of ICLR
- [5] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. 2022. Sampling Is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In *Proceedings of CIKM*. 2974–2983.
- [6] 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 SIGIR. 378–387.
- [7] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 108–116.
- [8] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of SIGIR*. 579–588.
- [9] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. ArXiv preprint (2019).
- [10] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. 2021. Rethinking Attention with Performers. In Proceedings of ICLR.
- [11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. CoRR (2022). arXiv:2210.11416
- [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 Fifteenth ACM Conference on Recommender Systems. 505–514.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT. 4171–4186.
- [14] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. In Proceedings of Sixth ACM Conference on Recommender Systems. 59–66.
- [15] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-Shot Recommender Systems. CoRR (2021). arXiv:2105.08318
- [16] 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.
- [17] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S. Sheng. 2023. Frequency Enhanced Hybrid Attention Network for Sequential Recommendation. In *Proceedings of SIGIR*. 78–88.
- [18] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In Proceedings of IJCAI. 2301–2307.
- [19] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation Degeneration Problem in Training Natural Language Generation Models. In Proceedings of ICLR.
- [20] 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 Sixteenth ACM Conference on Recommender Systems. 299–315.
- [21] Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension. In Proceedings of ACL. 6751–6761.
- [22] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In Proceedings of IEEE 16th International Conference on Data Mining. 191–200.
- [23] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of WWW. 173–182.
- [24] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In

- Proceedings of ICLR.
- [25] 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 KDD*. 585–593.
- [26] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2333–2338.
- [27] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of EACL*. 874–880.
- [28] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In Proceedings of IEEE International Conference on Data Mining. 197–206
- [29] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of EMNLP*. 6769–6781.
- [30] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In Proceedings of ICML (Proceedings of Machine Learning Research). 5156–5165.
- [31] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In Proceedings of ICLR.
- [32] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In Proceedings of KDD. 1748–1757.
- [33] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication. 208–211.
- [34] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of EMNLP*. 9119–9130.
- [35] Čanjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PA-RADE: Passage representation aggregation for document reranking. ArXiv preprint (2020).
- [36] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of CIKM*. 1419–1428
- [37] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *Proceedings of KDD*. 1258–1267.
- [38] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Non-invasive Self-attention for Side Information Fusion in Sequential Recommendation. CoRR (2021). arXiv:2103.03578
- [39] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 9 (2023), 1–35.
- [40] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In Proceedings of IEEE 16th International Conference on Data Mining. IEEE, 1053–1058.
- [41] Zhenghao Liu, Kaitao Zhang, Chenyan Xiong, Zhiyuan Liu, and Maosong Sun. 2021. OpenMatch: An Open Source Library for Neu-IR Research. In *Proceedings of SIGIR*. 2531–2535.
- [42] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of SIGIR*, 43–52.
- [43] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pretrained Text-to-Text Models. In Proceedings of ACL Findings. 1864–1874.
- [44] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. In *Proceedings of SIGIR*. 695–704.
- [45] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. In Proceedings of KDD. 2671–2679.
- [46] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User Behavior Retrieval for Click-Through Rate Prediction. In *Proceedings of SIGIR*. 2347–2356.
- [47] 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.
- [48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. J. Mach. Learn. Res. (2020), 140:1–140:67.

- [49] Steffen Rendle. 2010. Factorization Machines. In Proceedings of The 10th IEEE International Conference on Data Mining. 995–1000.
- [50] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. CoRR (2012). arXiv:1205.2618
- [51] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of SIGIR*. 253–260.
- [52] 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 CIKM*. 1441–1450.
- [53] 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.
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of NeurIPS*. 5998–6008.
- [55] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. CoRR (2020). arXiv:2006.04768
- [56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. CoRR (2019). arXiv:1910.03771
- [57] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *Proceedings of AAAI*. 346–353.
- [58] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *Proceedings of SIGIR*. 1611–1621.
- [59] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proceedings of ICLR*.
- [60] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian J. McAuley. 2019. CosRec: 2D Convolutional Neural Networks for Sequential Recommendation. In *Proceedings of CIKM*. 2173–2176.
- [61] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed H. Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In Proceedings of the 13th ACM Conference on Recommender Systems. 269–277.
- [62] Shi Yu, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2023. OpenMatch-v2: An All-in-one Multi-Modality PLM-based Information Retrieval Toolkit. In Proceedings of SIGIR. 3160–3164.
- [63] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 582–590.
- [64] Xu Yuan, Dongsheng Duan, Lingling Tong, Lei Shi, and Cheng Zhang. 2021. ICAI-SR: Item Categorical Attribute Integrated Sequential Recommendation. In Proceedings of SIGIR. 1687–1691.
- [65] 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. In *Proceedings of SIGIR*. 2639–2649.
- [66] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In Proceedings of NeurIPS.
- [67] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. TKDE (2022).
- [68] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In Proceedings of IJCAI. 4320–4326.
- [69] Yuhui Zhang, HAO DING, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language Models as Recommender Systems: Evaluations and Limitations. In Proceedings of I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop.
- [70] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In Proceedings of CIKM. 4653–4664.
- [71] 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 CIKM. 1893–1902.
- [72] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Ge Kaikai, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. In

- Proceedings of SIGIR. 1167-1176.
- [73] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity bias in dynamic recommendation. In *Proceedings of KDD*. 2439–2449.