Large Language Models as Data Augmenters for Cold-Start Item Recommendation

Jianling Wang
Google DeepMind
jianlingw@google.com

Haokai Lu
Google DeepMind
haokai@google.com

James Caverlee
Google DeepMind
caverlee@google.com

Ed H. Chi
Google DeepMind
edchi@google.com

Minmin Chen
Google DeepMind
minminc@google.com

ABSTRACT

The reasoning and generalization capabilities of LLMs can help us better understand user preferences and item characteristics, offering exciting prospects to enhance recommendation systems. Though effective while user-item interactions are abundant, conventional recommendation systems struggle to recommend cold-start items without historical interactions. To address this, we propose utilizing LLMs as data augmenters to bridge the knowledge gap on cold-start items during training. We employ LLMs to infer user preferences for cold-start items based on textual description of user historical behaviors and new item descriptions. The augmented training signals are then incorporated into learning the downstream recommendation models through an auxiliary pairwise loss. Through experiments on public Amazon datasets, we demonstrate that LLMs can effectively augment the training signals for cold-start items, leading to significant improvements in cold-start item recommendation for various recommendation models.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

Large Language Models, Cold-start Recommendation, Data Augmentation

ACM Reference Format:
Jianling Wang, Haokai Lu, James Caverlee, Ed H. Chi, and Minmin Chen. 2024. Large Language Models as Data Augmenters for Cold-Start Item Recommendation. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651532

1 INTRODUCTION

Large language models (LLMs) trained on massive amount of web data, embody comprehensive understandings of the world, and have exhibited remarkable reasoning and generalization abilities [1, 3]. They have revolutionized many application fields, from creative writing, conversational agents, to search engine design [11, 20]. With their potential to reason and generalize user preferences and item characteristics better, we explore their application for recommendation systems, especially under cold-start setup. Recommendation systems serve as essential conduits conveying interesting contents to users across a wide range of online platforms. These systems typically operate by analyzing users' historical interactions to infer their preferences and subsequently recommend items that align with those preferences. The most widely adopted model-based recommendation systems heavily rely on well-learned ID-based embeddings [9, 12] for both users and items to memorize and predict their relevance.

While effective when user item interactions are abundant, these ID embedding-based approaches face a critical challenge in recommending fresh and tail content, which lack the initial exposure and interaction data necessary for the model to learn accurate embeddings. This is also known as the *cold-start* problem. To mitigate this long-standing problem, content-based recommendation systems utilize item meta features to assist item representation learning, i.e., replacing the ID-based item embedding with transformation of meta features, or their combination [8, 22, 31]. The hope is to generalize the learning power from items with abundant interactions to cold-start items through shared meta features.

Recent breakthroughs in LLMs and other foundation models offer exciting prospects for enhancing recommendation systems. Works in [10, 16, 17] showcase the promise of turning user query and content features into text, and utilizing the generative LLM models to build ID-free recommendation systems. However, it is still necessary to fine-tune large pre-trained modality encoders for recommendation (i.e., for the parameters and architecture) [30], which is a resource-intensive step and requires a significant amount of engineering effort. Additionally, latency in serving LLMs or large foundation models per user request to obtain recommendation results as required by these approaches is often more than the O(100ms) response time expected on recommendation platforms. Therefore it is prohibitively expensive to meet the Query-Per-Second requirement of industrial recommendation systems.

To transfer the power of LLMs to address the long-standing recommendation cold-start problem, instead of plugging them at the serving phase, we look into their potential in filling in the data gap during the **training phase** for current recommendation systems. In previous works [10, 16, 17], it is observed that LLMs are capable of understanding users' behavior (with appropriate prompt) and generating the context for contents of interest to the users (i.e., directly generating the title of the items or related topics). The research

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05.
https://doi.org/10.1145/3589335.3651532

questions we are answering is: 1) can we utilize LLMs' reasoning and generalization capabilities to generate synthetic user behaviors on cold-start items (i.e., ask if a user would prefer to watch a new video about "prompt-engineering" taught by Andrew Ng or the new TV series acted by Selena Gomez given the user's watching history); 2) does augmenting the training data for cold-start items with these synthetic interactions improve learning for classic recommendation models? With such a model-agnostic approach, we could bypass the slow API calls for LLM inference during serving time and also address the data sparsity issue for cold-start items. Additionally, the augmentation procedure as a data pre-processing step can be easily integrated to any industrial-scale recommenation system, providing a scalable approach to exploit LLMs' generalization capabilities. Together we make the following contributions in this work: 1) we propose pairwise comparison prompting LLMs to infer a user's preference between an item pair based on the user's historical interactions; 2) we integrate these LLM-generated synthetic user preferences with a pairwise loss as a supplement to the regular recommendation task; 3) we experiment on two real-world datasets and show that these synthetic user preferences can significantly improve the performance for cold-start items, and LLMs, even with a small model size, can be effectively used as a data augmenter for cold-start recommendation.

2 RELATED WORK

LLMs for Recommendation Systems. There is increasing interest in adapting LLMs directly for recommendation [2, 4, 7, 10, 17]. Another line of research explores how to use LLMs to improve conventional recommendation systems by generating new features [15, 27] or improving the encoding of existing features [29]. The main advantage of augmenting conventional recommendation systems with LLMs through feature engineering or feature encoders lies in its serving efficiency, since directly serving LLMs to retrieve items from millions or even billions of contents on industrial recommendation platform is prohibitively expensive and prevents their wide adoption. Our work aims to augment conventional ID-based recommmendation with synthetic examples generated by LLMs during training, instead of directly serving LLMs.

Cold-start item recommendation has been a persistent challenge in the field of recommender systems. Specifically, we focus on the more challenging scenario where cold-start items are newly uploaded and lack any user feedback. Traditionally, recommendation systems have addressed this issue by incorporating side information about items [26] and learning the correlation between well-trained ID-based embeddings and side information [31]. While recent efforts enable models to learn item embedding with just few interaction [14, 24] via meta-learning, these methods still fall short in effectively handling items with no historical interactions. Our work investigates the feasibility of utilizing LLMs to generate synthetic training signals for cold-start items, allowing for direct learning of collaborative embeddings from these synthetic signals.

Data Augmentation has demonstrated its effectiveness on enhancing the training efficiency of neural models across various application domains [6, 13]. In the context of recommendation systems, CLS4Rec, inspired by similar concepts in [5], employs random item cropping, masking, and reordering to generate augmented views of user historical sequences, ultimately improving model robustness and accuracy. To address the challenge of less active users, Wang et al. [25] proposed a learning-to-learn pipeline for augmenting training data and enhancing model performance for this user group. However, our work is the first to generate augmented training data for user behavior understanding based on historical interactions, specifically aiming to bridge the knowledge gap for cold-start items.

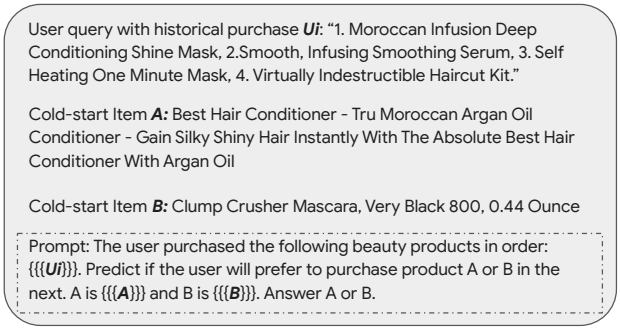


Figure 1: Pairwise comparison prompt for a user query.

3 PRELIMINARIES

Let $\mathcal{U} = \{u_1, u_2, ..., u_G\}$ represent the user set, $\mathcal{I}_{warm} = \{i_1, i_2, ..., i_P\}$ and $\mathcal{I}_{cold} = \{i_{P+1}, i_{P+2}, ..., i_{P+N}\}$ represent the warm and cold-start items on the platform respectively. Each item is mapped to a trainable embedding associated with its ID. The principle of generating personalized recommendation is to predict the compatibility between users and items, from which the items with high compatibility to a user would be retrieved from a large set of candidate items to make up the unique recommendation list for the user. As one of the most effective variants of matrix factorization [9], latent factor model-based recommendation has attracted lots of attention due to the Netflix Prize. The high level idea of latent factor models is to approximate compatibility between a user and an item with the dot product of the corresponding latent factor vectors. Given that $\mathbf{v}_u$ and $\mathbf{v}_i$ denote the latent factor vector for user u on item i, a latent factor model calculates their compatibility via $\hat{y}_{u,i} = \mathbf{v}_u^T \mathbf{v}_i$. Most of the widely-adopted recommendation frameworks can be regarded as extension of such latent factor model. Cold-start items, of course, have no training signal to obtain informative embeddings $\mathbf{v}_i$. To bridge the knowledge gap, we propose to generate synthetic data to simulate users' interactions on the cold-start items.

4 LLMS AS DATA AUGMENTERS

Augmented Data Generation. We focus on the PaLM family [1], and directly use their generation without any finetuning. We follow [10] and simply put the description of items that the user has interacted with into the prompt. Specifically, given a user query U_i in the training set, we adopt the descriptive item titles to denote each historical interactions. To infer users' preferences from this descriptive user query, we can either ask the user if he/she would like a specific cold-start item (pointwise) or ask them if he/she prefer cold-start item A or B (pairwise). LLMs have been shown to struggle with calibrated pointwise relevance estimation [19], but demonstrate

Table 1: Summary of cold-start recommendation and warm-start recommendation performance (%). The augmented data generated by LLMs significantly boost cold-start recommendation without sacrificing warm-start.

Task		Cold-start						Warm-start					
Dataset		**Beauty**			**Sports**			**Beauty**			**Sports**		
Metrics		R@5	R@10	R@50	R@5	R@10	R@50	R@5	R@10	R@50	R@5	R@10	R@50
NeuMF	w/o aug	0.14	0.22	0.48	0.01	0.02	0.13	3.44	5.36	18.76	2.65	4.76	17.98
	content-based	0.45	1.07	2.13	0.09	0.19	0.87	2.48	4.02	16.89	1.77	3.23	16.01
	w/ aug	1.19	2.32	10.53	0.22	0.41	2.11	3.35	5.21	18.03	2.32	4.75	17.67
SASRec	w/o aug	0.18	0.48	0.74	0.10	0.22	0.31	4.25	6.18	19.87	3.57	5.48	19.01
	content-based	0.56	1.26	4.77	0.15	0.30	0.96	2.90	5.01	16.97	1.89	3.09	16.33
	w/ aug	1.34	2.47	11.40	0.37	0.61	2.41	4.30	6.11	19.79	3.51	5.39	18.95

better capabilities [4, 20] at pairwise comparison task. Therefore, we probe LLMs to generate pairwise preference between cold-start items given a user query. In particular, we randomly sample an item pair (A, B) with $A, B \in \mathcal{I}_{cold}$, and construct the prompt as shown in Figure 1 to retrieve the user's preference between A and B. The pairwise comparison in the prompt ensures that we obtain training signal from every LLM call to indicate preference between two cold-start items for the user. In comparison, a pointwise inference on a random cold-start item for the user will most likely result in negative labels (not interested) while positive labels are rare.

Pairwise Comparison Loss. To incorporate this augmented signal during the training process, we add the pair-wise preference prediction on the cold-start item pairs as an auxiliary task complementary to the regular recommendation task. The answer returned by the LLM would be regarded as the *pos* item and the other item as the *neg*. And given that a user u prefers cold-start item *pos* than cold-start item *neg*, we have the following pairwise loss inspired by Bayesian Personalized Ranking (BPR) loss [21]:

$$\mathcal{L}_{aug} = -\sum_{(u,pos,neg)} \ln \sigma(\hat{y}_{u,pos} - \hat{y}_{u,neg}), \tag{1}$$

in which σ is the Sigmoid activation function. We then add this pairwise BPR loss to the sampled batch softmax [28] loss typically used in training recommendation model. The pairwise loss back-propagates gradients to the embeddings of the positive and negative cold-start items, which are often under-trained due to the lack of interaction, and thus training signals from the main task.

5 EXPERIMENTS

5.1 Experimental Setup

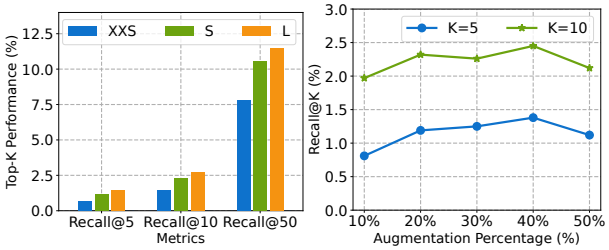
Data and Preprocessing. We use the public Amazon review datasets [18] for evaluating the performance of our method. We select the "*Beauty*" and "*Sports and Outdoors*" categories, which include users' ratings and reviews on items falling into these two categories. To split the datasets for training and testing, we follow the single-time-point split [23] and select the time-point to split the data by 7:3. The interaction data before the splitting time-point would be used for model training and the trained model is tested on interaction data after the splitting time-point. Specifically, we regard the items *showing up only in the testing data* as *cold-start items* and others as *warm-start* items, which is similar to the real-world setup. As an example, the split resulted in 55,255 warm-start items and 2,751 cold-start items in the "*Sports and Outdoors*" category. There are 224,956 user queries in this category. We randomly sample a subset of user queries, and randomly sampled two cold-start items for each query to generate augmented data examples.

Models and Parameters. We evaluate the generalizability of our proposed methods employing two established recommender backbones: Neural Matrix Factorization (**NeuMF**) [9] and **SASRec** [12]. NeuMF learns user embedding $\mathbf{v}_u$ and item embedding $\mathbf{v}_i$ from their IDs, and SASRec leverages a self-attention layer over user's historical interactions to encode sequential information in user embedding $\mathbf{v}_u$. These backbones represent core components in many recommender systems, and we test three variations built upon them: 1) Baseline (***w/o Aug***) trains recsys with the original training data; 2) ***content-based*** method employs bag-of-words representations for items, incorporating their category and title, which is a common approach for handling cold-start items and 3) LLM-augmentation (***w Aug***) incorporates LLM-generated augmentations and supplement the training process with pairwise comparison loss in Equ 1. We use ***PaLM2*** [1] with different model size (i.e., *XXS*, *S* and *L*) to investigate the performance on the synthetic data generation.

Top-K Evaluation Metrics. For each test user query, we retrieve the top-K items from $\mathcal{I} = \mathcal{I}_{cold} \cup \mathcal{I}_{warm}$ with the highest compatibility scores. And to compare the performance of the recommendation systems offline, we adopt recall (i.e., *R@K*) to check if the ground-truth item of the testing query shows up in the top-K list. We group the results by ground-truth label, i.e., grouping results when the purchased items are cold-start items in the "cold-start" columns, and the rest in the "warm-start" columns in Table 1.

5.2 Results and Analysis

In the overall comparison, we use *PaLM2-S* to generate the synthetic data and combine it with the regular training data. We randomly sample 20% of user queries and generate one augmented data example for each query. In Table 1, we report the results to test its effectiveness on both NeuMF and SASRec. Without any augmentation data examples, cold-start items lack training signals, leading to extremely low recall at all K values. While content-based leveraging item titles and descriptions can offer decent performance on cold-start items, it neglects collaborative signals, significantly hurting warm-start item recommendations. In contrast, the augmented training signals for cold-start items, when learned through the pairwise loss, can benefit the representation learning for both NeuMF



(a) (b)

Figure 2: Cold-start Recommendation Under Different Model Sizes and Augmentation Percentage.

and SASRec, and boost the performance on cold-start recommendation significantly. Furthermore, the augmented signals and pairwise comparison loss improve recall more at higher K values, as they enable the model to rank a wider range of cold-start items, including some less relevant ones (i.e., which still match users' preferences). The results suggest that LLMs is an effective data augmenter to fill up the missing knowledge on cold-start items. Although the augmented training signals for cold-start items hurt the warm-start item recommendations slightly for some recall metrics, the drop in performance is marginal compared to the huge gain obtained on cold-start recommendations.

To understand the impact of different LLM model sizes, we conduct experiments on Amazon-Beauty using NeuMF based recommender with *PaLM2* of various model sizes (i.e., XS, S and L) as the data augmenter. In Figure 2 (a), we find that the model size does influence the performance of augmentation. It is known that many abilities of LLMs are emergent as these models scale up [1, 3]. We hypothesize larger models are able to reason through the user historical behaviors better and infer the preferences more accurately. In Figure 2 (b), we also observe that by generating augmented training signals with more user queries, we can further increase the performance for cold-start recommendation. Even though adding more synthetic data beyond certain point (40%) did not lead to any further improvement.

6 CONCLUSION

Addressing the lack of user interactions with cold-start items is crucial for enhancing recommendation effectiveness. We propose to employ LLMs to generate augmented training signals for cold-start items in recommender systems. We use a pairwise comparison prompt to leverage LLMs to infer user preferences between pairs of cold-start items. This model-agnostic design provides informative training signals for cold-start items without introducing additional computational overhead during serving time. Experiments on public datasets show that our method generates effective augmented training signals and improves cold-start item recommendation.

REFERENCES

[1] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
[2] 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*.
[3] 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. In *NeurIPS*.
[4] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. *arXiv preprint arXiv:2305.02182*.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*.
[6] Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. 2022. Data augmentation for deep graph learning: A survey. In *ACM SIGKDD Explorations Newsletter*.
[7] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. VIP5: Towards Multimodal Foundation Models for Recommendation. *arXiv preprint arXiv:2305.14302*.
[8] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *ICCCA*.
[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
[10] 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. *ECIR*.
[11] Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. In *arXiv preprint arXiv:2305.06984*.
[12] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*.
[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *NeurIPS*.
[14] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *KDD*.
[15] Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. TagGPT: Large Language Models are Zero-shot Multimodal Taggers. *arXiv preprint arXiv:2304.03022*.
[16] 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. In *KDD*.
[17] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*.
[18] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*.
[19] 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. In *NeurIPS*.
[20] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.
[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.
[22] Tobias Schnabel, Mengting Wan, and Longqi Yang. 2022. Situating Recommender Systems in Practice: Towards Inductive Learning and Incremental Updates. In *arXiv preprint arXiv:2211.06365*.
[23] Aixin Sun. 2023. Take a Fresh Look at Recommender Systems from an Evaluation Standpoint. In *SIGIR*.
[24] Jianling Wang, Kaize Ding, and James Caverlee. 2021. Sequential recommendation for cold-start users with meta transitional learning. In *SIGIR*.
[25] Jianling Wang, Ya Le, Bo Chang, Yuyan Wang, Ed H Chi, and Minmin Chen. 2022. Learning to Augment for Casual User Recommendation. In *TheWebConf*.
[26] Jianling Wang, Ainur Yessenalina, and Alireza Roshan-Ghias. 2021. Exploring heterogeneous metadata for video recommendation with two-tower model. *arXiv preprint arXiv:2109.11059*.
[27] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. *arXiv preprint arXiv:2306.10933*.
[28] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *RecSys*.
[29] Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, and Qi Liu. 2022. Tiny-newsrec: Effective and efficient plm-based news recommendation. In *EMNLP*.
[30] 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*.
[31] Xu Zhao, Yi Ren, Ying Du, Shenzheng Zhang, and Nian Wang. 2022. Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder. In *SIGIR*.