Zero-Shot Next-Item Recommendation using Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated impressive performance in various natural language processing tasks when given appropriate input prompts without requiring fine-tuning on specific training data. However, their application in next-item recommendation remains unexplored due to the vast, task-specific recommendation space and unfamiliarity with user preferences. To address these issues, this paper introduces the **Zero-Shot Next-Item Recommendation (NIR)** strategy, using an external module for candidate item generation and a *3-step prompting* method for capturing user preferences and making ranked recommendations. Evaluations on MovieLens 100K and LastFM datasets using GPT-3.5 reveal that the proposed NIR competes well with strong sequential recommendation models, opening up new interesting research opportunities to leverage LLMs as recommender systems.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022), such as GPT-3 (Brown et al., 2020), have demonstrated impressive results in various natural language processing (NLP) tasks. Nevertheless, LLMs are usually very large and only accessible only via some API services. Hence, they cannot be fine-tuned like the earlier pre-trained language models (PTMs) (Devlin et al., 2018; Radford et al., 2019). Many works have also demonstrated that LLMs are capable of solving many known NLP problems through task-specific prompts under the zero-shot setting, i.e., without any examples or further fine-tuning (Brown et al., 2020; Chowdhery et al., 2022). Nevertheless, using LLMs to perform next-item recommendations is still a relatively new research topic which awaits investigation.

Unlike NLP tasks that rely on the inherent textual knowledge of LLMs, recommendation tasks require LLMs to utilize a user's past item interactions to make item recommendations. Direct methods, such as the Simple Prompting method in Section 3, yield poor recommendations (Zhang et al., 2021). Moreover, LLMs struggle to contribute to recommendations without prior knowledge of the items. In this research, we assume that recommended items should be included in the pre-training data of LLMs (e.g., reviews, Wikipedia pages, etc.). Examples of such items include movies, artists, songs, etc.. For illustration and evaluation, we focus on movie and artist recommendations using GPT-3.5.

In this paper, we introduce an approach for next-item recommendation called Next-Item Recommendation (NIR) prompting. It first limits the recommendation space for a user to items within a candidate item set by using user or item filtering techniques. Secondly, the NIR recommends items using a 3-step prompting method: (i) capturing user preferences (Step 1), (ii) selecting representative items from the user's interacted items (Step 2), and (iii) recommending a ranked list of items (Step 3). Finally, we use a formatting technique in Step 3 to ensure easier extraction of recommended items. Our experiments on MovieLens 100K and LastFM 2k with GPT-3.5 (`text-davinci-003`) indicate that NIR prompting is competitive compared to strong supervised learning baselines. Related work is detailed in Appendix Section A.

2 Zero-Shot NIR Prompting Strategy

This section presents our proposed zero-shot NIR prompting strategy. As shown in Figure 1, the proposed method has three main components:

Candidate set construction: This component performs user-filtering or item-filtering to create a candidate item set for the target user using the training data, which effectively narrows down the recommendation space. These candidate items are then used in the three-step prompting.

Three-step prompting: This component in-

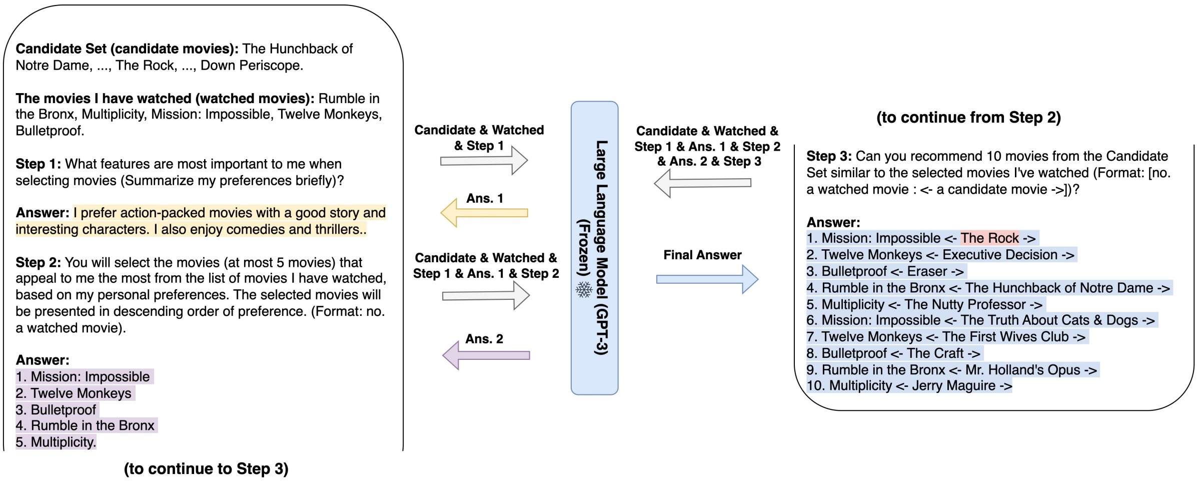

Figure 1: Zero-Shot NIR prompts. The ground truth movie in this example is The Rock

volves three instruction prompts corresponding to three subtasks. In the first subtask (*user preference subtask*), we design a user preference prompt (Step 1 prompt) to summarize the target user's preferences based on the previously interacted items. In the second subtask (*representative items subtask*), we then define the Step 2 prompt to combine the user preference prompt and its answer to request GPT-3.5 to list representative items based on user preference. In the third subtask (*item recommendation subtask*), we direct GPT-3.5 to recommend k items similar to the representative ones.

Answer extraction: This component extracts the recommended items from the textual results of the three-step GPT-3.5 prompting using a simple extraction rule.

2.1 Candidate Set Construction

In Section 1, we highlight the challenge of large recommendation spaces for LLM-based recommendation. Handling the vast number of recommendations is complex, and not all items can be fed to the LLM. For instance, 1,683 movies from the MovieLens 100K are too large to be fed into a prompt. Thus, in our approach, we build a candidate item set for the user based on the relevance to the user. Specifically, we employ *user filtering* and *item filtering* to determine candidate items.

User-Filtering. This principle assumes that the candidate items should also be liked by other users similar to the target user. Hence, we first represent every user by a multi-hot vector of their watched items. Users similar to the target user are then derived by cosine similarity between the target user's vector and vectors of other users. Next, we select the m most similar users and the candidate item set of size n_s is constructed by selecting the most popular items among the interacted items by the similar users.

Item-Filtering. Similar to user filtering, we represent each item by a multi-hot vector based on its interacted users. Using cosine similarity between two items, we select the n_m most similar items for each item in the target user's interaction history. We then generate a candidate item set of size n_s based on the "popularity" of these similar items among items in the target user's interaction history.

The constructed candidate item set is then incorporated into the prompts for recommendation using the sentence: "Candidate Set (candidate tt items):" as shown in Figure 1. Following the candidate set, the prompts also include the list of target user's previously interacted items.

2.2 Three-Step Prompting

Step 1: User Preference Prompting. To capture the user's preferences, we include the sentence "Step 1: What features are most important to me when selecting items (summarize my preferences briefly)?" into the first prompt. As shown in Figure 1, the answer returned by GPT-3.5 summarizes the target user preference (highlighted in yellow).
Step 2: Representative Item Selection Prompting. As the second step, this prompt includes the previous prompt text appended with the answer of Step 1, including the instruction: "Step 2: You will select the items ... that appeal to me the most ... presented in descending order of preference (...)" to determine the previously interacted items that best reflect the target user's preferences. Figure 1 shows the GPT-3.5's answers highlighted in purple.
Step 3: Recommendation Prompting. Again, this prompt includes the previous text appended with the answers of Step 2, including the instruction

Method	MovieLens 100K		LastFM 2K	
	HR	NDCG	HR	NDCG
POP	0.0519	0.0216	0.0755	0.0458
FPMC	0.1018	0.0463	0.0872	0.0449
GRU4Rec	0.1230	0.0559	0.0890	0.0480
SASRec	**0.1241**	**0.0573**	**0.1101**	**0.0539**
Simple Prompting	0.0297	0.0097	0.1032	0.0410
CS-Random-IF	0.0805	0.0352	0.0851	0.0440
CS-Random-UF	0.0954	0.0457	0.0869	0.0378
NIR-Single-IF	0.0975	0.0501	**0.1198**	**0.0624**
NIR-Single-UF	0.1135	0.0529	0.1140	0.0621
NIR-Multi-IF	0.1028	0.0505	0.1013	0.0512
NIR-Multi-UF	**0.1187**	**0.0546**	0.0936	0.0492

Table 1: HR@10 (HR) and NDCG@10 (NDCG) on the test sets of MovieLens 100K and LastFM. (Best results in each group of methods are **boldfaced**.

CSet	UPref	RItem	ML100K	LastFM2K	Average
–	–	–	0.0297	0.1032	0.0664
✓	–	–	0.1019	0.1093	0.1056
✓	✓	–	0.1081	0.1112	0.1096
✓	–	✓	0.1060	0.1102	0.1081
✓	✓	✓	**0.1135**	**0.1140**	**0.1137**

Table 2: Ablation study of the impact of Candidate Set (CSet), User Preference (UPref), and Representative Items (RItem) in the proposed NIR-Single-UF prompting on MovieLens100K (ML100K) and LastFM datasets. HR@10 is adopted for this evaluation.

"Step 3: Can you recommend 10 items from the Candidate Set similar to ...". This prompt explicitly instructs GPT-3.5 to generate 10 recommended items from the candidate set as highlighted in blue.

3 Experiments and Results

3.1 Experiment Setup.

We empirically investigate the performance of the zero-shot NIR strategy against fully trained and zero-shot baselines using the MovieLens 100K dataset (Harper and Konstan, 2015) (943 users and 1,682 movies) and Last.FM 2k dataset (Cantador et al., 2011) (1,892 users and 17,632 artists) for movie and artist recommendations, respectively.

We evaluate our proposed NIR-based methods including: (i) **Zero-Shot NIR-Single-IF/NIR-Single-UF** (that combines the 3 steps into a single prompt leaving out the intermediate answers, and prompts GPT-3.5 only once to generate n recommended items from IF/UF-based candidate set.); (ii) **Zero-Shot NIR-Multi-IF/NIR-Multi-UF** (that uses three separate prompts to guide GPT-3.5 step-by-step and incorporates intermediate answers to the subsequent prompts with the IF/UF-based candidate set.). NIR-Single can save some prompting cost compared with NIR-Multi.

The *strong next-item recommendation baselines* to be compared include: (i) **POP** (that recommends most popular items), (ii) **FPMC** (Rendle et al., 2010) (that combines matrix factorization and Markov chains), (iii) **GRU4Rec** (Hidasi et al., 2015) (a GRU-based sequential recommendation model), and **SASRec** (Kang and McAuley, 2018) (a sequential recommendation model based on self-attention). As FPMC and GRU4Rec are fully trained models, they are expected to outperform zero-shot methods. The zero-shot baseline methods to be compared include: (i) **Simple Prompting** (that prompts LLMs to recommend n items directly), (ii) **CS-Random-IF** (that randomly selects n items from the item filtering-based candidate set), and (iii) **CS-Random-UF** (that randomly selects n items from the UF-based candidate set).

We utilize the GPT-3.5 `text-davinci-003` (175B) with public APIs[1], setting the temperature to 0 for consistent results. For $*$-**UF**'s, default values are: most similar users (m) as 12, and candidate items (n_s) as 19. For $*$-**IF**'s, we use: most similar items (n_m) as 10 and candidate items (n_s) as 19. We apply a leave-one-out strategy for performance measurement: the last item in each user sequence is test data, the penultimate is validation, and others form the training set. Evaluation metrics include *Hit Ratio (HR) at 10* and *Normalized Discounted Cumulative Gain (NDCG) at 10*, following SASRec (Kang and McAuley, 2018).

3.2 Experiment Results

Main results. Table 1 reveals that our zero-shot NIR-based methods significantly surpass POP. Notably, Zero-Shot NIR-Single-UF, NIR-Multi-IF, and NIR-Multi-UF even outperform the fully trained FPMC. Although the three Zero-Shot NIR-based methods perform slightly worse than the strong sequential recommendation model SASRec, they still compete strongly with SASRec. Among zero-shot methods, CS-Random-UF(IF) surpasses Simple Prompting, demonstrating that candidate sets enhance recommendation performance. Our NIR-based prompts outperform Simple Prompting and CS-Random-IF/UF, indicating that combining user preferences and other strategies enrich LLM recommendations. Additionally, Multi-IF(UF) excels over Single-IF(UF) on MovieLens 100K, but not LastFM 2K. Simple prompting leads

[1]https://beta.openai.com/docs/models/gpt-3

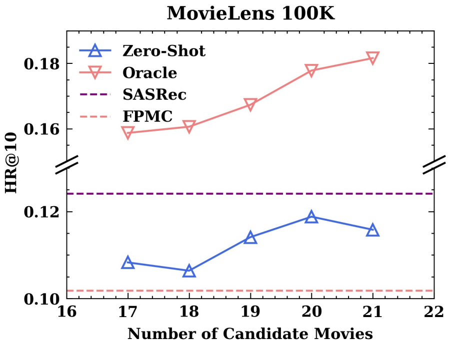

Figure 2: HR@10 of Full-Trained SASRec, FPMC and NIR-Single-UF prompting with varying number of candidate movies n_s on MovieLens 100K.

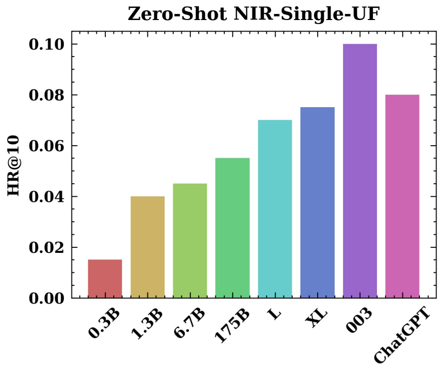

Figure 3: HR@10 of NIR-Single-UF prompting using backbone LLMs with different sizes. 0.3B: GPT-3 ada, 1.3B: GPT-3 `babbage`, 6.7B: GPT-3 `curie`, 175B: GPT-3 `davinci`, X: Instruct GPT-3 `text-davince-001`, XL: Instruct GPT-3 `text-davinci-002`, 003: GPT-3.5 `text-davinci-003`.

in HR@10 on LastFM but lags in NDCG@10. UF-based NIR prompts generally perform better than IF-based ones, though IF-based methods are better in a zero-shot setting on LastFM 2K.

Effects of NIR Prompt Components. Our proposed methods, NIR-Single-UF/IF and NIR-Multi-UF/IF, involve candidate set construction and a three-step prompting process. We evaluate the effectiveness of these components on MovieLens 100K and LastFM 2K datasets with HR@10. Results (Table 2) reveal that each step enhances recommendation accuracy. The Simple Prompting method, which employs a candidate set, performed better than the one without it on average (HR@10=0.1056 vs. HR@10=0.0664), highlighting the importance of the candidate set. Our findings show that integrating candidate sets and specific prompting steps improve performance, suggesting that a narrowed recommendation space and clear guidelines improve GPT-3.5's output.

Impact of Candidate Set Size n_s. In this study, we examine how the candidate set size affects the performance of NIR-based methods on the MovieLens 100K dataset. We tested the NIR-Single-UF method with candidate set sizes ranging from 17 to 21. The results, depicted in Figure 2, show that an optimal candidate set size is around 20; both smaller and larger sizes diminish performance, though it remains between the levels of SASRec and FPMC. Similar results were seen with the LastFM dataset. Moreover, we observe the oracle's performance continues to improve with larger candidate set ($n_s = 21$). Nevertheless, NIR-Single-UF could not exploit this for performance improvement. Furthermore, while an oracle model, which returns the true item when present in the candidate set, improves its performance with a larger candidate set, NIR-Single-UF does not. This indicates potential for further enhancements in the zero-shot NIR approach. We thus believe there are ample room for the zero-shot NIR approach to further improve.

Impact of Backbone LLMs. In this study, we investigate the impact of LLM model size and capability on NIR-based prompting methods for recommendations using various models, such as different versions of GPT-3.5, accessed via OpenAI API on MovieLens 100K. Figure 3 ranks these models by capability, from `GPT-3 ada` (lowest) to ChatGPT (highest). Testing on a subset of 200 examples from the MovieLens 100K dataset shows an improvement in performance from ada to `text-davinci-003`. However, ChatGPT underperforms `text-davinci-003`, possibly due to ChatGPT's flexible generation nature. These results indicate that more capable LLMs typically yield better recommendation results.

4 Conclusion

In this paper, we propose a three-step prompting strategy called Next-Item Recommendation (NIR) for LLM to make next-item recommendation for user-item interaction sequences. We evaluate our approach using GPT-3.5 as the LLM on both Movielen 100K and LastFM 2K datasets, and obtain promising accuracy. Our results show the potential of using LLMs in zero-shot recommendation and call for further exploration of using LLMs in recommendation tasks. This work can be extended in several directions, including the few-shot approach (instead of zero-shot), choice of LLMs, recommendation of proprietary items, and explainable LLM-based recommendations.

5 Limitations

Our proposed prompting method partially relies on handcrafted prompts when writing the prompting questions. However, handcrafted prompts are usually based on the personal knowledge and experience of the exports, which can introduce subjective biases.

References

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*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA. ACM.

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*, pages 378–387.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.

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*.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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*.

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). *arXiv preprint arXiv:2203.13366*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

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)*, pages 191–200. IEEE.

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

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*.

Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 505–514.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. *arXiv preprint arXiv:2202.07371*.

Zhiwei Liu, Yongjun Chen, Jia Li, Man Luo, S Yu Philip, and Caiming Xiong. 2021. Self-supervised learning for sequential recommendation with model augmentation.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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*, pages 811–820.

Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-shot recommendation as language modeling. In *European Conference on Information Retrieval*, pages 223–230. Springer.

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*, pages 1441–1450.

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*, pages 565–573.

Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1110.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–735.

Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized prompts for sequential recommendation. *arXiv preprint arXiv:2205.09666*.

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*.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395*.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4321–4330.

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*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

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 *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

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*, pages 1893–1902.

A Related Works

A.1 Supervised Learning-based Recommender Systems

Next-item recommendation is an important and well studied research problem. Early research works proposed Markov Chains to model low-order relationships between items for next-item recommendation (Rendle et al., 2010; He and McAuley, 2016). With the advancement of neural models, deep neural networks (Hidasi et al., 2015; Tang and Wang, 2018; Kang and McAuley, 2018; Huang et al., 2018; Wang et al., 2020; Sun et al., 2019; Chang et al., 2021) have been applied to the modeling of sequential patterns which leads to improved recommendation accuracy. Recent research has also explored the use of data augmentation and contrastive learning to enhance the representations of users and items, thereby making further improvement to recommendation performance (Zhou et al., 2020; Xie et al., 2020; Liu et al., 2021; Yao et al., 2021; Wu et al., 2021). Nevertheless, all the above works require model training using users' historical item-interactions. In other words, they are not suitable for zero- or few-shot setting. To the best of our knowledge, there has been very little research on zero- and few-shot recommendation. While LLMs are known to be good zero/few-shot NLP problem solvers, there has been very few works that attempt to use LLMs as recommenders.

A.2 LLM-based Recommender Systems

Among the early efforts in LM-based recommendation (Li et al., 2022; Zhang et al., 2021; Sileo et al., 2022; Cui et al., 2022; Geng et al., 2022; Wu et al., 2022), Zhang et al. (2021) proposed to use GPT-2 (Radford et al., 2019) or BERT (Devlin et al., 2018) as the backbone recommender, making the next-movie prediction based on five previously watched movies by the target user. However, the huge recommendation space and inadequate user preference modeling make the LLMs perform poorly. With newer LLMs such as GPT-3 (Brown et al., 2020), OPT (Zhang et al., 2022), and PaLM (Chowdhery et al., 2022) which have shown significantly improved results in various NLP tasks, our work chooses GPT-3 to be the LLM for developing more effective zero/few-shot recommendation methods.

LLM-based recommender systems can be categorized into (a) LLM-augmented recommender systems (Gao et al., 2023; Xi et al., 2023), and (b) LLM-only recommender systems (Hou et al., 2023; Dai et al., 2023; Bao et al., 2023; Zhang et al., 2023). KAR (Xi et al., 2023) leverages LLMs for open-world knowledge and improving recommendation accuracy and versatility. Chat-REC is a LLM-based recommender system with conversational chat interface (Gao et al., 2023). It augments a supervised learning recommender system by selecting a smaller set of candidate items from the latter and reranking them for the target user. Chat-REC also provides explanation to the recommended items. Hence, Chat-REC still requires fully supervised learning which could incur significant overhead. For LLM-only recommender systems, Dai et al. (2023) conduct an empirical analysis on ChatGPT's recommendation abilities in three ranking policies. Hou et al. (2023) explore LLMs (e.g., GPT-4) as ranking models in recommender systems, revealing promising zero-shot abilities but position biases. Instead of designing the prompting strategy from scratch, our proposed NIR prompting strategy incorporates user-filtering and item-filtering to derive a candidate item set. This way, it mimics well-known recommendation techniques and leverages its item knowledge and reasoning capability to deliver more accurate recommendation results.