Identify User Intention for Recommendation using Chain-of-Thought Prompting in LLM

Gangmin Li[1[0000-0003-4006-7472]],

Fan Yang[2[0000-0002-7763-4242]], and Yong Yue[2[0000-0001-7695-4538]]

[1] HeXie Management Research Centre, College of Industry-Entrepreneurs (CIE), Xi'an Jiaotong-Liverpool University, Suzhou， Jiangsu， 215123, China
[2] Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou， Jiangsu， 215123, China
gangmin.li,Fan.yang,Yong.yue@xjtlu.edu.cn

Abstract. It is crucial for a recommendation system to accurately identify a user's intention since it dictates the user's selection once facing multiple candidates. Traditional user intention identification uses the user's selection among multiple items. This technique relies primarily on historical behavioral data and results in problems such as the cold start, unintended choice, and failure when items are new. Motivated by recent advancements in Large Language Models (LLMs) like ChatGPT, we present an approach for user intention identification by embracing LLMs with Chain-of-Thought (CoT) prompting. We use the initial user profile as inputs to LLMs and design a collection of prompts to align the LLM's response through various recommendation tasks encompassing rating prediction, search and browse history, user clarification, etc. to identify the user's intention and then the same input with generated user intention feed to the LLM to produce recommendations. We tested our approach with real-world datasets to demonstrate the improvements in the recommendation comparing the recommendation without the intention merged into a user model. The results indicate the clear improvements.

Keywords: Personalized Recommendation, Generative User Modeling, User Intention Identification, Large Language Models, Chain-Of-Thought Prompting.

1 Introduction

Recommendation has become an essential service for many internet service providers [1]. It triggered much research on user modeling since providing accurate services can only be achieved by a better understanding of user's intention [2]. Traditional retrieval-based user recommendations have limitations and flaws because they fail to capture user intention dynamically. Identifying user's intention from the user's past actions proves to be inadequate. This is because the user's actions may sometimes exhibit the user's temporary ethical concerns and moral factors, other times they reflect user's personal biased and unfair discriminations and also sometimes the user's actions are purely the user's temporary and random non-intentional temperaments. Therefore, it is hard to identify the user's true intention simply from some past actions. The problems associated with the technique have been revealed in recommendations when new products or items become available. Users often find themselves trapped in past selections and fail to provide opportunities for exploring new and more preferable items in current recommendation systems.

In response to the above challenges, we propose a new method called "User Intention Identification with Chain-of-Thought Prompting"(UII-COT), which is an improved version of the UIGRec (User Intention with the Chain-of-Thought). It leverages large language models (LLMs) and uses the chain-of-thought(CoT) prompting method in natural language to reason the data and information provided at three levels: user profiles, interests, and preferences to figure out user intentions. We integrate them into a user model. We take an initial existing user model as input to LLMs and design a set of prompts on four separate tasks: rating prediction, direct recommendation, sequential recommendation, and explanation generation. Unlike traditional recommendation methods, we do not tune LLMs during the process and only rely on the prompts to adapt LLMs to understand the user's intention.

To assess the proposed approach, we conduct experiments with a diverse set of recommendation scenarios using real-world datasets. The results of our experiments indicate that the proposed approach exhibits promising performance compared to several competing baselines. Moreover, the experimental results demonstrate the effectiveness of UII-COT in enhancing the LLM's capability to identify user's intentions that lead to users' decisions and actions.

2 Related Work

Large Language Models (LLMs) originated in Natural Language Processing (NLP) and have been used in many application domains. They have recently gained significant attention in the domain of Recommendation Systems (RS). The efforts using LLMs for recommendation can be categorized into five major areas: LLMs for recommendation subtasks, LLMs to generate recommendations, fine-tuning LLMs, prompt engineering, and user modelling.

2.1 LLMs For Recommendation Subtasks

Several studies have conducted using LLMs to overcome traditional problems in recommendation. Hou et al. [3] assessed the zero-shot ranking capability of LLMs in recommendation systems for candidate retrieval and sorting. Chat-REC [4] testing chat-style recommendations by deploying in-context learning on top of LLMs, transforming user's profiles and historical interactions into prompts to generate chat-style recommendations. Wang et al. [5] employed traditional collaborative filtering models to generate candidate items. They then utilized step-by-step prompts to extract user preferences, select representative historical interactions, and recommend and rank a list of ten items. Kang et al. [6] evaluated LLMs' performance in rating prediction. They revealed that augmenting model parameters in cold-start scenarios leads to enhanced recommendations. Arkadeep et al. [7] have explored how we can use LLMs to generate detailed descriptions of the items. However, LLMs' performance in zero-shot settings is noticeably inferior to traditionally trained models.

2.2 LLMs To Generate Recommendations

LLMs' proficient text generation capabilities are also used for generating recommendations. GeneRec [8] introduced a novel generative recommendation paradigm, where user instructions and feedback were pre-processed as generation dependencies. An AI generator was employed to customize existing items or create new ones based on user needs. GPT4Rec [9] utilized a flexible generation framework for recommendation tasks. It first requested GPT2 to generate a hypothetical “search query” based on user-historical interactions and their titles. Then, a search engine (BM25) was employed to retrieve relevant items according to this query. GENRE [10] is a generative news recommendation framework based on LLMs, utilizing the metadata of news items interacted with by the user (titles, summaries, categories, etc.) to construct prompts.

2.3 Fine-Tune LLMs For Recommendations

Some research aligns LLMs with recommendation tasks through fine-tuning. InstructRec [11] proposed expressing requirements through natural language instructions, leveraging LLMs to achieve personalized recommendations by analyzing these instructions. It combines user preferences, intentions, and task formats to adapt to various interaction scenarios. TALLRec [12] is an efficient fine-tuning framework comprising instruction tuning and recommendation tuning. In the instruction tuning stage, TALLRec fine-tunes LLMs using Alpaca's instruction data. In the recommendation tuning stage, TALLRec formalizes recommendation data into the format used in instruction tuning. Through these two stages of efficient tuning, LLMs can be well-adapted to recommendation systems and demonstrate robust cross-domain generalization capabilities. Li et al. [13] propose distilling a specific task's discrete prompt to a set of continuous prompt vectors to bridge IDs and words and then reduce the inference time.

2.4 Prompt Engineering For Recommendation

Formulating natural language queries to guide the LLMs toward desired outcomes is called prompt engineering. Notably, models like OpenAI's ChatGPT showcase impressive language generation capabilities, demonstrating an understanding of complex prompts and generating text that resembles human-like responses. Brown et al. [14] highlight the potential of these models for tasks through “few-shot learning” without the need for explicit fine-tuning. In contrast, prompt learning harnesses the LLMs' ability to comprehend and respond to prompts in a specific manner, effectively influencing the model's output. Prompt learning has found practical applications in various recommendation tasks [3, 4, 9, 15]. However, some challenges have been identified in prompt learning. Schick et al. [16] discovered that the choice of prompts significantly impacts performance, and designing effective prompts often requires human expertise. Additionally, as noted by Gao et al. [17], continuous prompts may lead to overfitting.

By constructing prompts for recommendation tasks, several fundamental approaches to prompting the model include zero-shot prompting, few-shot prompting, and self-consistency. Zero-shot prompting directly provides context information and task descriptions for LLMs without reference examples. Conversely, few-shot prompting offers a few prompts to assist LLMs in better understanding user intentions [14]. The self-prompting strategy implies that knowledge for answering questions can be acquired by prompting LLMs multiple times. LLMs may be required to generate relevant knowledge, providing necessary information for concepts in the original problem [18]. Simultaneously, the randomness and self-consistency generated by LLMs enable them to produce multiple inference chains. The majority voting method is then applied to the results obtained from all chains to derive final outputs.

2.5 Chain-Of-Thought Prompting

Chain-of-thought prompting is a special prompting engineering technique. It elicits LLMs' ability to solve problems step by step [19] and explicitly decomposes the reasoning and analysis process using the least-to-most prompting strategy [20]. In recommender systems, explicit steps can also be provided manually by researchers to assist in solving recommendation tasks [5]. Moreover, a recent study [21] demonstrates that Chain of Thought (COT) paradigms yield the highest accuracy and have the potential to apply to scenarios that favour more recent content, thereby offering a more balanced and up-to-date recommendation experience.

2.6 User Modeling for Recommendation

User modelling plays a pivotal role in recommendation systems as it captures user interests, preferences, and decisions once facing choices. Numerous studies have explored with diverse approaches. Some research focuses on learning user choice representations from their past behaviours and others incorporating extra information like general knowledge and social networks to model users. Chen et al. [22] propose the Social Attentional Memory Network (SAMN), a novel social-aware recommendation model. Lin et al. [5] present a recommendation model with Implicit Preference Communities that leverages user ratings and social connections. Huang et al. [23] introduce the Explainable Interaction-driven User Modelling (EIUM) algorithm, utilizing Knowledge Graphs (KG) for constructing effective and explainable sequential recommenders. Wang et al. [24] propose KGIE, a knowledge graph-enhanced sequential recommendation model that enhances user interest modelling through knowledge-enriched item sets.

2.7 User’s Interests Modelling

Above all, existing efforts using LLMs for recommendation focused on how to adopt LLMs with well-known recommendation methods to increase matching between user’s past behaviour with a user model. They ignored the user’s intention, which we believe is a crucial factor that led to mismatch between user’s selection and what they really want. Cognitively a user’s desire defines the user’s intention and it is this intention that generates multiple interests, and the subsequent selections when the user’s interests can be fulfilled with multiple items. Researchers are trying to explore the connection between user’s interests and their selections but fail to link the underlay intention and the dynamic change of the interests. Wang et al. [25] propose the Multi-Interest News Sequence (MINS) model for news recommendation. Portman et al. [26] introduce MiCRO, a generative statistical framework that models multi-interest user preferences and temporal multi-interest item representations. Yang et al. [27] propose KEMIM, a knowledge-enhanced user multi-interest modelling approach for recommender systems, which leverages a knowledge graph to discover explicit user interests, expand potential interests using relationship paths, and combine user attribute features. Zhang et al. [28] propose using LLMs with long- and short-term feature-wise attention layers to capture users’ long- and short-term preferences. These efforts again failed to capsulate the intention which is the key aspect which enables the utilization of users’ preferences in an adaptive way. Based on the above analyses, we set our task for trying LLMs for identify user intention for recommendations.

3 Methodology

Motivated by the recent advancements in LLMs, Our UII-COT framework builds on our previous work on user’s intention generation by an LLM through CoT [27]. We use an initial user profile as input to the LLM and design a collection of prompts to align the LLM's response through various recommendation subtasks such as rating prediction, clarification, and explanation. It is hoped that through iterations LLM can identify a user's intention.

3.1 Overall Framework

Figure 1 illustrates our proposed UII-COT framework, which uses an LLM and CoT. The framework incorporates appropriate prompt strategies to explore the utilization of the LLM in the identification of user intention and subsequential recommendation tasks. The UII-COT framework consists of three modules: (1) understanding user intention with Chain-of-Thought prompting; (2) prompt construction for recommendation, and (3) recommendation evaluation.

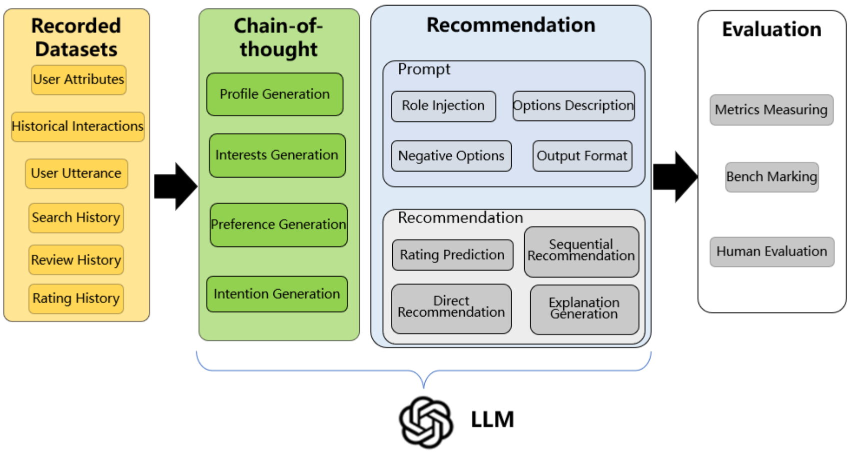

Fig. 1. An overview of the UII-COT framework.

3.2 Identify User Intention with Chain-Of-Thought Prompting

User intention understanding is a complex and difficult process that involves gathering user profiles, tracking user's past actions and extracting their interests and preferences. In our approach, instead of directly requesting the LLM for the user intention output, we expect the LLM to infer and integrate different aspects of user intention. To achieve this, we construct prompts in four logical steps as follows, ensuring a coherent flow of information:

Step 1. Given a set of user attributes, historical interactions between a user and items, and item metadata (e.g., item descriptions), the LLM follows the prompts to generate a natural language description that summarizes the user's profiles, supporting future recommendations.
Step 2. Building upon the description of user profiles, historical interactions, reviews, and ratings, we instruct the LLM to generate a description of the user's interests.
Step 3. In addition to describing user interests, we go further with some options for choices and try to figure out user preferences.
Step 4. With the three aspects of the user model as contextual information, we ultimately task the LLM with inferring a complete and coherent user intention by integrating these different elements.

Figure 2 is the illustration of user intention generation with CoT prompting. By employing this approach, we facilitate a logical and coherent flow of information, allowing the LLM to accurately summarize user profiles, describe user interests and preferences in detail, convey users' current intentions, and even a step further that we can continue our CoT to generate a new and complete user model by integrating these aspects seamlessly.

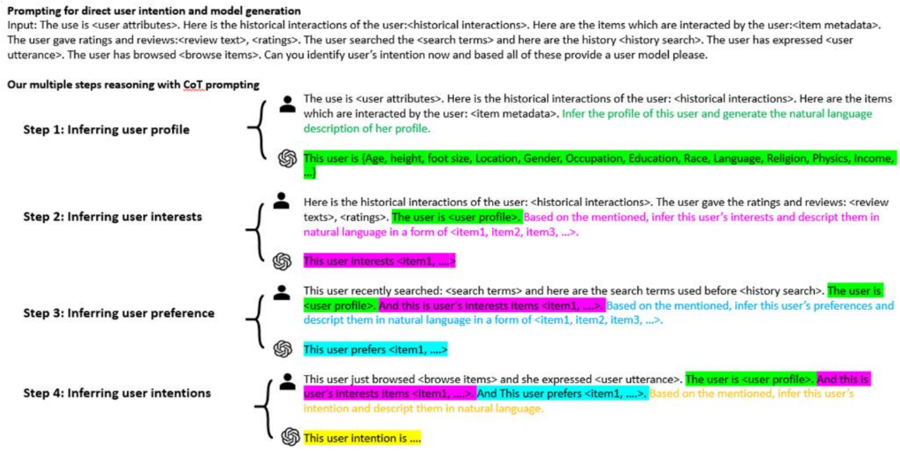

Fig. 2. The illustration of generative user intention with CoT prompting.

3.3 Prompt Construction for Recommendation

With the same inspiration, LLMs are used to generate recommendations with user intention. The recommendation relies on the model's comprehension of user intention and available item options. By designing appropriate prompt formats, we can effectively communicate with LLMs and ensure the models fully understand the task and generate efficacy output.

The basic construction of prompts for recommendation generation involves four essential components: injected identity, candidate description prompt, item should be avoided prompt, and output format prompt. We present a sample prompt for a typical recommendation task in Figure 3. In this task, the LLM is guided in selecting the most suitable item from a list of potential candidates. The recommendation incorporates intentions identified by the LLM through CoT and injects a role of the LLM as an expert in recommendation. The task description prompt is formulated as follows: Choose the top 10 items to recommend to the user and rank them in order of the degree of fit into the user's intention from highest to lowest.

Zero-shot: You are an expert in recommending movies. You must choose 10 movies for recommendation to a user and sort them in order of priority from highest to lowest based the match of user intention given as <user intention>, your output should be formatted in a JSON array. Do not include titles that user rated below 5 in his review and rating history. Unordered candidates are listed as follows: <candidates items>.

Few-shot: You are an expert in recommending movies. You must choose 10 movies for recommendation to a user and sort them in order of priority from highest to lowest based the match of user intention given as <user intention >, your output should be formatted in a JSON array. Do not include titles that user rated below 5 in his review and rating history. In addition, considering user recent search item in <item list>, Unordered candidates are listed as follows: <candidates items>.

Fig. 3. Prompt examples for the UII-COT.

Where, the yellow highlights represent the injected identity Prompt, the green highlights indicate the task description prompt, the blue highlights represent the task output format prompt, the grey highlights indicate the task boundary prompt (negative list), and the red highlights represent the few-shot prompt.

4 Experiments and Evaluation

We have conducted experiments and human evaluation on real-world datasets to evaluate our proposed UII-COT through recommendation results comparison with the benchmarking on selected Metrics. Human evaluation was done with selected human participants and their declared intention matched their identified intention through LLM by feeding their selections (for movies).

4.1 Experimental Setup

Human Evaluation. We evaluate our UII-COT performance on selected users with their declared intention and the intention identified by an LLM through the recommendation results. The data sets are the TG-ReDial dataset [29], the Douban Movie dataset [30], the Douban Book dataset [31], and the Amazon Beauty dataset [32].

The TG-ReDial dataset was created in a semi-automatic way by involving reasonable and controllable human annotation on personal movie selections. The movie-watching records were collected from real users on the Douban website. The dataset contains an average of 1,482 users and 202.7 watching records for each user. The movie information was extracted from movie tags on Douban (e.g., genre, director, and starring). Douban Movie was collected from the Douban website. Movie and actor data were collected in early August 2019. Movie review data (users, ratings, comments) were collected in early September 2019, a total of 9.45 million data, including 140,000 movies, 70,000 actors, 630,000 users, 4.16 million movie ratings, and 4.42 million movie reviews. Douban Book comes from the Douban website, including the book title, author, publisher, number of reviewers, ratings, price, number of pages, etc., of 60,000 Douban books. Amazon Beauty dataset contains 22,363 users, 12,101 items, and 198,502 product reviews from Amazon.com. We compare the recommendations generated by LLM and the actual user selections. When using LLM as a recommender, we first identify the user's intention, instead of comparing the user's intention directly we compare the recommendations generated by the LLM with that intention with the user's selections in accuracy, completeness, consistency, and interpretability. we employ well-known and widely used collaborative filtering methods: BPR-MF [33] and BPR-MLP [34] as baselines and use top-k hit rate (HR@{5,10}) and normalized discounted cumulative gain (NDCG@{5,10}) metrics for evaluation.

Implementation Details. To validate the effectiveness of our proposed approach, we employ the GPT-3.5-turbo model. To enable ChatGPT to learn user's intentions implicitly, we gather n items that users have interacted with and include k shots of historical records. Due to GPT-3.5-turbo's maximum context length of 4096 tokens, we set our n = 9 and k = 2. We also set the maximum length of input tokens max_tokens=1024, which gives us 3072 tokens to carry the contextual content. we randomly split each dataset into training (80%), validation (10%), and testing (10%) sets and ensured that there was at least one instance included in the training set for each user and item. We randomly choose 100 records from the test set to evaluate. Furthermore, we establish a candidate list of length 100, comprising one positive item and 99 negative samples, by setting the number of negative samples to 99. Additionally, we incorporate the candidate pool in the request, setting the number of shots to 1. We present the user's historical interacted items sequentially to ChatGPT, prompting it to predict the title of the next potential interaction. Specifically, we use the Sentires toolbox [35] to extract feature words and rating score explanations from review splits, resulting in explanation splits that are a subset of the original review divides. We randomly choose outcomes from several ways. We also employed three human evaluators assess and rank each outcome. After collecting the manually annotated results, we calculate the average top1 ratio and ranking position for each technique to evaluate their generation performance. We ran all the experiments with different random seeds three times and reported the average results with standard deviation to prevent extreme cases. For LLM-based methods, we show the standard deviation for few-shot prompting exemplars with three different random seeds.

4.2 Performance Comparison

The experimental findings are summarized in Table 1. The experimental results unequivocally demonstrate that UII-COT outperforms the state-of-the-art baseline and ChatGPT across all four datasets when utilizing a limited number of cues. However, it falls short of baseline performance in the zero-shot setting. In contrast, ChatGPT exhibits significantly lower performance than the baseline in both few-shot and zero-shot. This discrepancy may stem from the fact that when the textual descriptions of the item pool are input into the large language model, the model tends to prioritize semantic similarity among these texts rather than capturing the transformational relationships between items. Consequently, the large language model struggles to accurately discern features of items that align with the user's intention. Conversely, UII-COT excels in effectively capturing and comprehending user intention, interests and preferences. It proceeds to match items based on robust and complete user model which include all the past data and well obtained general knowledge. Therefore, resulting in more precise and contextually relevant recommendations.

Table 1. Performance comparison on recommendation (Shown is the mean ± s.d. of 3 runs with different random seeds)

		SOTA baselines		ChatGPT		UII-COT	
		BPR-MF	BPR-MLP	Zero-shot	Few-shot	Zero-shot	Few-shot
TG-ReDial	HR@5	0.1624± 0.0011	0.1587± 0.0009	0.0417	0.0549± 0.0003	0.0637	**0.1889**± 0.0028
	HR@10	0.2800±	0.2773±	0.0852	0.1130±	0.1061	**0.3071**±

		0.0016	0.0012		0.0008		0.0045
	NDCG@5	0.0975±0.0004	0.0936±0.0005	0.0309	0.0416±0.0003	0.0513	**0.1189**±0.0019
	NDCG@10	0.1415±0.0003	0.1400±0.0007	0.0404	0.0548±0.0005	0.0619	**0.1673**±0.0023
Douban-Movie	**HR@5**	0.1702±0.0005	0.1653±0.0014	0.0460	0.0579±0.0003	0.0582	**0.1795**±0.0030
	HR@10	0.2885±0.0015	0.2832±0.0006	0.0906	0.1193±0.0004	0.1111	**0.3322**±0.0047
	NDCG@5	0.1041±0.0008	0.0974±0.0011	0.0351	0.0470±0.0002	0.0574	**0.1228**±0.0030
	NDCG@10	0.1577±0.0015	0.1509±0.0009	0.0438	0.0575±0.0005	0.0661	**0.1905**±0.0022
Douban-Book	**HR@5**	0.1370±0.0016	0.1321±0.0003	0.0322	0.0560±0.0002	0.0511	**0.1675**±0.0012
	HR@10	0.2598±0.0014	0.2517±0.0013	0.0769	0.1005±0.0004	0.0939	**0.2983**±0.0039
	NDCG@5	0.0969±0.0002	0.0912±0.0007	0.0215	0.0328±0.0003	0.0427	**0.1183**±0.0015
	NDCG@10	0.1494±0.0016	0.1476±0.0008	0.0402	0.0536±0.0004	0.0614	**0.1679**±0.0023
Amazon-Beauty	**HR@5**	0.1426±0.0013	0.1392±0.0015	0.0217	0.0349±0.0004	0.0598	**0.1806**±0.0029
	HR@10	0.2573±0.0017	0.2542±0.0009	0.0652	0.0930±0.0003	0.1025	**0.2984**±0.0026
	NDCG@5	0.0857±0.0002	0.0848±0.0002	0.0111	0.0216±0.0005	0.0498	**0.1173**±0.0013
	NDCG@10	0.1224±0.0005	0.1215±0.0005	0.0252	0.0398±0.0003	0.0603	**0.1701**±0.0027

5 Conclusion

We employ large language models to understand user's intentions by feeding it with user's attributes, past selections, reviews and ratings. We deployed a CoT prompting to identify the user's current intention and integrate that intention into the user profile for a recommendation. We demonstrated how to adapt large language models for recommendations, we have designed a set of prompts where we take a natural language user profile as input, and the language model generates recommendation results based on specific instructions to identify the user's intention first. The experimental findings demonstrate the immense potential of large language models in identify user intention for recommendation. Furthermore, with that user intention integrated into a user model then using that model and LLM to generate recommendations has demonstrated improved recommendation performance compared to classical methods and LLM alone. We believe that exploring enhanced integration between recommendation systems and large language models holds great promise in future work. Finally, the success of LLMs in other areas has introduced novel human-machine interaction paradigms, necessitating research into new prompts and user interactions with the LLMs that can help to understand users, particularly in the context of recommendations.

Acknowledgment

This work was supported in part by the AI University Research Centre (AI-URC) through the XJTLU Key Program Special Fund (KSF-A-17).

References

1. Davidson J, Liebald B, Liu J, et al.: The YouTube video recommendation system. Proceedings of the fourth ACM conference on Recommender systems. 2010: 293-296. (2010).
2. Jin X, Zhou Y, Mobasher B.: Task-oriented web user modeling for recommendation. International Conference on User Modeling. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 109-118. (2005).
3. Hou Y, Zhang J, Lin Z, et al.: Large language models are zero-shot rankers for recommender systems[J]. arXiv preprint arXiv:2305.08845, (2023).
4. Gao Y, Sheng T, Xiang Y, et al.: Chat-rec: Towards interactive and explainable llms-augmented recommender system[J]. arXiv preprint arXiv:2303.14524, (2023).
5. Wang L, Lim E P.: Zero-Shot Next-Item Recommendation using Large Pretrained Language Models[J]. arXiv preprint arXiv:2304.03153, (2023).
6. Kang W C, Ni J, Mehta N, et al.: Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction[J]. arXiv preprint arXiv:2305.06474, (2023).
7. Acharya A, Brijraj S, and Naoyuki O.: LLM based generation of item-description for recommendation system. Proceedings of the 17th ACM Conference on Recommender Systems. 2023: 1204-1207. (2003).
8. Wang W, Lin X, Feng F, et al.: Generative recommendation: Towards next-generation recommender paradigm[J]. arXiv preprint arXiv:2304.03516, (2023).
9. Li J, Zhang W, Wang T, et al.: GPT4Rec: A generative framework for personalized recommendation and user interests interpretation[J]. arXiv preprint arXiv:2304.03879, (2023).
10. Liu Q, Chen N, Sakai T, et al.: A First Look at LLM-Powered Generative News Recommendation[J]. arXiv preprint arXiv:2305.06566, (2023).
11. Zhang J, Xie R, Hou Y, et al.: Recommendation as instruction following: A large language model empowered recommendation approach[J]. arXiv preprint arXiv:2305.07001, (2023).
12. Bao K, Zhang J, Zhang Y, et al.: Tallrec: An effective and efficient tuning framework to align large language model with recommendation[J]. arXiv preprint arXiv:2305.00447, (2023).
13. Li, Lei, Yongfeng Zhang, and Li Chen.: Prompt distillation for efficient LLM-based recommendation. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 1348-1357. (2023).
14. Brown T, Mann B, Ryder N, et al.: Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901. (2020).
15. Dai S, Shao N, Zhao H, et al.: Uncovering ChatGPT's Capabilities in Recommender Systems[J]. arXiv preprint arXiv:2305.02182, (2023).
16. T, Schütze H.: Exploiting cloze questions for few shot text classification and natural language inference[J]. arXiv preprint arXiv:2001.07676, (2020).
17. Gao T, Fisch A, Chen D.: Making pre-trained language models better few-shot learners[J]. arXiv preprint arXiv:2012.15723, (2020).
18. Wang X, Wei J, Schuurmans D, et al.: Self-consistency improves chain of thought reasoning in language models[J]. arXiv preprint arXiv:2203.11171, (2022).

19. Wei J, Wang X, Schuurmans D, et al.: Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in Neural Information Processing Systems, 2022, 35: 24824-24837. (2022).
20. Zhou, Denny, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans et al.: “Least-to-most prompting enables complex reasoning in large language models.” arXiv preprint arXiv:2205.10625 (2022).
21. Deldjoo, Yashar.: Understanding Biases in ChatGPT-based Recommender Systems: Provider Fairness, Temporal Stability, and Recency[J]. arXiv preprint arXiv:2401.10545, (2024).
22. Chen C, Zhang M, Liu Y, et al.: Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. Proceedings of the twelfth ACM international conference on web search and data mining. 2019: 177-185. (2019).
23. Huang X, Fang Q, Qian S, et al.: Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. proceedings of the 27th ACM international conference on multimedia. 2019: 548-556. (2019).
24. Wang C, Zhu Y, Liu H, et al.: Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 1889-1898.(2021).
25. Wang R, Lu W.: Modeling multi-interest news sequence for news recommendation[J]. arXiv preprint arXiv:2207.07331, (2022).
26. Portman F, Ragain S, El-Kishky A.: MiCRO: Multi-interest Candidate Retrieval Online[J]. arXiv preprint arXiv:2210.16271, (2022).
27. Yang F, Yue Y, Li G, et al.: KEMIM: Knowledge-enhanced User Multi-interest Modeling for Recommender Systems[J]. IEEE Access, (2023).
28. Zhang, Jianqing, Dongjing Wang, and Dongjin Yu.: TLSAN: Time-aware long-and short-term attention network for next-item recommendation[J]. Neurocomputing. 2021, 441: 179-191.(2021).
29. Zhou K, Zhou Y, Zhao W X, et al.: Towards topic-guided conversational recommender system[J]. arXiv preprint arXiv:2010.04125, (2020).
30. https://github.com/csuldw/AntSpider
31. https://www.heywhale.com/mw/dataset/5cbeb2088c90d7002c822b1
32. https://jmcauley.ucsd.edu/data/amazon/index_2014.html
33. Rendle S, Freudenthaler C, Gantner Z, et al.: BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint arXiv:1205.2618, (2012).
34. Cheng H T, Koc L, Harmsen J, et al.: Wide & deep learning for recommender systems. Proceedings of the 1st workshop on deep learning for recommender systems. 2016: 7-10. (2016).
35. Geng S, Liu S, Fu Z, et al.: Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). Proceedings of the 16th ACM Conference on Recommender Systems. 2022: 299-315. (2022).