World Wide Web (2024) 27:60
https://doi.org/10.1007/s11280-024-01291-2

A survey on large language models for recommendation

Likang Wu[1,2] · Zhi Zheng[1,2] · Zhaopeng Qiu[2] · Hao Wang[1] · Hongchao Gu[1] · Tingjia Shen[1] · Chuan Qin[2] · Chen Zhu[2] · Hengshu Zhu[2] · Qi Liu[1] · Hui Xiong[3] · Enhong Chen[1]

Received: 17 June 2024 / Revised: 9 July 2024 / Accepted: 17 July 2024 /
Published online: 22 August 2024

Abstract

Large Language Models (LLMs) have emerged as powerful tools in the field of Natural Language Processing (NLP) and have recently gained significant attention in the domain of Recommendation Systems (RS). These models, trained on massive amounts of data using self-supervised learning, have demonstrated remarkable success in learning universal representations and have the potential to enhance various aspects of recommendation systems by some effective transfer techniques such as fine-tuning, prompt tuning, etc. The crucial aspect of harnessing the power of language models in enhancing recommendation quality is the utilization of their high-quality representations of textual features and their extensive coverage of external knowledge to establish correlations between items and users. To provide a comprehensive understanding of the existing LLM-based recommendation systems, this survey presents a taxonomy that categorizes these models into two major paradigms, respectively Discriminative LLM for Recommendation (DLLM4Rec) and Generative LLM for Recommendation (GLLM4Rec), with the latter being systematically sorted out for the first time. Furthermore, we systematically review and analyze existing LLM-based recommendation systems within each paradigm, providing insights into their methodologies, techniques, and performance. Additionally, we identify key challenges and several valuable findings to provide researchers and practitioners with inspiration. We have also created a GitHub repository to index relevant papers and resources on LLMs for recommendation (https://github.com/WLiK/LLM4Rec-Awesome-Papers).

Keywords Large language models · Recommendation system

1 Introduction

Recommendation systems play a critical role in assisting users in finding relevant and personalized items or content. With the emergence of Large Language Models (LLMs) in Natural Language Processing (NLP), there has been a growing interest in harnessing the power of these models to enhance recommendation systems.

Zhi Zheng and Zhaopeng Qiu are contributed equally to this work

Extended author information available on the last page of the article

The key advantage of incorporating LLMs into recommendation systems lies in their ability to extract high-quality representations of textual features and leverage the extensive external knowledge encoded within them [1]. And this survey views LLM as the Transformer-based model with a large number of parameters, trained on massive datasets using self/semi-supervised learning techniques, e.g., BERT, GPT series, PaLM series, etc[1]. Unlike traditional recommendation systems, the LLM-based models capture contextual information, and comprehend user queries, item descriptions, and other textual data more effectively [2]. By understanding the context, LLM-based RS can improve the accuracy and relevance of recommendations, leading to enhanced user satisfaction. Meanwhile, facing the common data sparsity issue of limited historical interactions [3], LLMs also bring new possibilities to recommendation systems through zero/few-shot recommendation capabilities [4]. These models can generalize to unseen candidates due to the extensive pre-training with factual information, domain expertise, and common-sense reasoning, enabling them to provide reasonable recommendations even without prior exposure to specific items or users.

The aforementioned strategies are already well-applied in discriminative models. However, with the evolution of AI learning paradigms, generative language models have started to gain prominence [5] as shown in Figure 1. A prime example of this is the emergence of ChatGPT and other comparable models, which have significantly disrupted human life and work patterns. Furthermore, the fusion of generative models with recommendation systems offers the potential for even more innovative and practical applications. For instance, the interpretability of recommendations can be improved, as LLM-based systems are able to provide explanations based on their language generation capabilities [6], helping users understand the factors influencing the recommendations. Moreover, generative language models enable more personalized and context-aware recommendations, such as users' customizable prompts [7] in the chat-based recommendation system, enhancing user engagement and satisfaction with the diversity of results.

Motivated by the remarkable effectiveness of the aforementioned paradigms in solving data sparsity and efficiency issues, the adaptation of language modeling paradigms for recommendation has emerged as a promising direction in both academia and industry, significantly advancing the state-of-the-art in the research of recommendation systems. In the early stage, there are a few studies that review relevant papers in this domain [1, 8]. Zeng et al. [8] summarizes some research on the pre-training of recommendation models and discusses knowledge transfer methods between different domains. Liu et al. [1] proposes an orthogonal taxonomy to divide existing pre-trained language model-based recommendation systems w.r.t. their training strategies and objectives, analyzes and summarizes the connection between pre-trained language model-based training paradigms and different input data types. However, both of these surveys primarily focus on the transfer of training techniques and strategies in pretraining language models, rather than exploring the potential of language models and their capabilities, i.e., LLM-based way. Additionally, they lack a comprehensive overview of the recent advancements and systematic introductions of generative large language models in the recommendation field. To address this issue, we delve into LLM-based recommendation systems, categorizing them into discriminative LLMs for recommendation and generative LLMs for recommendation, and the focus of our review is on the latter. Recently, there have been several reviews introducing the application of large language models in recommendation systems or related technologies [9–12]. However, our paper is the first to comprehensively summarize three representative modeling paradigms of applying large language models in recommendation systems. Its distinct characteristics of being concise yet broadly and accu-

[1] https://en.wikipedia.org/wiki/Large_language_model

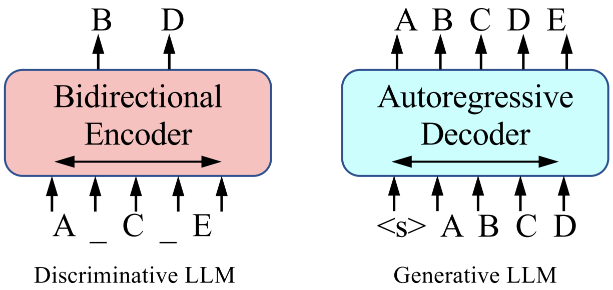


Figure 1 Two major training paradigms of large language models: Discriminative LLM (e.g. BERT) and Generative LLM (e.g. GPT)

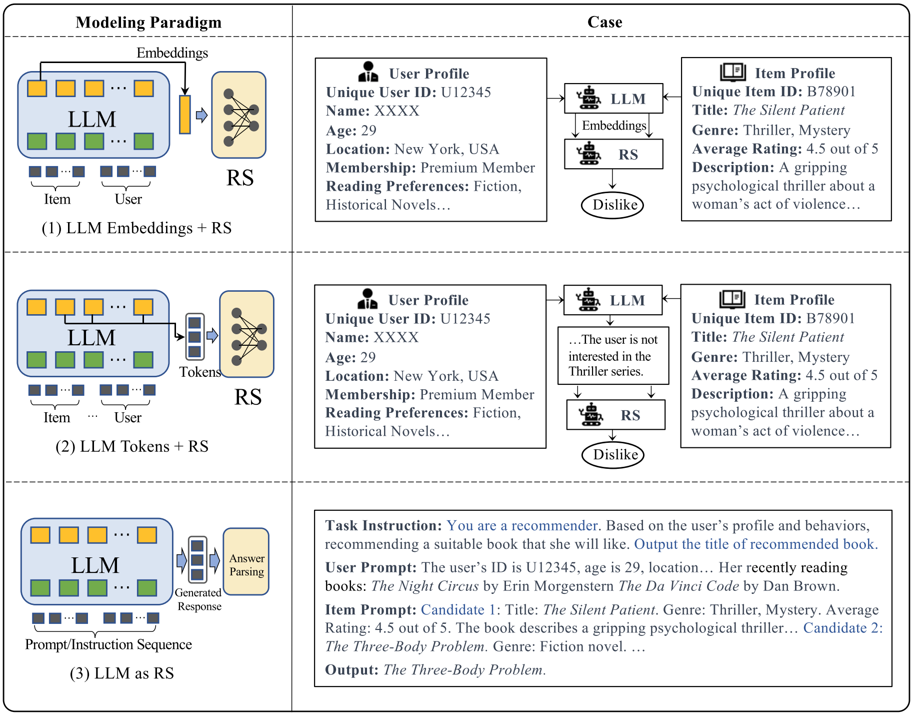
rately covered have garnered significant attention in the industry. Additionally, our paper thoroughly reviews the major challenges faced by large language models in the recommendation field, providing valuable insights to guide future research directions in this area. The main contributions of our survey are summarized as follows:

- We present a systematic survey of the current state of LLM-based recommendation systems, focusing on expanding the capacity of language models. We provide a systematic overview of related advancements and applications by analyzing the existing methods.
- From the perspective of modeling paradigms, we categorize the current studies of large language model recommendations into three distinct schools of thought. Any existing method can be fittingly placed within these categories, thereby providing a clear and organized overview of this burgeoning field.
- Our survey critically analyzes the advantages, disadvantages, and limitations of existing methods. We identify key challenges faced by LLM-based recommendation systems and propose valuable findings that can inspire further research in this potential field.

2 Modeling paradigms and taxonomy

The basic framework of all large language models is composed of several transformer blocks, e.g., GPT, PaLM, LLaMA, etc. The input of this architecture is generally composed of token embeddings or position embeddings and so on, while the expected output embedding or tokens can be obtained at the output module. Here, both the input and output data types are textual sequences. As shown in (1)-(3) in Figure 2, for the adaption of language models in recommendations, i.e., the modeling paradigm, existing work can be roughly divided into the following three categories:

1. **LLM Embeddings + RS**. This modeling paradigm views the language model as a feature extractor, which feeds the features of items and users into LLMs and outputs corresponding embeddings. A traditional RS model can utilize knowledge-aware embeddings for various recommendation tasks.
2. **LLM Tokens + RS**. Similar to the former method, this approach generates tokens based on the inputted items' and users' features. The generated tokens capture potential preferences through semantic mining, which can be integrated into the decision-making process of a recommendation system.
3. **LLM as RS**. Different from (1) and (2), this paradigm aims to directly transfer pre-trained LLM into a powerful recommendation system. The input sequence usually consists of the



User Profile
Unique User ID: U12345
Name: XXXX
Age: 29
Location: New York, USA
Membership: Premium Member
Reading Preferences: Fiction, Historical Novels…

Item Profile
Unique Item ID: B78901
Title: *The Silent Patient*
Genre: Thriller, Mystery
Average Rating: 4.5 out of 5
Description: A gripping psychological thriller about a woman's act of violence…

…The user is not interested in the Thriller series.

Task Instruction: You are a recommender. Based on the user's profile and behaviors, recommending a suitable book that she will like. Output the title of recommended book.

User Prompt: The user's ID is U12345, age is 29, location… Her recently reading books: *The Night Circus* by Erin Morgenstern *The Da Vinci Code* by Dan Brown.

Item Prompt: Candidate 1: Title: *The Silent Patient*. Genre: Thriller, Mystery. Average Rating: 4.5 out of 5. The book describes a gripping psychological thriller… Candidate 2: *The Three-Body Problem*. Genre: Fiction novel. …

Output: *The Three-Body Problem*.

Figure 2 Three representative modeling paradigms of the research for large language models on recommendation systems

profile description, behavior prompt, and task instruction. The output sequence is expected to offer a reasonable recommendation result.

In practical applications, the choice of large language models significantly influences the design of modeling paradigms in recommendation systems. As shown in Figure 3, in this paper, we categorize existing works into two main categories, discriminative LLMs and generative LLMs for recommendation, respectively. The taxonomy of development styles of LLMs for recommendation can be further subdivided based on the training manner, and the distinction among different manners is illustrated clearly in Figure 4. Generally, discriminative language models are well-suited for embedding within the paradigm (1), while the response generation capability of generative large language models further supports paradigms (2) or (3).

3 Discriminative LLMs for recommendation

Discriminative large language models, such as those from the BERT series [13], are particularly adept at natural language understanding tasks and are frequently used as embedding backbones for various downstream applications. This includes recommendation systems, where pre-trained models like BERT are commonly fine-tuned to align their representations with domain-specific data. Additionally, some research explores training strategies like prompt tuning and adapter tuning. The representative approaches and common-used datasets are listed in Tables 1 and 2.

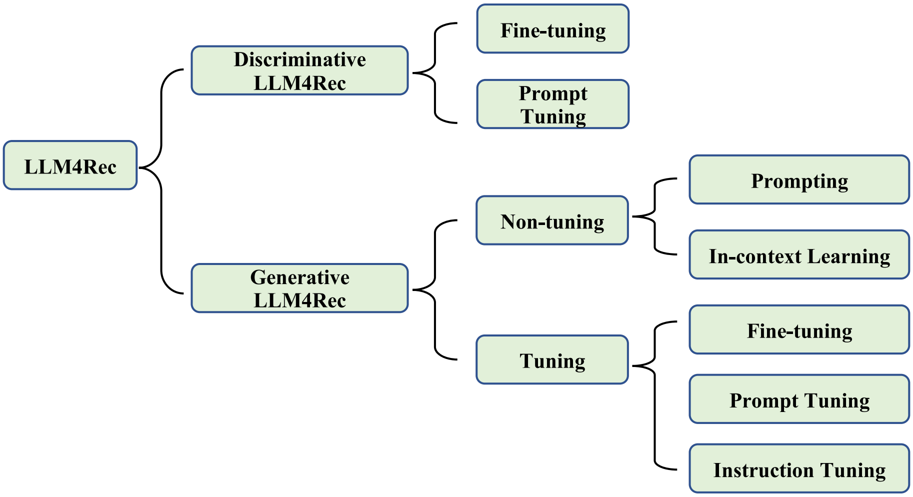


Figure 3 A taxonomy of the research for large language models on recommendation systems

3.1 Fine-tuning

Fine-tuning pre-trained language models is a universal technique that has gained significant attention in various natural language processing (NLP) tasks, including recommendation systems. The idea behind fine-tuning is to take a language model, which has already learned rich linguistic representations from large-scale text data, and adapt it to a specific task or

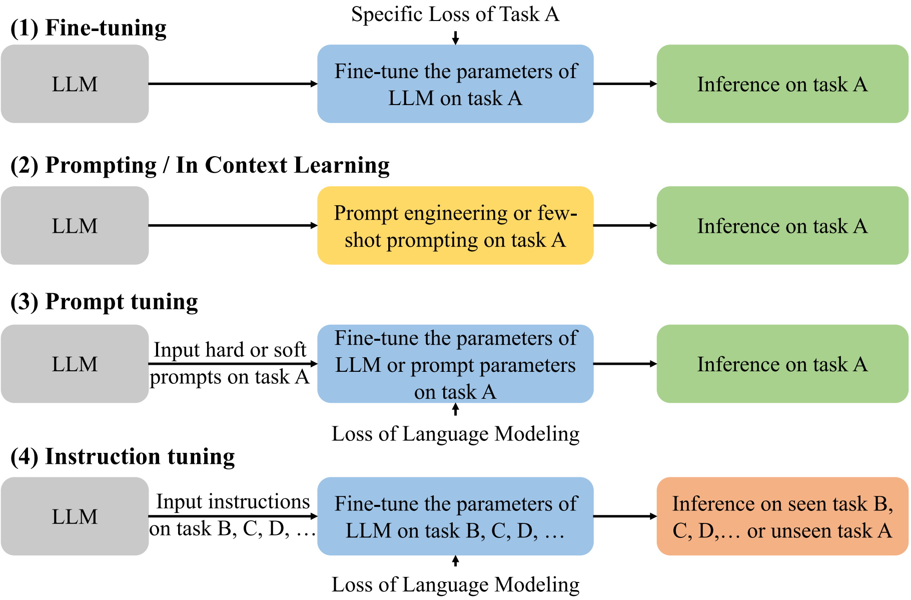

(1) Fine-tuning
Specific Loss of Task A
LLM
Fine-tune the parameters of LLM on task A
Inference on task A
(2) Prompting / In Context Learning
LLM
Prompt engineering or few-shot prompting on task A
Inference on task A
(3) Prompt tuning
LLM
Input hard or soft prompts on task A
Fine-tune the parameters of LLM or prompt parameters on task A
Inference on task A
Loss of Language Modeling
(4) Instruction tuning
LLM
Input instructions on task B, C, D, …
Fine-tune the parameters of LLM on task B, C, D, …
Inference on seen task B, C, D,… or unseen task A
Loss of Language Modeling

Figure 4 Detailed explanation of five different training (domain adaption) manners for LLM-based recommendations

Table 1 A list of representative LLM-based recommendation methods and their features

Adaption	Paper	Base model	Recommendation task	Modeling paradigm
Discriminative LLMs for recommendation				
Fine-tuning	[28]	BERT/UniLM	News recommendation	LLM embeddings + RS
	[14]	BERT	User representation	LLM embeddings + RS
	[19]	BERT	Group recommendation	LLM as RS
	[20]	BERT	Search/Matching	LLM embeddings + RS
	[21]	BERT	CTR prediction	LLM embeddings + RS
	[114]	BERT/RoBERTa	Conversational RS	LLM Embeddings + RS
Prompt tuning	[38]	BERT	Sequential recommendation	LLM as RS
	[36]	DistilBERT/GPT-2	Conversational RS	LLM as RS
	[37]	BERT	Conversational RS	LLM embeddings + RS
	[35]	BERT	Conversational RS	LLM as RS
Generative LLMs for recommendation				
Non-tuning	[58]	ChatGPT	News recommendation	LLM tokens + RS
	[92]	DialoGPT/RoBERTa	Converational RS	LLM tokens + RS / LLM as RS
	[4]	GPT-2	Sequential recommendation	LLM as RS
	[56]	GPT-3.5	Sequential recommendation	LLM Tokens + RS / LLM as RS
	[6]	ChatGPT/GPT-3.5	Sequential recommendation	LLM as RS
	[68]	ChatGPT	Generative recommendation	LLM as RS
	[47]	ChatGPT	Sequential recommendation	LLM as RS
	[48]	ChatGPT/GPT-3.5	Passage reranking	LLM as RS
	[115]	T5/GPT-3.5/GPT-4	Passage reranking	LLM as RS
	[41]	ChatGPT	Five tasks	LLM as RS
	[42]	ChatGPT/GPT-3.5	Sequential recommendation	LLM as RS

Table 1 continued

Adaption	Paper	Base model	Recommendation task	Modeling paradigm
	[46]	ChatGLM	CTR Prediction	LLM Tokens + RS
	[70]	ChatGPT	Recommendation agent	LLM Tokens + RS
	[71]	ChatGPT	Recommendation agent	LLM Tokens + RS
Tuning	[109]	FLAN-T5	Three tasks	LLM as RS
	[78]	FLAN-T5/ChatGPT	Rating prediction	LLM as RS
	[81]	LLaMA-7B	Movie/Book RS	LLM as RS
	[108]	GPT-4	Explainable RS	LLM Tokens + RS
	[2]	T5	Five tasks	LLM as RS
	[110]	M6	Five tasks	LLM as RS
	[93]	BART/LLaMA	Text-based/Sequential recommendation	LLM as RS
	[106]	BELLE	Job recommendation	LLM as RS
	[105]	BELLE	Generative recommendation	LLM Tokens +RS
	[116]	UniTRecAU	Text-based recommendation	LLM as RS
	[94]	LLaMA-7B	Movie/Game RS	LLM as RS
	[32]	OPT	Text-based recommendation	LLM Embeddings +RS
	[112]	T5-small	Three tasks	LLM as RS
	[117]	RoBERTa/GLM	CTR prediction	LLM Embeddings +RS

Note that, here the target of tuning/non-tuning denotes the used LLM module in the following methods

Table 2 A list of common datasets used in existing LLM-based recommendation methods

Name	Scene	Tasks	Information	URL
Amazon review [118]	Commerce	Seq Rec / CF Rec	This is a large crawl of product reviews from Amazon. Ratings: 82.83 million, Users:20.98 million, Items: 9.35 million, Timespan: May 1996 - July 2014	http://jmcauley.ucsd.edu/data/amazon/
Amazon-M2 [87]	Commerce	Seq Rec / CF Rec	A large dataset of anonymized user sessions with their interacted products collected from multiple language sources at Amazon. It includes 3,606,249 train sessions, 361,659 test sessions, and 1,410,675 products.	https://arxiv.org/abs/2307.09688
Amazon review 2023 [119]	Commerce	Seq Rec / CF Rec	The dataset comprises over 570 million reviews and 48 million items from 33 categories.	https://amazon-reviews-2023.github.io
Steam [120]	Game	Seq Rec / CF Rec	Reviews represent a great opportunity to break down the satisfaction and dissatisfaction factors around games. Reviews: 7,793,069, Users: 2,567,538, Items: 15,474, Bundles: 615	https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data
MovieLens	Movie	General	The dataset consists of 4 sub-datasets, which describe users’ ratings to moives and free-text tagging activities from MovieLens, a movie recommendation service.	https://grouplens.org/datasets/movielens/

Table 2 continued

Name	Scene	Tasks	Information	URL
Yelp	Commerce	General	There are 6,990,280 reviews, 150,346 businesses, 200,100 pictures, 11 metropolitan areas, 908,915 tips by 1,987,897 users. Over 1.2 million business attributes like hours, parking, availability, etc.	https://www.yelp.com/dataset
Douban [121]	Movie, Music, Book	Seq Rec / CF Rec	This dataset includes three domains, i.e., movie, music, and book, and different kinds of raw information, i.e., ratings, reviews, item details, user profiles, tags (labels), and date.	https://github.com/ MarkWuNLP/ MultiTurnResponse Selection
MIND [122]	News	General	MIND contains about 160k English news articles and more than 15 million impression logs generated by 1 million users. Every news contains textual content including title, abstract, body, category, and entities.	https://msnews.github.io/assets/doc/ACL2020_MIND.pdf
U-NEED [123]	Commerce	Conversa -tion Rec	U-NEED consists of 7,698 fine-grained annotated pre-sales dialogues, 333,879 user behaviors, and 332,148 product knowledge tuples.	https://github.com/LeeeeoLiu/U-NEED

Table 2 continued

Name	Scene	Tasks	Information	URL
KuaiSAR [124]	Video	Search and Rec	KuaiSAR contains genuine search and recommendation behaviors of 25,877 users, 6,890,707 items, 453,667 queries, and 19,664,885 actions within a span of 19 days on the Kuaishou app.	https://kuaisar.github.io/
Tenrec [125]	Video, Article	General	Tenrec is a large-scale benchmark dataset for recommendation systems. It contains around 5 million users and 140 million interactions.	https://tenrec0.github.io/
PixelRec [126]	Video	Seq Rec / CF Rec	PixelRec is a massive image-centric recommendation dataset that includes approximately 200 million user-image interactions, 30 million users, and 400,000 cover images. The texts and other aggregated attributes of videos are also included.	https://github.com/westlake-repl/PixelRec

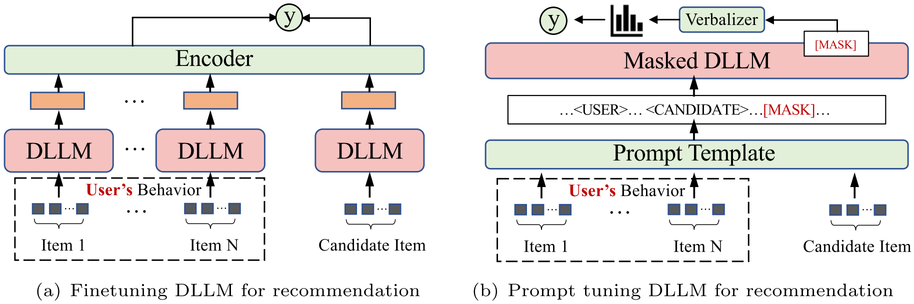
domain by further training it on task-specific data. The classic architect is shown in Figure 5 (a).

The process of fine-tuning involves initializing the pre-trained language model with its learned parameters and then training it on a recommendation-specific dataset. This dataset typically includes user-item interactions, textual descriptions of items, user profiles, and other relevant contextual information. During fine-tuning, the model's parameters are updated based on the task-specific data, allowing it to adapt and specialize for target recommendation tasks. The learning objectives in the pre-training and fine-tuning stages can vary, as they are aimed at different optimization targets.

Since the fine-tuning strategy is flexible, most BERT-enhanced recommendation methods can be summarized into this track. For the basic representation task, [14] proposed a novel pre-training and fine-tuning-based approach U-BERT to learn users' representation, which leveraged content-rich domains to complement those users' feature with insufficient behavior data. A review co-matching layer is designed to capture implicit semantic interactions between the reviews of users and items. Similarly, in UserBERT [15], two self-supervision tasks are incorporated for user model pre-training on unlabeled behavior data to empower user modeling. This model utilizes medium-hard contrastive learning, masked behavior prediction, and behavior sequence matching to train accurate user representation via captured inherent user interests and relatedness.

The pre-trained BERT achieved outstanding breakthroughs in the ranking task as well. BECR [16] proposed a lightweight composite re-ranking scheme that combined deep contextual token interactions and traditional lexical term-matching features at the same time. With a novel composite token encoding, BECR effectively approximates the query representations using pre-computable token embeddings based on uni-grams and skip-n-grams, allowing for a reasonable tradeoff between ad-hoc ranking relevance and efficiency. Besides, [17] proposed an end-to-end multi-task learning framework for product ranking with fine-tuned domain-specific BERT to address the issue of vocabulary mismatch between queries and products. The authors utilized the mixture-of-experts layer and probability transfer between tasks to harness the abundant engagement data. In a more specific situation of code example recommendations, the authors revealed that utilizing natural language queries (BERT + LSH) yields better ranking results compared to API-based queries [18].

There are also many related studies in other specific tasks or scenarios, e.g., group recommendation [19], search/matching [20], CTR prediction [21]. Especially, the "pre-train, fine-tuning" mechanism played an important role in several sequential or session-based recommendation systems, such as BERT4Rec [22], RESETBERT4Rec [23], and Adapter



(a) Finetuning DLLM for recommendation (b) Prompt tuning DLLM for recommendation

Figure 5 Discriminative LLMs for recommendation

Tuning [24, 25]. However, the above models only leveraged the advantages of the training strategy rather than expanding the large language model into the recommendation field, so it was not the focus of our discussion. The sequence representation learning model UniSRec [26] developed a BERT-fine-tuned framework, which associated description text of items to learn transferable representations across different recommendation scenarios. Considering the binding between item text and item representations might be too tight, leading to potential problems such as over-emphasizing the effect of text features and exaggerating the negative impact of domain gap, Hou et al [27] proposed to learn distinguishable vector-quantized item codes for transferable sequential recommenders. For the content-based recommendation, especially news recommendation, NRMS [28], Tiny-NewsRec [29], PREC [30], exploited large language models to empower news recommendation via handling known domain shift problems or reducing transfer cost. Specifically, to answer the crucial question that *Can a purely modality-based recommendation model (MoRec) outperforms or matches a pure ID-based model (IDRec) by replacing the itemID embedding with a SOTA modality encoder?*, [31] conducted large-scale experiments and found that modern MoRec could already perform on par or better than IDRec with the typical recommendation architecture (i.e., Transformer backbone) even in the non-cold-start item recommendation setting with the SOTA and E2E-trained Modality Encoder. The subsequent exploration [32] based on larger-scale language model encoders, e.g. OPT [33], further validated the viewpoint.

In summary, the integration of BERT fine-tuning into recommendation systems fuses the powerful external knowledge and personalized user preference, which primarily aims to promote recommendation accuracy and simultaneously obtains a little cold-start handling capability for new items with limited historical data.

3.2 Prompt tuning

Instead of adapting LLMs to different downstream recommendation tasks by designing specific objective functions, prompt tuning [34] aims to align the objective of recommendation tuning with pre-trained loss, using hard/soft prompts and a label word verbalizer. As shown in Figure 5(b), due to the mask-based training commonly employed in DLLM, the role of the mentioned verbalizer is to establish a mapping between the words predicted by DLLM at the [MASK] position and the actual labels. This association allows for the linkage between the language model and the task, ensuring their alignment. For example, [35] leveraged BERT's Masked Language Modeling (MLM) head to uncover its understanding of item genres using cloze-style prompts. They further utilized BERT's Next Sentence Prediction (NSP) head and similarity (SIM) of representations to compare relevant and non-relevant search and recommendation query-document inputs. The experiment showed that BERT, even without any fine-tuning, can prioritize relevant items in the ranking process. Yang et al. [36] developed a conversational recommendation system with prompts, where a BERT-based item encoder directly mapped the metadata of each item to an embedding. Similarly, Shen et al [37] developed a conversational recommendation system that incorporates user-item attribute fairness analysis. They achieved this by employing constructed prompt templates with placeholders (referred to as template-based result generation). These templates include non-preferential information such as names or relationships, which can implicitly indicate characteristics like race, gender, sexual orientation, geographical context, and religion. The analysis demonstrated that by combining train side masking and test side neutralization of non-preferential entities, the observed biases can be eliminated without compromising recommendation performance. Recently, Prompt4NR [38] pioneered the application of the prompt learning paradigm for news recommendation. This framework redefined the objective of predicting user clicks on candidate news as a cloze-style mask-prediction task. The experiments found that the performance of recommendation systems is noticeably enhanced through the utilization of multi-prompt ensembling, surpassing the results achieved with a single prompt on discrete and continuous templates. This highlights the effectiveness of prompt ensembling in combining multiple prompts to make more informed decisions.

4 Generative LLMs for recommendation

Compared to discriminative models, generative models excel in natural language generation. Thus, rather than aligning the representations learned by LLMs to the recommendation domain, generative model-based approaches often reframe recommendation tasks as natural language tasks. Techniques such as in-context learning, prompt tuning, and instruction tuning are then employed to adapt LLMs for directly generating recommendation results, like a Q&A problem. Moreover, with the impressive capabilities demonstrated by ChatGPT, this type of work has received increased attention recently.

As shown in Figure 3, according to whether tuning parameters, these generative LLM-based approaches can be further subdivided into two paradigms: *non-tuning paradigm* and *tuning paradigm*. **Here the tuning/non-tuning target denotes the used LLM module in the following methods.** The following two sub-sections will address their details, respectively. The representative approaches and common-used datasets are also listed in Tables 1 and 2.

4.1 Non-tuning paradigm

The LLMs have shown strong zero/few-shot abilities in many unseen tasks [39, 40]. Hence, some recent works assume LLMs already have the recommendation abilities, and attempt to trigger these abilities by introducing specific prompts. They employ the recent practice of Instruction and In-Context Learning [39] to adopt the LLMs to recommendation tasks without tuning model parameters. According to whether the prompt includes the demonstration examples, the studies in this paradigm mainly belong to the following two categories: *prompting* and *in-context learning*.

4.1.1 Prompting

This category of work aims to design more suitable instructions and prompts to help LLMs better understand and solve the recommendation tasks. [41] systematically evaluated the performance of ChatGPT on five common recommendation tasks, i.e., *rating prediction*, *sequential recommendation*, *direct recommendation*, *explanation generation*, and *review summarization*. They proposed a general recommendation prompt construction framework, which consists of the following elements: (1) task description, adapting recommendation tasks to natural language processing tasks; (2) behavior injection, incorporating user-item interaction to aid LLMs in capturing user preferences and needs; (3) format indicator, constraining the output format and making the recommendation results more comprehensible and assessable. Similarly, [42] conducted an empirical analysis of ChatGPT's recommendation abilities on three common information retrieval tasks, including point-wise, pair-wise, and list-wise ranking. They proposed different prompts for different kinds of tasks and introduced the *role* instructions (such as *You are a news recommendation system now.*) at the beginning of the prompts to enhance the domain adaption ability of ChatGPT. Lin and Zhang [43] explored the feasibility of developing an Artificial General Recommender (AGR) using Large Language Models (LLMs) from the perspective of ten fundamental principles, such as *contextual memory*, *repair mechanism* and *feedback mechanism*.

To evaluate the enhancement of different prompting inputs, [44] designed three prompt templates for the case of Items only (the attribute of items), Language only (the description of user's preference), and combined Language+Items in their experiments. After analyzing the performance of language models, they discovered that zero-shot and few-shot strategies are highly effective for making recommendations based solely on language-based preferences (without considering item preferences). In fact, these strategies have proven to be remarkably competitive in comparison to item-based collaborative filtering methods, particularly in near cold-start scenarios. Meanwhile, to summarize the user's intention by prompt based on their interaction data, MINT [45] employed InstructGPT, a 175B parameter LLM, to generate a synthetic narrative query. This query was then filtered using a smaller language model, and retrieval models were trained on both the synthetic queries and user items. The results indicate that the resulting models outperformed several strong baseline models and ablated models. In a one-shot setup, these models matched or even outperformed a 175B LLM that was directly used for narrative-driven recommendation. However, these methods have not considered decomposing the topics in a textual description, which would result in noisy and target-unclear prompts. KAR [46] solved this issue by introducing factorization prompting to elicit accurate reasoning on user preferences and factual knowledge.

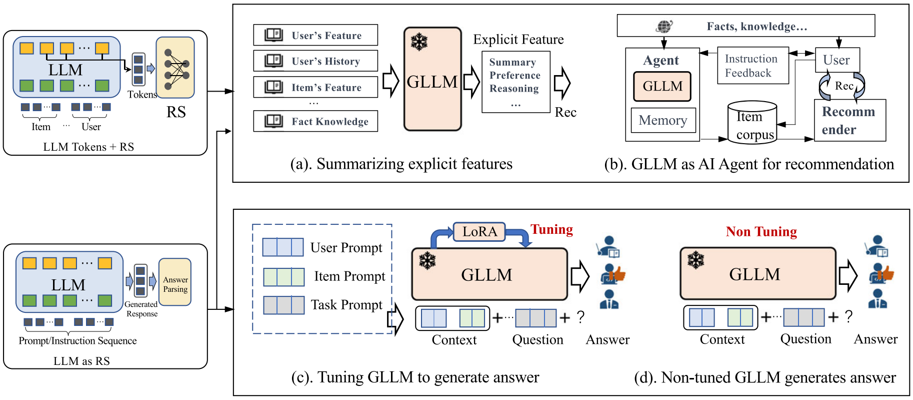
Instead of proposing a general framework, some works focus on designing effective prompts for specific recommendation tasks. Sileo [4] mined the movie recommendation prompts from the pre-training corpus of GPT-2. Hou [47] introduced two prompting methods to improve the sequential recommendation ability of LLMs: *recency-focused sequential prompting*, enabling LLMs to perceive the sequential information in the user interaction history, and *bootstrapping*, shuffling the candidate item list multiple times and taking the average scores for ranking to alleviate the position bias problem. Due to the limited number of input tokens allowed for the LLMs, it is hard to input a long candidate list in the prompt. To solve this problem, [48] proposed a sliding window prompt strategy, which only ranks the candidates in the window each time, then slides the window in back-to-first order, and finally repeats this process multiple times to obtain the overall ranking results. Yang et al. [49] designed a sequence-residual prompt to use LLMs to improve the interpretability of traditional sequential recommender. [50] proposed a multi-perspective criteria ensemble framework that improves the consistency and comprehensiveness of pointwise LLM rankers by simulating a virtual annotation team with diverse expertise. For conversational recommendation tasks, [51] provided empirical evidence that LLMs can outperform specialized models without the need for fine-tuning. Additionally, the authors constructed a new real-world dataset by extracting conversations from the popular website Reddit.

In addition to taking LLMs as recommendation systems, some studies also utilize LLMs to construct model features to improve the conventional recommender system. Archaya et al. [52–54] and LLM-Rec [55] used LLMs and prompting strategies to conduct content augmentation to enhance the features of item perspective. From the user feature perspective, NIR [56] designed prompts to generate user preference description and LLMRG [57] used ChatGPT and knowledge base to construct reasoning graph to enhance user representations. GENRE [58] introduced three prompts to employ LLMs to conduct three feature enhancement sub-tasks for news recommendation from both user and item perspectives. Specifically, it used ChatGPT to refine the news titles according to the abstract, extract profile keywords from the user reading history, and generate synthetic news to enrich user historical interactions.

Similarly, LLMRec [59] and RLMRec [60] first used ChatGPT to generate textual features for users and items and then took these features to enhance the ID-based representation learning.

In practice, in addition to the ranking model, the whole recommendation system generally consists of multiple important components, such as a content database and a candidate retrieval model. Hence, another line of using LLMs for recommendation is taking them as the controllers of the whole system. ChatREC [6], RAH [61], BiLLP [62] and InteRecAgent [63] designed the interactive recommendation framework around LLMs, which understands user requirements through multi-turn dialogues, and calls existing recommendation systems and various tools, such as database, retriever, memory, to provide results. The agent can play a crucial role in such conversational situations, thus several advanced versions are developed to optimize the fusion between chat intelligence and recommender module [64, 65]. What's more, some recent agent-based models [66, 67] propose recommendation frameworks for LLM-based agent platforms, emphasizing personalized agent services through enhanced interaction and collaboration among users, agent recommenders, and agent items. GeneRec [68] proposed a generative recommendation framework and used LLMs to control when to recommend existing items or to generate new items by AIGC models. Furthermore, [69], RecAgent [70] and Agent4Rec [71] further utilized LLM as intelligent simulator to develop a virtual recommendation environment. The simulator typically consists of two main modules: user and recommender. The user module enables browsing the recommendation site, interaction with other users, and posting on social media. The recommender module offers tailored search and recommendation lists, supporting various model designs for recommendation. Users in the environment interact based on LLM-generated responses, evolving organically to mirror real-world behavior. These projects show potential utilization across several applications, such as simulating the feedback for RL-based recommendations, tracking information dissemination process among the users on social media, investigating the filter bubble effect, and unveiling the underlying causal relationships embedded within recommender system scenarios. Instead of using LLMs as controllers, UniLLMRec [72] proposed an end-to-end chain-like recommendation framework that leverages LLMs to efficiently integrate recall, ranking, and re-ranking tasks.

In summary, these studies employ natural language prompts to leverage the zero-shot capabilities of LLMs for recommendation tasks, offering a cost-effective and pragmatic approach (Figure 6).



(a). Summarizing explicit features

(b). GLLM as AI Agent for recommendation

(c). Tuning GLLM to generate answer

(d). Non-tuned GLLM generates answer

Figure 6 Generative LLMs for recommendation

4.1.2 In-context learning

In-context learning is a technique used by GPT-3 and other LLMs to quickly adapt to new tasks and information. With a few demonstration input-label pairs, they can predict the label for an unseen input without additional parameter updates [73]. Hence, some works attempt to add demonstration examples in the prompt to make LLMs better understand the recommendation tasks. For sequential recommendation, [47] introduced demonstration examples by augmenting the input interaction sequence itself. In detail, they paired the prefix of the input interaction sequence and the corresponding successor as examples. Wang and Lim [74] investigated the effects of instruction format, task consistency, demonstration selection, and number of demonstrations [41] and [42] designed the demonstration example templates for various recommendation tasks and the experimental results also showed the in-context learning method will improve the recommendation abilities of LLMs on most tasks. In addition, a suitable demonstration can be used to control the output format and content of the LLM [75], which can improve the regular evaluation metric. This is crucial for developing a stable and robust recommender system.

However, in comparison to prompting, only a few studies have explored the use of In-context Learning of Language Models (LLMs) in recommendation tasks. Numerous open questions remain, including the selection of demonstration examples and the influence of the number of demonstration examples on recommendation performance.

4.2 Tuning paradigm

As we mentioned above, LLMs have strong zero/few-shot abilities, and their recommendation performance can significantly surpass random guessing with appropriate prompt design. However, it is not surprising that recommendation systems constructed in this manner fail to surpass the performance of recommendation models trained specifically for a given task on specific data. Therefore, many researchers aim to enhance the recommendation ability of LLMs by further fine-tuning or prompt learning. In this paper, we categorize the paradigm of the tuning methods into three different types, respectively fine-tuning, prompt tuning, and instruction tuning. Specifically, in the fine-tuning paradigm, the usage methods for discriminative and generative large language models are notably similar. The LLMs mainly serve as encoders to extract representations of users or items, and the parameters of the LLMs are subsequently fine-tuned on the specific loss functions of downstream recommendation tasks. Meanwhile, in the prompt tuning and instruction tuning paradigms, the output of the large models is consistently textual, and their parameters are trained using the loss of language modeling. The primary distinction between the prompt tuning and instruction tuning training paradigms is that prompt tuning predominantly focuses on a specific task, e.g., rating prediction, while the LLMs are trained for multiple tasks with different types of instructions under the instruction tuning paradigm. Therefore, the LLMs can get better zero-shot abilities by instruction tuning. In the subsequent sections, we will delve into representative works of these three paradigms in detail.

4.2.1 Fine-tuning

Since under the fine-tuning paradigm, the utilization and training methodologies of generative LLMs are fundamentally similar to the discriminative LLMs discussed in Section 3.1 [76], therefore, we will only introduce a few representative studies in this subsection. For example, [77] proposed GPTRec, which is a generative sequential recommendation model based on GPT-2. In contrast with BERT4Rec, which is based on discriminative LLM, GPTRec is based on generative LLM, uses SVD Tokenisation for memory efficiency, and is more flexible using the Next-K generation strategy. Kang et al. [78] proposed to format the user historical interactions as prompts, where each interaction is represented by information about the item, and formulated the rating prediction task as two different tasks, respectively multi-class classification and regression. Kang et al. [78] further investigated various LLMs in different sizes, ranging from 250M to 540B parameters and evaluate their performance in zero-shot, few-shot, and fine-tuning scenarios, and found that the FLAN-T5-XXL (11B) model with fine-tuning can achieve the best result. [32] studied the influence of LLMs, such as GPT-3 with 175-billion parameters, on text-based collaborative filtering (TCF). Liu et al. [58] proposed initially using closed-source LLMs such as ChatGPT to supplement item information in recommendation systems, obtaining closed-source tokens. subsequently, open-source LLMs like LLaMA were used for representing items and users, resulting in open-source embeddings. Finally, [58] employed the fine-tuning paradigm to train the recommendation model based on the user and item embeddings. Li et al. [32] found that using more powerful LLMs as text encoders can result in higher recommendation accuracy. However, an extremely large LM may not result in a universal representation of users and items, and the simple ID-based collaborative filtering still remains a highly competitive approach in the warm item recommendation setting.

Few projects fully train large-scale recommenders to address their business needs. Meta AI, for example, redefines recommendation problems as sequential transduction tasks within a generative modeling framework (“Generative Recommenders”) [79]. They introduced a novel architecture, HSTU, designed for high cardinality and non-stationary streaming recommendation data. HSTU outperforms baselines on synthetic and public datasets by up to 65.8% in NDCG and is 5.3x to 15.2x faster than FlashAttention2-based Transformers on 8192-length sequences.

4.2.2 Prompt tuning

In this paradigm, LLMs typically take the user/item information as input, and output the user preference (e.g., like or unlike, ratings) for the items, or output items that the user may be interested in. Prompt tuning involves designing specific prompts to guide LLMs in generating more relevant recommendations. The effectiveness of prompt designs can significantly impact performance. Generally,

- **Prompt Design**: Crafting prompts that closely resemble natural user queries can improve the relevance of recommendations. For instance, using prompts like “Recommend me a movie similar to ‘Inception’ ” helps the model generate more precise recommendations.
- **Optimization Techniques**: Techniques such as prompt ensembling, where multiple prompts are used to guide the model, have shown to enhance performance. Additionally, iterative refinement of prompts [80] based on user feedback can lead to more accurate and personalized recommendations.

For specific research, [81] proposed TALLRec which is trained by two tuning stages. TALLRec is first fine-tuned based on the self-instruct data by Alpaca [82]. Then, TALLRec is further fine-tuned by recommendation tuning, where the input is the historical sequence of users and the output is the “yes or no" feedback. Ji et al. [83] presented an LLM-based generative recommendation method named GenRec that utilized the generation ability of generative LLM to directly generate the target item to recommend. Specifically, [83] proposed to use input generation function to convert items into prompts, and use LLMs to generate the next item. Chen [84] proposed a multi-step approach to harness the potential of LLMs for recommendation. Specifically, [84] first proposed to leverage LLMs to generate a summary of a user's preferences. For example, by analyzing a user's music and TV viewing history, the LLM can generate a summary like "pop music" and "fantasy movies". Then, a retrieval module is utilized to get a much smaller candidate pool. Finally, the interaction history, natural language user profile, and retrieved candidates are utilized to construct a natural language prompt that can be fed into the LLM for recommendation. Similarly, drawing inspiration from the successful application of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) models in user modeling, [85] proposed two unique techniques for user preference summarization, respectively hierarchical summarization and recurrent summarization. Zheng et al. [85] further utilized SFT techniques to finetune the final recommendation model. Chu et al. [86] proposed to combine the user features and behavioral sequences as the input text. Chu et al. [86] also proposed to name the attributes in user features and items in the behavioral sequences as entities, and keep the entities as complete units in the input text. Jin et al. [87] proposed to generate the title of the next product of interest for the user with the help of LLMs. They fine-tune a mT5 model using a generative objective defined on their dataset. However, a simple heuristic method that takes the last product title as a result surpasses the performance of the fine-tuned language model. [88] proposed RecLLM, which contains a dialogue management module that uses an LLM to converse with the user, a ranker module that uses an LLM to match the user preferences, and a controllable LLM-based user simulator to generate synthetic conversations for tuning system modules.

Further, [89] proposed PBNR, which can describe user behaviors and news textually in the designed prompts. Specifically, the personalized prompts are created by designing input-target templates, wherein the relevant fields in the prompts are replaced with corresponding information from the raw data. To enhance the performance of LLMs on the recommendation task, PBNR incorporates the ranking loss and the language generation loss throughout the training. Li et al. [90] proposed to regard the recommendation task as a query generation and searching problem. They further utilized the LLMs to produce diverse and interpretable user interests representations, i.e., the queries. Yue et al. [91] argued that despite using instructions to prompt LLMs, the generated output does not directly provide ranking scores for candidates. In order to get the ranking scores of different items efficiently, [91] utilized the output from the LLM head, i.e., output scores over all items, as the ranking scores for candidates. Li et al. [11] focused on using large language models for Point-of-Interest (POI) recommendation tasks. The proposed framework constructed prompts to retain the heterogeneity of Location-based Social Networks (LBSN) data, avoiding the loss of contextual information and enabling an understanding of the intrinsic meaning of context. Additionally, by utilizing the concept of prompt-based trajectory similarity, it integrates historical trajectories with different users' trajectory information, thus alleviating the cold start problem and improving the prediction accuracy for trajectories of various lengths.

In addition to directly fine-tuning the LLMs, some studies also proposed to utilize prompt learning to achieve better performance. For example, [92] designed a unified conversational recommendation system named UniCRS based on knowledge-enhanced prompt learning. In this paper, the authors proposed to freeze the parameters of LLMs, and train the soft prompts for response generation and item recommendation by prompt learning. Li et al. [7] proposed to provide user-understandable explanations based on the generation ability of LLMs. The authors tried both discrete prompt learning and continuous prompt learning, and further proposed two training strategies, respectively sequential tuning and recommendation as regularization.

Another noteworthy point is that how to control the output of large language models remains an unresolved challenge. When LLMs are used to directly generate items that users might be interested in, they may produce items that are out-of-corpus. To address this difficulty, some researchers have focused on how to combine prompting tuning with grounding methods, so that the results generated by the LLMs can be precisely aligned with the items in the item database. For example, [93] pointed out that item indexing and generation grounding are two essential steps for bridging LLMs and recommendation models. Lin et al. [93] first designed a multi-facet item indexing paradigm, which contains numeric ID, item title, and item attribute. Then, [93] pointed out that out-of-corpus identifiers and over-reliance on the quality of initially generated tokens are two critical problems in the generation process. To solve these problems, [93] proposed an FM-index-based multi-facet grounding method which can solve the above two problems simultaneously. Bao et al. [94] proposed a bi-step grounding paradigm. Specifically, [94] first proposed empowering LLMs to generate meaningful tokens through prompt tuning. Then, the output of LLMs was aligned with the real-world items by calculating the L2 distance between their embeddings, and some statistical information like the popularity factor was also utilized to rewight the L2 distance.

Furthermore, some studies have focused on integrating traditional collaborative models, especially ID-based recommendation models, with LLM-based recommendation models by prompt tuning. For instance, [95, 96] proposed to expand the vocabulary of LLMs to include user and item IDs. Subsequently, the embedding of these new tokens is trained through multi-step or joint training methods with both traditional and LLM-based recommendation models. [97] proposed to use both soft and hard prompting strategies to effectively learn user/item collaborative/content token embedding via language modeling on RS-specific corpora. Liao et al. [98] proposed a hybrid item representation method, which integrates both textual tokens and behavioral tokens derived from the ID-based item embedding learned by traditional recommender models. To align the ID representation with the LLM token space, [98] designed an adapter based on a trainable linear projector. Liao et al. [98] further designed a curriculum prompt tuning, i.e., gradually shifting the learning focus from text-only prompting to hybrid prompting for better performance. Similarly, [99] extracted the ID embeddings from a pre-trained sequential recommendation model, and employed a linear projection to convert the ID embeddings into the same dimension with the LLM token space. For the prediction, [99] employed an item linear projection to replace the original prediction layer in LLMs via a weight matrix to get the ranking score of all the candidate items. LLMGR [100] proposed integrating ID and graph embeddings with LLMs to leverage the complementary strengths of LLMs in natural language understanding and GNNs in relational data processing [101, 102]. Qu et al. [103] found that existing sequence recommendation methods based on pre-trained language models have not yet fully utilized the capabilities of language models and suffer from parameter redundancy. Based on this finding, the authors suggest using behavior-tuned PLMs (behavior-tuned pre-trained language models) to initialize item embeddings, thereby enhancing the capabilities of traditional sequence recommendation models such as SASRec, improving performance without increasing additional inference costs. Additionally, Google [104] introduces Semantic IDs, which are semantically meaningful tuples of codewords used as unique identifiers for each item, instead of randomly generated atomic IDs. They demonstrate that a sequence-to-sequence model combined with hierarchical Semantic IDs provides better generalization, thereby improving the retrieval of cold-start items in recommendations.

Lastly, we propose that most of the aforementioned methods are recommendations for general tasks using large language models. However, as previously mentioned, a significant advantage of large language models is their ability to efficiently align model parameters with specific domains, and some works mainly focus on the application of LLMs in specific domains. Take online recruitment as an example, within the realm of job-resume matching, the generative recommendation model GIRL [105] pioneers the use of LLM to generate potential job descriptions (JDs), enhancing the explainability and appropriateness of recommendations. GLRec [106] introduced the meta-path prompt constructor, a novel approach that employed LLM recommenders to interpret behavior graphs. This method also incorporated a path augmentation module to mitigate prompt bias. Subsequently, an LLM-based framework was introduced to align unpaired low-quality resumes with high-quality generated ones using Generative Adversarial Networks (GANs). This alignment process refined resume representations, leading to improved recommendation outcomes [107].

4.2.3 Instruction tuning

In this paradigm, LLMs are fine-tuned for multiple tasks with different types of instructions. In this way, LLMs can better align with human intent and achieve better zero-shot ability. Generally, instruction tuning involves training LLMs with a variety of tasks using specific instructions to improve their adaptability and performance in recommendation tasks. Key aspects include:

- **Instruction Design**: Instructions should clearly define the recommendation task and expected output. For example, “Given the user’s viewing history, recommend a new movie and explain why it was chosen” helps the model understand both the task and the need for interpretability [108].
- **Optimization Strategies**: Fine-tuning LLMs on diverse instructional data improves their ability to generalize across different recommendation scenarios. This includes training on tasks like rating prediction, sequential recommendation, and review summarization, which enhances the model’s versatility and zero-shot capabilities [109].

For detailed research, [2] proposed to fine-tune a T5 model on five different types of instructions, respectively sequential recommendation, rating prediction, explanation generation, review summarization, and direct recommendation. After the multitask instruction tuning on recommendation datasets, the model can achieve the capability of zero-shot generalization to unseen personalized prompts and new items. Similarly, [110] proposed to fine-tune an M6 model on three types of tasks, respectively scoring tasks, generation tasks, and retrieval tasks. Zhang et al. [109] first designed a general instruction format from three types of key aspects, respectively preference, intention, and task form. Then, [109] manually designed 39 instruction templates and automatically generated a large amount of user-personalized instruction data for instruction tuning on a 3B FLAN-T5-XL model. The experiment results demonstrated that this approach can outperform several competitive baselines including GPT-3.5. Yin et al. [111] proposed to extract heterogeneous knowledge from the Meituan dataset, and constructed behavior text by prompt engineering. Then, [111] utilized instruction tuning to make the LLMs more effective for tasks in recommendation scenarios. Li et al. [112] proposed to distill the discrete prompt for a specific task to a set of continuous prompt vectors so as to bridge IDs and words, and [112] leveraged instruction tuning to solve three different recommendation tasks, respectively sequential recommendation, top-N recommendation, and explanation generation. Lu et al. [113] proposed that previous work has mainly focused on improving the accuracy of LLM-based recommendations without adequately addressing their instruction-following capabilities. Therefore, this paper introduces a reinforcement learning (RL) training strategy to significantly enhance the instruction-following capabilities of LLMs while performing recommendation tasks.

5 Findings

In this survey, we systematically reviewed the application paradigms and adaptation strategies of large language models in recommendation systems, especially for generative language models. We have identified their potential to improve the performance of traditional recommendation models in specific tasks. However, it is necessary to note that the overall exploration in this field is still in the early stage. Researchers may find it challenging to determine the most worthwhile problems and pain points to investigate. To address this, we have summarized the common findings presented by numerous studies on large-scale model recommendations. As shown in Figure 7, these findings highlight certain technical challenges and present potential opportunities for further advancements in the field.

5.1 Model bias

Position bias In the generative language modeling paradigm of recommendation systems, various information such as user behavior sequences and recommended candidates are input to the language model in the form of textual sequential descriptions [127], which can introduce some position biases inherent in the language model itself [128]. For example, the order of candidates affects the ranking results of LLM-based recommendation models, i.e., LLM often prioritizes the items in the top order. And the model usually cannot capture the behavior order of the sequence well. Hou et al. [26] used the random sampling-based bootstrapping to alleviate the position bias of candidates and emphasized the recently interacted items to enhance behavior order. However, these solutions are not adaptive enough to adapt to different contexts, and more robust learning strategies are needed in the future.

Popularity bias The ranking results of LLMs are influenced by the popularity levels of the candidates. Popular items, which are often extensively discussed and mentioned in the pre-training corpora of LLMs, tend to be ranked higher. This can lead to a lack of diversity in the responses and potentially marginalize less popular or minority viewpoints. Addressing this issue is challenging as it is closely tied to the composition of the pre-trained corpus.

Fairness bias Pre-trained language models have exhibited fairness issues related to sensitive attributes [129, 130], which are influenced by the training data or the demographics of the individuals involved in certain task annotations [131]. These fairness concerns can result in

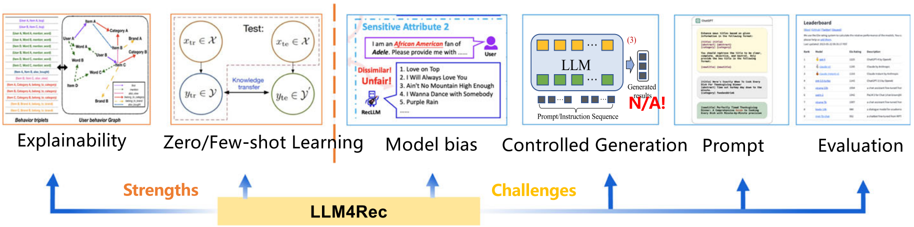


Figure 7 The major strengths and technical challenges of LLM4Rec

models making recommendations that assume users belong to a specific group, potentially leading to controversial issues when deployed commercially. One example is the bias in recommendation results caused by gender or race [37]. Addressing these fairness issues is crucial and necessary to ensure equitable and unbiased recommendations.

Personalization bias Introducing collaborative filtering signals into large language models (LLMs) for recommendation purposes presents several challenges, particularly when compared to traditional ID-based recommendation models. While LLMs have the potential to generate highly personalized content by understanding nuanced textual inputs, translating this capability into personalized recommendations is challenging. Traditional models, with their direct mapping of user-item interactions, can sometimes more straightforwardly personalize recommendations. Integrating with ID-based recommendation models or directly training and learning ID-based tokens are potential approaches in this field [32].

5.2 Recommendation prompt designing

User/Item representation In practice, recommendation systems typically utilize a large number of discrete and continuous features to represent users and items. However, most existing LLM-based work only uses the name to represent items, and a list of item names to represent users, which is insufficient for modeling users and items accurately. Additionally, it is critical to translate a user's heterogeneous behavior sequence (such as clicks, adding to cart, and purchases in the e-commerce domain) into natural language for preference modeling. ID-like features have been proven effective in traditional recommendation models, but incorporating them into prompts to improve personalized recommendation performance is also challenging.

Limited context length The context length limitation of LLMs will constrain the length of users' behavioral sequences and the number of candidate items, resulting in suboptimal performance [109]. Existing work has proposed some techniques to alleviate this problem, such as selecting representative items from user behavior sequence [56] and sliding window strategy for candidate list [48]. Recently, there have been efforts to extend the context length limitations of LLMs. For example, the LongLLaMA model [132] is a large language model capable of handling long contexts of 256k tokens or even more, which is fine-tuned using the Focused Transformer (FoT) method. However, the effectiveness of these methods in application to recommendation scenarios still merits further validation and study.

5.3 Promising ability

Zero/Few-shot recommendation ability The experimental results on multiple domain datasets indicate that LLMs possess impressive zero/few-shot abilities in various recommendation tasks [41, 42, 47]. It is worth noting that few-shot learning, which is equivalent to in-context learning, does not change the parameters of LLMs. This suggests LLMs have the potential to mitigate the cold-start problem with limited data. However, there are still some open questions, such as the need for clearer guidance in selecting representative and effective demonstration examples for few-shot learning, as well as the need for experimental results across more domains to further support the conclusion regarding the zero/few-shot recommendation abilities.

Explainable ability Generative LLMs exhibit a remarkable ability for natural language generation. Thus, a natural thought is to use LLMs to conduct explainable recommendations via a text-generation manner [133, 134]. Liu et al. [41] conduct a comparison experiment among

ChatGPT and some baselines on explanation generation task. The results demonstrate that even without fine-tuning and under the in-context learning setting, ChatGPT still performs better than some traditional supervised methods. Moreover, according to human evaluation, ChatGPT's explanations are deemed even clearer and more reasonable than the ground truth. Encouraged by these exciting preliminary explorations and experimental results, the performance of fine-tuned LLMs in explainable recommendations is expected to be promising.

5.4 Evaluation issues

Generation controlling As we mentioned before, many studies have employed large-scale models as recommendation systems by providing carefully designed instructions. For these LLMs, the output should strictly adhere to the given instruction format, such as providing binary responses (yes or no) or generating a ranked list. However, in practical applications, the output of LLMs may deviate from the desired output format. For instance, the model may produce responses in incorrect formats or even refuse to provide an answer [42]. And, generative models struggle to perform well in list-wise recommendation tasks due to their training data and autoregressive training mode, which make them less capable of handling ranking problems with multiple items. This issue cannot be resolved through fine-tuning, as there is no ground truth for ranking multiple items in a sequence in real-world scenarios. Therefore, it is difficult to apply autoregressive training logic based on sequence. PRP (Pairwise Ranking Prompting) [115] proposes pairwise ranking for listwise tasks with LLM, which enumerates all pairs and performs a global aggregation to generate a score for each item. However, this logic is time consuming in the inference process. Therefore, addressing the challenge of ensuring better control over the output of LLMs is a pressing issue that needs to be resolved.

Evaluation criteria If the tasks performed by LLMs are standard recommendations, such as rating prediction or item ranking, we can employ existing evaluation metrics for evaluation, e.g., NDCG, MSE, etc. However, LLMs also have strong generative capabilities, making them suitable for generative recommendation tasks [68]. Following the generative recommendation paradigm, LLMs can generate items that have never appeared in the historical data and recommend them to users. In this scenario, evaluating the generative recommendation capability of LLMs remains an open question.

For the evaluation of generative recommenders, given the unique nature of generative LLMs, LANE [108] delves into specific evaluation criteria such as coherence, relevance, diversity, and novelty of generated recommendation reasons. They introduce methods for qualitative assessment, including user studies and expert evaluations, to capture the subtleties of generative outputs that quantitative metrics might miss.

Datasets Current research in this area primarily evaluates the recommendation and zero/few-shot capabilities of LLMs using datasets like MovieLens and Amazon Books. However, this approach presents two potential issues. First, these datasets are relatively small compared to real-world industrial datasets and may not fully capture the recommendation capabilities of LLMs. Second, the items in these datasets, such as movies and books, might have appeared in the LLMs' pre-training data, introducing bias in evaluating their few-shot and zero-shot learning capabilities. Consequently, there is a lack of suitable benchmarks for more comprehensive evaluation.

In addition to the aforementioned prominent findings, there are also some limitations associated with the capabilities of large language models. For example, the challenge of knowledge forgetting may arise when training models for specific domain tasks or updating model knowledge [135]. Another issue is the distinct performances caused by varying sizes of language model parameters, where using excessively large models would result in excessive computational costs for research and deployment in recommendation systems [47]. These challenges also present valuable research opportunities in the field.

6 Conclusion

In this paper, we reviewed the research area of large language models (LLMs) for recommendation systems. We classified existing work into discriminative models and generative models, and then illustrated them in detail by the domain adaption manner. To prevent conceptual confusion, we provided definitions and distinctions of fine-tuning, prompting, prompt tuning, and instruction tuning in LLM-based recommendations. To the best of our knowledge, our survey is the first systematic and up-to-date review specifically dedicated to generative LLMs for recommendation systems, which further summarized the common findings and challenges presented by numerous related studies. Therefore, this survey provided researchers with a valuable resource for gaining a comprehensive understanding of LLM recommendations and exploring potential research directions.

Looking to the future, as computational capabilities continue to advance and the realm of artificial intelligence expands, we anticipate even more sophisticated applications of LLMs in recommendation systems. There is a promising horizon where the adaptability and precision of these models will be harnessed in more diverse domains, possibly leading to real-time, personalized recommendations that consider multi-modal inputs. Moreover, as ethical considerations gain prominence, future LLM-based recommendation systems might also integrate fairness, accountability, and transparency more intrinsically.

In conclusion, while we have made substantial strides in understanding and implementing LLMs in recommendation systems, the journey ahead is replete with opportunities for innovation and refinement. Our survey, we hope, will serve as a foundational stepping stone for the next wave of discoveries in this dynamic and ever-evolving field.

Author Contributions Likang Wu, Zhi Zheng, and Zhaopeng Qiu wrote the main manuscript text. Hongchao Gu and Tingjia Shen developed the GitHub project. Chuan Qin, Chen Zhu, and Qi Liu prepared the paper revision. Hao Wang, Hengshu Zhu, Hui Xiong, and Enhong Chen organized this project. All authors reviewed the manuscript.

Funding This research was partially supported by grants from National Key Research and Development Program of China (Grant No. 2021YFF0901003).

Data Availability No datasets were generated or analysed during the current study.

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Competing Interests The authors declare no competing interests.

Ethical Approval Not applicable.

References

1. Liu, P., Zhang, L., Gulla, J.A.: Pre-train, prompt and recommendation: a comprehensive survey of language modelling paradigm adaptations in recommender systems. arXiv:2302.03735 (2023)
2. Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (RLP): a unified pretrain, personalized prompt & predict paradigm (P5). In: RecSys, pp. 299–315 (2022)
3. Da'u, A., Salim, N.: Recommendation system based on deep learning methods: a systematic review and new directions. Artificial Intelligence Review. **53**(4), 2709–2748 (2020)
4. Sileo, D., Vossen, W., Raymaekers, R.: Zero-shot recommendation as language modeling. In: ECIR (2). Lecture Notes in Computer Science, vol. 13186, pp. 223–230 (2022)
5. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv:2303.18223 (2023)
6. Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., Zhang, J.: Chat-rec: towards interactive and explainable llms-augmented recommender system. arXiv:2303.14524 (2023)
7. Li, L., Zhang, Y., Chen, L.: Personalized prompt learning for explainable recommendation. ACM Transactions on Information Systems. **41**(4), 1–26 (2023)
8. Zeng, Z., Xiao, C., Yao, Y., Xie, R., Liu, Z., Lin, F., Lin, L., Sun, M.: Knowledge transfer via pre-training for recommendation: a review and prospect. Frontiers in Big Data. **4**, 602071 (2021)
9. Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Zhang, H., Liu, Y., Wu, C., Li, X., Zhu, C., Guo, H., Yu, Y., Tang, R., Zhang, W.: How Can Recommender Systems Benefit from Large Language Models: A Survey (2024)
10. Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J., Li, Q.: Recommender Systems in the Era of Large Language Models (LLMs) (2024)
11. Li, L., Zhang, Y., Liu, D., Chen, L.: Large Language Models for Generative Recommendation: A Survey and Visionary Discussions (2024)
12. Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., Pu, Y., Lei, Y., Chen, X., Wang, X., Lian, D., Chen, E.: When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities (2023)
13. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1), pp. 4171–4186 (2019)
14. Qiu, Z., Wu, X., Gao, J., Fan, W.: U-BERT: pre-training user representations for improved recommendation. In: AAAI, pp. 4320–4327 (2021)
15. Wu, C., Wu, F., Yu, Y., Qi, T., Huang, Y., Xie, X.: Userbert: Contrastive user model pre-training. arXiv:2109.01274 (2021)
16. Yang, Y., Qiao, Y., Shao, J., Yan, X., Yang, T.: Lightweight composite re-ranking for efficient keyword search with BERT. In: WSDM, pp. 1234–1244 (2022)
17. Wu, X., Magnani, A., Chaidaroon, S., Puthenputhussery, A., Liao, C., Fang, Y.: A multi-task learning framework for product ranking with BERT. In: WWW, pp. 493–501 (2022)
18. Rahmani, S., Naghshzan, A., Guerrouj, L.: Improving code example recommendations on informal documentation using bert and query-aware lsh: a comparative study. arXiv:2305.03017 (2023)
19. Zhang, S., Zheng, N., Wang, D.: GBERT: pre-training user representations for ephemeral group recommendation. In: CIKM, pp. 2631–2639 (2022)
20. Yao, S., Tan, J., Chen, X., Zhang, J., Zeng, X., Yang, K.: Reprbert: distilling BERT to an efficient representation-based relevance model for e-commerce. In: KDD, pp. 4363–4371 (2022)
21. Muhamed, A., Keivanloo, I., Perera, S., Mracek, J., Xu, Y., Cui, Q., Rajagopalan, S., Zeng, B., Chilimbi, T.: Ctr-bert: cost-effective knowledge distillation for billion-parameter teacher models. In: NeurIPS Efficient Natural Language and Speech Processing Workshop (2021)
22. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: CIKM, pp. 1441–1450 (2019)
23. Zhao, Q.: Resetbert4rec: a pre-training model integrating time and user historical behavior for sequential recommendation. In: SIGIR, pp. 1812–1816 (2022)
24. Fu, J., Yuan, F., Song, Y., Yuan, Z., Cheng, M., Cheng, S., Zhang, J., Wang, J., Pan, Y.: Exploring adapter-based transfer learning for recommender systems: empirical studies and practical insights. arXiv:2305.15036 (2023)
25. Hu, J., Xia, W., Zhang, X., Fu, C., Wu, W., Huan, Z., Li, A., Tang, Z., Zhou, J.: Enhancing sequential recommendation via llm-based semantic embedding learning. In: Companion Proceedings of the ACM on Web Conference 2024, pp. 103–111 (2024)
26. Hou, Y., Mu, S., Zhao, W.X., Li, Y., Ding, B., Wen, J.: Towards universal sequence representation learning for recommender systems. In: KDD, pp. 585–593 (2022)

27. Hou, Y., He, Z., McAuley, J., Zhao, W.X.: Learning vector-quantized item representation for transferable sequential recommenders. In: Proceedings of the ACM Web Conference 2023. WWW '23, pp. 1162–1171, New York, USA (2023)
28. Wu, C., Wu, F., Qi, T., Huang, Y.: Empowering news recommendation with pre-trained language models. In: SIGIR, pp. 1652–1656 (2021)
29. Yu, Y., Wu, F., Wu, C., Yi, J., Liu, Q.: Tiny-newsrec: effective and efficient plm-based news recommendation. In: EMNLP, pp. 5478–5489 (2022)
30. Liu, Q., Zhu, J., Dai, Q., Wu, X.: Boosting deep CTR prediction with a plug-and-play pre-trainer for news recommendation. In: COLING, pp. 2823–2833 (2022)
31. Yuan, Z., Yuan, F., Song, Y., Li, Y., Fu, J., Yang, F., Pan, Y., Ni, Y.: Where to go next for recommender systems? id-vs. modality-based recommender models revisited. arXiv:2303.13835 (2023)
32. Li, R., Deng, W., Cheng, Y., Yuan, Z., Zhang, J., Yuan, F.: Exploring the upper limits of text-based collaborative filtering using large language models: discoveries and insights. arXiv:2305.11700 (2023)
33. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: open pre-trained transformer language models. arXiv:2205.01068 (2022)
34. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3045–3059 (2021)
35. Penha, G., Hauff, C.: What does BERT know about books, movies and music? probing BERT for conversational recommendation. In: RecSys, pp. 388–397 (2020)
36. Yang, B., Han, C., Li, Y., Zuo, L., Yu, Z.: Improving conversational recommendation systems' quality with context-aware item meta-information. In: Findings of the Association for Computational Linguistics: NAACL 2022, pp. 38–48 (2022)
37. Shen, T., Li, J., Bouadjenek, M.R., Mai, Z., Sanner, S.: Towards understanding and mitigating unintended biases in language model-driven conversational recommendation. Information Processing & Management. **60**(1), 103139 (2023)
38. Zhang, Z., Wang, B.: Prompt learning for news recommendation. arXiv:2304.05263 (2023)
39. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: NeurIPS (2020)
40. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P.F., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: NeurIPS (2022)
41. Liu, J., Liu, C., Lv, R., Zhou, K., Zhang, Y.: Is chatgpt a good recommender? A preliminary study. CoRR. arXiv:2304.10149 (2023)
42. Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., Xu, J.: Uncovering chatgpt's capabilities in recommender systems. arXiv:2305.02182 (2023)
43. Lin, G., Zhang, Y.: Sparks of artificial general recommender (AGR): early experiments with chatgpt. CoRR. arXiv:2305.04518 (2023)
44. Sanner, S., Balog, K., Radlinski, F., Wedin, B., Dixon, L.: Large language models are competitive near cold-start recommenders for language-and item-based preferences. In: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 890–896 (2023)
45. Mysore, S., McCallum, A., Zamani, H.: Large language model augmented narrative driven recommendations. arXiv:2306.02250 (2023)
46. Xi, Y., Liu, W., Lin, J., Zhu, J., Chen, B., Tang, R., Zhang, W., Zhang, R., Yu, Y.: Towards open-world recommendation with knowledge augmentation from large language models. ArXiv. arXiv:2306.10933 (2023)
47. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J.J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. arXiv:2305.08845 (2023)
48. Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., Ren, Z.: Is chatgpt good at search? investigating large language models as re-ranking agent. arXiv:2304.09542 (2023)
49. Yang, Z., Wu, J., Luo, Y., Zhang, J., Yuan, Y., Zhang, A., Wang, X., He, X.: Large language model can interpret latent space of sequential recommender. arXiv:2310.20487 (2023)
50. Guo, F., Li, W., Zhuang, H., Luo, Y., Li, Y., Yan, L., Zhang, Y.: Generating diverse criteria on-the-fly to improve point-wise LLM rankers. arXiv:2404.11960 (2024)
51. He, Z., Xie, Z., Jha, R., Steck, H., Liang, D., Feng, Y., Majumder, B.P., Kallus, N., McAuley, J.J.: Large language models as zero-shot conversational recommenders. In: CIKM, pp. 720–730 (2023)

52. Acharya, A., Singh, B., Onoe, N.: LLM based generation of item-description for recommendation system. In: RecSys, pp. 1204–1207 (2023)
53. Gao, S., Fang, J., Tu, Q., Yao, Z., Chen, Z., Ren, P., Ren, Z.: Generative news recommendation. CoRR. arXiv:2403.03424 (2024)
54. Yang, S., Ma, W., Sun, P., Ai, Q., Liu, Y., Cai, M., Zhang, M.: Sequential recommendation with latent relations based on large language model. arXiv:2403.18348 (2024)
55. Lyu, H., Jiang, S., Zeng, H., Xia, Y., Luo, J.: Llm-rec: personalized recommendation via prompting large language models. arXiv:2307.15780 (2023)
56. Wang, L., Lim, E.: Zero-shot next-item recommendation using large pretrained language models. arXiv:2304.03153 (2023)
57. Wang, Y., Chu, Z., Ouyang, X., Wang, S., Hao, H., Shen, Y., Gu, J., Xue, S., Zhang, J.Y., Cui, Q., Li, L., Zhou, J., Li, S.: Enhancing recommender systems with large language model reasoning graphs. arXiv:2308.10835 (2023)
58. Liu, Q., Chen, N., Sakai, T., Wu, X.: Once: boosting content-based recommendation with both open- and closed-source large language models. arXiv:2305.06566 (2023)
59. Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., Huang, C.: Llmrec: large language models with graph augmentation for recommendation. arXiv:2311.00423 (2023)
60. Ren, X., Wei, W., Xia, L., Su, L., Cheng, S., Wang, J., Yin, D., Huang, C.: Representation learning with large language models for recommendation. arXiv:2310.15950 (2023)
61. Shu, Y., Gu, H., Zhang, P., Zhang, H., Lu, T., Li, D., Gu, N.: Rah! recsys-assistant-human: a human-central recommendation framework with large language models. arXiv:2308.09904 (2023)
62. Shi, W., He, X., Zhang, Y., Gao, C., Li, X., Zhang, J., Wang, Q., Feng, F.: Large language models are learnable planners for long-term recommendation. (2024). https://api.semanticscholar.org/CorpusID:268230856
63. Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., Xie, X.: Recommender AI agent: integrating large language models for interactive recommendations. arXiv:2308.16505 (2023)
64. Jin, J., Chen, X., Ye, F., Yang, M., Feng, Y., Zhang, W., Yu, Y., Wang, J.: Lending interaction wings to recommender systems with conversational agents. Advances in Neural Information Processing Systems. **36** (2024)
65. Huang, D., Markovitch, D.G., Stough, R.A.: Can chatbot customer service match human service agents on customer satisfaction? an investigation in the role of trust. Journal of Retailing and Consumer Services. **76**, 103600 (2024)
66. Zhang, J., Bao, K., Wang, W., Zhang, Y., Shi, W., Xu, W., Feng, F., Chua, T.: Prospect personalized recommendation on large language model-based agent platform. arXiv:2402.18240 (2024)
67. Zhang, J., Hou, Y., Xie, R., Sun, W., McAuley, J., Zhao, W.X., Lin, L., Wen, J.-R.: Agentcf: collaborative learning with autonomous language agents for recommender systems. In: Proceedings of the ACM on Web Conference 2024, pp. 3679–3689 (2024)
68. Wang, W., Lin, X., Feng, F., He, X., Chua, T.: Generative recommendation: towards next-generation recommender paradigm. arXiv:2304.03516 (2023)
69. Yoon, S., He, Z., Echterhoff, J.M., McAuley, J.J.: Evaluating large language models as generative user simulators for conversational recommendation. arXiv:2403.09738 (2024)
70. Wang, L., Zhang, J., Chen, X., Lin, Y., Song, R., Zhao, W.X., Wen, J.-R.: Recagent: a novel simulation paradigm for recommender systems. arXiv:2306.02552 (2023)
71. Zhang, A., Sheng, L., Chen, Y., Li, H., Deng, Y., Wang, X., Chua, T.: On generative agents in recommendation. arXiv:2310.10108 (2023)
72. Zhang, W., Wu, C., Li, X., Wang, Y., Dong, K., Wang, Y., Dai, X., Zhao, X., Guo, H., Tang, R.: Tired of plugins? large language models can be end-to-end recommenders. arXiv:2404.00702 (2024)
73. Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., Wei, F.: Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. arXiv:2212.10559 (2022)
74. Wang, L., Lim, E.: The whole is better than the sum: using aggregated demonstrations in in-context learning for sequential recommendation. arXiv:2403.10135 (2024)
75. Wang, X., Tang, X., Zhao, W.X., Wang, J., Wen, J.-R.: Rethinking the evaluation for conversational recommendation in the era of large language models. arXiv:2305.13112 (2023)
76. Zhang, C., Wu, S., Zhang, H., Xu, T., Gao, Y., Hu, Y., Chen, E.: Notellm: a retrievable large language model for note recommendation. In: Companion Proceedings of the ACM on Web Conference 2024, pp. 170–179 (2024)
77. Petrov, A.V., Macdonald, C.: Generative sequential recommendation with gptrec. arXiv:2306.11114 (2023)
78. Kang, W., Ni, J., Mehta, N., Sathiamoorthy, M., Hong, L., Chi, E.H., Cheng, D.Z.: Do llms understand user preferences? evaluating llms on user rating prediction. arXiv:2305.06474 (2023)

79. Zhai, J., Liao, L., Liu, X., Wang, Y., Li, R., Cao, X., Gao, L., Gong, Z., Gu, F., He, J., et al.: Actions speak louder than words: trillion-parameter sequential transducers for generative recommendations. In: Forty-first International Conference on Machine Learning
80. Krishna, S., Agarwal, C., Lakkaraju, H.: Understanding the effects of iterative prompting on truthfulness. arXiv:2402.06625 (2024)
81. Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X.: Tallrec: an effective and efficient tuning framework to align large language model with recommendation. arXiv:2305.00447 (2023)
82. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford Alpaca: An Instruction-following LLaMA model. GitHub (2023)
83. Ji, J., Li, Z., Xu, S., Hua, W., Ge, Y., Tan, J., Zhang, Y.: Genrec: large language model for generative recommendation. arXiv e-prints, 2307 (2023)
84. Chen, Z.: Palr: Personalization aware llms for recommendation. arXiv:2305.07622 (2023)
85. Zheng, Z., Chao, W., Qiu, Z., Zhu, H., Xiong, H.: Harnessing large language models for text-rich sequential recommendation. In: Proceedings of the ACM on Web Conference 2024, pp. 3207–3216 (2024)
86. Chu, Z., Hao, H., Ouyang, X., Wang, S., Wang, Y., Shen, Y., Gu, J., Cui, Q., Li, L., Xue, S., et al.: Leveraging large language models for pre-trained recommender systems. arXiv:2308.10837 (2023)
87. Jin, W., Mao, H., Li, Z., Jiang, H., Luo, C., Wen, H., Han, H., Lu, H., Wang, Z., Li, R., et al.: Amazon-m2: a multilingual multi-locale shopping session dataset for recommendation and text generation. arXiv preprint arXiv:2307.09688 (2023)
88. Friedman, L., Ahuja, S., Allen, D., Tan, T., Sidahmed, H., Long, C., Xie, J., Schubiner, G., Patel, A., Lara, H., et al.: Leveraging large language models in conversational recommender systems. arXiv:2305.07961 (2023)
89. Li, X., Zhang, Y., Malthouse, E.C.: Pbnr: prompt-based news recommender system. arXiv:2304.07862 (2023)
90. Li, J., Zhang, W., Wang, T., Xiong, G., Lu, A., Medioni, G.: Gpt4rec: a generative framework for personalized recommendation and user interests interpretation. arXiv:2304.03879 (2023)
91. Yue, Z., Rabhi, S., Moreira, G.d.S.P., Wang, D., Oldridge, E.: Llamarec: two-stage recommendation using large language models for ranking. arXiv:2311.02089 (2023)
92. Wang, X., Zhou, K., Wen, J., Zhao, W.X.: Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In: KDD, pp. 1929–1937 (2022)
93. Lin, X., Wang, W., Li, Y., Feng, F., Ng, S.-K., Chua, T.-S.: A multi-facet paradigm to bridge large language model and recommendation. arXiv:2310.06491 (2023)
94. Bao, K., Zhang, J., Wang, W., Zhang, Y., Yang, Z., Luo, Y., Feng, F., He, X., Tian, Q.: A bi-step grounding paradigm for large language models in recommendation systems. arXiv:2308.08434 (2023)
95. Zhang, Y., Feng, F., Zhang, J., Bao, K., Wang, Q., He, X.: Collm: integrating collaborative embeddings into large language models for recommendation. arXiv:2310.19488 (2023)
96. Zhang, W., Liu, H., Du, Y., Zhu, C., Song, Y., Zhu, H., Wu, Z.: Bridging the information gap between domain-specific model and general llm for personalized recommendation. arXiv:2311.03778 (2023)
97. Zhu, Y., Wu, L., Guo, Q., Hong, L., Li, J.: Collaborative large language model for recommender systems. arXiv:2311.01343 (2023)
98. Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., He, X.: Llara: aligning large language models with sequential recommenders. arXiv:2312.02445 (2023)
99. Li, X., Chen, C., Zhao, X., Zhang, Y., Xing, C.: E4srec: an elegant effective efficient extensible solution of large language models for sequential recommendation. arXiv:2312.02443 (2023)
100. Guo, N., Cheng, H., Liang, Q., Chen, L., Han, B.: Integrating large language models with graphical session-based recommendation. arXiv:2402.16539 (2024)
101. Guan, Z., Wu, L., Zhao, H., He, M., Fan, J.: Enhancing collaborative semantics of language model-driven recommendations via graph-aware learning. arXiv:2406.13235 (2024)
102. Liu, Z., Wu, L., He, M., Guan, Z., Zhao, H., Feng, N.: Dr. e bridges graphs with large language models through words. arXiv:2406.15504 (2024)
103. Qu, Z., Xie, R., Xiao, C., Sun, X., Kang, Z.: The elephant in the room: rethinking the usage of pre-trained language model in sequential recommendation. arXiv:2404.08796 (2024)
104. Rajput, S., Mehta, N., Singh, A., Hulikal Keshavan, R., Vu, T., Heldt, L., Hong, L., Tay, Y., Tran, V., Samost, J., et al.: Recommender systems with generative retrieval. Advances in Neural Information Processing Systems. **36** (2024)
105. Zheng, Z., Qiu, Z., Hu, X., Wu, L., Zhu, H., Xiong, H.: Generative job recommendations with large language model. arXiv:2307.02157 (2023)
106. Wu, L., Qiu, Z., Zheng, Z., Zhu, H., Chen, E.: Exploring large language model for graph data understanding in online job recommendations. arXiv:2307.05722 (2023)

107. Du, Y., Luo, D., Yan, R., Liu, H., Song, Y., Zhu, H., Zhang, J.: Enhancing job recommendation through llm-based generative adversarial networks. arXiv:2307.10747 (2023)
108. Zhao, H., Zheng, S., Wu, L., Yu, B., Wang, J.: Lane: Logic alignment of non-tuning large language models and online recommendation systems for explainable reason generation. arXiv:2407.02833 (2024)
109. Zhang, J., Xie, R., Hou, Y., Zhao, W.X., Lin, L., Wen, J.: Recommendation as instruction following: a large language model empowered recommendation approach. arXiv:2305.07001 (2023)
110. Cui, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: M6-rec: generative pretrained language models are open-ended recommender systems. arXiv:2205.08084 (2022)
111. Yin, B., Xie, J., Qin, Y., Ding, Z., Feng, Z., Li, X., Lin, W.: Heterogeneous knowledge fusion: a novel approach for personalized recommendation via llm. In: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 599–601 (2023)
112. Li, L., Zhang, Y., Chen, L.: Prompt distillation for efficient llm-based recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1348–1357 (2023)
113. Lu, W., Lian, J., Zhang, W., Li, G., Zhou, M., Liao, H., Xie, X.: Aligning large language models for controllable recommendations. arXiv:2403.05063 (2024)
114. Xiao, S., Liu, Z., Shao, Y., Di, T., Middha, B., Wu, F., Xie, X.: Training large-scale news recommenders with pretrained language models in the loop. In: KDD, pp. 4215–4225 (2022)
115. Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X., et al.: Large language models are effective text rankers with pairwise ranking prompting. arXiv:2306.17563 (2023)
116. Mao, Z., Wang, H., Du, Y., Wong, K.-F.: Unitrec: a unified text-to-text transformer and joint contrastive learning framework for text-based recommendation. In: Annual Meeting of the Association for Computational Linguistics (2023). https://api.semanticscholar.org/CorpusID:258888030
117. Li, X., Chen, B., Hou, L., Tang, R.: Ctrl: connect tabular and language model for ctr prediction. arXiv:2306.02841 (2023)
118. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, pp. 507–517 (2016)
119. Hou, Y., Li, J., He, Z., Yan, A., Chen, X., McAuley, J.J.: Bridging language and items for retrieval and recommendation. arXiv:2403.03952 (2024)
120. Wan, M., McAuley, J.: Item recommendation on monotonic behavior chains. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 86–94 (2018)
121. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 496–505 (2017)
122. Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al.: Mind: a large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3597–3606 (2020)
123. Liu, Y., Zhang, W., Dong, B., Fan, Y., Wang, H., Feng, F., Chen, Y., Zhuang, Z., Cui, H., Li, Y., et al.: U-need: a fine-grained dataset for user needs-centric e-commerce conversational recommendation. arXiv:2305.04774 (2023)
124. Sun, Z., Si, Z., Zang, X., Leng, D., Niu, Y., Song, Y., Zhang, X., Xu, J.: Kuaisar: A unified search and recommendation dataset. (2023) https://doi.org/10.1145/3583780.3615123
125. Yuan, G., Yuan, F., Li, Y., Kong, B., Li, S., Chen, L., Yang, M., Yu, C., Hu, B., Li, Z., et al.: Tenrec: a large-scale multipurpose benchmark dataset for recommender systems. arXiv:2210.10629 (2022)
126. Cheng, Y., Pan, Y., Zhang, J., Ni, Y., Sun, A., Yuan, F.: An image dataset for benchmarking recommender systems with raw pixels. arXiv:2309.06789 (2023)
127. Harte, J., Zorgdrager, W., Louridas, P., Katsifodimos, A., Jannach, D., Fragkoulis, M.: Leveraging large language models for sequential recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 1096–1102 (2023)
128. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 8086–8098 (2022)
129. Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F., He, X.: Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. arXiv:2305.07609 (2023)
130. Deldjoo, Y., Noia, T.D.: Cfairllm: consumer fairness evaluation in large-language model recommender system. arXiv:2403.05668 (2024)
131. Ferrara, E.: Should chatgpt be biased? challenges and risks of bias in large language models. arXiv:2304.03738 (2023)

132. Tworkowski, S., Staniszewski, K., Pacek, M., Wu, Y., Michalewski, H., Miłoś, P.: Focused transformer: contrastive training for context scaling. arXiv:2307.03170 (2023)
133. Silva, Í., Marinho, L., Said, A., Willemsen, M.C.: Leveraging chatgpt for automated human-centered explanations in recommender systems. In: Proceedings of the 29th International Conference on Intelligent User Interfaces, pp. 597–608 (2024)
134. Wang, Y., Tian, C., Hu, B., Yu, Y., Liu, Z., Zhang, Z., Zhou, J., Pang, L., Wang, X.: Can small language models be good reasoners for sequential recommendation? In: Proceedings of the ACM on Web Conference 2024, pp. 3876–3887 (2024)
135. Jang, J., Ye, S., Yang, S., Shin, J., Han, J., Kim, G., Choi, S.J., Seo, M.: Towards continual knowledge learning of language models. In: ICLR (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Likang Wu[1,2] · Zhi Zheng[1,2] · Zhaopeng Qiu[2] · Hao Wang[1] · Hongchao Gu[1] · Tingjia Shen[1] · Chuan Qin[2] · Chen Zhu[2] · Hengshu Zhu[2] · Qi Liu[1] · Hui Xiong[3] · Enhong Chen[1]

🖂 Hao Wang
wanghao3@ustc.edu.cn

🖂 Hengshu Zhu
zhuhengshu@gmail.com

🖂 Hui Xiong
xionghui@ust.hk

🖂 Enhong Chen
cheneh@ustc.edu.cn

Likang Wu
wulk@mail.ustc.edu.cn

Zhi Zheng
zhengzhi97@mail.ustc.edu.cn

Zhaopeng Qiu
zhpengqiu@gmail.com

Hongchao Gu
hcgu@mail.ustc.edu.cn

Tingjia Shen
jts_stj@mail.ustc.edu.cn

Chuan Qin
chuanqin0426@gmail.com

Chen Zhu
zc3930155@gmail.com

Qi Liu
qiliuql@ustc.edu.cn

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, JinZhai Road, Hefei 230026, Anhui, China

[2] Career Science Lab, BOSS Zhipin, Beijing 100020, China

[3] Hong Kong University of Science and Technology (Guangzhou), Guangzhou 510000, Guangdong, China