Recommender Systems in the Era of Large Language Models (LLMs)

Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang,
Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li

Abstract—With the prosperity of e-commerce and web applications, Recommender Systems (RecSys) have become an important component in our daily life, providing personalized suggestions that cater to user preferences. While Deep Neural Networks (DNNs) have made significant advancements in enhancing recommender systems by modeling user-item interactions and incorporating their textual side information, these DNN-based methods still have some limitations, such as difficulties in effectively understanding users' interests and capturing textual side information, inabilities in generalizing to various seen/unseen recommendation scenarios and reasoning on their predictions, etc. Meanwhile, the emergence of Large Language Models (LLMs), such as ChatGPT and GPT4, has revolutionized the fields of Natural Language Processing (NLP) and Artificial Intelligence (AI), due to their remarkable abilities in fundamental responsibilities of language understanding and generation, as well as impressive generalization and reasoning capabilities. As a result, recent studies have attempted to harness the power of LLMs to enhance recommender systems. Given the rapid evolution of this research direction in recommender systems, there is a pressing need for a systematic overview that summarizes existing LLM-empowered recommender systems, so as to provide researchers and practitioners in relevant fields with an in-depth understanding. Therefore, in this survey, we conduct a comprehensive review of LLM-empowered recommender systems from various aspects including Pre-training, Fine-tuning, and Prompting. More specifically, we first introduce representative methods to harness the power of LLMs (as a feature encoder) for learning representations of users and items. Then, we review recent advanced techniques of LLMs for enhancing recommender systems from three paradigms, namely pre-training, fine-tuning, and prompting. Finally, we comprehensively discuss the promising future directions in this emerging field.

Index Terms—Recommender Systems, Large Language Models (LLMs), Pre-training and Fine-tuning, In-context Learning, Prompting.

✦

---

1 INTRODUCTION

Recommender Systems (RecSys) play a vital role in alleviating information overload for enriching users' online experience (*i.e.*, users need to filter overwhelming information to locate their interested information) [1], [2]. They offer personalized suggestions towards candidate items tailored to meet user preferences in various application domains, such as entertainment [3], e-commerce [4], and job matching [2]. For example, on movie recommendations (*e.g.*, *IMDB* and *Netflix*), the latest movies are recommended to users based on the content of movies and the past interaction histories of users, which help users discover new movies that accord with their interests. The basic idea of recommender systems is to make use of the interactions between users and items and their associated side information, especially textual information (*e.g.*, item titles or descriptions, user profiles, and user reviews for items), to predict the matching score between users and items (*i.e.*, the probability that the user would like the item) [5]. More specifically, collaborative behaviors between users and items have been leveraged to design various recommendation models, which can be further used to learn the representations of users and items [6], [7]. In addition, textual side information about users and items contains rich knowledge that can assist in the calculation of the matching scores, providing great opportunities to understand user preferences for advancing recommender systems [8].

Due to the remarkable ability of representation learning in various fields, Deep Neural Networks (DNNs) have been widely adopted to advance recommender systems [9], [10]. DNNs demonstrate distinctive abilities in modeling user-item interactions with different architectures. For example, as particularly effective tools for sequential data, Recurrent Neural Networks (RNNs) have been adopted to capture high-order dependencies in user interaction sequences [11], [12]. Considering users' online behaviors (*e.g.*, chick, purchase, socializing) as graph-structured data, Graph Neural Networks (GNNs) have emerged as advanced representation learning techniques to learn user and item representations [1], [6], [13]. Meanwhile, DNNs have also demonstrated advantages in encoding side information. For instance, a BERT-based method is proposed to extract and utilize textual reviews from users [14].

Despite the aforementioned success, most existing advanced recommender systems still face some intrinsic limitations. *First*, due to the limitations on model scale and data size, previous DNN-based models (*e.g.*, CNN and LSTM) and

- W. Fan, Z. Zhao, J. Li, Y. Liu, and Q. Li are with the Department of Computing, The Hong Kong Polytechnic University. E-mail: {wenqifan03, scofield.zzh}@gmail.com, {jiatong.li, yunqing617.liu}@connect.polyu.hk, csqli@comp.polyu.edu.hk.
- X. Mei is with the Department of Management and Marketing, The Hong Kong Polytechnic University. E-mail: michael.mei@polyu.edu.hk.
- Y. Wang is with National University of Defense Technology. E-mail: yiq@nudt.edu.cn.
- Z. Wen and F. Wang are with Amazon. E-mail: {zhenwen, feiww}@amazon.com.
- X. Zhao is with City University of Hong Kong. E-mail: xy.zhao@cityu.edu.hk.
- J. Tang is with Michigan State University. E-mail: tangjili@msu.edu.

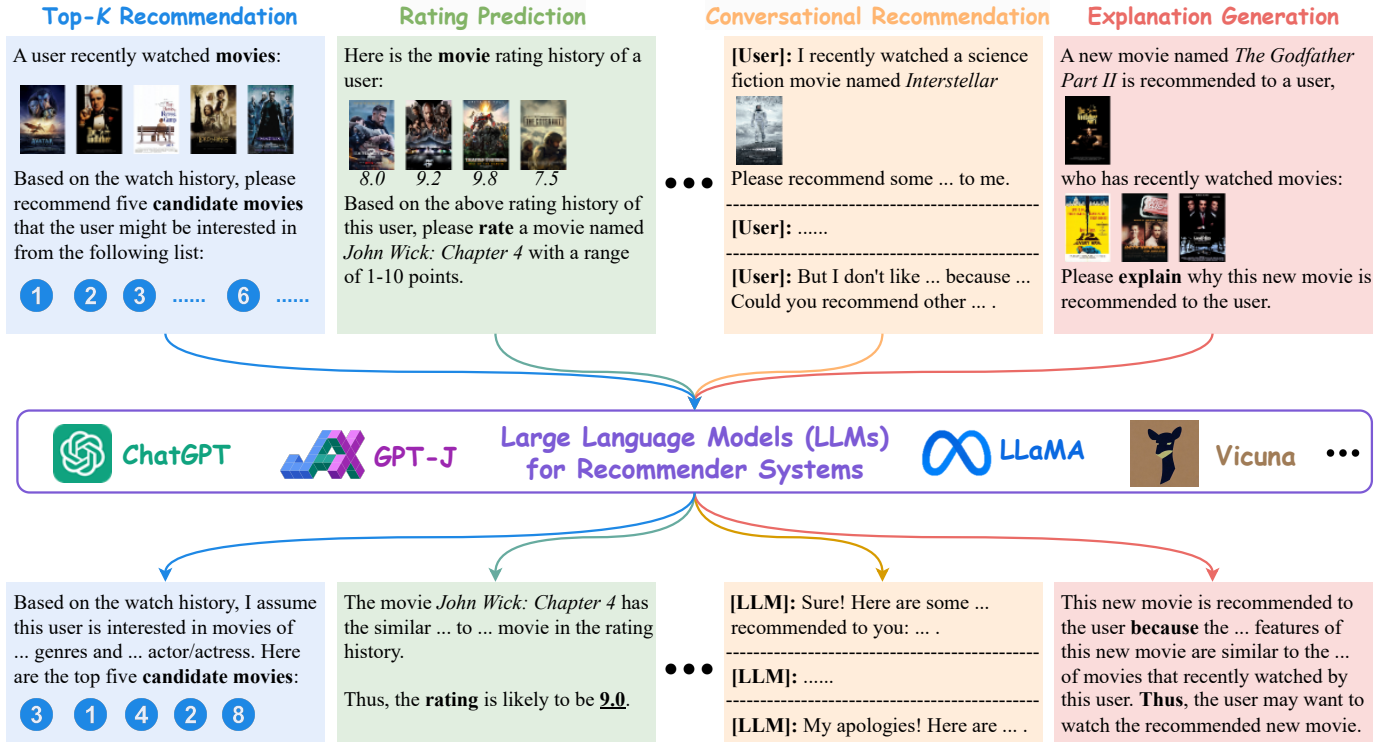*(Corresponding authors: Wenqi Fan and Qing Li.)*

Top-K Recommendation

A user recently watched **movies**:

Based on the watch history, please recommend five **candidate movies** that the user might be interested in from the following list:

① ② ③ ...... ⑥ ......

Rating Prediction

Here is the **movie** rating history of a user:

8.0   9.2   9.8   7.5

Based on the above rating history of this user, please **rate** a movie named *John Wick: Chapter 4* with a range of 1-10 points.

Conversational Recommendation

[User]: I recently watched a science fiction movie named *Interstellar*

Please recommend some ... to me.

[User]: ......

[User]: But I don't like ... because ... Could you recommend other ... .

Explanation Generation

A new movie named *The Godfather Part II* is recommended to a user,

who has recently watched movies:

Please **explain** why this new movie is recommended to the user.

ChatGPT   GPT-J   **Large Language Models (LLMs) for Recommender Systems**   LLaMA   Vicuna •••

Based on the watch history, I assume this user is interested in movies of ... genres and ... actor/actress. Here are the top five **candidate movies**:

③ ① ④ ② ⑧

The movie *John Wick: Chapter 4* has the similar ... to ... movie in the rating history.

Thus, the **rating** is likely to be **9.0**.

[LLM]: Sure! Here are some ... recommended to you: ... .

[LLM]: ......

[LLM]: My apologies! Here are ... .

This new movie is recommended to the user **because** the ... features of this new movie are similar to the ... of movies that recently watched by this user. **Thus**, the user may want to watch the recommended new movie.

Figure 1: Examples of the applications of LLMs for various recommendation tasks in the scenario of movie recommendations. LLMs can leverage textual data (or even multimodal data like images) for recommendation tasks.

pre-trained language models (*e.g.*, BERT) for recommender systems cannot sufficiently capture textual knowledge about users and items, demonstrating their inferior natural language understanding capability, which leads to sub-optimal prediction performance in various recommendation scenarios. *Second*, most existing RecSys methods have been specifically designed for their own tasks and have inadequate generalization ability to their unseen recommendation tasks. For example, a recommendation algorithm is well-trained on a user-item rating matrix for predicting movies' rating scores, while it is challenging for this algorithm to perform top-k movie recommendations along with certain explanations. This is due to the fact that the design of these recommendation architectures highly depends on task-specific data and domain knowledge toward specific recommendation scenarios such as top-k recommendations, rating predictions, and explainable recommendations. *Third*, most existing DNN-based recommendation methods can achieve promising performance on recommendation tasks needing simple decisions (*e.g.*, rating prediction, and top-k recommendations). However, they face difficulties in supporting complex and multi-step decisions that involve multiple reasoning steps. For instance, multi-step reasoning is crucial to trip planning recommendations, where RecSys should first consider popular tourist attractions based on the destination, then arrange a suitable itinerary corresponding to the tourist attractions, and finally recommend a journal plan according to specific user preferences (*e.g.*, cost and time for travel).

Recently, as advanced natural language processing techniques, Large Language Models (LLMs) with billion

parameters have generated large impacts on various research fields such as Natural Language Processing (NLP) [15], Computer Vision [16], and Molecule Discovery [17]. Technically, most existing LLMs are transformer-based models pre-trained on a vast amount of textual data from diverse sources, such as articles, books, websites, and other publicly available written materials. As the parameter size of LLMs continues to scale up with a larger training corpus, recent studies indicated that LLMs can lead to the emergence of remarkable capabilities [18], [19]. More specifically, LLMs have demonstrated the unprecedently powerful abilities of their fundamental responsibilities in language understanding and generation. These improvements enable LLMs to better comprehend human intentions and generate language responses that are more human-like in nature. Moreover, recent studies indicated that LLMs exhibit impressive generalization and reasoning capabilities, making LLMs better generalize to a variety of unseen tasks and domains. To be specific, instead of requiring extensive fine-tuning on each specific task, LLMs can apply their learned knowledge and reasoning skills to fit new tasks simply by providing appropriate instructions or a few task demonstrations. Advanced techniques such as in-context learning can further enhance such generalization performance of LLMs without being fine-tuned on specific downstream tasks [19]. In addition, empowered by prompting strategies such as chain-of-thought, LLMs can generate the outputs with step-by-step reasoning in complicated decision-making processes. Hence, given their powerful abilities, LLMs demonstrate great potential to revolutionize recommender systems.

Very recently, initial efforts have been made to explore

the potential of LLMs as a promising technique for the next-generation RecSys. For example, Chat-Rec [3] is proposed to enhance the recommendation accuracy and explainability by leveraging ChatGPT to interact with users through conversations and then refine the candidate sets generated by traditional RecSys for movie recommendations. Zhang et al. [20] employ T5 as LLM-based RecSys, which enables users to deliver their explicit preferences and intents in natural language as RecSys inputs, demonstrating better recommendation performance than merely based on user-item interactions. Figure 1 demonstrates some examples of applying LLMs for various movie recommendation tasks, including top-K recommendation, rating prediction, conversational recommendation, and explanation generation. Due to their rapid evolution, it is imperative to comprehensively review recent advances and challenges of LLMs-empowered recommender systems.

Therefore, in this survey, we provide a comprehensive overview of LLMs for recommender systems from the paradigms in terms of *pre-training*, *fine-tuning*, and *prompting*. The remaining part of this survey is organized as follows. First, we review the related works on RecSys and LLMs, and their combinations in Section 2. Then, two types of LLM-empowered RecSys that take advantage of LLMs to learn the representation of users and items are illustrated in Section 3, which are ID-based RecSys and textual side information-enhanced RecSys. Subsequently, we summarize the techniques for adopting LLMs to RecSys in terms of the pre-training & fine-tuning paradigm and the prompting paradigm in Sections 4 and 5, respectively. Finally, some challenges and potential future directions for LLM-empowered RecSys are discussed in Section 6.

Concurrent to our survey, Liu *et al.* [21] review the training strategies and learning objectives of the language modeling paradigm adaptations for recommender systems. Wu *et al.* [22] summarize the LLMs for recommender systems from discriminative and generative perspectives. Lin *et al.* [23] introduce two orthogonal perspectives: where and how to adapt LLMs in recommender systems.

2 RELATED WORK

In this section, we briefly review some related work on recommender systems and LLMs techniques.

2.1 Recommender Systems (RecSys)

To address the information overload problem, recommender systems have emerged as a crucial tool in various online applications by providing personalized content and services to individual users [24], [25]. Typically, most existing recommendation approaches can fall into two main categories: Collaborative Filtering (CF) and Content-based recommendation. As the most common technique, CF-based recommendation methods aim to find similar behavior patterns of users to predict the likelihood of future interactions [12], which can be achieved by utilizing the historical interaction behaviors between users and items, such as purchase history or rating data. For example, as one of the most popular CF methods, Matrix Factorization (MF) is introduced to learn representations of users and items by

using pure user-item interactions [7], [26]. In other words, unique identities of users and items (*i.e.*, discrete IDs) are encoded to continue embedding vectors so that the matching score can be calculated easily for recommendations [27], [28]. Content-based recommendation methods generally take advantage of additional knowledge about users or items, such as user demographics or item descriptions, to enhance user and item representations for improving recommendation performance [29]. Note that as textual information is one of the most available contents for users and items, we mainly focus on text as content in this survey.

Due to the remarkable representation learning capabilities, deep learning techniques have been effectively applied to develop recommender systems [5], [25]. For instance, NeuMF is proposed to model non-linear interactions between users and items by replacing the general inner product with DNNs [30]. Considering that data in RecSys can be naturally represented as graph-structured data, GNN techniques are treated as the main deep learning approaches for learning meaningful representations of nodes (*i.e.*, users and items) via message propagation strategies for recommender systems [1], [31]–[33]. In order to integrate textual knowledge about users and items, DeepCoNN is developed to use CNNs to encode users' reviews written for items with two parallel neural networks so as to contribute to rating predictions in recommender systems [8]. Meanwhile, a neural attention framework NARRE is introduced to simultaneously predict users' ratings towards items and generate review-level explanations for the predictions [34].

Recently, language models have been increasingly utilized in recommender systems due to their capacity to comprehend and produce human natural language. These models are designed to comprehend the semantics and syntax of human natural language, thereby enabling RecSys to provide more personalized recommendations, such as news recommendations [35], [36], and drug recommendations [37]. Specifically, a sequential recommendation method called BERT4Rec is proposed to adopt Bidirectional Encoder Representations from Transformers (*i.e.*, BERT) to model the sequential nature of user behaviors [38]. Furthermore, to take advantage of Transformer's capability for language generation, Li *et al.* [39] design a transformer-based framework to simultaneously make item recommendations and generate explanations in recommender systems.

2.2 Large Language Models (LLMs)

As a type of advanced Artificial Intelligence (AI) techniques, LLMs are trained on a large amount of textural data with billions of parameters to understand the patterns and structures of natural language. There are several classical types of pre-trained language models available, such as BERT (Bidirectional Encoder Representations from Transformers) [40], GPT (Generative Pre-trained Transformer) [41], and T5 (Text-To-Text Transfer Transformer) [42]. Typically, these language models fall into three main categories: encoder-only models, decoder-only models, and encoder-decoder models.

BERT, GPT, and T5 are distinct models based on the Transformer architecture [43]. More specifically, BERT, an encoder-only model, uses bi-directional attention to process

token sequences, considering both the left and right context of each token. It is pre-trained based on massive amounts of text data using tasks like masked language modeling and next-sentence prediction, thereby capturing the nuances of language and meaning in context. This process translates text into a vector space, facilitating nuanced and context-aware analyses. On the other hand, GPT, based on the transformer decoder architecture, uses a self-attention mechanism for one-directional word sequence processing from left to right. GPT is mainly adopted in language generation tasks, mapping embedding vectors back to text space, and generating contextually relevant responses. At last, T5, an encoder-decoder model, could handle any text-to-text task by converting every natural language processing problem into a text generation problem. For instance, it can re-frame a sentiment analysis task into a text sequence, like *'sentiment: I love this movie.'*, which adds *'sentiment:'* before *'I love this movie.'*. Then it will get the answer *'positive'*. By doing so, T5 uses the same model, objective, and training procedure for all tasks, making it a versatile tool for various NLP tasks.

Due to the increasing scale of models, LLMs have revolutionized the field of NLP by demonstrating unprecedented capabilities in understanding and generating human-like textual knowledge [18], [44]. These models (e.g., GPT-3 [15], LaMDA [45], PaLM [46], and Vicuna [47]) often based on transformer architectures, undergo training on extensive volumes of text data. This process enables them to capture complex patterns and nuances in human language. Recently, LLMs have demonstrated remarkable capabilities of ICL, a concept that is central to their design and functionality. ICL refers to the model's capacity to comprehend and provide answers based on the input context as opposed to merely relying on inside knowledge obtained through pre-training. Several works have explored the utilization of ICL in various tasks, such as SG-ICL [48] and EPR [49]. These works show that ICL allows LLMs to adapt their responses based on input context instead of generating generic responses. Another technique that can enhance the reasoning abilities of LLMs is chain-of-thought (CoT). This method involves supplying multiple demonstrations to describe the chain of thought as examples within the prompt, guiding the model's reasoning process [50]. An extension of the CoT is the concept of self-consistency, which operates by implementing a majority voting mechanism on answers [51]. Current researches continue to delve into the application of CoT in LLMs, such as STaR [52], THOR [53], and Tab-CoT [54]. By offering a set of prompts to direct the model's thought process, CoT enables the model to reason more effectively and deliver more accurate responses.

With the powerful abilities mentioned above, LLMs have shown remarkable potential in various fields, such as chemistry [17], education [55], and finance [56]. These models, such as ChatGPT, have also been instrumental in enhancing the functionality and user experience of RecSys. One of the key applications of LLMs in RecSys is the prediction of user ratings for items. This is achieved by analyzing historical user interactions and preferences, which in turn enhances the accuracy of the recommendations [57], [58]. LLMs have also been employed in sequential recommendations, which analyze the sequence of user interactions to predict their next preference, such as TALLRec [59], M6-Rec [60], PALR [61],

and P5 [62]. Moreover, LLMs, particularly ChatGPT, have been utilized to generate explainable recommendations. One such example is Chat-Rec [3], which leverages ChatGPT to provide clear and comprehensible reasoning behind its suggestions, thereby fostering trust and user engagement. Furthermore, the interactive and conversational capabilities of LLMs have been harnessed to create a more dynamic recommendation experience. For instance, UniCRS [63] develops a knowledge-enhanced prompt learning framework to fulfill both conversation and recommendation subtasks based on a pre-trained language model. UniMIND [64] proposes a unified multi-task learning framework by using prompt-based learning strategies in conversational recommender systems. Furthermore, it is worth noting that to investigate the potential of LLMs in learning on graphs, Chen *et al.* [18] introduce two possible pipelines: *LLMs-as-Enhancers* (*e.g.*, LLMs enhance the textual information of node attributes) and *LLMs-as-Predictors* (*e.g.*, LLMs serve as independent predictor in graph learning like link prediction problems), which provide guidance on the design of LLMs for graph-based recommendations.

3 DEEP REPRESENTATION LEARNING FOR LLM-BASED RECOMMENDER SYSTEMS

Users and items are atomic units of recommender systems. To denote items and users in recommender systems, the straightforward method assigns each item or user a unique index (*i.e.*, discrete IDs). To capture users' preferences towards items, ID-based recommender systems are proposed to learn representations of users and items from user-item interactions. In addition, since textual side information about users and items provides rich knowledge to understand users' interests, textual side information-enhanced recommendation methods are developed to enhance user and item representation learning in an end-to-end training manner for recommender systems. In this section, we will introduce these two categories that take advantage of language models in recommender systems. These two kinds of recommender systems are illustrated in Figure 2.

3.1 ID-based Recommender Systems

Recommender systems are commonly used to affect users' behaviors for making decisions from a range of candidate items. These user behaviors (*e.g.*, click, like, and subscription) are generally represented as user-item interactions, where users and items are denoted as discrete IDs. Modern recommendation approaches are proposed to model these behaviors by learning embedding vectors of each ID representation. Generally, in LLM-based recommendation systems, an item or a user can be represented by a short phrase in the format of "$[prefix]\_[ID]$", where the prefix denotes its type (*i.e.*, item or user) and the ID number helps identify its uniqueness.

As the early exploration of LLM-based methods, a unified paradigm called P5 is proposed to facilitate the transfer of various recommendation data formats [62], such as user-item interactions, user profiles, item descriptions, and user reviews, into natural language sequences by mapping users and items into indexes. Note that the pre-trained T5 backbone
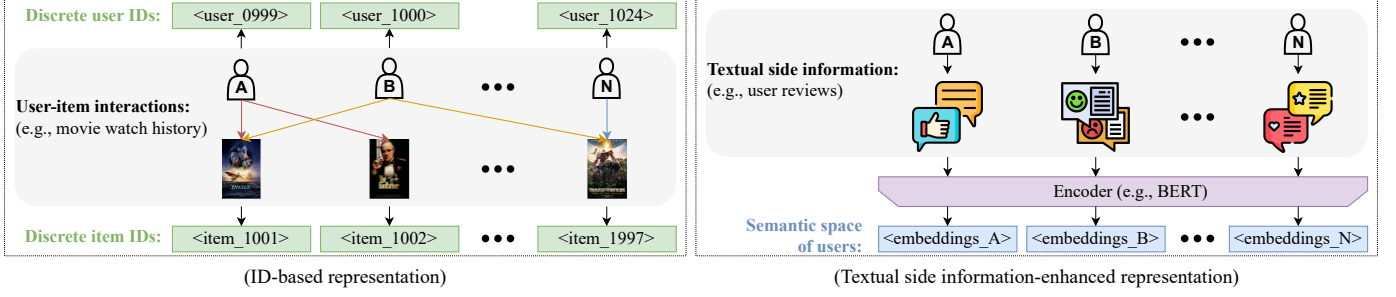
Figure 2: An illustration of two methods for representing users and items for LLM-based RecSys: *ID-based representation* (left) which denotes user-item interactions with discrete identities, and *Textual side information-enhanced representation* (right) which leverages textual side information of users and items, including user profiles, user reviews for items, item titles or descriptions.

is used to train the P5 with personalized prompts. Meanwhile, P5 incorporates the normal index phrase with a pair of angle brackets to treat these indexes as special tokens in the vocabulary of LLMs (e.g., $< item\_6637 >$), avoiding tokenizing the phrases into separate tokens. Based on P5, Hua et al. put forward four straightforward but effective indexing solutions [65]: sequential indexing, collaborative indexing, semantic (content-based) indexing, and hybrid indexing, underscoring the significance of indexing methods. Different from P5's randomly assigning numerical IDs to each user or item, Semantic IDs, a tuple of codewords with semantic meanings for each user or item, is proposed to serve as unique identifiers, each carrying semantic meaning for a particular user or item [66]. Meanwhile, to generate these codewords, a hierarchical method called RQ-VAE is also proposed [66] to leverage Semantic IDs, where recommendation data formats can be effectively transformed into natural language sequences for transformer-based models.

3.2 Textual Side Information-enhanced Recommender Systems

Despite the aforementioned success, ID-based methods suffer from intrinsic limitations. That is due to the fact that pure ID indexing of users and items is naturally discrete, which cannot provide sufficient semantic information to capture representations of users and items for recommendations. As a result, it is very challenging to perform relevance calculations based on index representations among users and items, especially when user-item interactions are severely sparse. Meanwhile, ID indexing usually requires modifying the vocabularies and altering the parameters of LLMs, which brings additional computation costs.

To address these limitations, a promising alternative solution is to leverage textual side information of users and items, which includes user profiles, user reviews for items, and item titles or descriptions. Specifically, given the textual side information of an item or a user, language models like BERT can serve as the text encoder to map the item or user into the semantic space, where we can group similar items or users and figure out their differences in a more fine-grained granularity. For instance, Li *et al.* have investigated the performance comparison between ID and modality-based recommender systems, showing that ID-based recommender

systems might be challenged by recommender systems that can better utilize side information [67]. Meanwhile, Unisec [68] is one such approach that takes advantage of item descriptions to learn transferable representations from various recommendation scenarios. More specifically, Unisec also introduces a lightweight item encoder to encode universal item representations by using parametric whitening and a mixture-of-experts (MoE) enhanced adaptor. Besides, text-based collaborative filtering (TCF) is also explored by prompting LLMs like GPT-3 [69]. Compared to the previous ID-based collaborative filtering, TCF methods demonstrate positive performance, proving the potential of textual side information-enhanced recommender systems.

However, solely relying on language models to encode item descriptions might excessively emphasize text features. To mitigate this issue, VQ-Rec [70] proposes to learn vector-quantized item representations, which can map item text into a vector of discrete indices (*i.e.*, item codes) and use them to retrieve item representations from a code embedding table in recommendations. Beyond text features, Fan *et al.* [71] propose a novel method for the Zero-Shot Item-based Recommendation (ZSIR), focusing on introducing a Product Knowledge Graph (PKG) to LLMs to refine item features. More specifically, user and item embeddings are learned via multiple pre-training tasks upon the PKG. Moreover, ShopperBERT [72] investigates modeling user behaviors to denote user representations in e-commerce recommender systems, which pre-trains user embedding through several pre-training tasks based on user purchase history. Furthermore, IDA-SR [72], an ID-Agnostic User Behavior Pre-training framework for Sequential Recommendation, directly retains representations from text information using pre-trained language models like BERT. Specifically, given an item i and its description with m tokens $D_i = \{t_1, t_2, ..., t_m\}$, an extra start-of-sequence token $[CLS]$ is added to the description $D_i = \{[CLS], t_1, t_2, ..., t_m\}$. Then, the description is fed as the input to LLMs. Finally, the embedding of the token $[CLS]$ could be used as the ID-agnostic item representation.

4 PRE-TRAINING & FINE-TUNING LLMS FOR REC-OMMENDER SYSTEMS

In general, there are three key manners in developing and deploying LLMs in recommendation tasks, namely,
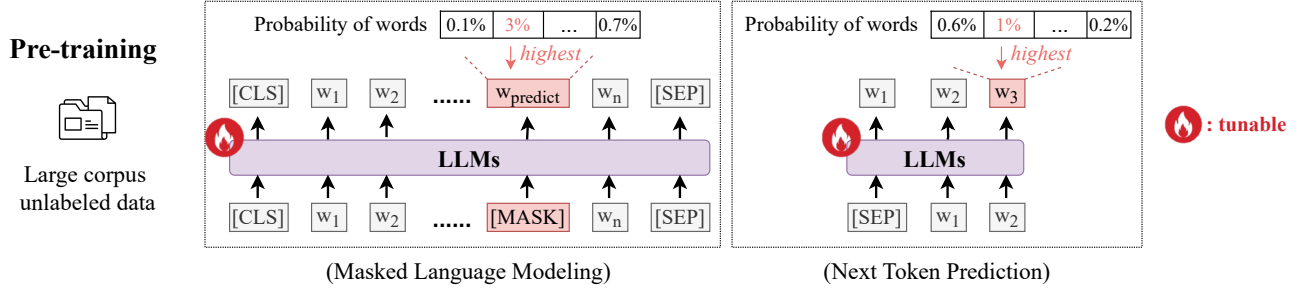
Figure 3: An illustration of two main pre-training methods of LLMs: *Masked Language Modeling* (left) which randomly masks tokens or spans in the sequence and requires LLMs to generate the masked tokens or spans based on the remaining context, and *Next Token Prediction* (right) which requires prediction for the next token based on the given context. In pre-training, LLMs are trained on a vast amount of corpus consisting of diverse and unlabeled data.

Table 1: Pre-training methods for LLM-empowered RecSys.

Paradigms	Methods	Pre-training Tasks	Code Availability
Pre-training	PTUM [73]	Masked Behavior Prediction	https://github.com/wuch15/PTUM
		Next K Behavior Prediction	
	M6 [60]	Auto-regressive Generation	Not available
	P5 [62]	Multi-task Modeling	https://github.com/jeykigung/P5

pre-training, fine-tuning, and *prompting*. In this section, we first introduce the pre-training and fine-tuning paradigms, which are shown in Figure 3 and Figure 4, respectively. More specifically, we will focus on the specific pre-training tasks applied in LLMs for recommender systems and fine-tuning strategies for better performance in downstream recommendation tasks. Note that the works mentioned below are summarized in Table 1 and Table 2.

4.1 Pre-training Paradigm for Recommender Systems

Pre-training is an important step in developing LLMs. It involves training LLMs on a vast amount of corpus consisting of diverse and unlabeled data. This strategy enables LLMs to acquire a broad understanding of various linguistic aspects, including grammar, syntax, semantics, and even common sense reasoning. Through pre-training, LLMs can learn to recognize and generate coherent and contextually appropriate responses. In general, there are two main methods to pre-train LLMs in the natural language domain, depending on the adopted model structure. One is *Masked Language Modeling* (MLM) for encoder-only or encoder-decoder Transformer structures, which randomly masks tokens or spans in the sequence and requires LLMs to generate the masked tokens or spans based on the remaining context [82]. The other is *Next Token Prediction* (NTP) for decoder-only Transformer structures, which requires prediction for the next token based on the given context [41].

In the context of recommender systems, most of the existing works follow the two classical pre-training strategies. Next, we will introduce representative methods. PTUM [73] proposes two similar pre-training tasks, Masked Behavior Prediction (MBP) and Next K behavior Prediction (NBP), to model user behaviors in recommender systems. Unlike language tokens, user behaviors are more diverse and thus more difficult to be predicted. In this case, instead of masking a span of tokens, PTUM only masks a single user behavior

with the goal of predicting the masked behavior based on the other behaviors in the interaction sequence of the target user. On the other side, NBP models the relevance between past and future behaviors, which is crucial for user modeling. The goal of NBP is to predict the next k behaviors based on the user-item interaction history.

M6 [60] also adopts two pre-training objectives motivated by the two classical pre-training tasks, namely a text-infilling objective and an auto-regressive language generation objective, corresponding to the above two pre-training tasks, respectively. To be more specific, the text-infilling objective exhibits the pre-training task of BART [83], which randomly masks a span with several tokens in the text sequence and predicts these masked spans as the pre-training target, providing the capability to assess the plausibility of a text or an event in the recommendation scoring tasks. Meanwhile, the auto-regressive language generation objective follows the Next Token Prediction task in natural language pre-training, but it is slightly different as it predicts the unmasked sentence based on the masked sequence.

Additionally, P5 adopts multi-mask modeling and mixes datasets of various recommendation tasks for pre-training. In this case, it can be generalized to various recommendation tasks and even unseen tasks with zero-shot generation ability [62]. Across different recommendation tasks, P5 applies a unified indexing method for representing users and items in language sequence as stated in Section 3 so that the Masked Language Modelling task could be employed.

4.2 Fine-tuning Paradigm for Recommender Systems

Fine-tuning is a crucial step in deploying pre-trained LLMs for specific downstream tasks. Especially for recommendation tasks, LLMs require fine-tuning to grasp more domain knowledge. Particularly, fine-tuning paradigm involves training the pre-trained model based on task-specific recommendation datasets that include user-item interaction behaviors (*e.g.*, purchase, click, ratings) and side knowledge
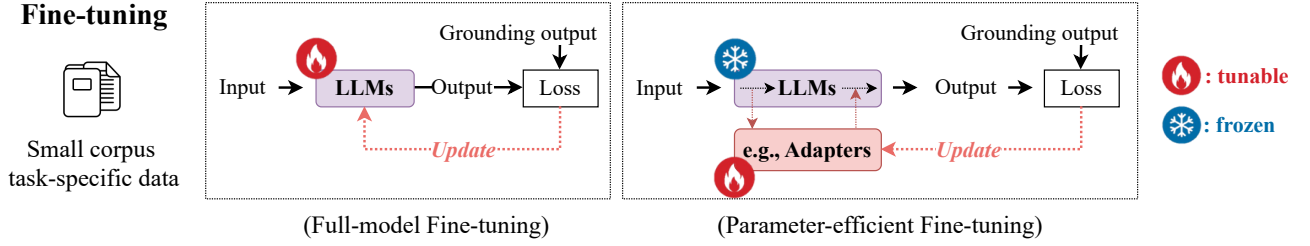
Figure 4: An illustration of two main fine-tuning methods of LLMs: Full-model Fine-tuning (left) which involves changing the entire model weights, and Parameter-efficient Fine-tuning (right) which involves fine-tuning a small proportion of model weights or a few extra trainable weights while fixing most of the parameters in LLMs. In fine-tuning, LLMs are trained on a relatively small amount of corpus (*i.e.,* compared to the amount of corpus for pre-training) of task-specific data.

Table 2: Fine-tuning methods applied in LLM-empowered RecSys.

Paradigms	Methods	References
Fine-tuning	Full-model Fine-tuning	[74], [75], [76], [77], [78], [79], and [80][1]
	Parameter-efficient Fine-tuning	[59][2], [81], and [60]

Code Availability: [1]https://github.com/veason-silverbullet/unitrec, [2]https://github.com/sai990323/tallrec

about users and items (*e.g.,* users' social relations and items' descriptions). This process allows the model to specialize its knowledge and parameters to improve performance in the recommendation domain. In general, fine-tuning strategies can be divided into two categories according to the proportion of model weights changed to fit the given task. One is *full-model fine-tuning*, which changes the entire model weights in the fine-tuning process. By considering the computation cost, the other is *parameter-efficient fine-tuning*, which aims to change only a small part of weights or develop trainable adapters to fit specific tasks.

4.2.1 Full-model Fine-tuning

As a straightforward strategy in deploying pre-trained LLMs to fit specific downstream recommendation tasks, full-model fine-tuning involves changing the entire model weights. For example, RecLLM [74] is proposed to fine-tune LaMDA as a Conversational Recommender System (CRS) for YouTube video recommendation. Meanwhile, GIRL [78] leverages a supervised fine-tuning strategy for instructing LLMs in job recommendation. However, directly fine-tuning LLMs might bring unintended bias into recommender systems, producing serious harm towards specific groups or individuals based on sensitive attributes such as gender, race and occupation. To mitigate such harmful effects, a simple LLMs-driven recommendation (LMRec) [75] is developed to alleviate the observed biases through train-side masking and test-side neutralization of non-preferential entities, which achieves satisfying results without significant performance drops. TransRec [76] studies pre-trained recommender systems in an end-to-end manner, by directly learning from the raw features of the mixture-of-modality items (*i.e.,* texts and images). In this case, without relying on overlapped users or items, TransRec can be effectively transferred to different scenarios. Additionally, Carranza *et al.* [77] propose privacy-preserving large-scale recommender systems by applying differentially private (DP) LLMs, which relieves certain challenges and limitations in DP training.

Contrastive learning has also emerged as a popular approach for fine-tuning LLMs in recommender systems. Several methods have been proposed in this direction. SBERT [79] introduces a triple loss function, where an intent sentence is paired with an anchor, and corresponding products are used as positive and negative examples in the e-commerce domain. Additionally, UniTRec [80] proposes a unified framework that combines discriminative matching scores and candidate text perplexity as contrastive objectives to improve text-based recommendations.

4.2.2 Parameter-efficient Fine-tuning

Full-model fine-tuning requires large computational resources as the size of LLMs scales up. Currently, it is infeasible for a single consumption-level GPU to fine-tune the most advanced LLMs, which usually have more than 10 billion parameters. In this case, Parameter-efficient Fine-tuning (PEFT) targets fine-tuning LLMs efficiently with lower requirements for computational resources. PEFT involves fine-tuning a small proportion of model weights or a few extra trainable weights while fixing most of the parameters in LLMs to achieve comparable performance with full-model fine-tuning.

Currently, the most popular PEFT methods lie in introducing extra trainable weights as adapters. The adapter structure is designed for embedding into the transformer structure of LLMs [84]. For each Transformer layer, the adapter module is added twice: the first module is added after the projection following the multi-head attention, and the other is added after the two feed-forward layers. During fine-tuning, the original weights of pre-trained LLMs are fixed, while the adapters and layer normalization layers are fine-tuned to fit downstream tasks. Thus, adapters contribute to the expansion and generalization of LLMs, relieving the problem of full-model fine-tuning and catastrophic forgetting. Inspired by the idea of adapters and low intrinsic ranks of weight matrices in LLMs, Low-Rank Adaptation of LLMs (LoRA) [85] introduces low-rank decomposition to simulate

the change of parameters. Basically, LoRA adds a new pathway to specific modules handling matrix multiplication in the original structure of the LLMs. In the pathway, two serial matrices first reduce the dimension to a pre-defined dimension of the middle layer and then increase the dimension back. In this case, the dimension of the middle layer could simulate the intrinsic rank.

In recommender systems, PEFT can greatly reduce the computational cost of fine-tuning LLMs for recommendation tasks, which requires less update and maintains most of the model capabilities. TallRec [59] introduces an efficient and effective tuning framework on the LLaMA-7B model and LoRA for aligning LLMs with recommendation tasks, which can be executed on a single RTX 3090. GLRec [81] also takes the adavantage of LoRA for fine-tuning and adapting LLMs as job recommender. Moreover, M6 [60] also applies LoRA fine-tuning, making it feasible to deploy LLMs in phone devices.

5 PROMPTING LLMs FOR RECOMMENDER SYSTEMS

Apart from the pre-training & fine-tuning paradigm, prompting serves as the latest paradigm for adapting LLMs to specific downstream tasks with the help of task-specific prompts. A prompt refers to a text template that can be applied to the input of LLMs. For example, a prompt *"The relation between _ and _ is _."* can be designed to deploy LLMs for relation extraction tasks. Prompting enables LLMs to unify different downstream tasks into language generation tasks, which are aligned to their objectives during pre-training [86].

To facilitate the performance of LLMs for RecSys, prompting techniques like In-context Learning (ICL) and Chain-of-Thought (CoT) are increasingly investigated to manually design prompts for various recommendation tasks. In addition, prompt tuning serves as an additive technique of prompting, by adding prompt tokens to LLMs and then updating them based on task-specific recommendation datasets. More recently, instruction tuning that combines the pre-training & fine-tuning paradigm with prompting [87] is explored to fine-tune LLMs over multiple recommendation tasks with instruction-based prompts, which enhances the *zero-shot* performance of LLMs on unseen recommendation tasks. Figure 5 compares the representative methods corresponding to each of the aforementioned three prompting techniques of LLMs, in terms of the input formation and parameter update of LLMs (*i.e.*, either tunable or frozen). In this section, we will discuss the prompting, prompt tuning, and instruction tuning techniques in detail, for improving the performance of LLMs on recommendation tasks. In summary, Table 3 categorizes the existing works according to the aforementioned three techniques, including the specific recommendation tasks and the LLM backbones considered in these works.

5.1 Prompting

The key idea of prompting is to keep LLMs frozen (*i.e.*, no parameters updates), and adapt LLMs to downstream tasks via task-specific prompts. To recap the development

of prompting strategies for adapting LLMs to downstream tasks, early-stage conventional prompting methods mainly target at unifying downstream tasks to language generation manners, such as text summarization, relation extraction, and sentiment analysis. Later on, ICL [15] emerges as a powerful prompting strategy that allows LLMs to learn new tasks (*i.e.*, tasks with knowledge demanding objectives) based on contextual information. In addition, another up-to-date prompting strategy named CoT [50] serves as a particularly effective method for prompting LLMs to address downstream tasks with complex reasoning.

5.1.1 Conventional Prompting

There are two major approaches for prompting pre-trained language models to improve the performance on specific downstream tasks. One approach is *prompt engineering*, which generates prompt by emulating text that language models encountered during pre-training (*e.g.*, text in NLP tasks). This allows pre-trained language models to unify downstream tasks with unseen objectives into language generation tasks with known objectives. For instance, Liu *et al.* [39] consider prompting ChatGPT to format the review summary task in recommendations into text summarization, with a prompt including *"Write a short sentence to summarize _."*. Another approach is *few-shot prompting*, where a few input-output examples (*i.e.*, shots) are provided to prompt and guide pre-trained language models to generate desired output for specific downstream tasks.

Due to the huge gap between language generation tasks (*i.e.*, the pre-training objectives of LLMs) and downstream recommendation tasks, these conventional prompting methods have only shown limited applications in specific recommendation tasks that have similar nature to language generation tasks, such as the review summary of users [39] and the relation labeling between items [4].

5.1.2 In-Context Learning (ICL)

Alongside the introduction of GPT-3 [15], ICL is proposed as an advanced prompting strategy, which significantly boosts the performance of LLMs on adapting to many downstream tasks. Gao *et al.* [86] attribute the success of ICL in prompting LLMs for downstream tasks to two designs: prompt and in-context demonstrations. In other words, the key innovation of ICL is to elicit the in-context ability of LLMs for learning (new or unseen) downstream tasks from context during the inference stage. In particular, two settings proposed in ICL are prevalently leveraged for prompting LLMs for RecSys. One is the few-shot setting, in which a few demonstrations with contexts and desired completions of the specific downstream tasks are provided along with prompts. The other is the zero-shot setting, where no demonstrations will be given to LLMs but only natural language descriptions of the specific downstream tasks are appended to the prompt. As shown in Figure 6, two brief templates of few-shot ICL and zero-shot ICL for recommendation tasks are provided, respectively.

- **Prompting LLMs for RecSys via Few-shot ICL.** A straightforward approach for prompting LLMs to downstream recommendation tasks is to teach LLMs how to act as RecSys. For instance, Liu *et al.* [39] employ
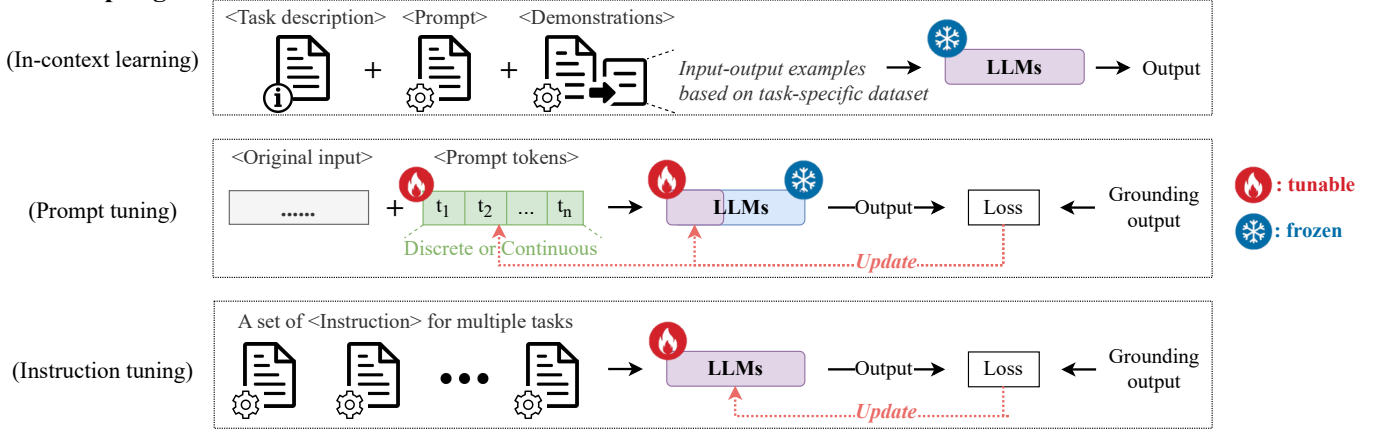
Prompting

Figure 5: An illustration of three representative methods of prompting LLMs: *in-context learning* (top) which requires no parameter update of LLMs, *prompt tuning* (middle) which adds new prompt tokens to LLMs and optimizes the prompt along with minimal parameter updates at the input layer of LLMs, and *instruction tuning* (bottom) which fine-tunes LLMs over multiple tasks-specific prompts, also known as instructions.

ChatGPT and propose separate task descriptions tailored to different recommendation tasks, including top-K recommendation, rating prediction, and explanation generation, to perform few-shot ICL based on corresponding input-output examples of each recommendation task. For instance, the user rating history is given as an example for rating prediction tasks. Similarly, other existing works propose their distinct insights into designing the in-context demonstrations for better recommendation performance. For example, a text description of role injection, such as *"You are a book rating expert."*, is proposed in [58] to augment the in-context demonstrations, which prevents LLMs from refusing to complete the recommendation tasks (*e.g.*, LLMs sometimes respond with *"As a language model, I don't have the ability to recommend ..."* for recommendation tasks). Apart from teaching LLMs to directly act as RecSys, few-shot ICL is also leveraged to guide LLMs to call traditional RecSys or external domain tools for recommendations. For example, a framework named Chat-Rec [3] is proposed to bridge ChatGPT and traditional RecSys via few-shot ICL, where ChatGPT learns to receive candidate items from traditional RecSys and then refines the final recommendation results. What's more, Zhang [101] designs a textual API call template for external graph reasoning tools and successfully teaches ChatGPT to use those templates through few-shot ICL to access the graph-based recommendation results generated by the external tools.

- **Prompting LLMs for RecSys via Zero-shot ICL.** Many existing works consider both few-shot ICL and zero-shot ICL settings at the same time to compare their performance under the same recommendation tasks. Typically, few-shot ICL can outperform zero-shot ICL since additional in-context demonstrations are provided to LLMs. Despite the reduction in performance, zero-shot ICL entirely relieves the requirement of task-specific recommendation datasets to form in-context
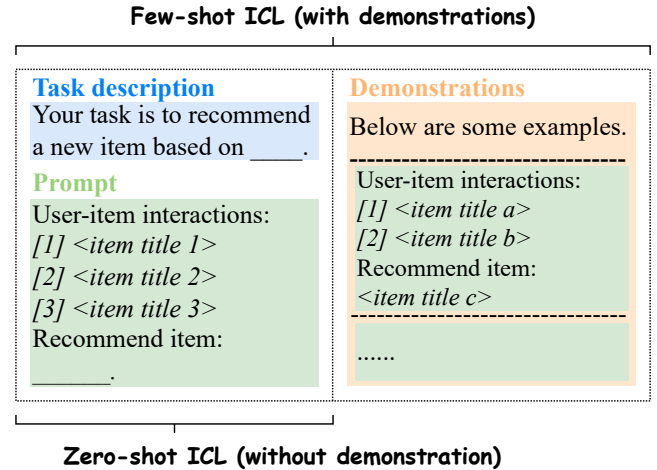

Figure 6: Brief templates of few-shot ICL and zero-shot ICL for recommendation tasks.

demonstrations and can be suitable for certain tasks like conversational recommendations, where users are not likely to provide any demonstration to LLMs. For example, Wang *et al.* [92] prompt ChatGPT for conversational recommendations with a zero-shot ICL template containing two parts: a text description of conversational recommendation tasks (*e.g.*, *"Recommend items based on user queries in the dialogue."*), and a format guideline in natural languages, such as *"The output format should be ⟨no.⟩ ⟨item title⟩."*, making the recommendation results easier to parse.

5.1.3 Chain-of-Thought (CoT) Prompting

Although ICL has shown great effectiveness in prompting LLMs for downstream tasks with in-context demonstrations, recent studies indicate that LLMs still have limited performance in reasoning-heavy tasks [50]. More specifically, by prompting LLMs with in-context examples of input-output

Table 3: An organization of representative methods of prompting LLMs for RecSys in terms of three paradigms: prompting, prompt tuning, and instruction tuning. We subsequently categorize existing works corresponding to each paradigm, including the specific recommendation tasks and the LLM backbones considered in these works.

Paradigms	Methods	Recommendation Tasks	LLM Backbones	References
Prompting	Conventional Prompting	Review Summary	ChatGPT	[39]
		Relation Labeling	ChatGPT	[4]
	In-Context Learning (ICL)	Top-K Recommendation	ChatGPT	[3], [39], [88][1], [89][2], [90][3], [91][4], [92][5], [93][6], [94]
			GPT-3	[95][7], [96]
			T5	[97], [98][8]
			PaLM	[99], [100]
		Rating Prediction	ChatGPT	[3], [58], [88][1], [101][9]
			ChatGLM	[102][10]
		Conversational Recommendation	ChatGPT	[3], [92][5], [93][6], [103]
		Explanation Generation	ChatGPT	[3], [39]
	Chain-of-Thought (CoT)	Top-K Recommendation	T5	[20]
Prompt Tuning	Hard Prompt Tuning	(Refer to ICL above, see Section 5.2.1 for explanations)		
	Soft Prompt Tuning	Top-K Recommendation	T5	[104]
			PaLM	[99]
			M6	[60]
Instruction Tuning	Full-model Tuning with Prompt	Top-K Recommendation	T5	[20]
			LLaMA	[61], [78]
		Rating Prediction	T5	[57]
	Parameter-efficient Model Tuning with Prompt	Top-K Recommendation	LLaMA	[105][11]
		Rating Prediction	LLaMA	[59][12], [81]

Code Availability: [1]https://github.com/rainym00d/LLM4RS, [2]https://github.com/jizhi-zhang/FaiRLLM,
[3]https://github.com/Jyonn/GENRE-requests, [4]https://github.com/RUCAIBox/LLMRank,
[5]https://github.com/RUCAIBox/iEvaLM-CRS, [6]https://github.com/Linxyhaha/GeneRec,
[7]https://github.com/AGI-Edgerunners/LLM-Next-Item-Rec, [8]https://github.com/JacksonWuxs/PromptRec,
[9]https://github.com/jwzhanggy/Graph_Toolformer,
[10]will be available at https://gitee.com/mindspore/models/tree/master/research/recommend/KAR,
[11]https://github.com/rutgerswiselab/GenRec, [12]https://anonymous.4open.science/r/LLM4Rec-Recsys.

pairs, the answers directly generated by LLMs often suffer from missing one or a few intermediate reasoning steps in multi-step problems like mathematical equations, leading to a broken reasoning logic that causes errors in the subsequent reasoning steps (*i.e.*, "one-step missing errors" [50]). Similar multi-step problems also exist in RecSys, such as the multi-step reasoning of user preferences based on the multi-turn dialogues in conversational recommendations. To address such limitations, CoT offers a special prompting strategy to enhance the reasoning ability of LLMs, by annotating intermediate reasoning steps to prompt. This enables LLMs to break down complicated decision-making processes and generate the final output with step-by-step reasoning.

Considering the suitable prompting strategies for adapting LLMs to various downstream tasks with complex reasoning, Zhao *et al.* [19] discuss the combination of ICL and CoT prompting under two major settings: Few-shot CoT and Zero-shot CoT, as illustrated below.

- ***Zero-shot CoT.*** By inserting tricky texts such as "*Let's think step by step*" and "*Therefore, the answer is*" to prompt, zero-shot CoT leads LLMs to generate task-specific reasoning steps independently, without providing any task-relevant instruction or grounding example.
- ***Few-shot CoT.*** Task-specific reasoning steps are manually designed for each demonstration in ICL, where the

original input-output examples are augmented to input-CoT-output manners. Besides, CoT can also augment the task descriptions in ICL demonstrations, by adding interpretable descriptions of reasoning steps based on task-specific knowledge.

In practice, the design of appropriate CoT reasoning steps highly depends on the contexts and objectives of the specific recommendation tasks. For example, a simple CoT template "*Please infer the preference of the user and recommend suitable items.*" is proposed to guide LLMs to first infer the user's explicit preference and then generate final recommendations [20]. So far, there is still a notable lack of research addressing the general format of CoT prompting for recommendations tasks. Next, we present a preliminary idea of CoT prompting, through an example in the context of e-commerce recommendations below.

*[**CoT Prompting**] Based on the user purchase history, let's think step-by-step. First, please infer the user's high-level shopping intent. Second, what items are usually bought together with the purchased items? Finally, please select the most relevant items based on the shopping intent and recommend them to the user.*

Despite the limited number of works on CoT prompting in the RecSys field, a recent research [106] has revealed the great effectiveness of adopting CoT prompting to facilitate

the graph reasoning ability of LLMs (T5 particularly) by modeling the reasoning steps as nodes and connecting the reasoning paths as edges instead of a sequential chain. We believe that similar ideas can be potentially transferred, and contribute to the CoT prompting for RecSys, based on the fact that recommendation tasks can be considered as a special case of link prediction problems in graph learning.

5.2 Prompt Tuning

In contrast to manually prompting LLMs for downstream tasks (*e.g.*, manually generate task-specific prompt in natural language), prompt tuning serves as an additive technique of prompting, which adds new prompt tokens to LLMs and optimizes the prompt based on the task-specific dataset. Generally, prompt tuning requires less task-specific knowledge and human effort than manually designing prompts for specific tasks and only involves minimal parameter updates of the tunable prompt and the input layer of LLMs. For example, AutoPrompt [107] takes the step of decomposing prompt into a set of vocabulary tokens, and finding the suitable tokens to language models via gradient-based search with respect to the performance on specific tasks.

According to the definition, prompts can be either discrete (*i.e.*, hard) or continuous (*i.e.*, soft) that guide LLMs to generate the expected output [108]. Thus, we categorize prompt tuning strategies for prompting LLMs for RecSys into hard prompt tuning and soft prompt tuning, as illustrated below.

5.2.1 Hard Prompt Tuning

Hard prompt tuning is to generate and update discrete text templates of prompt (*e.g.*, in natural language), for prompting LLMs to specific downstream tasks. Dong et al. [108] argue that ICL can be considered as a subclass of hard prompt tuning and regard the in-context demonstrations in ICL as a part of the prompt. From this perspective, ICL performs hard prompt tuning for prompting LLMs to downstream recommendation tasks by refining prompts in natural language based on task-specific recommendation datasets. Despite the effectiveness and convenience of generating or refining natural language prompts for downstream recommendation tasks, hard prompt tuning inevitably faces the challenge of discrete optimization, which requires laborious trial and error to discover the vast vocabulary space in order to find suitable prompts for specific recommendation tasks.

5.2.2 Soft Prompt Tuning

In contrast to discrete prompt, soft prompt tuning employs continuous vectors as prompt (*e.g.*, text embeddings), and optimizes the prompt based on task-specific datasets, such as using gradient methods to update the prompt with respect to a recommendation loss. In LLMs, soft prompt tokens are often concatenated to the original input tokens at the input layer (*e.g.*, tokenizer). During soft prompt tuning, only the soft prompt and minimal parameters at the input layer of LLMs will be updated.

To improve the recommendation performance of LLMs, some existing works combine advanced feature extraction and representation learning methods to better capture and embed task-specific information in RecSys into soft prompts. For instance, Wu *et al.* [109] apply contrastive learning to capture user representations and encode them into prompt tokens, and Wang *et al.* [63] and Guo *et al.* [110] share the similar idea of encoding mutual information in cross-domain recommendations into soft prompt. In addition to directly embedding task-specific information into soft prompt, soft prompt can also be learned based on task-specific datasets. For example, randomly initialized soft prompts are adopted to guide T5 to generate desired recommendation results [104], where the soft prompt is optimized in an end-to-end manner with respect to a recommendation loss based on the T5 output. Compared to the hard prompt, the soft prompt is more feasible for tuning on continuous space but in a cost of explainability [104]. In other words, compared to task-specific hard prompt in a natural language like *"Your task is to recommend ..."*, the relationships between the specific downstream tasks and the soft prompt written in continuous vectors are not interpretable to humans.

5.3 Instruction Tuning

Although prompting LLMs has demonstrated remarkable few-shot performance on unseen downstream tasks, recent studies demonstrated that prompting strategies have much poorer zero-shot ability [87]. To address the limitations, instruction tuning is proposed to fine-tune LLMs over multiple task-specific prompts. In other words, instruction tuning possesses features of both prompting and pre-training & fine-tuning paradigms. This helps LLMs gain better capabilities of exactly following prompts as instructions for diverse downstream tasks, which hence contributes to the enhanced zero-shot performance of LLMs on unseen tasks by accurately following new task instructions. The key insight of instruction tuning is to train LLMs to follow prompts as task instructions, rather than to solve specific downstream tasks. More specifically, instruction tuning can be divided into two stages: "instruction" (*i.e.*, prompt) generation and model "tuning", since the straightforward idea of instruction tuning is the combination of prompting and fine-tuning LLMs.

- **Instruction (Prompt) Generation Stage.** Formally, instruction tuning introduces a format of instruction-based prompt in natural language, which composes of task-oriented input (*i.e.*, task descriptions based on task-specific dataset) and desired target (*i.e.*, corresponding output based on task-specific dataset) pairs. Considering the instruction tuning of LLMs for downstream recommendation tasks, Zhang *et al.* [20] propose a recommendation-oriented instruction template, including user preferences, intentions, and task forms, which serves as a common template for generating instructions for various recommendation tasks. More directly, three-part instruction templates in the form of "task description-input-output" are used in [59], [61] to generate instructions based on task-specific recommendation datasets.

- **Model Tuning Stage.** The second stage is to fine-tune LLMs over multiple aforementioned instructions for downstream tasks, where we categorize the existing works on RecSys, as shown in Table 3, according to the LLMs fine-tuning manners: full-model tuning and

parameter-efficient model tuning (see Section 4.2 for explanations), since basically the same principles of fine-tuning LLMs are adopted in this stage. For example, Bao *et al.* [59] utilize LoRA to make the instruction tuning of LLaMA more lightweight for downstream recommendation tasks.

In addition to textual data in RecSys, instruction tuning is recently explored to enhance the graph understanding ability of LLMs for recommendation tasks. In particular, Wu *et al.* [81] propose an LLM-based prompt constructor to encode the paths of nodes (*e.g.*, candidate items) and edges (*e.g.*, relationships between items) in behavior graphs into natural language descriptions, which is subsequently used for instruction tuning an LLM-based recommender based on task-specific dataset.

6 FUTURE DIRECTIONS

In this survey, we have comprehensively reviewed the recent advanced techniques for LLM-enhanced recommender systems. Since the adaption of LLMs to recommender systems is still in the early stage, there are still many challenges and opportunities. In this section, we discuss some potential future directions in this field.

6.1 Hallucination Mitigation

Although LLMs are used in various fields, a significant challenge is the phenomenon of '*hallucination*', where language models generate outputs that are plausible-sounding but factually incorrect or not referable in the input data [111], [112]. For instance, considering a scenario where you are seeking today's news events, the LLMs erroneously recommend/generate news that, in fact, does not exist. The causes of this problem are manifold such as source-reference divergence existing in dataset, and training&modeling choices of neural network models [113]. Moreover, the hallucination issue poses severe threats to users and society, especially in high-stakes recommendation scenarios such as medical recommendations or legal advice, where the dissemination of incorrect information can have severe real consequences. To address such issues, employing factual knowledge graphs as supplementary factual knowledge during the training and inference stages of LLMs for RecSys is promising to mitigate the hallucination problem. In addition, the model's output stage can be scrutinized to verify the accuracy and factuality of the produced content.

6.2 Trustworthy Large Language Models for Recommender Systems

The development of LLMs for RecSys has brought significant benefits to humans, including economic value creation, time and effort savings, and social benefits. However, these data-driven LLMs for RecSys might also pose serious threats to users and society [5], [114], [115], due to unreliable decisions making, unequal treatment of various consumers or producers, a lack of transparency and explainability, and privacy issues stemming from the extensive use of personal data for customization, among other concerns. As a result, there is an increasing concern about the issue of trustworthiness in LLMs for RecSys to mitigate the negative impacts and enhance public trust in LLM-based RecSys techniques. Thus, it is desired to achieve trustworthiness in LLMs for RecSys from four of the most crucial dimensions, including *Safety&Robustness, Non-discrimination&Fairness, Explainability*, and *Privacy*.

6.2.1 Safety&Robustness

LLMs have been proven to advance recommender systems in various aspects, but they are also highly vulnerable to adversarial perturbations (*i.e.*, minor changes in the input) that can compromise the safety and robustness of their uses in safety-critical applications [44], [114]. These vulnerabilities towards noisy inputs are frequently carried out with malicious intent, such as to gain unlawful profits and manipulate markets for specific products [116]–[119]. Therefore, it is crucial to ensure that the output of LLMs for recommender systems is stable given small changes in the LLMs' input. In order to enhance model safety and robustness, GPT-4 integrates safety-related prompts during reinforcement learning from human feedback (RLHF) [120]. However, the RLHF method requires a significant number of experts for manual labeling, which might not be feasible in practice. An alternative solution might involve the automatic pre-processing of prompts designed for recommender tasks before input to LLMs. This could include pre-processing for malicious prompts or standardizing prompts with similar purposes to have the same final input, thus potentially improving safety and robustness. In addition, as one of the representative techniques, adversarial training [121] can be used to improve the robustness of LLM-based recommender systems.

6.2.2 Non-discrimination&Fairness

LLMs, trained on vast datasets, often inadvertently learn and perpetuate biases and stereotypes in the human data that will later reveal themselves in the recommendation results. This phenomenon can lead to a range of adverse outcomes, from the propagation of stereotypes to the unfair treatment of certain user groups [2], [122], [123]. For instance, in the context of recommender systems, these biases can manifest as discriminatory recommendations, where certain items are unfairly promoted or demoted based on these learned biases. More recently, a few studies such as FaiRLLM [89] and UP5 [104] explore the fairness problem in recommender systems brought by LLMs, which only focus on user-side and item generation task. Concurrently, Hou *et al.* [91] guide LLMs with prompts to formalize the recommendation task as a conditional ranking task to improve item-side fairness. However, studies on non-discrimination and fairness in LLMs for RecSys are at a preliminary stage, further research is still needed.

6.2.3 Explainability

Owing to privacy and security considerations, certain companies and organizations choose not to open-source their advanced LLMs, such as ChatGPT and GPT-4, indicating that the architectures and parameters of these LLMs for RecSys are not publicly available for the public to understand their complex internal working mechanisms. Consequently, LLMs for RecSys can be treated as the 'black box', complicating

the process for users trying to comprehend why a specific output or recommendation was produced. Recently, Bills *et al.* [124] try to use GPT-4 to generate natural language descriptions to explain the neuronal behavior in the GPT-2 model. While this study is foundational, it also introduces fresh perspectives for comprehending the workings of LLMs. Neurons exhibit intricate behaviors that may not be easily encapsulated through simple natural language. To this end, efforts should be made to understand how LLMs for RecSys function, so as to enhance the explainability of LLM-based recommender systems.

6.2.4 Privacy

Privacy is a paramount concern when it comes to LLMs for RecSys. The reasons for this are multifold. On the one hand, the success of LLMs for recommender systems highly depends on large quantities of data that are collected from a variety of sources, such as social media and books. Users' sensitive information (*e.g.*, email and gender) contained in data is likely to be used to train modern LLMs for enhancing prediction performance and providing personalized experiences, leading to the risk of leaking users' private information. On the other hand, these systems often handle sensitive user data, including personal preferences, online behaviors, and other identifiable information. If not properly protected, this data could be exploited, leading to breaches of privacy. Therefore, ensuring the privacy and security of this data is crucial. Carlini *et al.* [125] show that LLMs might reveal some uses' real identity or private information when generating text. Recently, Li *et al.* [126] introduce RAPT that allows users to customize LLMs with their private data based on prompt tuning. It provides a direction on how to protect user privacy at LLMs for RecSys.

6.3 Vertical Domain-Specific LLMs for Recommender Systems

General LLMs, such as ChatGPT, whose powerful generation and inference capabilities make them a universal tool in various areas. Vertical domain-specific LLMs are LLMs that have been trained and optimized for a specific domain or industry, such as health [127] and finance [56]. Compared to general LLMs for RecSys, vertical domain-specific LLM-empowered RecSys are more focused on the knowledge and skills of a particular domain and have a higher degree of domain expertise and practicality. Instead of sifting through irrelevant information, users can focus on content that is directly aligned with their work or personalized preferences. By providing tailored recommendations, vertical domain-specific LLMs for RecSys can save professionals a significant amount of time. More recently, existing works have presented vertical domain-specific LLMs that cover a wide range of areas, such as medical care [128], [129], law [130], [131], and finance [132]. Due to trained specifically, these vertical domain-specific LLMs can better understand and process domain-specific knowledge, terminology and context. Yet the requirement for vast amounts of domain-specific data to train these models poses significant challenges in data collection and annotation. As such, constructing high-quality domain datasets and using suitable tuning strategies for specific domains are necessary steps in the development of vertical domain-specific LLMs for RecSys. In particular, Jin *et al.* [133] propose a multilingual dataset named Amazon-M2 as a new setting of session-based recommendations from Amazon (*i.e.*, sessions containing the interacted items of users) and inspire the opportunities to leverage LLMs as RecSys to learn on session graphs with multilingual and textual data, such as item (node) attributes including product titles, prices, and descriptions across session graphs of users from different locales (multilingual).

6.4 Users&Items Indexing

Recent research suggests that LLMs may not perform well when dealing with long texts in RecSys, as it can be difficult to effectively capture user-item interaction information in long texts [91]. On the other hand, user-item interactions (e.g., click, like, and subscription) with unique identities (i.e., discrete IDs) in recommender systems contain rich collaborative knowledge and make great contributions to understanding and predicting user preferences, encompassing both explicit actions like ratings and reviews, as well as implicit behaviors like browsing history or purchase data. Several studies, including InstructRec [20], PALR [61], GPT4Rec [134] and UP5 [104], have attempted to utilize user-item history interaction information as text prompts inputted into LLMs (*e.g.*, ChatGPT) in order to make recommendations. To address the long text problem, one possible solution is to perform user and item indexing for learning collaborative knowledge by incorporating user-item interactions. Therefore, rather than merely using text formats to represent users and items, advanced methods for indexing users&items are desired to build LLM-based recommender systems.

6.5 Fine-tuning Efficiency

In the application of LLMs to RecSys, fine-tuning refers to the process of adapting a pre-trained LLM to a specific task or domain, such as recommending movies [61] or books [59]. This process allows the model to leverage the general language understanding capabilities learned during pre-training while specializing its knowledge to the task at hand. However, fine-tuning can be computationally expensive, particularly for very large models and large datasets in recommender systems. Therefore, improving the efficiency of fine-tuning is a key challenge. In this case, Fu *et al.* [135] use adapter modules, which are small, plug-in neural networks that can be optimized separately from the main model, to achieve parameter-efficient transfer learning. However, the current adapter tuning techniques for RecSys fall slightly behind full-model fine-tuning when it comes to cross-platform image recommendation. The exploration of adapter tuning effects for multi-modal (*i.e.*, both text and image) RecSys is a potential future direction. In addition, given that most typical adapter tuning does not help to speed up the training process in practice, it is important to explore effective optimization techniques to reduce the computational cost and time for RecSys through end-to-end training.

6.6 Data Augmentation

Most conventional studies in the recommender systems domain rely on real data-driven research, founded on the collection of user behavior data via user interaction in digital platforms or through the recruitment of annotators. Nonetheless, these approaches appear to be resource-intensive and may not be sustainable in the long term. The quality and variety of the input data directly influence the performance and versatility of the models. With the aim to overcome the shortcomings of real data-centric studies, Wang *et al.* [136] introduce RecAgent, a simulation paradigm for recommender systems based on LLMs, which includes a user module for browsing and communication on the social media, and a recommender module for providing search or recommendation lists. Additionally, LLM-Rec [96] incorporates four prompting strategies to improve personalized content recommendations, which demonstrates through experiments that diverse prompts and input augmentation techniques can enhance recommendation performance. Therefore, rather than solely deploying LLMs as recommender systems, utilizing them for data augmentation to bolster recommendations emerges as a promising strategy in the future.

7 CONCLUSION

As one of the most advanced AI techniques, LLMs have achieved great success in various applications, such as molecule discovery and finance, owing to their remarkable abilities in language understanding and generation, powerful generalization and reasoning skills, and prompt-adaptation to new tasks and diverse domains. Similarly, increasing efforts have been made to revolutionize recommender systems with LLMs, so as to provide high-quality and personalized suggestion services. Given the rapid evolution of this research topic in recommender systems, there is a pressing need for a systematic overview that comprehensively summarizes the existing LLM-empowered recommender systems. To fill the gap, in this survey, we have provided a comprehensive overview of LLM-empowered RecSys from *pre-training&fine-tuning* and *prompting* paradigms, so as to provide researchers and practitioners in relevant fields with an in-depth understanding. Nevertheless, the current research on LLMs for RecSys is still in its early stage which calls for more systematic and comprehensive studies of LLMs in this field. Therefore, we also discussed some potential future directions in this field.

ACKNOWLEDGMENTS

REFERENCES

[1] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[2] X. Chen, W. Fan, J. Chen, H. Liu, Z. Liu, Z. Zhang, and Q. Li, "Fairly adaptive negative sampling for recommendations," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 3723–3733.

[3] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-rec: Towards interactive and explainable llms-augmented recommender system," *arXiv preprint arXiv:2303.14524*, 2023.

[4] J. Chen, L. Ma, X. Li, N. Thakurdesai, J. Xu, J. H. Cho, K. Nag, E. Korpeoglu, S. Kumar, and K. Achan, "Knowledge graph completion models are few-shot learners: An empirical study of relation labeling in e-commerce with llms," *arXiv preprint arXiv:2305.09858*, 2023.

[5] W. Fan, X. Zhao, X. Chen, J. Su, J. Gao, L. Wang, Q. Liu, Y. Wang, H. Xu, L. Chen *et al.*, "A comprehensive survey on trustworthy recommender systems," *arXiv preprint arXiv:2209.10117*, 2022.

[6] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[7] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, "Deep adversarial social recommendation," in *28th International Joint Conference on Artificial Intelligence (IJCAI-19)*. International Joint Conferences on Artificial Intelligence, 2019, pp. 1351–1357.

[8] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 425–434.

[9] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[10] W. Fan, C. Liu, Y. Liu, J. Li, H. Li, H. Liu, J. Tang, and Q. Li, "Generative diffusion models on graphs: Methods and applications," *arXiv preprint arXiv:2302.02591*, 2023.

[11] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[12] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, and Q. Li, "Deep social collaborative filtering," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 305–313.

[13] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[14] Z. Qiu, X. Wu, J. Gao, and W. Fan, "U-bert: Pre-training user representations for improved recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4320–4327.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.

[16] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.

[17] J. Li, Y. Liu, W. Fan, X.-Y. Wei, H. Liu, J. Tang, and Q. Li, "Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective," *arXiv preprint arXiv:2306.06615*, 2023.

[18] Z. Chen, H. Mao, H. Li, W. Jin, H. Wen, X. Wei, S. Wang, D. Yin, W. Fan, H. Liu *et al.*, "Exploring the potential of large language models (llms) in learning on graphs," *arXiv preprint arXiv:2307.03393*, 2023.

[19] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[20] J. Zhang, R. Xie, Y. Hou, W. X. Zhao, L. Lin, and J.-R. Wen, "Recommendation as instruction following: A large language model empowered recommendation approach," *arXiv preprint arXiv:2305.07001*, 2023.

[21] P. Liu, L. Zhang, and J. A. Gulla, "Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems," *arXiv preprint arXiv:2302.03735*, 2023.

[22] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu *et al.*, "A survey on large language models for recommendation," *arXiv preprint arXiv:2305.19860*, 2023.

[23] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang *et al.*, "How can recommender systems benefit from large language models: A survey," *arXiv preprint arXiv:2306.05817*, 2023.

[24] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li, and K. Tang, "Disentangled contrastive learning for social recommendation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4570–4574.

[25] W. Fan, X. Liu, W. Jin, X. Zhao, J. Tang, and Q. Li, "Graph trend filtering networks for recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 112–121.

[26] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[27] X. Zhao, H. Liu, W. Fan, H. Liu, J. Tang, and C. Wang, "Autoloss: Automated loss function search in recommendations," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3959–3967.

[28] X. Zhao, H. Liu, W. Fan, H. Liu, J. Tang, C. Wang, M. Chen, X. Zheng, X. Liu, and X. Yang, "Autoemb: Automated embedding dimensionality search in streaming recommendations," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 896–905.

[29] F. Vasile, E. Smirnova, and A. Conneau, "Meta-prod2vec: Product embeddings using side-information for recommendation," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 225–232.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[31] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[32] Y. Ma and J. Tang, *Deep learning on graphs*. Cambridge University Press, 2021.

[33] T. Derr, Y. Ma, W. Fan, X. Liu, C. Aggarwal, and J. Tang, "Epidemic graph convolutional network," in *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, 2020, pp. 160–168.

[34] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1583–1592.

[35] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu *et al.*, "Mind: A large-scale dataset for news recommendation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3597–3606.

[36] C. Wu, F. Wu, Y. Huang, and X. Xie, "Personalized news recommendation: Methods and challenges," *ACM Transactions on Information Systems*, vol. 41, no. 1, pp. 1–50, 2023.

[37] S. Dongre and J. Agrawal, "Deep learning-based drug recommendation and adr detection healthcare model on social media," *IEEE Transactions on Computational Social Systems*, 2023.

[38] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[39] J. Liu, C. Liu, R. Lv, K. Zhou, and Y. Zhang, "Is chatgpt a good recommender? a preliminary study," *arXiv preprint arXiv:2304.10149*, 2023.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[41] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[42] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[44] Z. Zhang, G. Zhang, B. Hou, W. Fan, Q. Li, S. Liu, Y. Zhang, and S. Chang, "Certified robustness for large language models with self-denoising," *arXiv preprint arXiv:2307.07171*, 2023.

[45] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[46] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[47] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[48] H. J. Kim, H. Cho, J. Kim, T. Kim, K. M. Yoo, and S.-g. Lee, "Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator," *arXiv preprint arXiv:2206.08082*, 2022.

[49] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," *arXiv preprint arXiv:2112.08633*, 2021.

[50] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.

[51] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[52] E. Zelikman, Y. Wu, J. Mu, and N. Goodman, "Star: Bootstrapping reasoning with reasoning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 476–15 488, 2022.

[53] H. Fei, B. Li, Q. Liu, L. Bing, F. Li, and T.-S. Chua, "Reasoning implicit sentiment with chain-of-thought prompting," *arXiv preprint arXiv:2305.11255*, 2023.

[54] Z. Jin and W. Lu, "Tab-cot: Zero-shot tabular chain of thought," *arXiv preprint arXiv:2305.17812*, 2023.

[55] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023.

[56] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[57] W.-C. Kang, J. Ni, N. Mehta, M. Sathiamoorthy, L. Hong, E. Chi, and D. Z. Cheng, "Do llms understand user preferences? evaluating llms on user rating prediction," *arXiv preprint arXiv:2305.06474*, 2023.

[58] A. Zhiyuli, Y. Chen, X. Zhang, and X. Liang, "Bookgpt: A general framework for book recommendation empowered by large language model," *arXiv preprint arXiv:2305.15673*, 2023.

[59] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "Tallrec: An effective and efficient tuning framework to align large language model with recommendation," *arXiv preprint arXiv:2305.00447*, 2023.

[60] Z. Cui, J. Ma, C. Zhou, J. Zhou, and H. Yang, "M6-rec: Generative pretrained language models are open-ended recommender systems," *arXiv preprint arXiv:2205.08084*, 2022.

[61] Z. Chen, "Palr: Personalization aware llms for recommendation," *arXiv preprint arXiv:2305.07622*, 2023.

[62] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)," in *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022, pp. 299–315.

[63] X. Wang, K. Zhou, J.-R. Wen, and W. X. Zhao, "Towards unified conversational recommender systems via knowledge-enhanced prompt learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1929–1937.

[64] Y. Deng, W. Zhang, W. Xu, W. Lei, T.-S. Chua, and W. Lam, "A unified multi-task learning framework for multi-goal conversational recommender systems," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–25, 2023.

[65] W. Hua, S. Xu, Y. Ge, and Y. Zhang, "How to index item ids for recommendation foundation models," *arXiv preprint arXiv:2305.06569*, 2023.

[66] S. Rajput, N. Mehta, A. Singh, R. H. Keshavan, T. Vu, L. Heldt, L. Hong, Y. Tay, V. Q. Tran, J. Samost *et al.*, "Recommender systems with generative retrieval," *arXiv preprint arXiv:2305.05065*, 2023.

[67] Z. Yuan, F. Yuan, Y. Song, Y. Li, J. Fu, F. Yang, Y. Pan, and Y. Ni, "Where to go next for recommender systems? id- vs. modality-based recommender models revisited," *arXiv preprint arXiv:2303.13835*, 2023.

[68] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen, "Towards universal sequence representation learning for recommender systems," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 585–593.

[69] R. Li, W. Deng, Y. Cheng, Z. Yuan, J. Zhang, and F. Yuan, "Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and insights," *arXiv preprint arXiv:2305.11700*, 2023.

[70] Y. Hou, Z. He, J. McAuley, and W. X. Zhao, "Learning vector-quantized item representation for transferable sequential recommenders," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1162–1171.

[71] Z. Fan, Z. Liu, S. Heinecke, J. Zhang, H. Wang, C. Xiong, and P. S. Yu, "Zero-shot item-based recommendation via multi-task product knowledge graph pre-training," *arXiv preprint arXiv:2305.07633*, 2023.

[72] K. Shin, H. Kwak, K.-M. Kim, M. Kim, Y.-J. Park, J. Jeong, and S. Jung, "One4all user representation for recommender systems in e-commerce," *arXiv preprint arXiv:2106.00573*, 2021.

[73] C. Wu, F. Wu, T. Qi, J. Lian, Y. Huang, and X. Xie, "Ptum: Pre-training user model from unlabeled user behaviors via self-supervision," *arXiv preprint arXiv:2010.01494*, 2020.

[74] L. Friedman, S. Ahuja, D. Allen, T. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara *et al.*, "Leveraging large language models in conversational recommender systems," *arXiv preprint arXiv:2305.07961*, 2023.

[75] T. Shen, J. Li, M. R. Bouadjenek, Z. Mai, and S. Sanner, "Towards understanding and mitigating unintended biases in language model-driven conversational recommendation," *Information Processing & Management*, vol. 60, no. 1, p. 103139, 2023.

[76] J. Wang, F. Yuan, M. Cheng, J. M. Jose, C. Yu, B. Kong, Z. Wang, B. Hu, and Z. Li, "Transrec: Learning transferable recommendation from mixture-of-modality feedback," *arXiv preprint arXiv:2206.06190*, 2022.

[77] A. G. Carranza, R. Farahani, N. Ponomareva, A. Kurakin, M. Jagielski, and M. Nasr, "Privacy-preserving recommender systems with synthetic query generation using differentially private large language models," *arXiv preprint arXiv:2305.05973*, 2023.

[78] Z. Zheng, Z. Qiu, X. Hu, L. Wu, H. Zhu, and H. Xiong, "Generative job recommendations with large language model," *arXiv preprint arXiv:2307.02157*, 2023.

[79] H. Kim, J. Jeong, K.-M. Kim, D. Lee, H. D. Lee, D. Seo, J. Han, D. W. Park, J. A. Heo, and R. Y. Kim, "Intent-based product collections for e-commerce using pretrained language models," in *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2021, pp. 228–237.

[80] Z. Mao, H. Wang, Y. Du, and K.-f. Wong, "Unitrec: A unified text-to-text transformer and joint contrastive learning framework for text-based recommendation," *arXiv preprint arXiv:2305.15756*, 2023.

[81] L. Wu, Z. Qiu, Z. Zheng, H. Zhu, and E. Chen, "Exploring large language model for graph data understanding in online job recommendations," *arXiv preprint arXiv:2307.05722*, 2023.

[82] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019.

[83] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[84] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[85] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[86] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," *arXiv preprint arXiv:2012.15723*, 2020.

[87] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.

[88] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, and J. Xu, "Uncovering chatgpt's capabilities in recommender systems," *arXiv preprint arXiv:2305.02182*, 2023.

[89] J. Zhang, K. Bao, Y. Zhang, W. Wang, F. Feng, and X. He, "Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation," *arXiv preprint arXiv:2305.07609*, 2023.

[90] Q. Liu, N. Chen, T. Sakai, and X.-M. Wu, "A first look at llm-powered generative news recommendation," *arXiv preprint arXiv:2305.06566*, 2023.

[91] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao, "Large language models are zero-shot rankers for recommender systems," *arXiv preprint arXiv:2305.08845*, 2023.

[92] X. Wang, X. Tang, W. X. Zhao, J. Wang, and J.-R. Wen, "Rethinking the evaluation for conversational recommendation in the era of large language models," *arXiv preprint arXiv:2305.13112*, 2023.

[93] W. Wang, X. Lin, F. Feng, X. He, and T.-S. Chua, "Generative recommendation: Towards next-generation recommender paradigm," *arXiv preprint arXiv:2304.03516*, 2023.

[94] Y. Du, D. Luo, R. Yan, H. Liu, Y. Song, H. Zhu, and J. Zhang, "Enhancing job recommendation through llm-based generative adversarial networks," *arXiv preprint arXiv:2307.10747*, 2023.

[95] L. Wang and E.-P. Lim, "Zero-shot next-item recommendation using large pretrained language models," *arXiv preprint arXiv:2304.03153*, 2023.

[96] H. Lyu, S. Jiang, H. Zeng, Y. Xia, and J. Luo, "Llm-rec: Personalized recommendation via prompting large language models," *arXiv preprint arXiv:2307.15780*, 2023.

[97] M. Leszczynski, R. Ganti, S. Zhang, K. Balog, F. Radlinski, F. Pereira, and A. T. Chaganty, "Generating synthetic data for conversational music recommendation using random walks and language models," *arXiv preprint arXiv:2301.11489*, 2023.

[98] X. Wu, H. Zhou, W. Yao, X. Huang, and N. Liu, "Towards personalized cold-start recommendation with prompts," *arXiv preprint arXiv:2306.17256*, 2023.

[99] K. Christakopoulou, A. Lalama, C. Adams, I. Qu, Y. Amir, S. Chucri, P. Vollucci, F. Soldo, D. Bseiso, S. Scodel *et al.*, "Large language models for user interest journeys," *arXiv preprint arXiv:2305.15498*, 2023.

[100] S. Sanner, K. Balog, F. Radlinski, B. Wedin, and L. Dixon, "Large language models are competitive near cold-start recommenders for language-and item-based preferences," *arXiv preprint arXiv:2307.14225*, 2023.

[101] J. Zhang, "Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt," *arXiv preprint arXiv:2304.11116*, 2023.

[102] Y. Xi, W. Liu, J. Lin, J. Zhu, B. Chen, R. Tang, W. Zhang, R. Zhang, and Y. Yu, "Towards open-world recommendation with knowledge augmentation from large language models," *arXiv preprint arXiv:2306.10933*, 2023.

[103] G. Lin and Y. Zhang, "Sparks of artificial general recommender (agr): Early experiments with chatgpt," *arXiv preprint arXiv:2305.04518*, 2023.

[104] W. Hua, Y. Ge, S. Xu, J. Ji, and Y. Zhang, "Up5: Unbiased foundation model for fairness-aware recommendation," *arXiv preprint arXiv:2305.12090*, 2023.

[105] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, "Genrec: Large language model for generative recommendation," *arXiv e-prints*, pp. arXiv–2307, 2023.

[106] Y. Yao, Z. Li, and H. Zhao, "Beyond chain-of-thought, effective graph-of-thought reasoning in large language models," *arXiv preprint arXiv:2305.16582*, 2023.

[107] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[108] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[109] Y. Wu, R. Xie, Y. Zhu, F. Zhuang, X. Zhang, L. Lin, and Q. He, "Personalized prompts for sequential recommendation," *arXiv preprint arXiv:2205.09666*, 2022.

[110] L. Guo, C. Wang, X. Wang, L. Zhu, and H. Yin, "Automated prompting for non-overlapping cross-domain sequential recommendation," *arXiv preprint arXiv:2304.04218*, 2023.

[111] P. Manakul, A. Liusie, and M. J. Gales, "Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models," *arXiv preprint arXiv:2303.08896*, 2023.

[112] N. McKenna, T. Li, L. Cheng, M. J. Hosseini, M. Johnson, and M. Steedman, "Sources of hallucination by large language models on inference tasks," *arXiv preprint arXiv:2305.14552*, 2023.

[113] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[114] T. Y. Zhuo, Y. Huang, C. Chen, and Z. Xing, "Exploring ai ethics of chatgpt: A diagnostic analysis," *arXiv preprint arXiv:2301.12867*, 2023.

[115] H. Liu, Y. Wang, W. Fan, X. Liu, Y. Li, S. Jain, Y. Liu, A. Jain, and J. Tang, "Trustworthy ai: A computational perspective," *ACM Transactions on Intelligent Systems and Technology*, 2022.

[116] W. Fan, T. Derr, X. Zhao, Y. Ma, H. Liu, J. Wang, J. Tang, and Q. Li, "Attacking black-box recommendations via copying cross-domain user profiles," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1583–1594.

[117] J. Chen, W. Fan, G. Zhu, X. Zhao, C. Yuan, Q. Li, and Y. Huang, "Knowledge-enhanced black-box attacks for recommendations," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 108–117.

[118] W. Fan, X. Zhao, Q. Li, T. Derr, Y. Ma, H. Liu, J. Wang, and J. Tang, "Adversarial attacks for black-box recommender systems via copying transferable cross-domain user profiles," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[119] W. Fan, W. Jin, X. Liu, H. Xu, X. Tang, S. Wang, Q. Li, J. Tang, J. Wang, and C. Aggarwal, "Jointly attacking graph neural network and its explanations," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023.

[120] OpenAI, "Gpt-4 technical report," *OpenAI*, 2023.

[121] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, "Adversarial training towards robust multimedia recommender system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 855–867, 2019.

[122] G. Zhang, Y. Zhang, Y. Zhang, W. Fan, Q. Li, S. Liu, and S. Chang, "Fairness reprogramming," in *Thirty-sixth Conference on Neural Information Processing Systems*, 2022.

[123] H. Liu, J. Dacon, W. Fan, H. Liu, Z. Liu, and J. Tang, "Does gender matter? towards fairness in dialogue systems," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 4403–4416.

[124] S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders, "Language models can explain neurons in language models," *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2023.

[125] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models." in *USENIX Security Symposium*, vol. 6, 2021.

[126] Y. Li, Z. Tan, and Y. Liu, "Privacy-preserving prompt tuning for large language model services," *arXiv preprint arXiv:2305.06212*, 2023.

[127] A. J. Nastasi, K. R. Courtright, S. D. Halpern, and G. E. Weissman, "Does chatgpt provide appropriate and equitable medical advice?: A vignette-based, clinical evaluation across care contexts," *medRxiv*, pp. 2023–02, 2023.

[128] H. Zhang, J. Chen, F. Jiang, F. Yu, Z. Chen, J. Li, G. Chen, X. Wu, Z. Zhang, Q. Xiao *et al.*, "Huatuogpt, towards taming language model to be a doctor," *arXiv preprint arXiv:2305.15075*, 2023.

[129] H. Xiong, S. Wang, Y. Zhu, Z. Zhao, Y. Liu, Q. Wang, and D. Shen, "Doctorglm: Fine-tuning your chinese doctor is not a herculean task," *arXiv preprint arXiv:2304.01097*, 2023.

[130] H.-T. Nguyen, "A brief report on lawgpt 1.0: A virtual legal assistant based on gpt-3," *arXiv preprint arXiv:2302.05729*, 2023.

[131] Q. Huang, M. Tao, Z. An, C. Zhang, C. Jiang, Z. Chen, Z. Wu, and Y. Feng, "Lawyer llama technical report," *arXiv preprint arXiv:2305.15062*, 2023.

[132] H. Yang, X.-Y. Liu, and C. D. Wang, "Fingpt: Open-source financial large language models," *arXiv preprint arXiv:2306.06031*, 2023.

[133] W. Jin, H. Mao, Z. Li, H. Jiang, C. Luo, H. Wen, H. Han, H. Lu, Z. Wang, R. Li *et al.*, "Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation," *arXiv preprint arXiv:2307.09688*, 2023.

[134] J. Li, W. Zhang, T. Wang, G. Xiong, A. Lu, and G. Medioni, "Gpt4rec: A generative framework for personalized recommendation and user interests interpretation," *arXiv preprint arXiv:2304.03879*, 2023.

[135] J. Fu, F. Yuan, Y. Song, Z. Yuan, M. Cheng, S. Cheng, J. Zhang, J. Wang, and Y. Pan, "Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights," *arXiv preprint arXiv:2305.15036*, 2023.

[136] L. Wang, J. Zhang, X. Chen, Y. Lin, R. Song, W. X. Zhao, and J.-R. Wen, "Recagent: A novel simulation paradigm for recommender systems," *arXiv preprint arXiv:2306.02552*, 2023.

Wenqi Fan is a research assistant professor of the Department of Computing at The Hong Kong Polytechnic University (PolyU). He received his Ph.D. degree from the City University of Hong Kong (CityU) in 2020. From 2018 to 2020, he was a visiting research scholar at Michigan State University (MSU). His research interests are in the broad areas of machine learning and data mining, with a particular focus on Recommender Systems, Graph Neural Networks, and Trustworthy Recommendations. He has published innovative papers in top-tier journals and conferences such as TKDE, TIST, KDD, WWW, ICDE, NeurIPS, SIGIR, IJCAI, AAAI, RecSys, WSDM, etc. He serves as top-tier conference (senior) program committee members and session chairs (e.g., ICML, ICLR, NeurIPS, KDD, WWW, AAAI, IJCAI, WSDM, etc.), and journal reviewers (e.g., TKDE, TIST, TKDD, TOIS, TAI, etc.). More information about him can be found at https://wenqifan03.github.io.

Zihuai Zhao is currently a PhD student of the Department of Computing (COMP), Hong Kong Polytechnic University (PolyU), under the supervision of Prof. Qing Li and Dr. Wenqi Fan. Before joining the PolyU, he received both his Master's degree (MPhil in Electrical Engineering) and Bachelor's degree (B.Eng. (Hons) in Electrical Engineering) from the University of Sydney in 2023 and 2020, respectively. His research interest covers Recommender Systems, Natural Language Processing, and Deep Reinforcement Learning. He has published innovative works in top-tier journals such as IoT-J. For more information, please visit https://scofizz.github.io/.

Jiatong Li is currently a PhD student of the Department of Computing (COMP), The Hong Kong Polytechnic University (funded by HKPFS). Before joining the PolyU, he received my Master's degree of Information Technology (with Distinction) from the University of Melbourne, under the supervision of Dr. Lea Frermann. In 2021, he got his bachelor's degree in Information Security from Shanghai Jiao Tong University. His interest lies in Natural Language Processing, Drug Discovery, and Recommender Systems. He has published innovative works in top-tier conferences such as IJCAI and ACL. For more information, please visit https://phenixace.github.io/.

Yunqing Liu is currently a PhD student of the Department of Computing (COMP), Hong Kong Polytechnic University (PolyU), under the supervision of Dr. Wenqi Fan. Before joining the PolyU, he received his Master's degree in Computer Science from the University of Edinburgh (M.Sc. in Computer Science), under the supervision of Dr. Elizabeth Polgreen. In 2020, he got his bachelor's degrees from Wuhan University (B.Sc. in Chemistry and B.Eng. in Computer Science and Technology). His research interest includes Drug Discovery, Graph Neural Networks, and Natural Language Processing. He has published innovative works in top-tier conferences and journals such as IJCAI, EACL, EurJOC and Organic Letters. For more information, please visit https://liuyunqing.github.io/.

Xiaowei Mei received his PhD in Information Systems and Operations Management from the University of Florida. His current research aims to extend standard economic models of information systems in two directions: differentiating various forms of social contagion or peer effects in online and offline networks using empirical methods and big data analytic skills; and designing optimal market mechanisms in information systems using game theory, statistics and simulations methods. His work has been accepted by leading journals such as the Journal of Management Information Systems.

Yiqi Wang is an assistant professor at College of Computer, National University of Defense Technology (NUDT). She is currently working on graph neural networks including fundamental algorithms, robustness and their applications. She has published innovative works in top-tier conferences such as ICML, KDD, WWW, EMNLP, WSDM, and AAAI. She serves as top-tier conference program committee members (e.g., WWW, AAAI, IJCAI, CIKM, and WSDM) and journal reviewers (e.g., TIST, TKDD, TKDE and TOIS). She also serves as the leading tutor of tutorials in top-tier conferences (e.g., KDD 2020, AAAI2021, SDM 2021, KDD 2021 and ICAPS 2021).
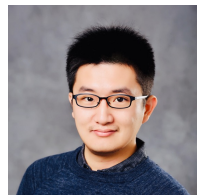
Zhen Wen is a Sr. Applied Science Manager at Amazon Prime Video, leading science efforts in video search, recommendation and promotions. chief scientist of Tencent news feeds product, serving more than 200 million users each day. Dr. Wen directs a team of AI scientists and engineers aiming at deep content understanding, to source and push content users find most relevant and interesting. Prior to his current role, he directed a team of AI scientists and engineers aiming at deep content understanding for short-form video recommendation at Tencent. He also held various science and technology roles at Alibaba Cloud, Google and IBM Research. Dr. Wen received PhD from University of Illinois at Urbana-Champaign. His work received best paper awards at International Conference On Information Systems and ACM Conference on Intelligent User Interfaces. Dr. Wen also received multiple Tencent Outstanding RD Award, IBM Outstanding Innovation Award, IBM Research Accomplishment Award, IBM invention achievement award. Dr. Wen served as an Associate Editor of IEEE Transactions on Multimedia.

Fei Wang is head of personalization science at Amazon Prime Video responsible for improving user's experience and engagement by developing a deep understanding of our customers and providing relevant, personalized and timely recommendations. Previously, he was a senior director with Visa Research leading a group of AI researchers to work on projects ranging from personalized restaurant recommendations, and fraud reduction, to credit risk prevention. With 50+ patents and 50+ research articles, he is also known for research on financial data mining, mobile healthcare, social computing and multimodal information retrieval. He has received a number of best paper awards from conferences like RecSys, Multimedia Information Retrieval and Computers in Cardiology.

Xiangyu Zhao is an assistant professor of the school of data science at City University of Hong Kong (CityU). Before CityU, he completed his Ph.D. at Michigan State University. His current research interests include data mining and machine learning, especially on Reinforcement Learning and its applications in Information Retrieval. He has published papers in top conferences (e.g., KDD, WWW, AAAI, SIGIR, ICDE, CIKM, ICDM, WSDM, RecSys, ICLR) and journals (e.g., TOIS, SIGKDD, SIGWeb, EPL, APS). His research received ICDM'21 Best-ranked Papers, Global Top 100 Chinese New Stars in AI, CCF-Tencent Open Fund, Criteo Research Award, and Bytedance Research Award. He serves as top data science conference (senior) program committee members and session chairs (e.g., KDD, AAAI, IJCAI, ICML, ICLR, CIKM), and journal reviewers (e.g., TKDE, TKDD, TOIS, CSUR). He is the organizer of DRL4KDD@KDD'19, DRL4IR@SIGIR'20, 2nd DRL4KD@WWW'21, 2nd DRL4IR@SIGIR'21, and a lead tutor at WWW'21 and IJCAI'21. More information about him can be found at https://zhaoxyai.github.io/.

Jiliang Tang is a University Foundation Professor in the computer science and engineering department at Michigan State University since 2022. He was an associate professor (2021-2022) and an assistant professor (2016-2021) in the same department. Before that, he was a research scientist in Yahoo Research and got his PhD from Arizona State University in 2015 under Dr. Huan Liu. His research interests include graph machine learning, trustworthy AI and their applications in education and biology. He was the recipient of various awards including 2022 AI's 10 to Watch, 2022 IAPR J. K. AGGARWAL Award, 2022 SIAM/IBM Early Career Research Award, 2021 IEEE ICDM Tao Li Award, 2021 IEEE Big Data Security Junior Research Award, 2020 ACM SIGKDD Rising Star Award, 2020 Distinguished Withrow Research Award, 2019 NSF Career Award, and 8 best paper awards (or runner-ups). His dissertation won the 2015 KDD Best Dissertation runner up and Dean's Dissertation Award. He serves as conference organizers (e.g., KDD, SIGIR, WSDM and SDM) and journal editors (e.g., TKDD, TOIS and TKDE). He has published his research in highly ranked journals and top conference proceedings, which have received tens of thousands of citations with h-index 82 (Google Scholar) and extensive media coverage. More details about him can be found at https://www.cse.msu.edu/~tangjili/.

Qing Li received the B.Eng. degree from Hunan University, Changsha, China, and the M.Sc. and Ph.D. degrees from the University of Southern California, Los Angeles, all in computer science. He is currently a Chair Professor (Data Science) and the Head of the Department of Computing, the Hong Kong Polytechnic University. He is a Fellow of IEEE and IET, a member of ACM SIGMOD and IEEE Technical Committee on Data Engineering. His research interests include object modeling, multimedia databases, social media, and recommender systems. He has been actively involved in the research community by serving as an associate editor and reviewer for technical journals, and as an organizer/co-organizer of numerous international conferences. He is the chairperson of the Hong Kong Web Society, and also served/is serving as an executive committee (EXCO) member of IEEE-Hong Kong Computer Chapter and ACM Hong Kong Chapter. In addition, he serves as a councilor of the Database Society of Chinese Computer Federation (CCF), a member of the Big Data Expert Committee of CCF, and is a Steering Committee member of DASFAA, ER, ICWL, UMEDIA, and WISE Society.