

# How Can Recommender Systems Benefit from Large Language Models: A Survey

JIANGHAO LIN\*, Shanghai Jiao Tong University, China

XINYI DAI\*, Noah's Ark Lab, Huawei, China

YUNJIA XI, Shanghai Jiao Tong University, China

WEIWEN LIU and BO CHEN, Noah's Ark Lab, Huawei, China

HAO ZHANG and YONG LIU, Noah's Ark Lab, Huawei, Singapore

CHUHAN WU and XIANGYANG LI, Noah's Ark Lab, Huawei, China

CHENXU ZHU and HUIFENG GUO, Noah's Ark Lab, Huawei, China

YONG YU, Shanghai Jiao Tong University, China

RUIMING TANG†, Noah's Ark Lab, Huawei, China

WEINAN ZHANG†, Shanghai Jiao Tong University, China

With the rapid development of online services and web applications, recommender systems (RS) have become increasingly indispensable for mitigating information overload and matching users' information needs by providing personalized suggestions over items. Although the RS research community has made remarkable progress over the past decades, conventional recommendation models (CRM) still have some limitations, *e.g.*, lacking open-domain world knowledge, and difficulties in comprehending users' underlying preferences and motivations. Meanwhile, large language models (LLM) have shown impressive general intelligence and human-like capabilities for various natural language processing (NLP) tasks, which mainly stem from their extensive open-world knowledge, logical and commonsense reasoning abilities, as well as their comprehension of human culture and society. Consequently, the emergence of LLM is inspiring the design of recommender systems and pointing out a promising research direction, *i.e.*, whether we can incorporate LLM and benefit from their common knowledge and capabilities to compensate for the limitations of CRM. In this paper, we conduct a comprehensive survey on this research direction, and draw a bird's-eye view from the perspective of the whole pipeline in real-world recommender systems. Specifically, we summarize existing research works from two orthogonal aspects: where and how to adapt LLM to RS. For the "**WHERE**" question, we discuss the roles that LLM could play in different stages of the recommendation pipeline, *i.e.*, feature engineering, feature encoder, scoring/ranking function, user interaction, and pipeline controller. For the "**HOW**" question, we investigate the training and inference strategies, resulting in two fine-grained taxonomy criteria, *i.e.*, whether to tune LLM or not during training, and whether to involve conventional recommendation models for inference. Detailed analysis and general development paths are provided for both "WHERE" and "HOW" questions, respectively. Then, we highlight the key challenges in adapting LLM to RS from three aspects, *i.e.*, efficiency, effectiveness, and ethics. Finally, we summarize the survey and discuss the future prospects. To further facilitate the research community of LLM-enhanced recommender systems, we actively maintain a GitHub repository for papers and other related resources in this rising direction<sup>1</sup>.

CCS Concepts: • **Information systems** → **Recommender systems**.

<sup>1</sup><https://github.com/CHANGEL/Awesome-LLM-for-RecSys>

\* Jianghao Lin and Xinyi Dai are the co-first authors.

† Ruiming Tang and Weinan Zhang are the co-corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

Additional Key Words and Phrases: Recommender Systems, Large Language Models

#### ACM Reference Format:

Jianghao Lin\*, Xinyi Dai\*, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang<sup>†</sup>, and Weinan Zhang<sup>‡</sup>. 2018. How Can Recommender Systems Benefit from Large Language Models: A Survey. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 45 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 INTRODUCTION

With the rapid development of online services, recommender systems (RS) have become increasingly important to match users' information needs [25, 41] and mitigate information overload [49, 110]. They offer personalized suggestions across diverse domains such as e-commerce [172], movie [48], music [179], etc. Despite the varied forms of recommendation tasks (e.g., top- $N$  recommendation, and sequential recommendation), the common learning objective for recommender systems is to estimate a given user's preference towards each candidate item, and finally arrange a ranked list of items presented to the user [108, 227].

Despite the remarkable progress of conventional recommender systems over the past decades, their recommendation performance is still suboptimal, hampered by two major drawbacks as follows: (1) Conventional recommender systems are domain-oriented systems generally built based on discrete ID features within specific domains [228]. Therefore, they lack open-domain world knowledge to obtain better recommendation performance (e.g., enhancing user interest modeling and item content understanding), and transferring abilities across different domains and platforms [13, 51, 119]. (2) Conventional recommender systems often aim to optimize specific user feedback such as clicks and purchases in a data-driven manner, where the user preference and underlying motivations are often implicitly modeled based on user behaviors collected online. As a result, these systems might lack recommendation explainability [11, 43], and cannot fully understand the complicated and volatile intent of users in various contexts. Moreover, users cannot actively guide the recommender system to follow their requirements and customize recommendation results by providing detailed instructions in natural language [39, 205, 208].

With the emergence of large foundation models in recent years, they provide promising and universal insights when handling many challenging problems in the data mining field [12, 186]. A representative form is the large language model (LLM), which has shown impressive general intelligence in various language processing tasks due to their huge memory of open-world knowledge, the ability of logical and commonsense reasoning, and the awareness of human society and culture [7, 66, 262]. By using natural language as a universal information carrier, knowledge in different forms, modalities, domains, and platforms can be generally integrated, exploited, and interpreted. Consequently, the rise of large language models is inspiring the design of recommender systems, *i.e.*, whether we can incorporate LLM and benefit from their common knowledge to address the aforementioned ingrained drawbacks of conventional recommender systems.

Recently, RS researchers and practitioners have made many pioneer attempts to employ LLM in current recommendation pipelines, and have achieved notable progress in boosting the performance of different canonical recommendation processes such as feature modeling [228] and ranking [3]. A few recent surveys also summarize the current state of this field, mainly from the perspective of how to adapt LLM (e.g., pretraining, finetuning, and prompting) [38, 224, 231] in specific modules for prediction or explanation [11, 90]. However, it still lacks a bird's-eye view of how recommender systems can embrace large language models, which is essential in building a technique map to systematically guide the research, practice, and service in LLM-empowered recommendation.

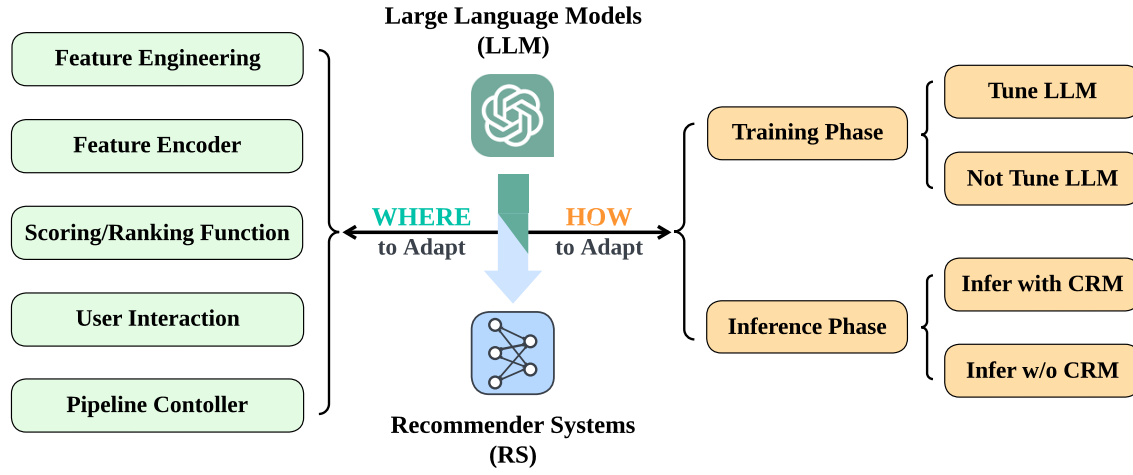


Fig. 1. The decomposition of our core research question about adapting large language models to recommender systems. We analyze the question from two orthogonal perspectives: (1) where to adapt LLM, and (2) how to adapt LLM. Note that CRM stands for conventional recommendation model.

Different from existing surveys on this topic, in this paper, we propose a systematic view of the LLM-enhanced recommendation, from the angle of the whole pipeline in industrial recommender systems. LLM is currently utilized in various stages of recommendation systems and are integrated with current systems via different techniques. To conduct a comprehensive review of latest research progress, as shown in Figure 1, we propose research questions about LLM-enhanced recommender systems from the following two perspectives:

- **“WHERE”** question focuses on where to adapt LLM for RS, and discusses the roles that LLM could play at different stages of current recommender system pipeline, *i.e.*, feature engineering, feature encoder, scoring/ranking function, user interaction, and pipeline controller.
- **“HOW”** question centers on how to adapt LLM for RS, where two orthogonal taxonomy criteria are carried out: (1) whether we will freeze the parameters of the large language model during the training phase, and (2) whether we will involve conventional recommendation models (CRM) during the inference phase.

From the two perspectives, we propose feasible and instructive suggestions for the evolution of existing online recommendation platforms in the era of large language models<sup>23</sup>.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the background and preliminary for recommender systems and large language models. Section 3 and Section 4 thoroughly analyze the aforementioned taxonomies from two perspectives (*i.e.*, **“WHERE”** and **“HOW”**), followed by detailed discussion and analysis of the general development path. In Section 5, we highlight the key challenges and future directions for the adaption of LLM to RS from three aspects (*i.e.*, efficiency, effectiveness, and ethics), which mainly arise from the real-world applications of recommender systems. Finally, Section 6 concludes this survey and draws a hopeful vision for future prospects in research communities of LLM-enhanced recommender systems. Furthermore, we give a comprehensive look-up table

<sup>2</sup>To provide a thorough survey and a clear development path, we broaden the scope of large language models, and bring those relatively smaller language models (e.g., BERT [28], GPT2 [158]) into the discussion as well.

<sup>3</sup>We focus on works that leverage LLM together with their pretrained parameters to handle textual features via prompting, and exclude works that simply apply pretraining paradigms from NLP domains to pure ID-based traditional recommendation models (e.g., BERT4Rec [181]). Interested readers can refer to [118, 240].

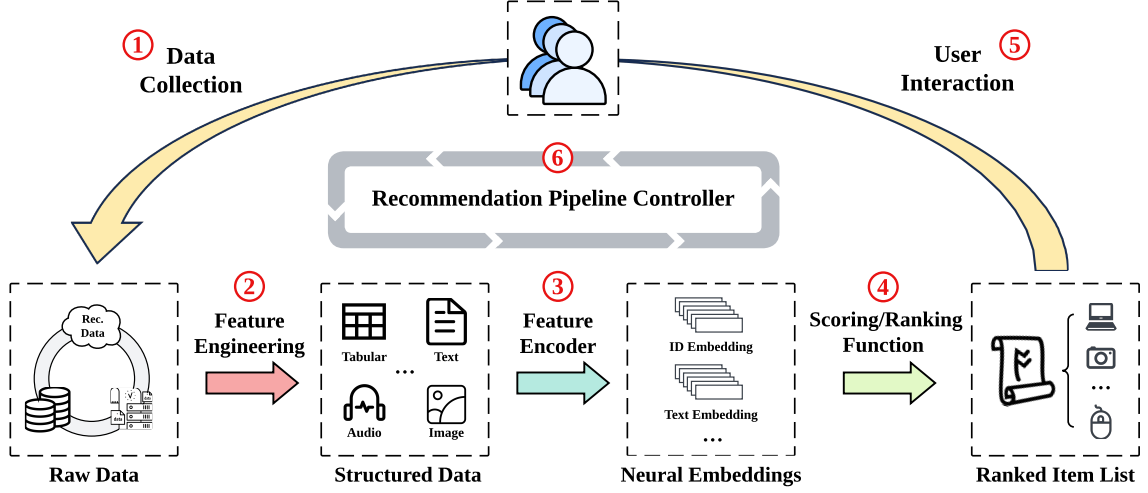


Fig. 2. The illustration of deep learning based recommender system pipeline. We characterize the modern recommender system as an information cycle that consists of six stages: data collection, feature engineering, feature encoder, scoring/ranking function, user interaction, and recommendation pipeline controller, which are denoted by different colors.

of related works that adapt LLM to RS in Appendix A (*i.e.*, Table 1), attaching the detailed information for each work, *e.g.*, the stage that LLM is involved in, LLM backbone, and LLM tuning strategy, etc.

## 2 BACKGROUND AND PRELIMINARY

Before elaborating on the detail of our survey, we would like to introduce the following background and basic concepts: (1) the general pipeline of modern recommender systems based on deep learning techniques, and (2) the general workflow and concepts for large language models.

### 2.1 Modern Recommender Systems

The core task of recommender systems is to provide a ranked list of items  $[i_k]_{k=1}^N$ ,  $i_k \in \mathcal{I}$  for the user  $u \in \mathcal{U}$  given a certain context  $c$ , where  $\mathcal{I}$  and  $\mathcal{U}$  are the universal sets of items and users, respectively. Note that scenarios like next item prediction are special cases for such a formulation with  $N = 1$ . We denote the goal as follows:

$$[i_k]_{k=1}^N \leftarrow \text{RS}(u, c, \mathcal{I}), u \in \mathcal{U}, i_k \in \mathcal{I}. \quad (1)$$

As shown in Figure 2, the modern deep learning based recommender systems can be characterized as an information cycle that encompasses six key stages: (1) Data Collection, where the users' feedback data is gathered; (2) Feature Engineering, which involves preparing and processing the collected raw data; (3) Feature Encoder, where data features are transformed into neural embeddings; (4) Scoring/Ranking Function, which selects and orders the recommended items; (5) User Interaction, which determines how users engage with the recommendations; and finally, (6) Recommendation Pipeline Controller, which serves as the central mechanism tying all the stages above together in a cohesive process. Next, we will briefly go through each of the stages as follows:

- **Data Collection.** The data collection stage gathers both explicit and implicit feedback from online services by presenting recommended items to users. The explicit feedback indicates direct user responses such as ratings, while

the implicit feedback is derived from user behaviors like clicks, downloads, and purchases. In addition to gathering user feedback, the data to be collected also encompasses a range of raw features including item attributes, user demographics, and contextual information. The collected raw data is stored in the database in certain formats such as JSON, ready for further processing.

- **Feature Engineering.** Feature engineering is the process of selecting, manipulating, transforming, and augmenting the raw data collected online into structured data that is suitable as inputs of neural recommendation models. As shown in Figure 2, the major outputs of feature engineering consist of various forms of features, which will be then encoded by feature encoders of different modalities, *e.g.*, language models for textual features, vision models for visual features, and conventional recommendation models (CRM) for ID features.
- **Feature Encoder.** Generally speaking, the feature encoder takes as input the processed features from the feature engineering stage, and generates the corresponding neural embeddings for scoring/ranking functions in the next stage. Various encoders are employed depending on the data modality. Typically, this process is executed as an embedding layer for one-hot encoded categorical features in standard recommendation models. Features of other modalities, such as text, vision, video, or audio, are further used and encoded to enhance content understanding.
- **Scoring/Ranking function.** Scoring/Ranking function serves as the core part of recommendation to select or rank the top-relevant items to satisfy users' information needs based on the neural embeddings generated by the feature encoders. Researchers develop various neural methods to precisely estimate the user preference and behavior patterns based on various techniques, *e.g.*, collaborative filtering [54, 180], sequential modeling [17, 122], graph neural networks [200, 209], etc.
- **User Interaction.** User interaction refers to the way we represent the recommended items to the target user, and the way users give their feedback back to the recommender system. While traditional recommendation pages basically involve a single list of items, various complex and multi-modal scenarios are recently proposed and studied [245]. For example, conversational recommendation provides natural language interface and enables multi-round interactive recommendation for the user [184]. Besides, multi-block page-level user interactions are also widely considered for nested user feedback [41, 168].
- **Recommendation Pipeline Control.** Pipeline controller monitors and controls the operations of the whole recommendation pipeline mentioned above. It can even provide fine-grained control over different stages for recommendation (*e.g.*, matching, ranking, reranking), or decide to combine different downstream models and APIs to accomplish the final recommendation tasks.

## 2.2 Large Language Models

Language models aim to conduct the probabilistic modeling of natural languages to predict the word tokens given a specific textual context. Nowadays, most language models are built based on transformer-like [194] architectures to proficiently model the context dependency for human languages. They are first pretrained on a massive amount of unlabeled text data, and then further finetuned with task-oriented data for different downstream applications. These pretrained language models (PLM) can be mainly classified into three categories: encoder-only models like BERT [28], decoder-only models like GPT [158], and encoder-decoder models like T5 [159].

Large language models (LLM) are the scaled-up derivatives of traditional pretrained language models mentioned above, in terms of both model sizes and data volumes, *e.g.*, GPT-3 [7], PaLM [19], LLaMA [192], ChatGLM [36, 248]. A typical LLM usually consists of billion-level or even trillion-level parameters, and is pretrained on much larger volumes of textual corpora with up-to trillions of tokens crawled from various Internet sources like Wikipedia, GitHub,

ArXiv, etc. As illustrated by the scaling law [59, 80], the scaling up of model size, data volume and training scale can continuously contribute to the growth of model performance for a wide range of downstream NLP tasks. Furthermore, researchers find that LLM can exhibit emergent abilities, e.g., few-shot in-context learning, instruction following and step-by-step reasoning, when the model size continues to scale up and reaches a certain threshold [217]

LLM has revolutionized the field of NLP by demonstrating impressive capabilities in understanding natural languages and generating human-like texts. Moreover, LLM has gone beyond the field of NLP and shown remarkable potential in various deep learning based applications, such information system [272], education [92], finance [225] and health-care [142, 187]. Therefore, recent studies start to investigate the application of LLM to recommender systems. Equipped with the extensive open-world knowledge and powerful emergent abilities like reasoning, LLM is able to analyze the individual preference based on user behavior sequences, and promote the content understanding and expansion for items, which can largely enhance the recommendation performance [3, 23, 228, 235]. Besides, LLM can also support more complex scenarios like conversational recommendation [43], explainable recommendation [11], as well as task decomposition and tool usage (e.g., search engines) [213] for recommendation enhancements.

### 3 WHERE TO ADAPT LARGE LANGUAGE MODELS

Based on the decomposition of modern recommender systems discussed in Section 2.1, we answer the “WHERE” question by elaborating on the adaptation of LLM to different parts of the recommendation pipeline: (1) feature engineering, (2) feature encoder, (3) scoring/ranking function, (4) user interaction, and (5) pipeline controller. It is worth noting that, the utilization of LLM in the same research work may involve multiple stages of the recommendation pipeline due the multi-task nature of LLM. For example, LLM is leveraged in both stages of feature engineering and scoring/ranking function in CUP [191].

#### 3.1 LLM for Feature Engineering

In the feature engineering stage, LLM takes as inputs the original features (e.g., item descriptions, user profiles, and user behaviors), and generates auxiliary textual features for data augmentation with varied goals, e.g., enriching the training data, alleviating the long-tail problem, etc. Different prompting strategies are employed to make full use of the open-world knowledge and reasoning ability exhibited by LLM. According to the type of data augmentation, the research works of this line can be mainly classified into two categories: (1) user- and item-level feature augmentation, (2) instance-level training sample generation.

**3.1.1 User- and Item-level Feature Augmentation.** Equipped with powerful reasoning ability and open-world knowledge, LLM is often treated as a flexible knowledge base [130]. Hence, it can provide auxiliary features for better user preference modeling and item content understanding. As a representative, KAR [228] adopts LLM to generate the user-side preference knowledge and item-side factual knowledge, which serve as the plug-in features for downstream conventional recommendation models. TF-DCon [222] leverages LLM to compress and condensate the training data from views of both user history and item content. SAGCN [114] introduces a chain-based prompting approach to uncover semantic aspect-aware interactions, which provides clearer insights into user behaviors at a fine-grained semantic level. CUP [191] adopts ChatGPT to summarize each user’s interests with a few short keywords according to the user review texts. In this way, the user profiling data is condensed within 128 tokens and thus can be further encoded with small-scale language models that are constrained by the context windows size (e.g., 512 for BERT [28]). Moreover, instead of using a frozen LLM for feature augmentation, LLaMA-E [173] and EcomGPT [103] finetune the base large

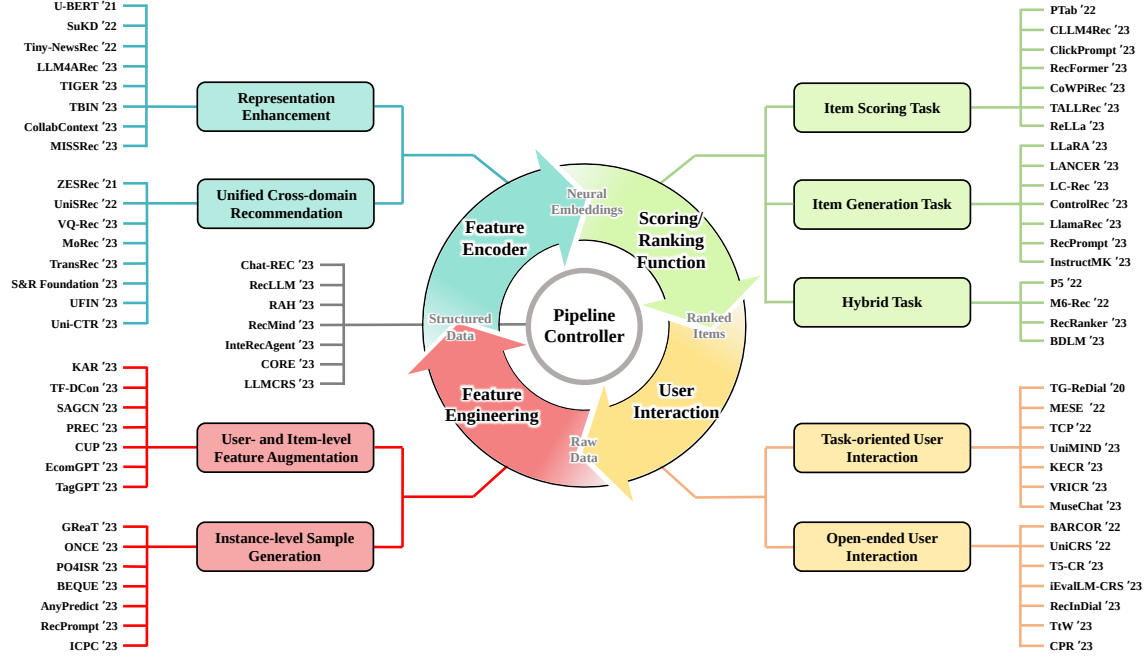


Fig. 3. The illustrative dissection of the “WHERE” research question. We show that LLM can be adapted to different stages of the recommender system pipeline as introduced in Section 2.1, *i.e.*, feature engineering, feature encoder, scoring/ranking function, user interaction, and pipeline controller. We provide finer-grained classification criteria for each stage, and list representative works denoted by different colors.

language models for various downstream generative tasks in e-commerce scenarios, *e.g.*, product categorization and intent speculation. Other works also utilize LLM to further enrich the training data from different perspectives, *e.g.*, [text refinement](#) [35, 127, 264], [knowledge graph completion and reasoning](#) [13, 22, 212, 219], [attribute generation](#) [6, 85, 238], and [user interest modeling](#) [20, 33, 132, 166].

**3.1.2 Instance-level Sample Generation.** Apart from feature-level augmentations, LLM is also leveraged to generate synthetic samples, which enrich the training dataset [141] and improve the model prediction quality [113, 185]. GReaT [5] tunes a generative language model to synthesize realistic tabular data as augmentations for the training phase. Carranza et al. [10] explore to train a [differentially private \(DP\)](#) large language model for synthetic user query generation, in order to address the privacy problem in recommender systems. ONCE [119] applies manually designed prompts to obtain additional news summarization, user profiles, and synthetic news pieces for news recommendation. AnyPredict [215] leverages LLM to consolidate datasets with different feature fields, and align out-domain datasets for a shared target task. Zhang et al. [250] further attempt to incorporate multiple large language models as agents to simulate the fine-grained user communication and interaction for more realistic recommendation scenarios. Moreover, RecPrompt [113] and PO4ISR [185] propose to perform automatic prompt template optimization with powerful LLM (*e.g.*, ChatGPT or GPT4), and therefore iteratively improve the recommendation performance with gradually better textual inputs for LLM-based recommenders. BEQUE [146] finetunes and deploys LLM for query rewriting in e-commercial scenarios to bridge the semantic gaps inherent in the semantic matching process, especially for long-tail queries. Li et al. [97] use



Chain-of-Thought [218] (CoT) technology to leverage LLM as agent to emulate various demographic profiles for robust and efficient query rewriting.

### 3.2 LLM as Feature Encoder

In conventional recommender systems, the structured data are usually formulated as one-hot encodings, and an embedding layer is adopted as the feature encoder to obtain dense embeddings. With the emergence of language models, researchers propose to adopt LLM as auxiliary textual feature encoder to gain two major benefits: (1) further enriching the user/item representations with semantic information for the later neural recommendation models; (2) achieving cross-domain<sup>4</sup> recommendation with natural language as the bridge, where ID feature fields might not be shared.

**3.2.1 Representation Enhancement.** For item representation enhancement, LLM is leveraged as feature encoder for scenarios with abundant textual features available (e.g., item title, body text, detailed description), including but not limited to: document ranking [124, 275], news recommendation [120, 167, 220, 221, 241, 254], tweet search [257], tag selection [52], nudge marketing [151], software purchase [77], social networking [72], code example recommendation [162], tour itinerary recommendation [58], and other general recommendation scenarios [14, 51, 145, 195, 198, 203]. While the item content is generally static, the user interest is highly dynamic and keeps evolving over time, therefore requiring sequential modeling over the fast-evolving user behaviors and underlying preferences [78, 148, 267]. For example, U-BERT [154] ameliorates the user representation by encoding review texts into a sequence of dense vectors via BERT [28], followed by specially designed attention networks for user interest modeling. LLM4ARec [91] uses GPT2 [158] to extract personalized aspect terms and latent vectors from user profiles and reviews to better assist recommendations. In some special cases, the semantic representation encoded by LLM is not directly used as the input for the later scoring/ranking function. Instead, it is converted into a sequence of discrete tokens through quantization to adapt to scoring/ranking functions that require discrete inputs (e.g., generative recommendation). TIGER [163] proposes to apply vector quantization techniques [193, 239, 247] over the semantic item representations to further compress each item into a tuple of discrete semantic tokens. Hence, the sequential recommendation can be expressed as a sequence modeling task over a list of discrete tokens, where classical transformer [194] architectures can be employed. Based on the idea of item vector quantization, LMIndexer [75] designs a self-supervised semantic indexing framework to capture the item’s semantic representation and the corresponding semantic tokens at the same time in an end-to-end manner.

**3.2.2 Unified Cross-domain Recommendation.** Apart from the user/item representation improvement, adopting LLM as feature encoder also enables transfer learning and cross-domain recommendation, where natural language serves as the bridge to align the heterogeneous information from different domains [93, 102, 202]. ZESRec [31] applies BERT [28] to convert item descriptions into universal semantic representations for zero-shot recommendation. In UniSRec [61], the item representations are learned for cross-domain sequential recommendation via a fixed BERT model followed by a lightweight MoE-enhanced network. Built upon UniSRec, VQ-Rec [60] introduces vector quantization techniques to better align the textual embeddings generated by LLM to the recommendation space. Uni-CTR [42] leverages layer-wise semantic representations from a shared LLM to sufficiently capture the commonalities among different domains, which leads to better multi-domain recommendation. Other works [47, 189] leverage unified cross-domain textual embeddings from a fixed LLM (e.g., ChatGLM [36], Sheared-LLaMA [229]) to tackle scenarios with cold-start users/items

<sup>4</sup>Different domains means data sources with different distributions, e.g., scenarios, datasets, platforms, etc.



or low-frequency long-tail features. Fu et al. [40] further explore layerwise adapter tuning on large language models to obtain better embeddings over textual features from different domains.

### 3.3 LLM as Scoring/Ranking Function

The ultimate goal of the scoring/ranking stage is highly tied with the general purpose of recommender systems as discussed in Section 2.1, *i.e.*, to provide a ranked list of items  $[i_k]_{k=1}^N, i_k \in \mathcal{I}$  for target user  $u \in \mathcal{U}$ , where  $\mathcal{I}$  and  $\mathcal{U}$  are the universal set of items and users (next item prediction is a special case where  $N = 1$ ). When directly adapting LLM as the scoring/ranking function, such a goal could be achieved through various kinds of tasks for LLM (*e.g.*, rating prediction, item ID generation). According to different tasks that LLM solves, we classify related research works into three categories: (1) item scoring task, (2) item generation task, and (3) hybrid task.

**3.3.1 Item Scoring Task.** In item scoring tasks, the large language model serves as a pointwise function  $F(u, i), \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$ , which estimates the utility score of each candidate item  $i$  for the target user  $u$ . Here  $\mathcal{U}$  and  $\mathcal{I}$  denote the universal set of users and items, respectively. The final ranked list of items is obtained by sorting the utility score calculated between the target user  $u$  and each item  $i$  in the candidate set  $C$ :

$$\begin{aligned} C &\leftarrow \text{Pre-filter}(u, \mathcal{I}), \\ [i_k]_{k=1}^N &\leftarrow \text{Sort}(\{F(u, i) \mid \forall i \in \mathcal{I}\}), N \leq |C|, \end{aligned} \quad (2)$$

where  $C$  is the candidate set obtained via a pre-filter function (*e.g.*, the retrieval and pre-ranking models for the ranking stage). The pre-filtering is conducted to reduce the number of candidate items, thus saving the computational cost. The pre-filter can be an identity-mapping function (*i.e.*,  $C = \mathcal{I}$ ) for the first retrieval stage for recommender systems.

Without loss of generality, the large language model takes as inputs the discrete tokens of textual prompt  $x$ , and generates the target token  $\hat{t}$  as the output for either the masked token in masked language modeling or the next token in causal language modeling. The process can be formulated as follows:

$$\begin{aligned} h &= \text{LLM}(x), \\ s &= \text{LM\_Head}(h) \in \mathbb{R}^V, \\ p &= \text{Softmax}(s) \in \mathbb{R}^V, \\ \hat{t} &\sim p, \end{aligned} \quad (3)$$

where  $h$  is the final representation,  $V$  is the vocabulary size, and  $\hat{t}$  is the predicted token sampled from the probability distribution  $p$ .

However, the item scoring task requires the model to do pointwise scoring for a given user-item pair  $(u, i)$ , and the output should be a real number  $\hat{y} = F(u, i)$ , instead of generated discrete tokens  $\hat{t}$ . The output  $\hat{y}$  should fall within a certain numerical range to indicate the user preference, *e.g.*,  $\hat{y} \in [0, 1]$  for click-through rate (CTR) estimation and  $\hat{y} \in [0, 5]$  for rating prediction. There are three major approaches to address such an issue that the output requires continuous numerical values while LLM produces discrete tokens.

The first type of solution [64, 79, 81, 96, 107, 115, 197, 199, 260, 271, 274] adopts the single-tower paradigm [155, 256]. To be specific, they directly abandon the language modeling decoder head (*i.e.*,  $\text{LM\_Head}(\cdot)$ ), and feed the final representation  $h$  of LLM in Eq. 3 into a delicately designed projection layer to calculate the final score  $\hat{y}$  for classification or regression tasks, *i.e.*,

$$\hat{y} = F(u, i) = \text{MLP}(h), \quad (4)$$

where MLP (short for multi-layer perceptron) is the projection layer. The input prompt  $x$  needs to contain information from both the user  $u$  and item  $i$  to support the preference estimation based on one single latent representation  $h$ . CoLLM [260] and E4SRec [96] construct personalized prompts with the help of pre-learned user & item ID embeddings for precise preference estimation. FLIP [199] and ClickPrompt [107] propose to conduct fine-grained knowledge alignment and fusion over the semantic and collaborative information in parallel and stacking paradigms, respectively. CER [157] reinforces the coherence between recommendations and their natural language explanations to improve the rating prediction performance. Kang et al. [79] finetune the large language model for rating prediction in a regression manner, which exhibits a surprising performance by scaling the model size of finetuned LLM up to 11 billion. Other typical examples in this line of research include: LSAT [174], BERT4CTR [197], CLLM4Rec [271], and PTab [115].

Similar to the first method, the second type of solution [87, 128, 188, 190, 191, 234] also discards the decoder head of LLM. However, what sets it apart is that it adopts the popular two-tower structure [53, 54, 209] in conventional recommender systems. They maintain both two separate towers to obtain the representations for user and item respectively, and the preference score is calculated via a certain distance metric between the two representations:

$$\hat{y} = F(u, i) = d(T_u(x_u), T_i(x_i)), \quad (5)$$

where  $d(\cdot, \cdot)$  is the distance metric function (e.g., cosine similarity, L2 distance).  $T_u(\cdot)$  and  $T_i(\cdot)$  are the user and item towers that consist of LLM backbones to extract the useful knowledge representations from both user and item texts (i.e.,  $x_u$  and  $x_i$ ). In this line of works, different auxiliary structures are designed to augment the dual-side information with LLM. For example, CoWPiRec [234] applies word graph neural networks to item texts within the user behavior sequence to amplify the semantic information correlation. By employing the encoder-decoder LLM, TASTE [128] first encodes each user behavior into a soft prompt vector and then leverages the decoder to extract the user preference from the sequence of soft prompts. Other typical examples include: RecFormer [87], LLM-Rec [188], and CUP [191].

Different from the aforementioned two solutions that both replace the original language modeling decoder head (i.e., LM\_Head( $\cdot$ )) with manually designed predictive modules, the last type of solution [3, 56, 57, 111, 130, 135, 149, 156, 176, 182, 223, 226, 259, 261, 265, 273] proposes to preserve the decoder head and perform preference estimation based on the probability distribution  $p \in \mathbb{R}^V$ . TALLRec [3], ReLLa [111], PromptRec [226], BTRec [57] and CR-SoRec [149] append a binary question towards the user preference after the textual description of user profile, user behaviors, and target item, and therefore convert the item scoring task into a binary question answering problem. Then, they can intercept the estimated score  $s \in \mathbb{R}^V$  or probability  $p \in \mathbb{R}^V$  in Eq. 3 and conduct a bidimensional softmax over the corresponding logits of the binary key answer words (i.e., the token used to denote label, for example, Yes/No) for pointwise scoring:

$$\hat{y} = \frac{\exp(p_{Yes})}{\exp(p_{Yes}) + \exp(p_{No})} \in (0, 1), \quad (6)$$

where  $p_{Yes}$  and  $p_{No}$  denote the logits for “Yes” and “No” tokens, respectively. Other typical examples that extract the softmax probabilities of corresponding label tokens for item scoring include TabLLM [56], Prompt4NR [261], and GLRec [223]. Moreover, another line of research intends to concatenate the item description (e.g., title) to the user behavior history with different templates, and estimates the score by calculating the overall perplexity [135, 156], log-likelihood [171, 176], or joint probability [259] of the prompting text as the final predicted score  $\hat{y}$  for user preference. Besides, Zhiyuli et al. [265] instruct LLM to predict the user rating in a textual manner, and restrict the output format as a value with two decimal places through manually designed prompts.

**3.3.2 Item Generation Task.** In item generation tasks, the large language model serves as a generative function  $F(u)$  to directly produce the final ranked list of items, requiring only one forward of function  $F(u)$ . Generally speaking, the item generation task highly relies on the intrinsic reasoning ability of LLM to infer the user preference and generate the ranked item list, the process of which can be formulated as:

$$[i_k]_{k=1}^N = F(u), \text{ s.t. } i_k \in \mathcal{I}. \quad (7)$$

According to whether a set of candidate items is provided for LLM to accomplish the item generation task, we can categorize the related solutions into two classes: (1) open-set item generation, and (2) closed-set item generation.

In **open-set item generation** tasks [2, 30, 45, 51, 65, 69, 73, 88, 89, 98, 105, 112, 137, 153, 170, 238, 249, 263, 270], LLM is required to directly generate the ranked item list that the user might prefer according to the user profile and behavior history *without a given candidate item set*. Since the candidate items are not provided in the input prompt, the large language model is actually not aware of the universal item pool  $\mathcal{I}$ , thus bringing the *generative hallucination problem* [137], where the generated items might fail to match the exact items in the item pool  $\mathcal{I}$ . Therefore, apart from the design of input prompt templates [62, 100] and finetuning algorithms [89], the *post-processing* operations for *item grounding and matching* after the item generation are also required to overcome the generative hallucination problem [137]. We formulate the process as follows:

$$\begin{aligned} [\hat{i}_k]_{k=1}^N &\leftarrow \text{LLM}(x_u), \\ [i_k]_{k=1}^N &\leftarrow \text{Match}([\hat{i}_k]_{k=1}^N, \mathcal{I}), \end{aligned} \quad (8)$$

where  $\text{Match}(\cdot, \cdot)$  is the matching function,  $\hat{i}_k$  is the LLM-generated items, and  $i_k$  is the actual item matched from  $\mathcal{I}$  according to  $\hat{i}_k$ . LANCER [73] employs knowledge-enhanced prefix tuning for generation ground and further applies cosine similarity to match the encoded representation of generated item text with the universal item pool  $\mathcal{I}$ . Di Palma et al. [30] leverage ChatGPT for user interest modeling and next item title generation with Damerau-Levenshtein distance [138] for item matching.

Apart from generating the items in textual manners, another line of research focuses on *aligning the language space with the ID-based recommendation space*, and therefore enables LLM to *generate the item IDs directly*. For instance, Hua et al. [65] explore better ways for item indexing (e.g., sequential indexing, collaborative indexing) in order to enhance the performance of such index generation tasks. LightLM [137] designs a lightweight LLM with carefully designed user & item indexing, and applies constrained beam search for open-set item ID generation. Besides, LLaRA [105] represents items in LLM’s input prompts using a novel hybrid approach that integrates ID-based item embeddings from traditional recommenders with textual item features. Other typical works for open-set item generation include: GenRec [71], TransRec [112], LC-Rec [263], ControlRec [153], and POD [89].

In **closed-set item generation** tasks [16, 45, 46, 62, 113, 130, 133, 178, 185, 196, 206, 214, 230, 236, 243, 251], LLM is required to *rank or select from a given candidate item set*. That is, we will first employ a lightweight retrieval model to pre-filter the universal item set  $\mathcal{I}$  into a limited number of candidate items denoted as  $C = \{i_j\}_{j=1}^J$ ,  $J \ll |\mathcal{I}|$ . The number of candidate items is usually set up to 20 due to the *context window limitation* of LLM. The content of candidate items is then presented in the input prompt for LLM to generate the ranked item list, which can be formulated as:

$$\begin{aligned} C &\leftarrow \text{Pre-filter}(u, \mathcal{I}), \\ [i_k]_{k=1}^N &\leftarrow \text{LLM}(u, C), \quad N \leq |C|, \end{aligned} \quad (9)$$

For example, [LlamaRec \[243\]](#) adopts [LRURec \[244\]](#) as the retriever, and finetunes LLaMA2 for listwise ranking over the pre-filtered items. [DRDT \[214\]](#) ranks the given candidates with iterative multi-round reflection to gradually refine the ranked list. [LiT5 \[178\]](#) proposes to distill the zero-shot ranking ability from a proficient LLM (e.g., RankGPT4 [183]) into a relatively smaller one (e.g., T5-XL [159]). [AgentCF \[252\]](#) incorporates LLM as the recommender by simulating user-item interactions in recommender systems through agent-based collaborative filtering. Other typical examples include: [JobRecoGPT \[46\]](#), [InstructMK \[196\]](#), [RecPrompt \[113\]](#), [PO4ISR \[185\]](#), etc.

In comparison of these two tasks, open-set generation tasks generally suffer from the generative hallucination problem, where the generated items might fail to match the exact items in the universal item pool. Therefore, the post-generation matching function is heavily required, which increases the inference overhead and might even hurt the final recommendation performance, especially for scenarios with item texts that largely differ from the language distribution of LLM. On the contrary, closed-set generation tasks use a lightweight retrieval model as the pre-filter to provide a clear set of candidate items, and therefore the large language model is able to mitigate the hallucination problem. However, the introduction of candidate items in the input prompt of LLM can cause other problems. Firstly, LLM cannot handle a large number of candidates (usually less than 20) due to the context window limitation, and the final recommendation performance can somehow be limited by the retrieval model (i.e., pre-filter). Moreover, [Ma et al. \[133\]](#) and [Hou et al. \[62\]](#) reveal that shuffling the order of candidate items in the prompt can affect the ranking output, leading to unstable recommendation results. The aforementioned issues of closed-set generation tasks intrinsically stem from the existence of candidate item set in the input prompt, which can be well solved in open-set generation tasks. In summary, we can observe that the open-set and closed-set generation tasks have complementary strengths and weaknesses compared with each other. Hence, the choice between them in practical applications actually depends on specific situations and problems we meet in real-world scenarios.

**3.3.3 Hybrid Task.** In hybrid tasks, the large language model serves in a multi-task manner, where both the item scoring and generation tasks could be handled by a single LLM through a unified language interface. The basis for supporting this hybrid functionality is that large language models are inherent multi-task learners [7, 143]. [P5 \[44\]](#), [M6-Rec \[23\]](#) and [InstructRec \[253\]](#) tune the encoder-decoder models for better alignment towards a series of recommendation tasks including both item scoring and generation tasks via different prompting templates. [RecRanker \[129\]](#) combines the pointwise scoring, pairwise comparison and listwise ranking tasks to explore the potential of LLM for top-N recommendation. [BDLM \[255\]](#) bridges the information gap between the domain-specific models and the general large language models for hybrid recommendation tasks via an information sharing module with memory storage mechanism. Other works [24, 116, 183] manually design task-specific prompts to call a unified central LLM (e.g., ChatGPT API) to perform multiple tasks, including but not restricted to pointwise rating prediction, pairwise item comparison, and listwise ranking list generation. There also exist benchmarks (e.g., [LLMRec \[117\]](#), [OpenP5 \[232\]](#)) that test the LLM-based recommenders on various recommendation tasks like rating prediction, sequential recommendation, and direct recommendation.

### 3.4 LLM for User Interaction

In many of practical applications, recommending is a one-shot interaction process, where the system monitors the user behaviors (e.g., click and purchase) over time and then presents a tailored set of relevant items in certain pre-defined situations. Such a one-turn interaction lacks effective and versatile ways to acquire user interests and detect the user's current situation or needs in complex scenarios. To this end, the advent of large language models presents a promising

alternative, by offering a more active and adaptive form of user interaction. Instead of relying solely on the past user behaviors passively, LLM could engage in real-time interactions with the users to gather more nuanced natural language feedback about their preferences.

In general, the user interaction based on LLM in recommendation is commonly formed as a multi-turn dialogue, which is covered in conversational recommender systems [27, 205, 211, 269]. During such a dialogue, LLM provides an unprecedented richness in understanding users' interests and requirements by integrating context in conversation and applying the extensive open-world knowledge. LLM can support a recommender to make highly relevant and tailored recommendations through eliciting the current preferences of user, providing explanations for the item suggestions, or processing feedback by users on the made suggestions [68]. In other words, the introduction of large language models makes recommender systems more feasible and user-friendly in terms of user interaction. Specifically, from the perspective of interactive content [94, 268], the modes of LLM-based user interaction can be categorized into (1) task-oriented user interaction, and (2) open-ended user interaction .

**3.4.1 Task-oriented User Interaction.** The task-oriented user interaction [27, 165, 201, 233, 258, 269] supposes that the user has a clear intent and the recommender system needs to support the user's decision making process or assist the user in finding relevant items. To be specific, LLM is integrated as a component of the recommender system, specially aiming at analyzing user intentions. As a typical work, TG-ReDial [269] proposes to incorporate topic threads to enforce natural semantic transitions towards the recommendation and develops a topic-guided conversational recommendation method. It deploys three BERT [28] modules to encode user profiles, dialogue history, and track conversation topics, respectively. Then, the encoded features are fed into a pre-set recommendation module to recommend items, followed by a GPT2 [158] module to aggregate the encoded knowledge for response generation. After each turn, the results are gathered and will be used to support the next round of dialogue interaction, such as understanding changes in user interest and analyzing user feedback, etc. The subsequent works roughly follow a similar process for task-oriented user interaction. While earlier works attempt to manage the dialogue understanding and response generation with relatively small language models (e.g., BERT and GPT2), recent works start to incorporate billion-level large language models for better conversational recommendation and improving the satisfaction of user interaction. MuseChat [34] builds a multi-modal LLM based on Vicuna-7B [18] to provide reasonable explanation for the music recommendation during the user dialogue. Liu et al. [126] leverage the complementary collaboration between conversational RS and LLM for e-commercial pre-sales dialogue understanding and generation. He et al. [55] construct a conversational recommendation dataset with more diverse textual contexts, and find that LLM is able to outperform finetuned traditional conversational recommenders in zero-shot settings. Other typical works for task-oriented user interaction include: MESE [233], KEER [165], UniMIND [27], VRICR [258], TCP [201].

**3.4.2 Open-ended User Interaction.** The task-oriented user interaction draws a strong assumption that the user engages in the recommender system with specific goals to seek certain items. Differently, the open-ended user interaction [83, 164, 205, 208, 210, 211] assumes that the user's intent is vague, and the system needs to gradually acquire user interests or guide the user through interactions (including topic dialogue, chitchat, QA, etc.) to achieve the goal of recommendation eventually. Consequently, the role of LLM for open-ended user interaction is no longer limited to a simple component for dialogue encoding and response generation as discussed in Section 3.4.1. Instead, LLM plays a key role in driving the interaction process by leading and acquiring the user interests for final recommendation. Specifically, BARCOR [208] proposes a unified framework based on BART [84] to first conduct user preference elicitation, and then perform response generation with recommended items, which aims to maximize the mutual information between conversation

interaction and item recommendation. T5-CR [164] focuses on user interaction modeling and formulates conversation recommendation as a language generation problem. It adopts T5 [160] to achieve dialogue context understanding, user preference elicitation, item recommendation and response generation in an end-to-end manner. Specifically, it adopts a special token symbol as the trigger to generate recommended item during the response generation. Wang et al. [210] investigate the ability of ChatGPT to converse with user for item recommendation and explanation generation through manually designed prompts without any demonstration (*i.e.*, zero-shot prompting). Then, they utilize LLM as an auxiliary user interaction component for dialogue understanding and user preference elicitation. Other related research works include: UniCRS [211], RecInDial [205], and TtW [83].

### 3.5 LLM for Pipeline Controller

As the model size scales up, LLM tends to exhibit emergent behaviors that may not be observed in previous smaller language models, *e.g.*, few-shot in-context learning, instruction following, step-by-step reasoning, and tool usage [217, 262]. With such emergent abilities, LLM is no longer just a part of the recommender system mentioned above, but could actively participate in the pipeline control over the system, possibly leading to a more interactive and explainable recommendation process [76]. Chat-REC [43] leverages ChatGPT to bridge the conversational interface and traditional recommender systems, where it is required to infer user preferences, decide whether or not to call the backend recommendation API, and further modify (*e.g.*, filter and rerank) the returned item candidates before presenting them to the user. These operations enable LLM to step beyond the role for user interaction in Section 3.4, and cast controls for the multi-stage recommendation pipeline with certain API calls and tool usage for conversational recommender systems. RecLLM [39] further extends the permission of LLM, and proposes a roadmap for building an integrated conversational recommender system, where LLM is able to manage the dialogue, understand user preference, arrange the ranking stage, and even provide a controllable LLM-based user simulator to generate synthetic conversations. RAH [175] designs the Learn-Act-Critic loop and a reflection mechanism for LLM-based agents to improve the alignment with user preferences during the interaction period. InteRecAgent [67] serves as the interactive agent for conversational recommendation with the users, and is accessible to a range of plug-in tools including but not limited to intention detection, information query, item retrieval, and item ranking. Besides, instead of allowing LLM to take over control of the entire recommendation pipeline, RecMind [213] makes finer-grained control over the recommendation process with task deconstruction. It proposes to address the user queries under self-inspiring prompting strategy and multi-step reasoning with tool usage (*e.g.*, expert models, SQL tool, search engine).

### 3.6 Discussion

We could observe that the development path about where to adapt LLM to RS is fundamentally aligned with the progress of large language models. Back in the year 2021 and early days of 2022, the parameter sizes of pretrained language models are still relatively small (*e.g.*, 110M for BERT-base, 1.5B for GPT2-XL). Therefore, earlier works usually tend to either incorporate these small-scale language models as simple textual feature encoders, or as scoring/ranking functions finetuned to fit the data distribution of recommender systems. In this way, the recommendation process is simply formulated as a one-shot straightforward predictive task, and can be better solved with the help of language models.

As the model size gradually increases, researchers discover that large language models have gained emergent abilities (*e.g.*, instruction following and reasoning), as well as a vast amount of open-world knowledge with powerful text generation capacities. Equipped with these amazing features brought by large-scale parameters, LLM starts to not only deepen its usage in the feature encoder and scoring/ranking function stage, but also further extend their roles into

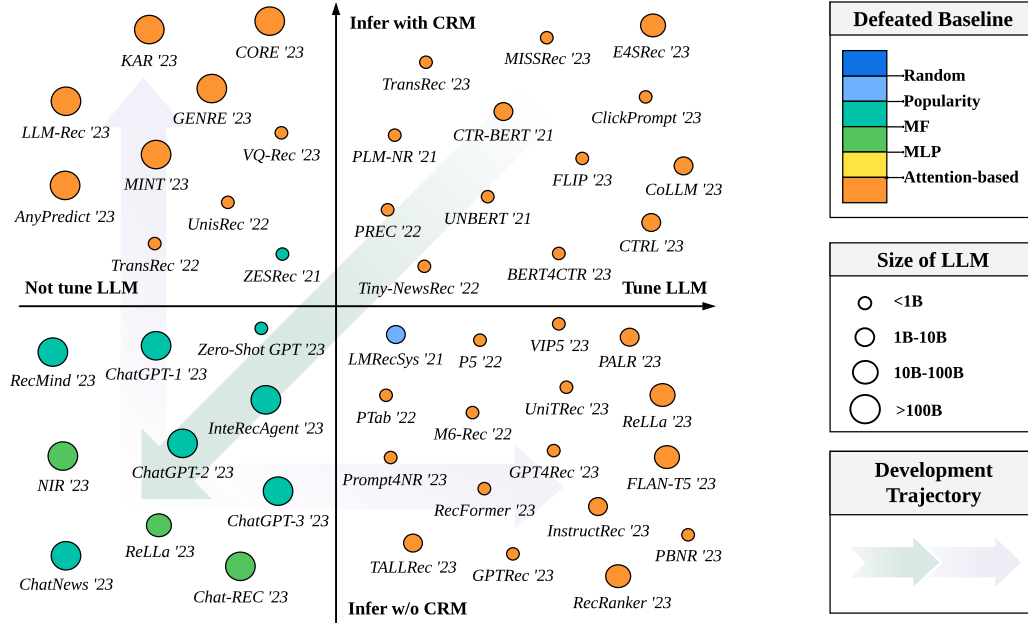


Fig. 4. Four-quadrant classification about how to adapt LLM to RS. Each circle in the quadrants denotes one research work with the corresponding model name attached below the circle. The size of each circle means the largest size of LLM leveraged in the research work. The color of each circle indicates the best compared baseline that the proposed model defeats as reported in the corresponding paper. For example, the **green** circle of **Chat-REC** in quadrant 3 denotes that it utilizes a large language model with size larger than 100B (i.e., ChatGPT) and defeats the MF baseline. Besides, we summarize the general development path with light-colored arrows. Abbreviations: MF is short for matrix factorization; MLP is short for multi-layer perceptron.

other stages of the recommendation pipeline. For instance, in the feature engineering stage, we could instruct LLM to generate reliable auxiliary features and synthetic data samples [119] to assist the model training and evaluation. In this way, the open-world knowledge from LLM is injected into the closed-domain recommendation models. Furthermore, large language models also revolutionize the user interaction with a more human-friendly natural language interface and free-form dialogue for various information systems. Not to mention, participating in the pipeline control further requires sufficient logical reasoning and tool utilization capabilities, which are possessed by large language models.

In summary, we believe that, as the abilities of large language models are further explored, they will form gradually deeper couplings and bindings with multiple stages of the recommendation pipeline. Even further, we might need to customize large language models specifically tailored to satisfy the unique requirements of recommender systems [106].

#### 4 HOW TO ADAPT LARGE LANGUAGE MODELS

To answer the “**HOW**” question about adapting LLM to RS, we carry out two orthogonal taxonomy criteria to distinguish the adaptation of LLM to RS, resulting in a four-quadrant classification shown in Figure 4:

- **Tune/Not Tune LLM** denotes whether we will tune LLM based on the in-domain recommendation data during the training phase. The definition of tuning LLM includes both full finetuning and other parameter-efficient finetuning methods (e.g., LoRA [63], prompt tuning [82]).



- **Infer with/without CRM** denotes whether we will involve conventional recommendation models (CRM) during the inference phase. Note that there are works that only use CRM to serve as independent pre-filter functions to generate the candidate item set for LLM [46, 196, 243]. We categorize them as “infer without CRM”, since the CRM is independent of LLM, and could be decoupled from the final recommendation task.

In Figure 4, we use different marker sizes to indicate the size of the large language model the research works adapt, and use different colors to indicate the best baseline they have defeated in terms of item recommendation. Thus, a few works are not presented in Figure 4 since they do not provide traditional recommendation evaluation, e.g., RecLLM [39] only investigates the system architecture design to involve LLM for RS pipeline control without experimental evaluation. Moreover, it is noteworthy that some research works might propose techniques that are applied across different quadrants. For instance, ReLLa [111] designs semantic user behavior retrieval to help LLM better comprehend and model the lifelong user behavior sequence in both zero-shot prediction (i.e., quadrant 3) and few-shot finetuning (i.e., quadrant 4) settings.

Given the four-quadrant taxonomy, we demonstrate that the overall development path in terms of “**HOW**” research question generally follows the light-colored arrows in Figure 4. Accordingly, we will introduce the latest research works in the order of quadrant 1, 3, 2, 4, followed by in-detail discussions for each quadrant subsection.

#### 4.1 Tune LLM & Infer with CRM (Quadrant 1)

Quadrant 1 refers to research works that not only finetune the large language models with in-domain recommendation data during the training phase, but also introduce conventional recommendation models to provide better collaborative knowledge. Based on their development over time, the works in quadrant 1 can be mainly classified into two stages.

Back to years 2021 and 2022, earlier works in quadrant 1 mainly focus on applying relatively smaller pretrained language models (e.g., BERT [28] and GPT2 [158]) to the downstream domains with abundant textual features, e.g., news recommendation [120, 220, 241, 254], web search [124, 275] and e-commercial advertisement [140, 198]. As discussed in Section 3.6, the primary roles of these small-scale language models are only limited to feature encoders for semantic representation enhancement. Consequently, a conventional recommendation model (CRM) is required to make the final recommendation, with generated textual representations as auxiliary inputs. Additionally, the small model size makes it affordable to fully finetune the language model during the training phase. UNBERT [254], PLM-NR [220], PREC [120] and Tiny-NewsRec [241] conduct full finetuning over small-scale language models (e.g., BERT [28], RoBERTa [125], UniLMv2 [1]) to enhance the content understanding for better news recommendation with sequential CRMs. Zou et al. [275] and Liu et al. [124] customize pretrained language models with further pretraining over the domain-specific corpus for web search scenarios. Moreover, TransRec [40] proposes layerwise adapter tuning over BERT [28] to ensure both the training efficiency and multi-modality enhanced representations. Although these earlier works can defeat strong baseline models with attention mechanisms by tuning language models and involving CRM, they only leverage small-scale language models as feature encoders, and thus the key capacities (e.g., reasoning, instruction following) of large foundation models remain underexplored.

At around the beginning of the year 2023, the rise of LLM (e.g., ChatGPT) demonstrates impressive emergent abilities like reasoning and instruction following, pointing out promising directions for LLM-enhanced recommendation. Therefore, researchers start to investigate the potential of incorporating billion-level large language models (e.g., LLaMA [192] and ChatGLM [36, 248]) to the field of recommender systems. Compared to earlier works with small-scale

language models that we have discussed above, there are two major differences to be clarified for these recent works that incorporate large language models:

- Due to the massive amount of model parameters possessed by LLM, we can hardly perform full finetuning on LLM as it can lead to an unaffordable cost in computational resources. Instead, parameter-efficient finetuning (PEFT) methods are commonly adopted for training efficiency with usually less than 1% parameters need to be updated, *e.g.*, low-rank adaption (LoRA) [63] and prompt tuning [82, 101].
- The role of LLM is no longer a simple tunable feature encoder for CRM. To make better use of the reasoning ability and open-world knowledge exhibited by LLM, researchers tend to place LLM and CRM on an equal footing (*e.g.*, both as the recommenders), mutually leveraging their respective strengths to collaborate and achieve improved recommendation performance. Moreover, as discussed in Section 3, LLM can also be finetuned for the stages of feature engineering [103], user interaction [83] and pipeline control [39] as well.

CoLLM [260] and E4SRec [96] adopt LoRA to finetune Vicuna-7B [18] and LLaMA2-13B [192] respectively, and build personalized prompts by injecting the user & item embedding from a pretrained CRM via a linear mapping layer. CTRL [95] conducts knowledge distillation between LLM and CRM for better alignment and interaction between the semantic and collaborative knowledge, where the size of involved LLM scales up to 6 billion (ChatGLM-6B [248]) with last-layer finetuning strategy. LLaMA-E [173] and EcomGPT [103] finetune the base large language models (*i.e.*, LLaMA-30B [192] and BLOOMZ-7.1B [139]) to assist the conventional recommendation models with augmented generative features, *e.g.*, item attributes and topics of user reviews.

As shown in Figure 4, since CRM is involved and LLM is tunable, the research works in quadrant 1 could better align to the data distribution of recommender systems and thus all achieve satisfying performance, even when the size of involved LLM is relatively small. Moreover, we can observe the clear trend that researchers intend to consider larger language models from the million level up to the billion level, thus benefiting from their vast amount of open-world semantic knowledge, as well as the instruction following and reasoning abilities. Nevertheless, when it comes to low-resource scenarios, the small-scale language model (*e.g.*, BERT) is still an economic choice to balance between LLM-based enhancement and computational efficiency.

#### 4.2 Not Tune LLM & Infer w/o CRM (Quadrant 3)

Quadrant 3 refers to research works that exclude the conventional recommendation model and solely adopt a frozen large language model as the recommender. This line of research generally emerges since the advent of large foundation models, especially ChatGPT, where researchers aim to analyze the zero-shot or few-shot performance of LLM in recommendation domains with LLM fixed and CRM excluded. It should be noted that, in the context of quadrant 3, the “few-shot” setting specifically refers to the in-context learning (ICL) approach for LLM, rather than tuning LLM based on a few training samples.

Earlier works [24, 116, 183, 206] investigate the zero-shot and few-shot recommendation settings based on the ChatGPT API, with delicate prompt engineering to instruct the LLM to model the user interest and perform tasks like rating prediction, pairwise comparison, and listwise ranking. However, the performance of these approaches is not satisfactory. Based on these previous works, several attempts [43, 111, 213, 252] are made to improve the zero-shot or few-shot recommendation performance of LLM. Lin et al. [111] identify the lifelong sequential behavior incomprehension problem of LLM, *i.e.*, LLM fails to extract the useful information from a textual context of long user behavior sequence for recommendation tasks, even if the length of context is far from reaching the context limitation of LLM. To mitigate

such an issue, ReLLa [111] proposes to perform semantic user behavior retrieval to replace the simply truncated top- $K$  recent behaviors with the top- $K$  semantically relevant behaviors towards the target item. In this way, the quality of data samples is improved, thus making it easier for LLM to comprehend the user sequence and achieve better zero-shot recommendation performance. RecMind [213] designs the self-inspiring prompt strategy and enables LLM to explicitly access the external knowledge with extra tools, such as SQL for recommendation database and search engine for web information. Chat-REC [43] instructs ChatGPT to not only serve as the score/ranking function, but also take control over the recommendation pipeline, *e.g.*, deciding when to call an independent pre-ranking model API.

As illustrated in Figure 4, although a larger model size might bring performance improvement, the zero-shot or few-shot learning of LLM in quadrant 3 is much inferior compared with the light-weight CRM tuned on the training data. Even when equipped with advanced techniques such as user behavior retrieval and tool usage, the performance of a frozen LLM without CRM is still suboptimal and far from the SOTA performance. The knowledge contained in LLM is global and factual, but recommendation is a personalized task that requires preference-oriented knowledge. This indicates the importance of in-domain collaborative knowledge from the training data of recommender systems, and that solely relying on a fixed large language model is currently unsuitable to well tackle the recommendation tasks. Consequently, there are two major approaches to further inject the in-domain collaborative knowledge for LLM to improve the recommendation performance: (1) involving CRM for inference, and (2) tuning LLM based on the training data, which refer to works of quadrant 2 and quadrant 4 in Figure 4, respectively.

#### 4.3 Not Tune LLM & Infer with CRM (Quadrant 2)

Research works in quadrant 2 utilize different key capabilities (*e.g.*, rich semantic information, reasoning ability) of LLM without finetuning to help CRM better accomplish the recommendation tasks. Similar to works in quadrant 1, the utilization of a frozen LLM in quadrant 2 generally demonstrates a development path in terms of the size of LLM evolving over time, *i.e.*, from small-scale language models to large language models.

Early works [31, 60, 61] propose to extract transferable text embeddings from a fixed BERT [28] model with rich semantic information. The text embeddings are then fed into several projection layers to better produce the cross-domain representations as the input of trainable conventional recommendation models. The projection layers are designed as a single-layer neural network for ZESRec [31], a mixture-of-expert (MoE) network for UniSRec [61], and a vector quantization based embedding lookup table for VQ-Rec [60]. We can observe from Figure 4 that the direct usage of a single-layer neural network as an adapter does not yield satisfactory results. However, with a carefully designed adapter module, the semantic representations from the fixed BERT parameters can be better aligned with the subsequent recommendation module, leading to impressive recommendation performances.

As discussed in Section 3.6, with the model size scaling up, the emergent abilities and abundant open-world knowledge enable large foundation models to extend their roles to other stages of the recommendation pipeline, such as feature engineering stage [85, 119, 215, 228] and user interaction [55, 61, 165, 210]. AnyPredict [215] leverages ChatGPT APIs to consolidate tabular samples to overcome the barrier across tables with varying schema, resulting in unified expanded training data for the follow-up conventional predictive models. ONCE [119] utilizes ChatGPT to perform news piece generation, user profiling, and news summarization, and thus augments the news recommendation model with LLM-generated features. KAR [228] and RLMRec [166] leverage LLM to enhance the user behavior modeling with specially designed input templates as well as chained prompting strategies, aiming to provide user-level feature augmentation for CRM. Wang et al. [210] and He et al. [55] investigate the integration of LLM (*e.g.*, ChatGPT and GPT4) to handle the open-ended free-form chatting during the user interaction of conversational recommendation.

In these works, although LLM is frozen, the involvement of CRM for the inference phase generally guarantees better recommendation performance, compared with works from quadrant 3 (*i.e.*, Not Tune LLM; Infer w/o CRM) in terms of the best baseline they defeat. When compared with quadrant 1 (*i.e.*, Tune LLM; Infer with CRM), since the large language model is fixed, the role of LLM in quadrant 2 is mostly auxiliary for CRM at different stages of the recommendation pipeline, including but not limited to feature engineering and feature encoder.

#### 4.4 Tune LLM & Infer w/o CRM (Quadrant 4)

Research works in quadrant 4 aim to finetune the large language models to serve as the scoring/ranking function based on the training data from recommender systems, excluding the involvement of CRM. Since CRM is excluded, we have to apply prompt templates to obtain textual input-output pairs, and therefore convert the recommendation tasks (*e.g.*, click-through rate estimation and next item prediction) into either a text classification task [3, 111] or a sequence-to-sequence task [44, 178, 196].

As an early attempt, LMRecSys [259] tunes language models to estimate the score of each candidate item via joint inference over multiple masked tokens, resulting in unsatisfying performance. The reason might be that its scoring manners are somehow problematic, where the authors simply pad or truncate the length of all the titles for items to 10 tokens. Prompt4NR [261] finetunes BERT by predicting the key answer words (*e.g.*, Yes/No, Good/Bad) based on the prompting templates. PTab [115] transforms tabular data into text and finetunes a BERT model based on the masked language modeling task followed by classification tasks. UniTRec [135] finetunes a BART [84] model with a joint contrastive loss to optimize the discriminative score and a perplexity-based score. RecFormer [87] adopts two-stage finetuning based on masked language modeling loss and item-item contrastive loss with LongFormer [4] as the backbone model. P5 [44], FLAN-T5 [79], and InstructRec [253] adopt T5 [159] as the backbone, and train the model in a sequence-to-sequence manner. GPT4Rec [88] tunes GPT [158] models as a generative function for next item prediction via causal language modeling.

The works mentioned above all adopt full finetuning over relatively small-scale language models (*e.g.*, 110M for BERT-base, 149M for LongFormer), which could be considerably expensive and unscalable as the size of the language model continuously increases up to tens of or even hundreds of billions. Although the large model parameter capacity enables proficient knowledge and capabilities, fully finetuning such a big model can lead to substantial resource consumption. As a result, parameter-efficient finetuning methods (PEFT) are usually required to efficiently adapt billion-level LLM to RS. Among those PEFT methods, low-rank adaption (LoRA) serves as the most popular choice. For instance, ReLLa [111], GenRec [70], BIGRec [2], RecSysLLM [21] and LSAT [174] adopt the LoRA [63] technique to finetune a base large language model (usually LLaMA-7b [192] or Vicuna-7B [18]) for item scoring or generation tasks. Apart from LoRA, M6-Rec [23] designs option tuning as an improved version of prompt tuning to empower M6 [109] for varied downstream tasks like item retrieval and ranking. VIP5 [45] performs layerwise adapter tuning to unify various modalities (*e.g.*, ID, text, and image) via a universal foundation model for recommendation tasks.

Although the introduction of PEFT alleviates the training inefficiency issue, the computational overhead can still be excessive for real-world applications where the number of training samples might scale up to billions. In such a situation, even PEFT methods like LoRA are not efficient enough for LLM to go over the entire training dataset. To this end, recent works start to investigate the strong inductive learning capability [150] of LLM by downsampling the whole training set into a small-scale subset [16, 79, 111, 129]. As a representative, ReLLa [111] uniformly samples less than 10% of the training instances and surprisingly finds that LLM, which is finetuned only based on less than 10% samples, is able to outperform the conventional recommendation baseline models that are trained on the entire training

dataset. Such a phenomenon about the strong few-shot inductive learning capability of LLM in recommendation is also validated by other related works [16, 79, 129]. As for different downsampling strategies, PALR [16] randomly selects 20% of the user to construct the training subset for efficient finetuning of LLaMA-7B [192]. RecRanker [129] designs an adaptive user sampling strategy, which consists of both importance-aware and clustering-based sampling followed the repetitive penalty.

As shown in Figure 4, the performance of finetuning LLM based on recommendation data is promising with proper task formulation, even if the model size is still relatively small (*i.e.*, less than 1 billion). Apart from the design of input prompt and model architecture to achieve superior recommendation performance, scalability and efficiency are also the major challenges in this line of research. That is, how to efficiently finetune a *large-scale* language model on a *large-scale* training dataset, where various PEFT methods and data downsampling strategies would be considered.

#### 4.5 Discussion

We first conclude the necessity of collaborative knowledge injection when adapting LLM to RS, and then summarize the overall development path in terms of the “**HOW**” question, as well as possible future directions. Next, we cast a discussion on the relationship between the recommendation performance and the size of the adapted LLM. Finally, we discuss an interesting property found about the hard sample reranking for large language models.

**4.5.1 Collaborative Knowledge is Needed.** From Figure 4, we could observe a clear performance boundary between works from quadrant 3 and quadrant 1, 2, 4. The research works from quadrant 3 are inferior even though they adapt large-scale models (*i.e.*, ChatGPT or GPT4), even when they are equipped with advanced techniques like user behavior retrieval and tool usage. This indicates that the recommender system is a highly specialized area, which demands a lot of in-domain collaborative knowledge. LLM cannot effectively learn such knowledge from its general pretraining corpus. Therefore, we have to involve in-domain collaborative knowledge for better performance when adapting LLM to RS, and there are generally two ways to achieve the goal (corresponding to quadrant 1, 2, 4):

- **Tune LLM** during the training phase, which injects collaborative knowledge from a data-centric aspect.
- **Infer with CRM** during the inference phase, which injects collaborative knowledge from a model-centric aspect.

Both two approaches emphasize the importance of in-domain collaborative knowledge when adapting LLM to RS. Based on the insights above, as shown in Figure 5, we draw a general development trend about the “**HOW**” research question on the basis of the four-quadrant classification. Starting from the early days of the year 2021, researchers usually intend to combine both small-scale LM and CRM to conduct joint optimization for recommendation (*i.e.*, Quadrant 1). Then, at around the beginning of the year 2023, several works begin to leverage a frozen LLM for recommendation without the help of CRM (*i.e.*, Quadrant 3), the inferior performance of which indicates the necessity of collaborative knowledge. To this end, two major solutions are proposed to conduct the in-domain collaborative knowledge injection via either involving CRM or tuning LLM, corresponding to Quadrants 2 and 4, respectively. Next, as we discover the golden principle for the adaptation of LLM to RS (*i.e.*, in-domain collaborative knowledge injection), the development path further moves back to Quadrant 1, where we aim to jointly optimize LLM and CRM for superior recommendation performance. Finally, in terms of how to adapt LLM to RS, the possible future direction might lie in the ways to better incorporate the collaborative knowledge from recommender systems with the general-purpose semantic knowledge and emergent abilities exhibited by LLM. For example, empowering agent-based LLM with external tools for more thorough access to recommendation data, as well as real-time web information from search engines.

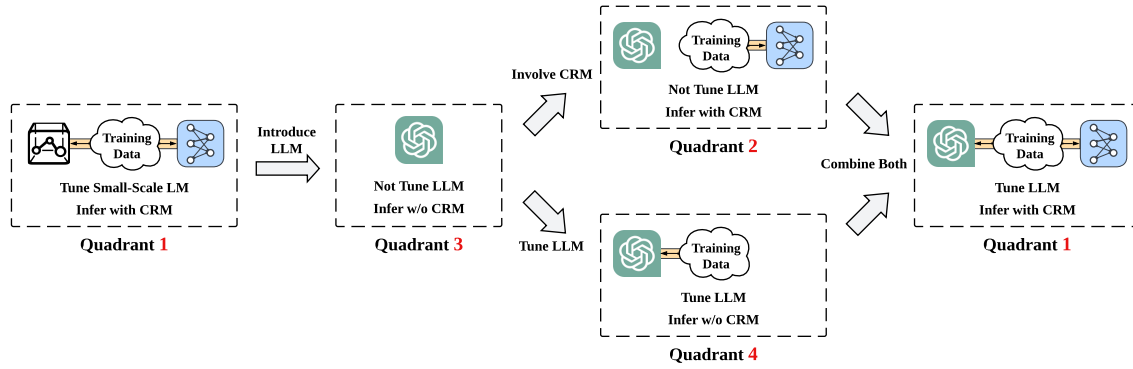


Fig. 5. The illustration of the development trend for adapting LLM to RS in terms of the “HOW” research question based on the four-quadrant classification. Earlier attempts generally perform joint optimization of small-scale language models and conventional recommendation models based on the training data (i.e., Quadrant 1). Then, researchers try to introduce a frozen LLM for recommendation without the help of CRM (i.e., Quadrant 3), which results in inferior performance. To this end, the golden principle, i.e., in-domain collaborative knowledge injection, is discovered, and a wide range of works start to explore the potential of LLM for RS by involving CRM (i.e., Quadrant 2), tuning LLM (i.e., Quadrant 4), or combining both strategies (i.e., back to Quadrant 1).

**4.5.2 Is Bigger Always Better?** By injecting in-domain collaborative knowledge from either data-centric or model-centric aspects, research works from quadrants 1, 2, and 4 can achieve satisfying recommendation performance compared with attention-based baselines, except for a few cases. Among these studies, although we could observe that the size of adapted LLM gradually increases according to the timeline, a fine-grained cross comparison among them (i.e., a unified benchmark) remains vacant. Hence, it is difficult to directly conclude that a larger model size of LLM can definitely yield better results for recommender systems. This gives rise to an open question: Is bigger language models always better for recommender systems? Or is it good enough to use small-scale language models in combination with collaborative knowledge injection? Our opinions towards the question are in two folds:

- Compared with small-scale language models, large language models are still irreplaceable in certain specific tasks where reasoning abilities are required. For example, textual feature augmentation, human-like user interaction & dialogue, and recommendation pipeline control. In these scenarios, it is usually necessary to involve LLM instead of small-scale LM to ensure task accomplishment and recommendation performance.
- When playing the same role in RS (e.g., feature encoder), it is generally a commonsense that LLM can achieve better performance than small-scale LM. However, small-scale LM would serve as a more economical choice to balance between performance enhancement and computational cost. Or to say, whether the additional computational cost brought by LLM is worth the performance gain is still not well verified, especially when having small-scale LM as the light-weight substitute.

**4.5.3 LLM is Good at Reranking Hard Samples.** Although LLM generally suffers from inferior performance for zero/few-shot learning since little in-domain collaborative knowledge is involved, researchers [62, 133] have found that large language models such as ChatGPT are more likely to be a good reranker for *hard samples*. They introduce the filter-then-rerank paradigm which leverages a pre-ranking function from traditional recommender systems (e.g., matching or pre-ranking stage in industrial applications) to pre-filter those easy negative items, and thus generates a set of candidates with harder samples for LLM to rerank. In this way, the listwise reranking performance of LLM (especially



ChatGPT-like APIs) could be promoted. This finding is instructive for industrial applications, where we could require LLM to only handle hard samples and leave other samples for light-weight models to save computational costs.

## 5 CHALLENGES FROM REAL-WORLD APPLICATIONS

In this section, we highlight the key challenges in adapting LLM to RS, which mainly arise from the unique characteristics of recommender systems and real-world applications. Accordingly, we will also discuss the preliminary efforts done by existing works, as well as other possible solutions. The following challenges are proposed from three aspects: (1) **efficiency** (training efficiency, and inference latency), (2) **effectiveness** (in-domain long text modeling, and ID indexing & modeling), and (3) **ethics** (fairness, and other potential risks from LLM).

### 5.1 Training Efficiency

There are two key aspects to improve the performance of modern deep learning based recommender systems: (1) enlarge the volumes of training data (e.g., billion-level training samples), and (2) increase the update frequency for model (from day-level to hour-level, or even minute-level). Both of these two factors highly require the training efficiency. Although, as suggested in Section 4.5, tuning LLM (possibly with CRM) is a promising approach to align LLM to RS for better performance, it actually brings prohibitive adaptation costs in terms of both time and computational resource consumption. Therefore, how to ensure the training efficiency when adapting LLM to RS is one of the key challenges for real-world applications.

Existing works mainly propose to leverage parameter-efficient finetuning (PEFT) methods (e.g., low-rank adaptation [63], option tuning [23], layerwise adapter tuning [45]), which mainly solve the memory usage problem, but the time consumption is still high especially for large-scale scenarios with massive users and items. From the perspective of real-world applications, we suggest adopting the asynchronous update strategy, when we leverage LLM for feature engineering and feature encoder. To be specific, we can cut down the training data volume and relax the update frequency for LLM (e.g., week-level) while maintaining full training data and high update frequency for CRM. The basis to support this approach is that researchers [16, 111, 266] point out that LLM has strong inductive learning capacities to produce generalized and reliable outputs via a handful of supervisions. In this way, LLM can provide aligned in-domain knowledge to CRM, while CRM acts as a frequently updated adapter for LLM.

### 5.2 Inference Latency

Online recommender systems are usually real-time services and extremely time-sensitive, where all the stages (e.g., matching, ranking, reranking) should be done within around tens of milliseconds. The involvement of LLM during the inference phase gives rise to the inference latency problem. The inference time of the LLM is fairly high, not to mention the additional time cost brought by prompt template generation.

Pre-computing and caching the outputs or middle representations of LLM serves as a common strategy to ensure low-latency inference when we have to involve LLM during the inference phase. When adapting the LLM as the scoring/ranking function, M6-Rec [23] proposes the multi-segment late interaction strategy. The textual features of user and item are split into finer-grained segments that are more static, e.g., by representing each clicked item as an individual segment. Then, we can pre-compute and cache the encoded representations of each segment using the first several transformer layers, while the rest of the layers are leveraged to perform late interaction among multiple segments when the recommendation request arrives. Other works like UniSRec [61] and VQ-Rec [60] simply adopt language models as feature encoders. Hence it is straightforward to directly cache the dense embeddings produced by



the language model. The pre-computing and caching strategy might be suitable for item-side information since they are generally static, but it can be suboptimal for user-side information since the user behaviors and interests are highly dynamic and quickly evolve over time. Hence, we have to find an appropriate caching frequency to balance between the performance and computational cost.

Moreover, we can also seek ways to reduce the size of model for the inference efficiency, where methods have been well explored in other deep learning domains, e.g., distillation [74], pruning [15], and quantization [246]. For instance, CTRL [95] and FLIP [199] propose to perform contrastive learning to distill the semantic knowledge from LLM to CRM. The CRM is then solely finetuned with improved parameter initialization for better recommendation performance, concurrently maintaining the low-latency inference. These strategies generally involve a tradeoff between the model performance and inference latency. Alternatively, we could involve LLM in the feature engineering stage and pre-store the outputs of LLM, which will bring a significantly smaller (but not entirely negligible) extra burden for inference. Besides, we can also introduce LLM to scenarios with relatively loose inference latency constraints like conversational recommender systems.

### 5.3 In-Domain Long Text Modeling

When adapting LLM, we have to construct in-domain textual inputs via prompting templates and insert proper instructions and demonstrations at the front if needed. However, the general guideline of real-world recommender systems requires longer user history, larger candidate set and more features to achieve better recommendation performance, possibly leading to long-text inputs for LLM. Such long-text inputs from RS domains (i.e., in-domain long texts) could result in two key challenges as follows.

Firstly, an excessively long-text input would cause the memory inefficiency problem (the space complexity of classical transformers are  $O(L^2)$  where  $L$  is the number of tokens), and might even break the context window limitation, leading to partial information lost and inferior outputs from LLM. Secondly, even if the length of input prompt does not exceed the context window, there may still exist issues for LLM to fully comprehend and reason on the recommendation data. Lin et al. [111] and Hou et al. [62] reveal that LLM has difficulty in dealing with long texts especially when we extend the text with longer user history or larger candidate set, even though the total number of input tokens is far from reaching the context window limitation (e.g., 512 for BERT, 4096 for ChatGPT). The reason might be that the distribution of in-domain long text is quite different from the pretraining corpora of LLM.

To this end, it is of great importance to investigate how to properly filter, select, and arrange the textual information as the input for LLM during prompting engineering, as well as how to instruct or tune the LLM to better align with the distribution of these in-domain long-text inputs. Besides, in NLP domains, a range of works are proposed to address the context window limitation (e.g., sliding windows [216], memory mechanism [32]), which could be considered in recommender systems. Moreover, recent works propose to combine the latent representations from CRM to compress the personalized input prompt for LLM, thus alleviating the long-text problem. For instance, CoLLM [260] and E4SRec [96] replace the textual description of each user behavior with one latent vector mapping from the embedding table of CRM via a linear projection layer, which greatly reduces the number of tokens for long user sequences. Inspired by prefix tuning [101, 123], ClickPrompt [107] transforms the sample-wise final representation of CRM into layer-wise prompts for LLM, making it easier to eliminate unnecessary features from the prompt templates.

#### 5.4 ID Indexing & Modeling

In recommender systems, there exists a kind of pure ID features that inherently contains no semantic information (e.g., user ID, item ID). If we include these ID features in the prompting text, the tokenization is actually unmeaningful to language models (e.g., user ID AX1265 might be tokenized as [AX, 12, 65]). Many works [23, 60] tend to directly abandon these ID features (e.g., replacing item IDs with item titles or descriptions) for unified cross-domain recommendation via the natural language interface, since the IDs are usually not shared across different domains. However, some works [44, 242] point out that ID features can greatly promote the recommendation performance, although sacrificing the cross-domain generalization ability. Therefore, it is still an open question about whether we should retain the ID features or not, which divides the research regarding ID indexing & modeling into two directions.

On the one hand, we could sacrifice the cross-domain generalization ability to obtain better in-domain recommendation performance by keeping the ID features. P5 [44] and its variants [45, 64, 65] preserve the ID features as textual inputs in the prompting templates. P5 designs a whole-word embedding layer to assign the same whole-word embedding for tokens from the same ID feature. The whole-word embeddings will be added to the token embeddings in the same way as position embeddings in language models. Based on P5, Hua et al. [65] further explore various item ID indexing strategies (e.g., sequential indexing, collaborative indexing) to ensure the IDs of similar items consist of similar sub-tokens. RecFormer [87] and UniSRec [61] omit the item IDs in prompting texts, but introduce additional ID embeddings at either bottom embedding layer or top projection layer. Other works [96, 260, 263] seek to integrate ID embeddings from conventional recommendation models and therefore make the input prompt of LLM free of pure ID features. In summary, researchers in this line should focus on how to associate LLM with ID features via carefully designed ID indexing & modeling strategies.

On the other hand, we could abandon the ID features to achieve unified cross-domain recommendation via natural language interface. Maintaining a unified model to serve various domains is very promising, especially when we involve large language model [23, 60, 111]. In this direction, in order to achieve similar performance to those works that keep ID features, researchers could investigate ways to introduce ID features in an implicit manner [95, 199], e.g., apply contrastive learning to align between the semantic and collaborative knowledge, and therefore avoid involving ID features for large language models.

#### 5.5 Fairness

Researchers have discovered that bias in the pretraining corpus could mislead LLM to generate harmful or offensive content, e.g., discriminating against disadvantaged groups [26, 169]. Although there are strategies (e.g., RLHF [143]) to reduce the harmfulness of LLM, existing works have already detected the unfairness problem in recommender systems brought by LLM from both user-side [64, 251] and item-side [62] perspectives.

The user-side fairness in recommender systems requires similar users to be treated similarly at either individual level or group level. The user sensitive attributes should not be preset during recommendation (e.g., gender, race). For instance, Salinas et al. [170] reveal the demographic bias of LLM through job recommendations, where LLM tends to provide unequal opportunities for people with different genders or from different countries. Xu et al. [230] study the traceback, degree, and impact of the implicit user unfairness of LLM for recommendation, and find that LLM will implicitly infer the gender, race or nationality from user name. Li et al. [100] further study to mitigate the provider bias [8, 152] in news recommendation by either explicitly specifying the number of articles from both popular and unpopular providers, or explicitly indicating the priority of less popular providers. To tackle such a user-side unfairness

problem, UP5 [64] proposes counterfactually fair prompting (CFP), which consists of a personalized prefix prompt and a prompt mixture to ensure fairness w.r.t. a set of sensitive attributes. Besides, Zhang et al. [251] introduce a benchmark named FaiRLLM, where authors comprise carefully crafted metrics and a dataset that accounts for eight sensitive attributes in recommendation scenarios where LLM is involved. Yet these studies only focus on the fairness issue in specific recommendation tasks (e.g., item generation task) with limited evaluation metrics.

The item-side fairness in recommender systems ensures that each item or item group should receive a fair chance to be recommended (e.g., proportional to its merits or utility) [121, 144, 177]. However, how to improve item-side fairness in LLM remains less explored. As a preliminary study, Hou et al. [62] observe that the popularity bias occurs when LLM serves as a ranking function, and alleviate the bias to some extents by designing prompts to guide the LLM focusing on users' historical interactions. Another related work [128] alleviates the item popularity bias by representing long-tail items using full-text modeling and bringing the benefits of LLM to recommender systems, but it neglects the intrinsic item-side bias within LLM itself. Further studies on popularity bias and other potential item-wise fairness issues when adapting LLM to RS are still needed.

## 5.6 Other Potential Risks from LLM

Apart from the fairness problem, researchers have identified many other potential risks that intrinsically stem from large language models, e.g., hallucination [86, 204]. When we adapt LLM to RS, some of these biases might be magnified and hurt the reliability of the system. Hence, in this section, we discuss the new challenges for building harmless and trustworthy LLM-enhanced recommender systems from three perspectives: hallucination, privacy, and explainability.

**5.6.1 Hallucination.** Hallucination refers to the phenomenon that large language models generate output texts that appear credible but are actually incorrect or lack of factual basis [86, 204]. The hallucination problem of LLM can mislead the recommender system with erroneous information, possibly resulting in recommendation performance degeneration. For instance, when adapting LLM to the feature engineering stage of RS for enhancing the item content understanding, a hallucinative output from LLM might erroneously provide fake attributes or descriptions for the given item, adversely affecting the performance of recommendation models. Furthermore, the hallucination problem can cause severe risks to individuals, particularly in critical recommendation scenarios like healthcare suggestions, legal guidance and education. In these areas, the spread of inaccurate information can lead to serious real repercussions in society. Therefore, to counteract the hallucination, it is crucial to verify the correctness and factualness of the generated content from LLM, possibly with the help of external resources like knowledge graphs as the additional verifiable information [134, 136].

**5.6.2 Privacy.** The data privacy serves as a long-standing problem in machine learning [37], and is becoming increasingly important for recommender systems in the era of large language models due to the following two concerns. Firstly, the success of LLM highly relies on the extensive pretraining corpus collected from diverse online sources, some of which might contain users' sensitive information, e.g., the user's email address from social media platforms. Secondly, apart from the sensitive information in pretraining corpora, LLM is also frequently leveraged to process or even finetuned on the user behavior data from the recommender system, which encompasses personal preferences, online activities and other identifiable information. The accessibility of LLM to these user-sensitive data resources would pose the potential risk of exposing private user information, leading to privacy violations [9, 237]. Consequently, safeguarding the confidentiality and security of the data is essential for privacy preservation and building a trustworthy recommender system. As preliminary studies, DPLLM [10] finetunes a differentially private (DP) large language model

for privacy-preserved synthetic user query generation in recommender systems. Li et al. [104] propose to personalize LLM based on the user’s own private data through prompt tuning with a privatized token reconstruction task.

**5.6.3 Explainability.** Generating user-friendly explanations regarding why an item is recommended plays a crucial role in enhancing user trust and facilitating more informed decision makings during recommendation [131]. We discuss the explainability property for LLM-enhanced recommender systems from the following two perspectives. *Firstly*, LLM can make conventional recommender systems more explainable. Several works have revealed that LLM is capable of generating reasonable explanations based on the recommendation output [22, 46, 157], as well as interpreting the latent representations of CRM after careful alignments [81]. For instance, Rahdari et al. [161] propose the Logic-Scaffolding framework to combine the aspect-based explanation and chain-of-thought prompting for LLM to generate recommendation explanations through intermediate reasoning steps. *Secondly*, although LLM helps improve the explainable recommendation, LLM itself is still a black box that lacks explainability for the recommender system, especially when we involve closed-source large language models like ChatGPT and GPT4 [12]. This is potentially risky if the behavior of LLM is unexplainable and uncontrollable when building a reliable and trustworthy LLM-enhanced recommender system. Based on the two insights above, we argue that the future directions for LLM-enhanced explainable recommendation generally lies in two folds: (1) design better strategies to prompt and acquire recommendation explanations from LLM, and meanwhile (2) seek better ways to enhance the interpretability of LLM itself.

## 6 CONCLUSION AND FUTURE PROSPECTS

In conclusion, large language models have demonstrated impressive human-like capabilities due to their extensive open-world knowledge, the logical and commonsense reasoning ability, and the comprehension of human culture and society [50, 207, 262]. As a result, the emergence of large language models is opening up a promising research direction for LLM-enhanced recommender systems. This survey proposes a systematic view of the LLM-enhanced recommendation from the perspective of the whole pipeline in industrial recommender systems. We comprehensively summarize the latest research progress in adapting large language models to recommender systems from two aspects: where and how to adapt LLM to RS.

- For the “**WHERE**” question, we analyze the roles that LLM could play at different stages of the recommendation pipeline, *i.e.*, feature engineering, feature encoder, scoring/ranking function, user interaction, and pipeline controller.
- For the “**HOW**” question, we analyze the training and inference strategies, resulting in two orthogonal classification criteria, *i.e.*, whether to tune LLM during training, and whether to involve CRM for inference.

Detailed discussions and insightful development paths are also provided for each taxonomy perspective. As for future prospects, apart from the three aspects we have already highlighted in Section 5 (*i.e.*, **efficiency**, **effectiveness** and **ethics**), we would like to further express our hopeful vision for the future development of combining large language models and recommender systems:

- **A unified public benchmark** is of an urgent need to provide reasonable and convincing evaluation protocols, since (1) the fine-grained cross comparison among existing works remains vacant, and (2) it is quite expensive and difficult to reproduce the experimental results of recommendation models combined with LLM. Although there exist some benchmarks for LLM-enhanced RS (*e.g.*, LLMRec [117], OpenP5 [232]), they generally concentrate on a certain aspect of LLM-enhanced RS. For instances, OpenP5 [232] and LLMRec [117] only focus on the generative recommendation paradigms that adopt LLM as the scoring/ranking function without help of CRM. Consequently, a

unified comparison for the adaptations of LLM to different recommendation pipeline stages (e.g., feature engineering, feature encoder) still remains to be explored.

- **A customized large foundation model** for recommendation domains, which can take over control of the entire recommendation pipeline. Currently, research works that involve LLM in the pipeline controller stage generally adopt a frozen general-purpose large foundation model like ChatGPT and GPT4 to connect the different stages. By constructing in-domain instruction data and even customizing the model structure for collaborative knowledge, there is a hopeful vision that we can acquire a large foundation model specially designed for recommendation domains, enabling a new level of automation in recommender systems.

## REFERENCES

- [1] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International conference on machine learning*. PMLR, 642–652.
- [2] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangna He, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434* (2023).
- [3] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangna He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. *arXiv preprint arXiv:2305.00447* (2023).
- [4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).
- [5] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language Models are Realistic Tabular Data Generators. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=cEygmqNOel>
- [6] Alexander Brinkmann, Roei Shraga, Reng Chiz Der, and Christian Bizer. 2023. Product Information Extraction using ChatGPT. *arXiv preprint arXiv:2306.14921* (2023).
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [8] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced neighborhoods for multi-sided fairness in recommendation. In *Conference on fairness, accountability and transparency*. PMLR, 202–214.
- [9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.
- [10] Aldo Gael Carranza, Reza Farahani, Natalia Ponomareva, Alex Kurakin, Matthew Jagielski, and Milad Nasr. 2023. Privacy-Preserving Recommender Systems with Synthetic Query Generation using Differentially Private Large Language Models. *arXiv preprint arXiv:2305.05973* (2023).
- [11] Junyi Chen. 2023. A Survey on Large Language Models for Personalized and Explainable Recommendations. *arXiv:2311.12338* [cs.IR]
- [12] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Defu Lian, and Enhong Chen. 2023. When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities. *arXiv:2307.16376* [cs.IR]
- [13] Jiao Chen, Luyi Ma, Xiaohan Li, Nikhil Thakurdesai, Jianpeng Xu, Jason HD Cho, Kaushiki Nag, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2023. 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).
- [14] Shuwei Chen, Xiang Li, Jian Dong, Jin Zhang, Yongkang Wang, and Xingxing Wang. 2023. TBN: Modeling Long Textual Behavior Data for CTR Prediction. *arXiv preprint arXiv:2308.08483* (2023).
- [15] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems* 33 (2020), 15834–15846.
- [16] Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. *arXiv preprint arXiv:2305.07622* (2023).
- [17] Mingyue Cheng, Qi Liu, Wenyu Zhang, Zhiding Liu, Hongke Zhao, and Enhong Chen. 2024. A general tail item representation enhancement framework for sequential recommendation. *Frontiers of Computer Science* 18, 6 (2024), 1–12.
- [18] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. <https://lmsys.org/blog/2023-03-30-vicuna/>
- [19] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.
- [20] Konstantina Christakopoulou, Alberto Lalam, Cj Adams, Iris Qu, Yifat Amir, Samer Chucui, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, et al. 2023. Large Language Models for User Interest Journeys. *arXiv preprint arXiv:2305.15498* (2023).
- [21] Zhixuan Chu, Hongyan Hao, Xin Ouyang, Simeng Wang, Yan Wang, Yue Shen, Jinjie Gu, Qing Cui, Longfei Li, Siqiao Xue, et al. 2023. Leveraging large language models for pre-trained recommender systems. *arXiv preprint arXiv:2308.10837* (2023).

- [22] Zhixuan Chu, Yan Wang, Qing Cui, Longfei Li, Wenqing Chen, Sheng Li, Zhan Qin, and Kui Ren. 2024. LLM-Guided Multi-View Hypergraph Learning for Human-Centric Explainable Recommendation. *arXiv preprint arXiv:2401.08217* (2024).
- [23] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *arXiv preprint arXiv:2205.08084* (2022).
- [24] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. *arXiv preprint arXiv:2305.02182* (2023).
- [25] Xinyi Dai, Jianghao Lin, Weinan Zhang, Shuai Li, Weiwen Liu, Ruiming Tang, Xiuqiang He, Jianye Hao, Jun Wang, and Yong Yu. 2021. An adversarial imitation click model for information retrieval. In *Proceedings of the Web Conference 2021*. 1809–1820.
- [26] Yashar Deldjoo. 2024. Understanding Biases in ChatGPT-based Recommender Systems: Provider Fairness, Temporal Stability, and Recency. *arXiv preprint arXiv:2401.10545* (2024).
- [27] Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. 2023. A Unified Multi-Task Learning Framework for Multi-Goal Conversational Recommender Systems. *ACM Trans. Inf. Syst.* 41, 3 (feb 2023), 25 pages.
- [28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [29] Dario Di Palma. 2023. Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1369–1373.
- [30] Dario Di Palma, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, Tommaso Di Noia, and Eugenio Di Sciascio. 2023. Evaluating chatgpt as a recommender system: A rigorous approach. *arXiv preprint arXiv:2309.03613* (2023).
- [31] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-shot recommender systems. *arXiv preprint arXiv:2105.08318* (2021).
- [32] Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. CogLix: Applying bert to long texts. *Advances in Neural Information Processing Systems* 33 (2020), 12792–12804.
- [33] Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. 2024. User Embedding Model for Personalized Language Prompting. *arXiv preprint arXiv:2401.04858* (2024).
- [34] Zhikang Dong, Bin Chen, Xiulong Liu, Pawel Polak, and Peng Zhang. 2023. MuseChat: A Conversational Music Recommendation System for Videos. *arXiv preprint arXiv:2310.06282* (2023).
- [35] Yingpeng Du, Di Luo, Rui Yan, Hongzhi Liu, Yang Song, Hengshu Zhu, and Jie Zhang. 2023. Enhancing job recommendation through llm-based generative adversarial networks. *arXiv preprint arXiv:2307.10747* (2023).
- [36] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.
- [37] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
- [38] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). *arXiv:2307.02046* [cs.IR]
- [39] Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *arXiv preprint arXiv:2305.07961* (2023).
- [40] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. 2023. Exploring Adapter-based Transfer Learning for Recommender Systems: Empirical Studies and Practical Insights. *arXiv preprint arXiv:2305.15036* (2023).
- [41] Lingyue Fu, Jianghao Lin, Weiwen Liu, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. An F-shape Click Model for Information Retrieval on Multi-block Mobile Pages. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1057–1065.
- [42] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2023. A Unified Framework for Multi-Domain CTR Prediction via Large Language Models. *arXiv preprint arXiv:2312.10743* (2023).
- [43] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).
- [44] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.
- [45] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. VIP5: Towards Multimodal Foundation Models for Recommendation. *arXiv preprint arXiv:2305.14302* (2023).
- [46] Preetam Ghosh and Vaishali Sadaphal. 2023. JobRecoGPT-Explainable job recommendations using LLMs. *arXiv preprint arXiv:2309.11805* (2023).
- [47] Yuqi Gong, Xichen Ding, Yehui Su, Kaiming Shen, Zhongyi Liu, and Guannan Zhang. 2023. An Unified Search and Recommendation Foundation Model for Cold-Start Scenario. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4595–4601.
- [48] Mahesh Goyani and Neha Chaurasiya. 2020. A review of movie recommendation system: Limitations, Survey and Challenges. *ELCVIA: electronic letters on computer vision and image analysis* 19, 3 (2020), 0018–37.
- [49] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).



- [50] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints* (2023).
- [51] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1096–1102.
- [52] Junda He, Bowen Xu, Zhou Yang, DongGyun Han, Chengran Yang, and David Lo. 2022. PTM4Tag: sharpening tag recommendation of stack overflow posts with pre-trained models. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. 1–11.
- [53] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. 639–648.
- [54] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
- [55] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 720–730.
- [56] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 5549–5581.
- [57] Ngai Lam Ho, Roy Ka-Wei Lee, and Kwan Hui Lim. 2023. BTRec: BERT-Based Trajectory Recommendation for Personalized Tours. *arXiv preprint arXiv:2310.19886* (2023).
- [58] Ngai Lam Ho and Kwan Hui Lim. 2023. Utilizing Language Models for Tour Itinerary Recommendation. *arXiv preprint arXiv:2311.12355* (2023).
- [59] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).
- [60] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*. 1162–1171.
- [61] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.
- [62] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845* (2023).
- [63] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [64] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. 2023. UP5: Unbiased Foundation Model for Fairness-aware Recommendation. *arXiv preprint arXiv:2305.12090* (2023).
- [65] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to Index Item IDs for Recommendation Foundation Models. *arXiv preprint arXiv:2305.06569* (2023).
- [66] Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards Reasoning in Large Language Models: A Survey. *arXiv preprint arXiv:2212.10403* (2022).
- [67] Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505* (2023).
- [68] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *ACM Comput. Surv.* 54, 5 (may 2021), 36 pages.
- [69] Jihwan Jeong, Yinlam Chow, Guy Tennenholtz, Chih-Wei Hsu, Azamat Tulepbergenov, Mohammad Ghavamzadeh, and Craig Boutilier. 2023. Factual and Personalized Recommendations using Language Models and Reinforcement Learning. *arXiv preprint arXiv:2310.06176* (2023).
- [70] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Genrec: Large language model for generative recommendation. *arXiv e-prints* (2023), arXiv–2307.
- [71] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Text based Large Language Model for Recommendation. *arXiv preprint arXiv:2307.00457* (2023).
- [72] Julie Jiang and Emilio Ferrara. 2023. Social-LLM: Modeling User Behavior at Scale using Language Models and Social Network Data. *arXiv preprint arXiv:2401.00893* (2023).
- [73] Junzhe Jiang, Shang Qu, Mingyue Cheng, and Qi Liu. 2023. Reformulating Sequential Recommendation: Learning Dynamic User Interest with Content-enriched Language Modeling. *arXiv preprint arXiv:2309.10435* (2023).
- [74] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351* (2019).
- [75] Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, et al. 2023. Language Models As Semantic Indexers. *arXiv preprint arXiv:2310.07815* (2023).
- [76] Jiarui Jin, Xianyu Chen, Fanghua Ye, Mengyue Yang, Yue Feng, Weinan Zhang, Yong Yu, and Jun Wang. 2023. Lending Interaction Wings to Recommender Systems with Conversational Agents. *arXiv preprint arXiv:2310.04230* (2023).
- [77] Angela John, Theophilus Aidoo, Hamayoon Behmanush, Irem B Gunduz, Hewan Shrestha, Maxx Richard Rahman, and Wolfgang Maaß. 2024. LLMRS: Unlocking Potentials of LLM-Based Recommender Systems for Software Purchase. *arXiv preprint arXiv:2401.06676* (2024).



- [78] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [79] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. *arXiv preprint arXiv:2305.06474* (2023).
- [80] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).
- [81] Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2023. RecExplainer: Aligning Large Language Models for Recommendation Model Interpretability. *arXiv preprint arXiv:2311.10947* (2023).
- [82] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).
- [83] Megan Leszczynski, Ravi Ganti, Shu Zhang, Krisztian Balog, Filip Radlinski, Fernando Pereira, and Arun Tejasvi Chaganty. 2023. Talk the Walk: Synthetic Data Generation for Conversational Music Recommendation. *ArXiv abs/2301.11489*.
- [84] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*. Association for Computational Linguistics, 7871–7880.
- [85] Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. TagGPT: Large Language Models are Zero-shot Multimodal Taggers. *arXiv preprint arXiv:2304.03022* (2023).
- [86] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 6449–6464.
- [87] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731* (2023).
- [88] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. *arXiv preprint arXiv:2304.03879* (2023).
- [89] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1348–1357.
- [90] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023. Large Language Models for Generative Recommendation: A Survey and Visionary Discussions. *arXiv:2309.01157* [cs.IR]
- [91] Pan Li, Yuyan Wang, Ed H Chi, and Minmin Chen. 2023. Prompt Tuning Large Language Models on Personalized Aspect Extraction for Recommendations. *arXiv preprint arXiv:2306.01475* (2023).
- [92] Qingyao Li, Lingyue Fu, Weiming Zhang, Xianyu Chen, Jingwei Yu, Wei Xia, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023. Adapting Large Language Models for Education: Foundational Capabilities, Potentials, and Challenges. *arXiv:2401.08664* [cs.AI]
- [93] Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. 2023. Exploring the Upper Limits of Text-Based Collaborative Filtering Using Large Language Models: Discoveries and Insights. *arXiv preprint arXiv:2305.11700* (2023).
- [94] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 9748–9758.
- [95] Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. CTRL: Connect Tabular and Language Model for CTR Prediction. *arXiv preprint arXiv:2306.02841* (2023).
- [96] Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023. E4SRec: An Elegant Effective Efficient Extensible Solution of Large Language Models for Sequential Recommendation. *arXiv preprint arXiv:2312.02443* (2023).
- [97] Xiaopeng Li, Lixin Su, Pengyue Jia, Xiangyu Zhao, Suqi Cheng, Junfeng Wang, and Dawei Yin. 2023. Agent4Ranking: Semantic Robust Ranking via Personalized Query Rewriting Using Multi-agent LLM. *arXiv preprint arXiv:2312.15450* (2023).
- [98] Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023. Exploring Fine-tuning ChatGPT for News Recommendation. *arXiv preprint arXiv:2311.05850* (2023).
- [99] Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023. PBNR: Prompt-based News Recommender System. *arXiv preprint arXiv:2304.07862* (2023).
- [100] Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023. A Preliminary Study of ChatGPT on News Recommendation: Personalization, Provider Fairness, Fake News. *arXiv preprint arXiv:2306.10702* (2023).
- [101] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).
- [102] Youhua Li, Hanwen Du, Yongxin Ni, Pengpeng Zhao, Qi Guo, Fajie Yuan, and Xiaofang Zhou. 2023. Multi-Modality is All You Need for Transferable Recommender Systems. *arXiv preprint arXiv:2312.09602* (2023).
- [103] Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. EcomGPT: Instruction-tuning Large Language Model with Chain-of-Task Tasks for E-commerce. *arXiv preprint arXiv:2308.06966* (2023).
- [104] Yansong Li, Zhixing Tan, and Yang Liu. 2023. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212* (2023).
- [105] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2023. LLaRA: Aligning Large Language Models with Sequential Recommenders. *arXiv preprint arXiv:2312.02445* (2023).

- [106] Guo Lin and Yongfeng Zhang. 2023. Sparks of Artificial General Recommender (AGR): Early Experiments with ChatGPT. *arXiv preprint arXiv:2305.04518* (2023).
- [107] Jianghao Lin, Bo Chen, Hangyu Wang, Yunjia Xi, Yanru Qu, Xinyi Dai, Kangning Zhang, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. ClickPrompt: CTR Models are Strong Prompt Generators for Adapting Language Models to CTR Prediction. *arXiv preprint arXiv:2310.09234* (2023).
- [108] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.
- [109] Junyang Lin, Rui Men, An Yang, Chang Zhou, Yichang Zhang, Peng Wang, Jingren Zhou, Jie Tang, and Hongxia Yang. 2021. M6: Multi-modality-to-multi-modality multitask mega-transformer for unified pretraining. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3251–3261.
- [110] Jianghao Lin, Yanru Qu, Wei Guo, Xinyi Dai, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. MAP: A Model-agnostic Pretraining Framework for Click-through Rate Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1384–1395.
- [111] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. *arXiv preprint arXiv:2308.11131* (2023).
- [112] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A multi-facet paradigm to bridge large language model and recommendation. *arXiv preprint arXiv:2310.06491* (2023).
- [113] Dairui Liu, Boming Yang, Honghui Du, Derek Greene, Aonghus Lawlor, Ruihai Dong, and Irene Li. 2023. RecPrompt: A Prompt Tuning Framework for News Recommendation Using Large Language Models. *arXiv preprint arXiv:2312.10463* (2023).
- [114] Fan Liu, Yaqi Liu, Zhiyong Cheng, Liqiang Nie, and Mohan Kankanahalli. 2023. Understanding Before Recommendation: Semantic Aspect-Aware Review Exploitation via Large Language Models. *arXiv preprint arXiv:2312.16275* (2023).
- [115] Guang Liu, Jie Yang, and Ledell Wu. 2022. PTab: Using the Pre-trained Language Model for Modeling Tabular Data. *arXiv preprint arXiv:2209.08060* (2022).
- [116] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *arXiv preprint arXiv:2304.10149* (2023).
- [117] Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, et al. 2023. Llmrec: Benchmarking large language models on recommendation task. *arXiv preprint arXiv:2308.12241* (2023).
- [118] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems. *arXiv preprint arXiv:2302.03735* (2023).
- [119] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. A First Look at LLM-Powered Generative News Recommendation. *arXiv preprint arXiv:2305.06566* (2023).
- [120] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiaoming Wu. 2022. Boosting Deep CTR Prediction with a Plug-and-Play Pre-trainer for News Recommendation. In *Proceedings of the 29th International Conference on Computational Linguistics*. 2823–2833.
- [121] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. 2019. Personalized fairness-aware re-ranking for microlending. In *Proceedings of the 13th ACM conference on recommender systems*. 467–471.
- [122] Weiwen Liu, Wei Guo, Yong Liu, Ruiming Tang, and Hao Wang. 2023. User Behavior Modeling with Deep Learning for Recommendation: Recent Advances. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1286–1287.
- [123] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602* (2021).
- [124] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3365–3375.
- [125] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv abs/1907.11692* (2019).
- [126] Yuanxing Liu, Wei-Nan Zhang, Yifan Chen, Yuchi Zhang, Haopeng Bai, Fan Feng, Hengbin Cui, Yongbin Li, and Wanxiang Che. 2023. Conversational Recommender System and Large Language Model Are Made for Each Other in E-commerce Pre-sales Dialogue. *arXiv preprint arXiv:2310.14626* (2023).
- [127] Zhenghao Liu, Zulong Chen, Moufeng Zhang, Shaoyang Duan, Hong Wen, Liangyue Li, Nan Li, Yu Gu, and Ge Yu. 2023. Modeling User Viewing Flow using Large Language Models for Article Recommendation. *arXiv preprint arXiv:2311.07619* (2023).
- [128] Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. 2023. Text Matching Improves Sequential Recommendation by Reducing Popularity Biases. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1534–1544.
- [129] Sichun Luo, Bowei He, Haohan Zhao, Yinya Huang, Aojun Zhou, Zongpeng Li, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2023. RecRanker: Instruction Tuning Large Language Model as Ranker for Top-k Recommendation. *arXiv preprint arXiv:2312.16018* (2023).
- [130] Sichun Luo, Yuxuan Yao, Bowei He, Yinya Huang, Aojun Zhou, Xinyi Zhang, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2024. Integrating Large Language Models into Recommendation via Mutual Augmentation and Adaptive Aggregation. *arXiv:2401.13870* [cs.IR]

- [131] Yucong Luo, Mingyue Cheng, Hao Zhang, Junyu Lu, and Enhong Chen. 2023. Unlocking the Potential of Large Language Models for Explainable Recommendations. *arXiv preprint arXiv:2312.15661* (2023).
- [132] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780* (2023).
- [133] Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559* (2023).
- [134] Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896* (2023).
- [135] Zhiming Mao, Huimin Wang, Yiming Du, and Kam-fai Wong. 2023. UniTRec: A Unified Text-to-Text Transformer and Joint Contrastive Learning Framework for Text-based Recommendation. *arXiv preprint arXiv:2305.15756* (2023).
- [136] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of Hallucination by Large Language Models on Inference Tasks. *arXiv preprint arXiv:2305.14552* (2023).
- [137] Kai Mei and Yongfeng Zhang. 2023. LightLM: A Lightweight Deep and Narrow Language Model for Generative Recommendation. *arXiv preprint arXiv:2310.17488* (2023).
- [138] Frederic P Miller, Agnes F Vandome, and John McBrewhster. 2009. Levenshtein distance: Information theory, computer science, string (computer science), string metric, damerau? Levenshtein distance, spell checker, hamming distance.
- [139] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786* (2022).
- [140] Aashiq Muhamed, Iman Keivanloo, Sujan Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. 2021. CTR-BERT: Cost-effective knowledge distillation for billion-parameter teacher models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*.
- [141] Sheshera Mysore, Andrew McCallum, and Hamed Zamani. 2023. Large Language Model Augmented Narrative Driven Recommendations. *arXiv preprint arXiv:2306.02250* (2023).
- [142] Oded Nov, Nina Singh, and Devin M Mann. 2023. Putting ChatGPT’s medical advice to the (Turing) test. *medRxiv* (2023), 2023–01.
- [143] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.
- [144] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of the web conference 2020*. 1194–1204.
- [145] Bo Peng, Ben Burns, Ziqi Chen, Srinivasan Parthasarathy, and Xia Ning. 2023. Towards Efficient and Effective Adaptation of Large Language Models for Sequential Recommendation. *arXiv preprint arXiv:2310.01612* (2023).
- [146] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Enhong Chen, et al. 2023. Large Language Model based Long-tail Query Rewriting in Taobao Search. *arXiv preprint arXiv:2311.03758* (2023).
- [147] Aleksandr V Petrov and Craig Macdonald. 2023. Generative Sequential Recommendation with GPTRec. *arXiv preprint arXiv:2306.11114* (2023).
- [148] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.
- [149] Tushar Prakash, Raksha Jalan, Brijraj Singh, and Naoyuki Onoe. 2023. CR-SoRec: BERT driven Consistency Regularization for Social Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 883–889.
- [150] Michael J Prince and Richard M Felder. 2006. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of engineering education* 95, 2 (2006), 123–138.
- [151] Sayan Putatunda, Anwesha Bhowmik, Girish Thiruvankadam, and Rahul Ghosh. 2023. A BERT based Ensemble Approach for Sentiment Classification of Customer Reviews and its Application to Nudge Marketing in e-Commerce. *arXiv preprint arXiv:2311.10782* (2023).
- [152] Tao Qi, Fangzhao Wu, Chuhan Wu, Peijie Sun, Le Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2022. Profairrec: Provider fairness-aware news recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1164–1173.
- [153] Junyan Qiu, Haitao Wang, Zhaolin Hong, Yiping Yang, Qiang Liu, and Xingxing Wang. 2023. ControlRec: Bridging the Semantic Gap between Language Model and Personalized Recommendation. *arXiv preprint arXiv:2311.16441* (2023).
- [154] Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: Pre-training user representations for improved recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4320–4327.
- [155] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.
- [156] Zekai Qu, Ruobing Xie, Chaojun Xiao, Yuan Yao, Zhiyuan Liu, Fengzong Lian, Zhanhui Kang, and Jie Zhou. 2023. Thoroughly Modeling Multi-domain Pre-trained Recommendation as Language. *arXiv preprint arXiv:2310.13540* (2023).
- [157] Jakub Raczynski, Mateusz Lango, and Jerzy Stefanowski. 2023. The Problem of Coherence in Natural Language Explanations of Recommendations. *arXiv preprint arXiv:2312.11356* (2023).

- [158] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
- [159] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
- [160] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (jan 2020), 67 pages.
- [161] Behnam Rahdari, Hao Ding, Ziwei Fan, Yifei Ma, Zhuotong Chen, Anoop Deoras, and Branislav Kveton. 2023. Logic-Scaffolding: Personalized Aspect-Instructed Recommendation Explanation Generation using LLMs. *arXiv preprint arXiv:2312.14345* (2023).
- [162] Sajjad Rahmani, AmirHossein Naghshzhan, and Latifa Guerrouj. 2023. Improving Code Example Recommendations on Informal Documentation Using BERT and Query-Aware LSH: A Comparative Study. *arXiv preprint arXiv:2305.03017* (2023).
- [163] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q Tran, Jonah Samost, et al. 2023. Recommender Systems with Generative Retrieval. *arXiv preprint arXiv:2305.05065* (2023).
- [164] Naveen Ram, Dima Kuzmin, Ellie Ka-In Chio, Moustafa Farid Alzantot, Santiago Ontañón, Ambarish Jash, and Judith Yue Li. 2023. Multi-Task End-to-End Training Improves Conversational Recommendation. *ArXiv abs/2305.06218* (2023).
- [165] Xuhui Ren, Tong Chen, Quoc Viet Hung Nguyen, Li zhen Cui, Zi-Liang Huang, and Hongzhi Yin. 2023. Explicit Knowledge Graph Reasoning for Conversational Recommendation. *ArXiv abs/2305.00783* (2023).
- [166] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation Learning with Large Language Models for Recommendation. *arXiv preprint arXiv:2310.15950* (2023).
- [167] Xie Runfeng, Cui Xiangyang, Yan Zhou, Wang Xin, Xuan Zhanwei, Zhang Kai, et al. 2023. Lkpn: Llm and kg for personalized news recommendation framework. *arXiv preprint arXiv:2308.12028* (2023).
- [168] Hitesh Sagtani, Olivier Jeunen, and Aleksei Ustimenko. 2024. Learning-to-Rank with Nested Feedback. *arXiv preprint arXiv:2401.04053* (2024).
- [169] Chandan Kumar Sah, Dr Lian Xiaoli, and Muhammad Mirajul Islam. 2024. Unveiling Bias in Fairness Evaluations of Large Language Models: A Critical Literature Review of Music and Movie Recommendation Systems. *arXiv preprint arXiv:2401.04057* (2024).
- [170] Abel Salinas, Parth Vipul Shah, Yuzhong Huang, Robert McCormack, and Fred Morstatter. 2023. The unequal opportunities of large language models: Revealing demographic bias through job recommendations. *arXiv preprint arXiv:2308.02053* (2023).
- [171] Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language-and item-based preferences. In *Proceedings of the 17th ACM conference on recommender systems*. 890–896.
- [172] J Ben Schafer, Joseph A Konstan, and John Riedl. 2001. E-commerce recommendation applications. *Data mining and knowledge discovery* 5 (2001), 115–153.
- [173] Kaize Shi, Xueyao Sun, Dingxian Wang, Yinlin Fu, Guandong Xu, and Qing Li. 2023. LLaMA-E: Empowering E-commerce Authoring with Multi-Aspect Instruction Following. *arXiv preprint arXiv:2308.04913* (2023).
- [174] Tianhao Shi, Yang Zhang, Zhijian Xu, Chong Chen, Fuli Feng, Xiangnan He, and Qi Tian. 2023. Preliminary Study on Incremental Learning for Large Language Model-based Recommender Systems. *arXiv preprint arXiv:2312.15599* (2023).
- [175] Yubo Shu, Hansu Gu, Peng Zhang, Haonan Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2023. RAH! RecSys-Assistant-Human: A Human-Central Recommendation Framework with Large Language Models. *arXiv preprint arXiv:2308.09904* (2023).
- [176] Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-Shot Recommendation as Language Modeling. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*. Springer, 223–230.
- [177] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.
- [178] Manveer Singh Tamber, Ronak Pradeep, and Jimmy Lin. 2023. Scaling Down, LiTting Up: Efficient Zero-Shot Listwise Reranking with Seq2seq Encoder-Decoder Models. *arXiv e-prints* (2023), arXiv–2312.
- [179] Yading Song, Simon Dixon, and Marcus Pearce. 2012. A survey of music recommendation systems and future perspectives. In *9th international symposium on computer music modeling and retrieval*, Vol. 4. 395–410.
- [180] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).
- [181] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.
- [182] Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Instruction distillation makes large language models efficient zero-shot rankers. *arXiv preprint arXiv:2311.01555* (2023).
- [183] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *arXiv preprint arXiv:2304.09542* (2023).
- [184] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*. 235–244.
- [185] Zhu Sun, Hongyang Liu, Xinghua Qu, Kaidong Feng, Yan Wang, and Yew-Soon Ong. 2023. Large Language Models for Intent-Driven Session Recommendations. *arXiv preprint arXiv:2312.07552* (2023).

- [186] Zhaoxuan Tan and Meng Jiang. 2023. User Modeling in the Era of Large Language Models: Current Research and Future Directions. *arXiv:2312.11518* [cs.CL]
- [187] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360* (2023).
- [188] Zuoli Tang, Zhaoxin Huan, Zihao Li, Xiaolu Zhang, Jun Hu, Chilin Fu, Jun Zhou, and Chenliang Li. 2023. One Model for All: Large Language Models are Domain-Agnostic Recommendation Systems. *arXiv preprint arXiv:2310.14304* (2023).
- [189] Zhen Tian, Changwang Zhang, Wayne Xin Zhao, Xin Zhao, Ji-Rong Wen, and Zhao Cao. 2023. UFIN: Universal Feature Interaction Network for Multi-Domain Click-Through Rate Prediction. *arXiv preprint arXiv:2311.15493* (2023).
- [190] Ghazaleh Haratinezhad Torbati, Anna Tiginova, and Gerhard Weikum. 2023. Unveiling challenging cases in text-based recommender systems. In *3rd Workshop Perspectives on the Evaluation of Recommender Systems*. CEUR-WS. org.
- [191] Ghazaleh Haratinezhad Torbati, Anna Tiginova, Andrew Yates, and Gerhard Weikum. 2023. Recommendations by Concise User Profiles from Review Text. *arXiv preprint arXiv:2311.01314* (2023).
- [192] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [193] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. Neural Discrete Representation Learning. *arXiv:1711.00937* [cs.LG]
- [194] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [195] Chen Wang, Liangwei Yang, Zhiwei Liu, Xiaolong Liu, Mingdai Yang, Yueqing Liang, and Philip S Yu. 2023. Collaborative Contextualization: Bridging the Gap between Collaborative Filtering and Pre-trained Language Model. *arXiv preprint arXiv:2310.09400* (2023).
- [196] Dui Wang, Xiangyu Hou, Xiaohui Yang, Bo Zhang, Renbing Chen, and Daiyue Xue. 2023. Multiple Key-value Strategy in Recommendation Systems Incorporating Large Language Model. *arXiv preprint arXiv:2310.16409* (2023).
- [197] Dong Wang, Kavé Salamatian, Yunqing Xia, Weiwei Deng, and Qi Zhang. 2023. BERT4CTR: An Efficient Framework to Combine Pre-trained Language Model with Non-textual Features for CTR Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5039–5050.
- [198] Dong Wang, Shaoguang Yan, Yunqing Xia, Kavé Salamatian, Weiwei Deng, and Qi Zhang. 2022. Learning Supplementary NLP Features for CTR Prediction in Sponsored Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4010–4020.
- [199] Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2023. FLIP: Towards Fine-grained Alignment between ID-based Models and Pretrained Language Models for CTR Prediction. *arXiv e-prints* (2023), arXiv–2310.
- [200] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 417–426.
- [201] Jian Wang, Dongding Lin, and Wenjie Li. 2022. Follow Me: Conversation Planning for Target-driven Recommendation Dialogue Systems. *ArXiv abs/2208.03516* (2022).
- [202] Jie Wang, Fajie Yuan, Mingyue Cheng, Joemon M Jose, Chenyun Yu, Beibei Kong, Xiangnan He, Zhijin Wang, Bo Hu, and Zang Li. 2022. Transrec: Learning transferable recommendation from mixture-of-modality feedback. *arXiv preprint arXiv:2206.06190* (2022).
- [203] Jinpeng Wang, Ziyun Zeng, Yunxiao Wang, Yuting Wang, Xingyu Lu, Tianxiang Li, Jun Yuan, Rui Zhang, Hai-Tao Zheng, and Shu-Tao Xia. 2023. MISSRec: Pre-training and Transferring Multi-modal Interest-aware Sequence Representation for Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6548–6557.
- [204] Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. 2023. Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint arXiv:2308.15126* (2023).
- [205] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Daxin Jiang, and Kam-Fai Wong. 2022. RecInDial: A Unified Framework for Conversational Recommendation with Pretrained Language Models. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 489–500.
- [206] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot Next-Item Recommendation using Large Pretrained Language Models. *arXiv preprint arXiv:2304.03153* (2023).
- [207] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432* (2023).
- [208] Tingting Wang, Shang-Yu Su, and Yun-Nung (Vivian) Chen. 2022. BARCOR: Towards A Unified Framework for Conversational Recommendation Systems. *ArXiv abs/2203.14257* (2022).
- [209] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
- [210] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji rong Wen. 2023. Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models. *ArXiv abs/2305.13112* (2023).
- [211] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing

- Machinery, 1929–1937.
- [212] Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Y Zhang, Qing Cui, et al. 2023. Enhancing recommender systems with large language model reasoning graphs. *arXiv preprint arXiv:2308.10835* (2023).
  - [213] Yancheng Wang, Ziyang Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. RecMind: Large Language Model Powered Agent For Recommendation. *ArXiv abs/2308.14296* (2023).
  - [214] Yu Wang, Zhiwei Liu, Jiaqiao Zhang, Weiran Yao, Shelby Heinecke, and Philip S Yu. 2023. DRDT: Dynamic Reflection with Divergent Thinking for LLM-based Sequential Recommendation. *arXiv preprint arXiv:2312.11336* (2023).
  - [215] Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. 2023. AnyPredict: Foundation Model for Tabular Prediction. *arXiv preprint arXiv:2305.12081* (2023).
  - [216] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167* (2019).
  - [217] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).
  - [218] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
  - [219] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Llmrec: Large language models with graph augmentation for recommendation. *arXiv preprint arXiv:2311.00423* (2023).
  - [220] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1652–1656.
  - [221] Chuhan Wu, Fangzhao Wu, Tao Qi, Chao Zhang, Yongfeng Huang, and Tong Xu. 2022. MM-Rec: Visiolinguistic Model Empowered Multimodal News Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2560–2564.
  - [222] Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. 2023. Leveraging Large Language Models (LLMs) to Empower Training-Free Dataset Condensation for Content-Based Recommendation. *arXiv preprint arXiv:2310.09874* (2023).
  - [223] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2023. Exploring large language model for graph data understanding in online job recommendations. *arXiv preprint arXiv:2307.05722* (2023).
  - [224] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A Survey on Large Language Models for Recommendation. *arXiv preprint arXiv:2305.19860* (2023).
  - [225] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *arXiv:2303.17564* [cs.LG]
  - [226] Xuansheng Wu, Huachi Zhou, Wenlin Yao, Xiao Huang, and Ninghao Liu. 2023. Towards Personalized Cold-Start Recommendation with Prompts. *arXiv preprint arXiv:2306.17256* (2023).
  - [227] Yunjia Xi, Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Rui Zhang, Ruiming Tang, and Yong Yu. 2023. A Bird’s-eye View of Reranking: from List Level to Page Level. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1075–1083.
  - [228] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. *arXiv preprint arXiv:2306.10933* (2023).
  - [229] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694* (2023).
  - [230] Chen Xu, Wenjie Wang, Yuxin Li, Liang Pang, Jun Xu, and Tat-Seng Chua. 2023. Do LLMs Implicitly Exhibit User Discrimination in Recommendation? An Empirical Study. *arXiv preprint arXiv:2311.07054* (2023).
  - [231] Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Mingchen Cai, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Prompting Large Language Models for Recommender Systems: A Comprehensive Framework and Empirical Analysis. *arXiv:2401.04997* [cs.IR]
  - [232] Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2023. OpenP5: Benchmarking Foundation Models for Recommendation. *arXiv preprint arXiv:2306.11134* (2023).
  - [233] Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving Conversational Recommendation Systems’ Quality with Context-Aware Item Meta-Information. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, 38–48.
  - [234] Shenghao Yang, Chenyang Wang, Yankai Liu, Kangping Xu, Weizhi Ma, Yiqun Liu, Min Zhang, Haitao Zeng, Junlan Feng, and Chao Deng. 2023. Collaborative Word-based Pre-trained Item Representation for Transferable Recommendation. *arXiv preprint arXiv:2311.10501* (2023).
  - [235] Zhengyi Yang, Jiancan Wu, Yanchen Luo, Jizhi Zhang, Yancheng Yuan, An Zhang, Xiang Wang, and Xiangnan He. 2023. Large language model can interpret latent space of sequential recommender. *arXiv preprint arXiv:2310.20487* (2023).
  - [236] Jing Yao, Wei Xu, Jianxun Lian, Xiting Wang, Xiaoyuan Yi, and Xing Xie. 2023. Knowledge Plugins: Enhancing Large Language Models for Domain-Specific Recommendations. *arXiv preprint arXiv:2311.10779* (2023).
  - [237] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. 2023. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003* (2023).
  - [238] Bin Yin, Junjie Xie, Yu Qin, Zixiang Ding, Zhichao Feng, Xiang Li, and Wei Lin. 2023. Heterogeneous knowledge fusion: A novel approach for personalized recommendation via llm. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 599–601.



- [239] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2021. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627* (2021).
- [240] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-supervised learning for recommender systems: A survey. *arXiv preprint arXiv:2203.15876* (2022).
- [241] Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, and Qi Liu. 2022. Tiny-NewsRec: Effective and Efficient PLM-based News Recommendation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5478–5489.
- [242] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835* (2023).
- [243] Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking. *arXiv preprint arXiv:2311.02089* (2023).
- [244] Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. 2023. Linear Recurrent Units for Sequential Recommendation. *arXiv preprint arXiv:2310.02367* (2023).
- [245] Naila Zaafira. 2023. SIAK-NG User Interface Design with Design Thinking Method to Support System Integration. *arXiv preprint arXiv:2309.12316* (2023).
- [246] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*. IEEE, 36–39.
- [247] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. SoundStream: An End-to-End Neural Audio Codec. *arXiv:2107.03312* [cs.SD]
- [248] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414* (2022).
- [249] Jianyang Zhai, Xiaowu Zheng, Chang-Dong Wang, Hui Li, and Yonghong Tian. 2023. Knowledge Prompt-tuning for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6451–6461.
- [250] An Zhang, Leheng Sheng, Yuxin Chen, Hao Li, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2023. On generative agents in recommendation. *arXiv preprint arXiv:2310.10108* (2023).
- [251] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. *arXiv preprint arXiv:2305.07609* (2023).
- [252] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Agentcf: Collaborative learning with autonomous language agents for recommender systems. *arXiv preprint arXiv:2310.09233* (2023).
- [253] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).
- [254] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *IJCAI*. 3356–3362.
- [255] Wenxuan Zhang, Hongzhi Liu, Yingpeng Du, Chen Zhu, Yang Song, Hengshu Zhu, and Zhonghai Wu. 2023. Bridging the Information Gap Between Domain-Specific Model and General LLM for Personalized Recommendation. *arXiv preprint arXiv:2311.03778* (2023).
- [256] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584* (2021).
- [257] Xinyang Zhang, Yury Malkov, Omar Florez, Serim Park, Brian McWilliams, Jiawei Han, and Ahmed El-Kishky. 2022. TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations. *arXiv preprint arXiv:2209.07562* (2022).
- [258] Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023. Variational Reasoning over Incomplete Knowledge Graphs for Conversational Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 231–239.
- [259] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. (2021).
- [260] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488* (2023).
- [261] Zizhuo Zhang and Bang Wang. 2023. Prompt Learning for News Recommendation. *arXiv preprint arXiv:2304.05263* (2023).
- [262] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).
- [263] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Adapting large language models by integrating collaborative semantics for recommendation. *arXiv preprint arXiv:2311.09049* (2023).
- [264] Zhi Zheng, Zhaopeng Qiu, Xiao Hu, Likang Wu, Hengshu Zhu, and Hui Xiong. 2023. Generative job recommendations with large language model. *arXiv preprint arXiv:2307.02157* (2023).
- [265] Aakas Zhiyuli, Yanfang Chen, Xuan Zhang, and Xun Liang. 2023. BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model. *arXiv preprint arXiv:2305.15673* (2023).
- [266] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206* (2023).



- [267] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.
- [268] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph Based Semantic Fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 1006–1014.
- [269] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 4128–4139.
- [270] Peilin Zhou, Meng Cao, You-Liang Huang, Qichen Ye, Peiyan Zhang, Junling Liu, Yueqi Xie, Yining Hua, and Jaeboum Kim. 2023. Exploring recommendation capabilities of gpt-4v (ision): A preliminary case study. *arXiv preprint arXiv:2311.04199* (2023).
- [271] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2023. Collaborative large language model for recommender systems. *arXiv preprint arXiv:2311.01343* (2023).
- [272] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2024. Large Language Models for Information Retrieval: A Survey. [arXiv:2308.07107](#) [cs.CL]
- [273] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Berdersky. 2023. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. *arXiv preprint arXiv:2310.14122* (2023).
- [274] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023. A setwise approach for effective and highly efficient zero-shot ranking with large language models. *arXiv preprint arXiv:2310.09497* (2023).
- [275] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in Baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4014–4022.

## A LOOK-UP TABLE FOR MENTIONED WORKS

To provide easy reference for readers and further facilitate the research community on LLM-enhanced recommender systems, we construct a comprehensive lookup table that contains detailed information for works we mentioned in this paper. As shown in Table 1, we first classify the research works according to the stage where their adapted LLM is involved. Different stages are separated and denoted by different colors. Then, we provide the detailed information for each work (*i.e.*, each row) including but not limited to the size of LLM and the LLM tuning strategy.

Table 1. The look-up table for works on adapting large language models (LLM) to recommender systems (RS) mentioned in this paper. We use the following abbreviations. **FFT**: full finetuning. **PT**: prompt tuning. **LAT**: layerwise adapter tuning. **OT**: option tuning. **T-FEW**: few-shot parameter efficient tuning. Note that only the largest models used in the corresponding papers are listed. If the version of the pretrained language model is not specified, we assume it to be the base version. We use *N/A* to denote works that do not name the proposed method.

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
<b>Feature Engineering (User- and Item-level Feature Augmentation)</b>				
LLM4KGC [13]	PaLM (540B) ChatGPT	Frozen	N/A	E-commerce
TagGPT [85]	ChatGPT	Frozen	Item Tagging	Food, Video
ICPC [20]	LaMDA (137B)	FFT/PT	User Profiling	N/A
KAR [228]	ChatGPT	Frozen	Reranking CTR Prediction Rating Prediction	N/A
PIE [6]	ChatGPT	Frozen	Attribute Extraction	E-commerce
LGIR [35]	GhatGLM (6B)	Frozen	Top-N RS	Job
GIRL [264]	BELLE (7B)	FFT	CTR Prediction	Job
LLM-Rec [132]	text-davinci-003	Frozen	Top-N RS	Movie, Food
HKFR [238]	ChatGPT	Frozen	Top-N RS	E-commerce
LLaMA-E [173]	LLaMA (30B)	LoRA	E-commerce Authoring	E-commerce
EcomGPT [103]	BLOOMZ (7.1B)	FFT	E-commerce NLP Tasks	E-commerce
TF-DCon [222]	ChatGPT	Frozen	Dataset Condensation	Movie, Book, News
RLMRec [166]	ChatGPT	Frozen	Top-N RS	E-commerce, Book, Game
LLMRec [219]	ChatGPT	Frozen	Top-N RS	Movie, Video
LLMRG [212]	GPT4	Frozen	Top-N RS	E-commerce, Movie
CUP [191]	ChatGPT	Frozen	Top-N RS	Book
SINGLE [127]	Vicuna (13B)	Frozen	Sequential RS	News
SAGCN [114]	ChatGPT	Frozen	Top-N RS	E-commerce
UEM [33]	FLAN-T5-base (250M)	FFT	User Profiling	Movie

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
LLMHG [22]	GPT4	FFT	Sequential RS	E-commerce, Movie
Llama4Rec [130]	LLaMA2 (7B)	FFT	Top-N RS Sequential RS Rating Prediction	E-commerce, Movie
<b>Feature Engineering (Instance-level Sample Generation)</b>				
GReaT [5]	GPT2-Medium (355M)	FFT	N/A	Tabular
ONCE [119]	ChatGPT	Frozen	Retrieval Sequential RS	News
AnyPredict [215]	ChatGPT	Frozen	N/A	Tabular
DPLLM [10]	T5-XL (3B)	FFT	Retrieval Privacy	Web Search
MINT [141]	text-davinci-003	Frozen	Narrative-Driven RS	POI
Agent4Rec [250]	ChatGPT	Frozen	RS Simulation	Movie
RecPrompt [113]	GPT4	Frozen	Top-N RS	News
PO4ISR [185]	GPT4	Frozen	Session-based RS	E-commerce, Movie, Game
BEQUE [97]	ChatGLM (6B)	FFT	Query Rewriting	E-commerce
Agent4Ranking [97]	GPT4	Frozen	Query Rewriting	Web Search
<b>Feature Encoder (Representation Enhancement)</b>				
U-BERT [154]	BERT-base (110M)	FFT	Rating Prediction	Business E-commerce
UNBERT [254]	BERT-base (110M)	FFT	Sequential RS	News
PLM-NR [220]	RoBERTa-base (125M)	FFT	Sequential RS	News
Pyramid-ERNIE [275]	ERNIE (110M)	FFT	Ranking	Web Search
ERNIE-RS [124]	ERNIE (110M)	FFT	Retrieval	Web Search
CTR-BERT [140]	Customized BERT (1.5B)	FFT	CTR Prediction	E-commerce
SuKD [198]	RoBERTa-large (355M)	FFT	CTR Prediction	Advertisement
PREC [120]	BERT-base (110M)	FFT	CTR Prediction	News
MM-Rec [221]	BERT-base (110M)	FFT	Sequential RS	News
Tiny-NewsRec [241]	UniLMv2-base (110M)	FFT	Sequential RS	News
PTM4Tag [52]	CodeBERT (125M)	FFT	Top-N RS	posts
TwHIN-BERT [257]	BERT-base (110M)	FFT	Social RS	posts
LSH [162]	BERT-base (110M)	FFT	Top-N RS	Code

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
LLM2BERT4Rec [51]	text-embedding-ada-002	Frozen	Sequential RS	E-commerce
LLM4ARec [91]	GPT2 (110M)	PT	Aspect-based RS	E-commerce, Movie
TIGER [163]	Sentence-T5-base (223M)	Frozen	Sequential RS	E-commerce
TBIN [14]	BERT-base (110M)	Frozen	CTR Prediction	E-commerce
LKPNR [167]	LLaMA2 (7B)	Frozen	Sequential RS	News
SSNA [145]	DistilRoBERTa-base (83M)	LAT	Sequential RS	E-commerce
CollabContext [195]	Instructor-XL (1.5B)	Frozen	Top-N RS	E-commerce
LMIndexer [75]	T5-base (223M)	FFT	Sequential RS Product Search Document Retrieval	E-commerce
Stack [151]	BERT-base (110M)	Frozen	Top-N RS	E-commerce
N/A [58]	BERT-base (110M)	FFT	Sequential RS	POI
UEM [33]	Sentence-T5-base (223M)	Frozen	User Profiling	Movie
Social-LLM [72]	SBERT-MPNet-base (110M)	Frozen	Social RS	Social Network
LLMRS [77]	MPNet (110M)	Frozen	Sequential RS	E-commerce
<b>Feature Encoder (Unified Cross-domain Recommendation)</b>				
ZESRec [31]	BERT-base (110M)	Frozen	Sequential RS	E-commerce
UniSRec [61]	BERT-base (110M)	Frozen	Sequential RS	E-commerce
VQ-Rec [60]	BERT-base (110M)	Frozen	Sequential RS	E-commerce
IDRec vs MoRec [242]	BERT-base (110M)	FFT	Sequential RS	E-commerce, News, Video
TransRec [40]	RoBERTa-base (125M)	LAT	Cross-domain RS Sequential RS	E-commerce, News, Video
TCF [93]	OPT-175B (175B)	Frozen/FFT	Top-N RS	Fashion, News, Video
S&R Foundation [47]	ChatGLM (6B)	Frozen	CTR Prediction Ranking Relevance Prediction	E-commerce
MISSRec [203]	CLIP-B/32 (400M)	FFT	Sequential RS	E-commerce
UFIN [189]	FLAN-T5-base (250M)	Frozen	CTR Prediction	E-commerce, Movie
PMMRec [102]	RoBERTa-large (355M)	Top-2-layer FT	Multi-modal RS	E-commerce, Video
Uni-CTR [42]	Sheared-LLaMA (1.3B)	LoRA	CTR Prediction	E-commerce
<b>Scoring/Ranking Function (Item Scoring Task)</b>				
LMRecSys [259]	GPT2-XL (1.5B)	FFT	Top-N RS	Movie

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
PTab [115]	BERT-base (110M)	FFT	N/A	Tabular
UniTRec [135]	BART (406M)	FFT	Sequential RS	News, Social Media
Prompt4NR [261]	BERT-base/110M	FFT	Sequential RS	News
RecFormer [87]	LongFormer/149M	FFT	Sequential RS	Product
TabLLM [56]	T0 (11B)	T-FEW	N/A	Tabular
Zero-shot GPT [176]	GPT2-Medium (355M)	Frozen	Rating Prediction	Movie
FLAN-T5 [79]	FLAN-T5-XXL (11B)	FFT	Rating Prediction	Book, Movie
BookGPT [265]	ChatGPT	Frozen	Sequential RS Top-N RS Summary Recommendation	Book
TALLRec [3]	LLaMA (7B)	LoRA	Sequential RS	Book, Movie
PBNR [99]	T5-small (60M)	FFT	Sequential RS	News
CR-SoRec [149]	BERT-base (110M)	FFT	Social RS	Social Media, E-commerce
PromptRec [226]	LLaMA (7B)	Frozen	CTR Prediction	E-commerce, Movie
GLRec [223]	BELLE-LLaMA (7B)	LoRA	Top-N RS	Job
BERT4CTR [197]	RoBERTa-large (355M)	FFT	CTR Prediction	Advertisement
ReLLa [111]	Vicuna (13B)	LoRA	Sequential RS	Movie, Book
TASTE [128]	T5-base (223M)	FFT	Sequential RS	E-commerce
N/A [190]	BERT-base (110M)	FFT	Top-N RS	Book
ClickPrompt [107]	RoBERTa-large (355M)	FFT	CTR Prediction	E-commerce, Movie, Book
SetwiseRank [274]	FLAN-T5-XXL (11B)	Frozen	Ranking	Web Search
UPSR [156]	T5-base (223M)	FFT	Sequential RS	E-commerce
LLM-Rec [188]	OPT (6.7B)	LoRA	Sequential RS	E-commerce
LLMRanker [273]	FLAN PaLM2 S	Frozen	Ranking	Web Search
CoLLM [260]	Vicuna (7B)	LoRA	CTR Prediction	Movie, Book
FLIP [199]	RoBERTa-large (355M)	FFT	CTR Prediction	Movie, Book
BTRec [57]	BERT-base (110M)	FFT	Sequential RS	POI
CLLM4Rec [271]	GPT2 (110M)	FFT	Sequential RS	E-commerce
CUP [191]	BERT-base (110M)	Last-layer FT	Top-N RS	Book
N/A [182]	FLAN-T5-XL (3B)	FFT	Ranking	Web Search
CoWPiRec [234]	BERT-base (110M)	FFT	Sequential RS	E-commerce

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
RecExplainer [81]	Vicuna-v1.3 (7B)	LoRA	Sequential RS	E-commerce
E4SRec [96]	LLaMA2 (13B)	LoRA	Sequential RS	E-commerce
CER [157]	GPT2 (110M)	FFT	Explainable RS	E-commerce, Movie
LSAT [174]	LLaMA (7B)	LoRA	Sequential RS	E-commerce, Movie
Llama4Rec [130]	LLaMA2 (7B)	FFT	Top-N RS Sequential RS Rating Prediction	Movie, Book
Scoring/Ranking Function (Item Generation Task)				
GPT4Rec [88]	GPT2 (110M)	FFT	Sequential RS	E-commerce
UP5 [64]	T5-base (223M)	FFT	Retrieval Sequential RS	Movie, Insurance
VIP5 [45]	T5-base (223M)	LAT	Sequential RS Top-N RS Explanation Generation	E-commerce
P5-ID [65]	T5-small (61M)	FFT	Sequential RS	Business E-commerce
FaiRLLM [251]	ChatGPT	Frozen	Top-N RS	Music, Movie
PALR [16]	LLaMA (7B)	FFT	Sequential RS	E-commerce, Movie
ChatGPT-3 [62]	ChatGPT	Frozen	Sequential RS	E-commerce, Movie
AGR [106]	ChatGPT	Frozen	Conversational RS	N/A
NIR [206]	GPT-3 (175B)	Frozen	Sequential RS	Movie
GPTRec [147]	GPT2-medium (355M)	FFT	Top-N RS	E-commerce, Movie, Music
ChatNews [100]	ChatGPT	Frozen	Sequential RS	News
N/A [171]	PaLM (62B)	Frozen	Sequential RS	Movie
LLMSeqPrompt [51]	OpenAI ada model	FT	Sequential RS	E-commerce
GenRec [70]	LLaMA (7B)	LoRA	Sequential RS	E-commerce, Movie
HKFR [238]	ChatGLM (6B)	LoRA	Top-N RS	POI
N/A [170]	ChatGPT	Frozen	Fair RS	Job
BIGRec [2]	LLaMA (7B)	LoRA	Sequential RS	Movie, Game
KP4SR [249]	T5-small (60M)	FFT	Sequential RS	Movie, Music, Book

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
RecSysLLM [21]	GLM (10B)	LoRA	Top-N RS Sequential RS Explanation Generation Review Summarization	E-commerce
POD [89]	T5-small (60M)	FFT	Top-N RS Sequential RS Explanation Generation	E-commerce
N/A [30]	ChatGPT	Frozen	Reranking Top-N RS	Movie, Music, Book
RaRS [29]	ChatGPT	Frozen	Top-N RS	Movie, Book
JobRecoGPT [46]	GPT4	Frozen	Top-N RS	Job
LANCER [73]	GPT2 (110M)	Prefix Tuning	Sequential RS	Movie, Books, News
TransRec [112]	LLaMA (7B)	LoRA	Sequential RS	E-commerce
AgentCF [252]	text-davinci-003 gpt-3.5-turbo	Frozen	Sequential RS	E-commerce
P4LM [69]	PaLM2-XS	FFT	Top-N RS	Movie
InstructMK [196]	LLaMA (7B)	FFT	Top-N RS	Movie
LightLM [137]	T5-small (60M)	FFT	Top-N RS	E-commerce
LlamaRec [243]	LLaMA2 (7B)	QLoRA	Sequential RS	E-commerce, Movie, Game
N/A [270]	GPT-4V	Frozen	Multi-modal RS	Culture, Art, Media Entertainment, Retail
N/A [98]	ChatGPT	gpt-3.5-turbo FT API	Top-N RS	News
N/A [230]	ChatGPT	Frozen	Fair RS	News, Job
LC-Rec [263]	LLaMA (7B)	LoRA	Sequential RS	E-commerce
DOKE [263]	ChatGPT	Frozen	Top-N RS	E-commerce, Movie
ControlRec [153]	T5-base (223M)	FFT	Sequential RS	E-commerce
LLaRA [105]	LLaMA2 (7B)	LoRA	Sequential RS	Movie, Game
PO4ISR [185]	ChatGPT	Frozen	Session-based RS	E-commerce, Movie, Game
DRDT [214]	ChatGPT	Frozen	Sequential RS	E-commerce, Movie
RecPrompt [113]	GPT4	Frozen	Top-N RS	News
LiT5 [178]	T5-XL (3B)	FFT	Ranking	Web Search
STELLA [133]	ChatGPT	Frozen	Sequential RS	Movie, Book, Music, News



Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
Llama4Rec [130]	LLaMA2 (7B)	FFT	Top-N RS Sequential RS Rating Prediction	E-commerce, Movie
Scoring/Ranking Function (Hybrid Task)				
P5 [44]	T5-base (223M)	FFT	Rating Prediction Top-N RS Sequential RS Explanation Generation Review Summarization	E-commerce, Business
M6-Rec [23]	M6-base (300M)	OT	Retrieval Ranking Explanation Generation Conversational RS	E-commerce
InstructRec [253]	Flan-T5-XL (3B)	FFT	Sequential RS Product Search Personalized Search Matching-then-reranking	E-commerce
ChatGPT-1 [116]	ChatGPT	Frozen	Rating Prediction Top-N RS Sequential RS Explanation Generation Review Summarization	E-commerce
ChatGPT-2 [24]	ChatGPT	Frozen	Pointwise Scoring Pairwise Comparison Listwise Ranking	E-commerce, Movie, News
ChatGPT-4 [183]	ChatGPT	Frozen	Passage Reranking	Web Search
BDLM [255]	Vicuna (7B)	FFT	Top-N RS	E-commerce, Movie, Luxury
RecRanker [129]	LLaMA2 (13B)	FFT	Pointwise Scoring Pairwise Comparison Listwise Ranking	Movie, Book
User Interaction (Task-oriented User Interaction)				
TG-ReDial [269]	BERT-base (110M) GPT2 (110M)	Unknown	Conversational RS	Movie
TCP [201]	BERT-base (110M)	FFT	Conversational RS	Movie, Music, Food Restaurant, News, Weather
MESE [233]	DistilBERT (67M) GPT2 (110M)	FFT	Conversational RS	Movie

Table 1 continued from previous page

Model Name	LLM Backbone	LLM Tuning Strategy	RS Task	RS Scenario
UniMIND [27]	BART-base (139M)	FFT	Conversational RS	Movie, Music, Food Restaurant, News, Weather
VRICR [258]	BERT-base (110M)	FFT	Conversational RS	Movie
KECR [165]	BERT-base (110M) GPT2 (110M)	Frozen	Conversational RS	Movie
N/A [55]	GPT4	Frozen	Conversational RS	Movie
MuseChat [34]	Vicuna (7B)	LoRA	Conversational RS	Music, Video
N/A [126]	Chinese-Alpaca (7B)	LoRA	Conversational RS	E-commerce
<b>User Interaction (Open-ended User Interaction)</b>				
BARCOR [208]	BART-base (139M)	Selective-layer FT	Conversational RS	Movie
RecInDial [205]	DialoGPT (110M)	FFT	Conversational RS	Movie
UniCRS [211]	DialoGPT-small (176M)	Frozen	Conversational RS	Movie
T5-CR [164]	T5-base (223M)	FFT	Conversational RS	Movie
TtW [83]	T5-base (223M) T5-XXL (11B)	FFT Frozen	Conversational RS	Music
T5-CR [210]	ChatGPT	Frozen	Conversational RS	Movie, Books, Sports, Music
<b>Pipeline Controller</b>				
Chat-REC [43]	ChatGPT	Frozen	Rating Prediction Top-N RS	Movie
RecLLM [39]	LLaMA (7B)	FFT	Conversational RS	Video
RAH [175]	GPT4	Frozen	Top-N RS	Movie, Book, Game
RecMind [213]	ChatGPT	Frozen	Rating Prediction Top-N RS Sequential RS Explanation Generation Review Summarization	Beauty, Business
InteRecAgent [67]	GPT4	Frozen	Conversational RS	E-commerce, Movie, Game
CORE [76]	N/A	N/A	Conversational RS	E-commerce, Movie, Music, Book