World Wide Web (2024) 27:42
https://doi.org/10.1007/s11280-024-01276-1

When large language models meet personalization: perspectives of challenges and opportunities

Jin Chen[1] · Zheng Liu[2] · Xu Huang[3] · Chenwang Wu[3] · Qi Liu[3] · Gangwei Jiang[3] · Yuanhao Pu[3] · Yuxuan Lei[3] · Xiaolong Chen[3] · Xingmei Wang[3] · Kai Zheng[4] · Defu Lian[3] · Enhong Chen[3]

Received: 22 November 2023 / Revised: 15 March 2024 / Accepted: 14 May 2024 /
Published online: 28 June 2024

Abstract
The advent of large language models marks a revolutionary breakthrough in artificial intelligence. With the unprecedented scale of training and model parameters, the capability of large language models has been dramatically improved, leading to human-like performances in understanding, language synthesizing, common-sense reasoning, etc. Such a major leap forward in general AI capacity will fundamentally change the pattern of how personalization is conducted. For one thing, it will reform the way of interaction between humans and personalization systems. Instead of being a passive medium of information filtering, like conventional recommender systems and search engines, large language models present the foundation for active user engagement. On top of such a new foundation, users' requests can be proactively explored, and users' required information can be delivered in a natural, interactable, and explainable way. For another thing, it will also considerably expand the scope of personalization, making it grow from the sole function of collecting personalized information to the compound function of providing personalized services. By leveraging large language models as a general-purpose interface, the personalization systems may compile user's requests into plans, calls the functions of external tools (e.g., search engines, calculators, service APIs, etc.) to execute the plans, and integrate the tools' outputs to complete the end-to-end personalization tasks. Today, large language models are still being rapidly developed, whereas the application in personalization is largely unexplored. Therefore, we consider it to be right the time to review the challenges in personalization and the opportunities to address them with large language models. In particular, we dedicate this perspective paper to the discussion of the following aspects: the development and challenges for the existing personalization system, the newly emerged capabilities of large language models, and the potential ways of making use of large language models for personalization.

Keywords Large language models · Personalization systems · Recommender systems · Tool-learning · AIGC

Jin Chen and Zheng Liu contributed equally to this work.

Extended author information available on the last page of the article

1 Introduction

The emergence of large language models [1], which have demonstrated remarkable progress in understanding human expression, is profoundly impacting the AI community. These models, equipped with vast amounts of data and large-scale neural networks, exhibit impressive capabilities in comprehending human language and generating text that closely resembles our own. Among these abilities are reasoning [2], few-shot learning [3], and the incorporation of extensive world knowledge within pre-trained models [1]. This marks a significant breakthrough in the field of artificial intelligence, leading to a revolution in our interactions with machines. Consequently, large language models have become indispensable across various applications, ranging from natural language processing and machine translation to creative content generation and chatbot development. The introduction of ChatGPT, in particular, has gained significant attention from the human community, prompting reflections on the transformative power of large language models and their potential to push the boundaries of what artificial intelligence (AI) can achieve. This disruptive technology holds the promise of transforming how we interact with and leverage AI in countless domains, opening up new possibilities and opportunities for innovation. As these language models continue to advance and evolve, they are likely to shape the future of artificial intelligence, empowering us to explore uncharted territories and unlock even greater potential in human-machine collaboration.

Personalization, the art of tailoring experiences to individual preferences, stands as an essential and dynamic connection that bridges the gap between humans and machines. In today's technologically driven world, personalization plays a pivotal role in enhancing user interactions and engagements with a diverse array of digital platforms and services. By adapting to individual preferences, personalization systems empower machines to cater to each user's unique needs, leading to more efficient and enjoyable interactions. Moreover, personalization goes beyond mere content recommendations; it encompasses various facets of user experiences, encompassing user interfaces, communication styles, and more. As artificial intelligence continues to advance, personalization becomes increasingly sophisticated in handling large volumes of interactions and diverse user intents. This calls for the development of more advanced techniques to tackle complex scenarios and provide even more enjoyable and satisfying experiences. The pursuit of improved personalization is driven by the desire to better understand users and cater to their ever-evolving needs. As technology evolves, personalization systems will likely continue to evolve, ultimately creating a future where human-machine interactions are seamlessly integrated into every aspect of our lives, offering personalized and tailored experiences that enrich daily routines.

Large language models, with their deep and broad capabilities, have the potential to revolutionize personalization systems, transforming the way humans interact and expanding the scope of personalization. The interaction between humans and machines can no longer be simply classified as active and passive, just like traditional search engines and recommendation systems. However, these large language models go beyond simple information filtering and they offer a diverse array of additional functionalities. Specifically, user intent will be actively and comprehensively explored, allowing for more direct and seamless communication between users and systems through natural language. Unlike traditional technologies that rely on abstract and less interpretable ID-based information representation, large language models enable a more profound understanding of users' accurate demands and interests. This deeper comprehension paves the way for higher-quality personalized services, meeting users' needs and preferences in a more refined and effective manner. Moreover, the integration of

various tools is greatly enhanced by the capabilities of large language models, significantly broadening the possibilities and scenarios for personalized systems. By transforming user requirements into plans, including understanding, generating, and executing them, users can access a diverse range of information and services. Importantly, users remain unaware of the intricate and complex transformations happening behind the scenes, as they experience a seamless end-to-end model. The potential of large language models in personal is largely unexplored.

This paper addresses the challenges in personalization and explores the potential solutions using large language models. In the existing related work, LaMP [4] introduces a novel benchmark for training and evaluating language models in producing personalized outputs for information retrieval systems. On the other hand, other related surveys [5–7] focus mainly on traditional personalization techniques, such as recommender systems. From the perspective of learning mechanisms, LLM4Rec [5] delves into both Discriminative LLM for Recommendation and Generative LLM for Recommendation. Regarding the adaptation of LLM for recommender systems in terms of 'Where' and 'How', Li et al [6] concentrate on the overall pipeline in industrial recommender phases. Fan et al [7], on the other hand, conduct a review with a focus on pre-training, fine-tuning, and prompting approaches. While these works discuss pre-trained language models like Bert for ease of analysis, they dedicate limited attention to the emergent capabilities of large language models. This paper aims to fill this gap by examining the unique and powerful abilities of large language models in personalization, and further expand the scope of personalization with tools.

The remaining of this survey is organized as follows: we review the personalization and large language models in Section 2 to overview the development and challenges. Then we carefully discuss the potential actors of large language models for personalization from Section 3, following the simple utilization of emergent capabilities and the complex integration with other tools. We also discuss the potential challenges when large language models are adapted for personalization. The whole architecture of this paper is shown in Figure 1.

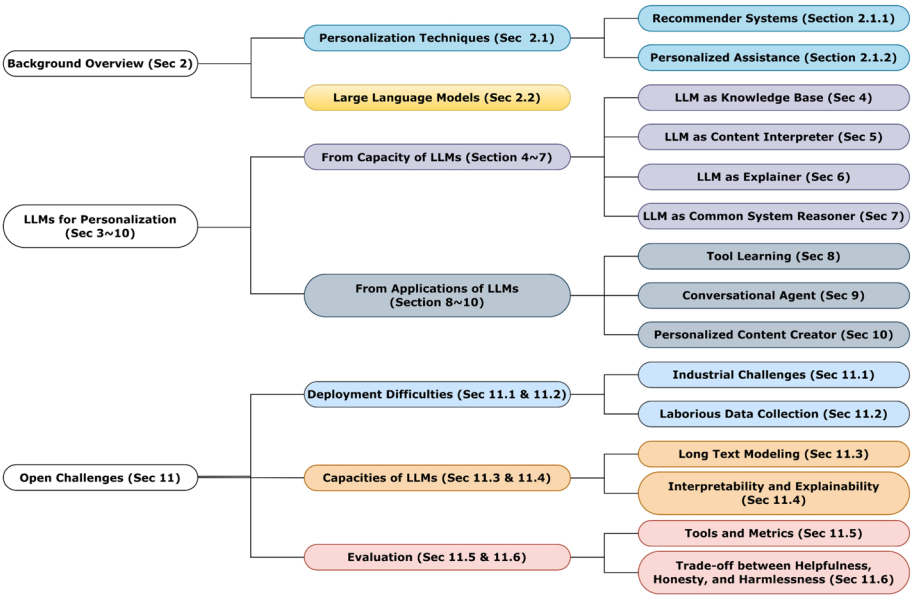

Figure 1 Overview of this paper

2 Background overview

2.1 Personalization techniques

Personalization, a nuanced art that tailors experience to the unique preferences and needs of individual users, has become a cornerstone of modern artificial intelligence. In this section, we explore the captivating world of personalized techniques and their profound impact on user interactions with AI systems. We will delve into two key aspects of personalization: recommender systems and personalized assistance. These techniques not only enhance user satisfaction but also exemplify the evolution of AI, where machines seamlessly integrate with our lives, understanding us on a profound level. By tailoring recommendations, providing customized assistance, and delivering personalized search results, AI systems have the potential to create a truly immersive and individualized user experience.

2.1.1 Recommender systems

Recommender systems play a pivotal role in personalization, revolutionizing the way users discover and engage with content. These systems aim to predict and suggest items of interest to individual users, such as movies, products, or articles, based on their historical interactions and preferences.

Regarding the development of recommender systems, they have evolved significantly over the years, with collaborative filtering [8, 9] being one of the earliest and most influential approaches. Collaborative filtering relies on user-item interaction data to identify patterns and make recommendations based on users with similar preferences. Traditional solutions, such as matrix factorization [10] and user/item-based approaches [11], extract potentially interesting items based on the idea that users who have shown similar preferences in the past are likely to have similar preferences in the future. While effective, collaborative filtering has limitations, such as the "cold start" problem for new users and items. To address these limitations, content-based filtering [12] emerged, which considers the content of items to make recommendations. It leverages the features and attributes of items to find similarities and make personalized suggestions. These features can be grouped into user-side information, such as user profiles, item-side information [13, 14], such as item brands and item categories, and interaction-based information [15], such as reviews and comments. However, content-based filtering may struggle to capture complex user preferences and discover diverse recommendations restricted by the limited feature representations.

In recent years, deep learning has gained significant attention in the field of recommender systems due to its ability to model complex patterns and interactions in user-item data [16]. Deep learning-based methods have shown promising results in capturing sequential, temporal, and contextual information, as well as extracting meaningful representations from large-scale data. With the introduction of deep networks, high-order interactions between features of users and items are well captured to extract user interest. Deep learning-based methods offer approaches to capture high-order interactions by employing techniques like attention mechanisms [17, 18] and graph-based networks [19] to mine complex relationships between user and item. These methods have been shown to enhance recommendation performance by considering higher-order dependencies and inter-item relationships. Another area of deep learning-based recommender systems is sequential recommenders, specifically designed to handle sequential user-item interactions, such as user behavior sequences over time. Self-Attentions [20] and Gated Recurrent Units (GRUs) [21] are popular choices for

modeling sequential data in recommender systems. These models excel in capturing temporal dependencies and context, making them well-suited for tasks like next-item recommendation and session-based recommendation. Sequential-based models can take into account the order in which items are interacted with and learn patterns of user behavior that evolve. Furthermore, the rise of language models like BERT has further advanced recommender systems by enabling a better understanding of both natural language features and user sequential behaviors [22]. These language models can capture deep semantic representations and world knowledge, enriching the recommendation process and facilitating more personalized and context-aware recommendations. Overall, the application of deep learning techniques in recommender systems has opened new avenues for research and innovation, promising to revolutionize the field of personalized recommendations and enhance user experiences.

2.1.2 Personalized assistance

Personalization Assistance refers to the use of artificial intelligence and machine learning techniques to tailor and customize experiences, products, or content based on individual preferences, behavior, and characteristics of users [23]. By analyzing individual preferences, behaviors, and characteristics, it creates a personalized ecosystem that enhances user engagement and satisfaction. In contrast to traditional recommender systems, which rely on predicting user interests passively, personalized assistance takes a more proactive approach. It ventures into the realm of predicting users' next intentions or actions by utilizing contextual information, such as historical instructions and speech signals. This deeper level of understanding enables the system to cater to users' needs in a more anticipatory and intuitive manner. At the core of this capability lies the incorporation of cutting-edge technologies like natural language processing (NLP) and computer vision. These advanced tools empower the system to recognize and interpret user intentions, whether conveyed through spoken or written language, or even visual cues. Moreover, the potential of personalized assistance extends beyond static recommendations to dynamic and context-aware interactions. As the system becomes more and more familiar with a user's preferences and patterns, it adapts and refines its recommendations in real-time, keeping pace with the ever-changing needs and preferences of the user.

Conversational Recommender Systems [24, 25] mark a remarkable stride forward in the realm of personalized assistance. By engaging users in interactive conversations, these systems delve deeper into their preferences and fine-tune their recommendations accordingly. Leveraging the power of natural language understanding, these conversational recommenders adeptly interpret user queries and responses, culminating in a seamless and engaging user experience. Notable instances of personalized assistance products, such as Siri [1] and Microsoft Cortana[2], have already proven their effectiveness on mobile devices. Additionally, the integration of large language models like ChatGPT further elevates the capabilities of conversational recommenders, promising even more enhanced user experiences. As this technology continues to progress, we can anticipate its growing significance across diverse industries, including healthcare, education, finance, etc. While the growth of conversational recommenders and personalized assistance promises immense benefits, it is imperative to develop these products responsibly. Upholding user privacy and ensuring transparent data handling practices are essential to maintain user trust and safeguard sensitive information.

[1] https://www.apple.com/siri/

[2] https://www.microsoft.com/en-us/cortana

2.2 Large language models

Language models perform the probabilistic modeling for the generation of natural language [26, 27], i.e., presented with one specific context, the language models make predictions for the words which are to be generated for the future steps. Nowadays, language models are mostly built upon deep neural networks, where two features need to be emphasized. First of all, the majority of language models are based on transformers or their close variations [28, 29]. Such types of neural networks are proficient at modeling context dependency within natural languages and exhibit superior and consistently improved performances when being scaled up. Secondly, the language models are pre-trained at scale with a massive amount of unlabeled corpus [3, 30–32]. The pre-trained models are further fine-tuned with task-oriented data to adapt to different downstream applications.

There has been tremendous progress in language models in recent years, where the emergence of large language models, represented by GPT-3 [3], marks an important milestone for the entire AI community. The large language models (LLMs), as the name suggests, are massively scaled-up derivatives of conventional language models. Particularly, the backbone networks and the training data have been largely magnified. For one thing, although there are no specific criteria for the minimum number, a typical LLM usually consists of no less than several billion and up to trillions of model parameters, which are orders larger than before [33]. For another thing, the pre-training is conducted based on much more unsupervised corpora, with hundreds of billions or trillions of tokens carefully filtered from sources like Common Crawl, GitHub, Wikipedia, Books, ArXiv, etc [1]. The impact of scaling is illustrated by the scaling laws [34, 35], which numerically uncover the power-law relationship between model size, data volume, training scale and the growth of the model's performance.

The scaling up of network and training data lead to the leap-forward of large language models' capability. They not only become more proficient at conventional skills, like understanding people's intent and synthesising human-like languages, but also process capabilities which are rarely exhibited by those smaller models. Such a phenomenon is referred as the emergent abilities of LLMs, where three representatives capabilities are frequently discussed. One is the in-context learning capability [3], where LLMs may quickly learn from the few-shot examples provided in the prompt. Another one is the instruct following capability [36, 37]. After fine-tuned with diversified tasks in the form of instruction tuning, the LLMs are made proficient to follow the human's instructions. Thus, they may handle different tasks presented in an ad-hoc manner. Last but not least, LLMs are found to be able to conduct step-by-step reasoning [38, 39]. With certain types of prompting strategies, like Chain-of-Thought (CoT), LLMs may iteratively approach the final answer of some complex tasks, like mathematical word problems, by breaking down the tasks into sub-problems and figuring out the plausible intermediate answers for each of the sub-problems.

Thanks to the superior capabilities of understanding, reasoning, and generating, large language models, especially the chat models produced by instruction tuning, are presented as foundamental building blocks for many personalization services. One direct scenario is the conversational search and recommendation [40]. Once built upon large language models, the search and recommendation systems will be able to engage with users via interactions, present outputs in a verbalized and explainable way, receive feedback from the user and make adjustments on top of the feedback, etc. The above changes will bring about a paradigm shift for personalization services, from passively making search and recommendation, to proactively figuring out user's need and seeking for user's preferred items. In broader scopes, the LLMs may go beyond simply making personalized search and recommendation, but play as personalized assistants to help users with their task completions. The LLMs may take

notes of users' important information within their memory, make personalized plans based on memorized information when new demands are raised, and execute plans by leveraging tools like search engines and recommendation systems.

Yet, we have to confront the reality that applying LLMs for personalization is not a trivial problem. To name a quite few of the open challenges. Firstly, personalization calls for the understanding of user preference, which is more domain-specific knowledge rather than the common-sense knowledge learned by LLMs. The effective and efficient adaptation of LLMs for personalized services remains to be resolved. Besides, the LLMs could memorize user's confidential information while providing personalized services. Thus, it raises concerns for privacy protection. The LLMs are learned from Internet data; due to the exposure bias, it is almost inevitable to make unfair predictions for minorities. To address the above challenges, benchmarks and evaluation datasets are needed by the research communities. However, such resources are far from complete at present. To fully support personalization with LLMs, methodological and experimental frameworks need to be systematically established for all these perspectives.

3 LLMs for personalization

In the following sections, we delve into the potential of large language models for personalization, examining their evolution from simple use cases, like utilizing word knowledge as features, to more intricate integration with other tool modules to act as agents. Specifically, we focus on the progression of emergent capabilities, starting from basic world knowledge and understanding user intent, and advancing to high-level reasoning abilities. We explore how large language models can contribute to constructing a knowledge base that enriches common-sense knowledge about various items. Additionally, we discuss how the understanding capability of large language models can empower content interpreters and explainers for in-depth analysis of interactions. Furthermore, we observe attempts to leverage the reasoning ability of large language models for system reasoners to provide recommendation results. These increasingly sophisticated capabilities enable complex utilization of large language models with other tool modules, enabling them to better comprehend user intentions and fulfill user instructions. Consequently, we also explore the integration of large language models with other tools for personalization, including tool learning, conversational agents and personalized content creators. The overview of this chapter is depicted in Figure 2. Our comprehensive survey aims to provide a deeper understanding of the current landscape, shedding light on the opportunities and challenges associated with incorporating large language models into personalization.

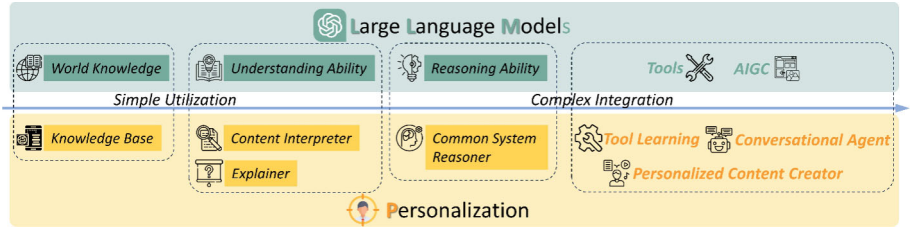

Figure 2 The overview of LLM for personalization

4 LLMs as knowledge base

The knowledge base provides rich information with semantics, attracting increasing attention for the usage of the knowledge base in the recommender systems. Particularly, the knowledge graphs, where nodes represent entities and edges represent relations in the heterogeneous information graph, are the common format of knowledge bases and are introduced as side information to enhance the performance of recommenders. Knowledge graphs help understand the mutual relations between users and items and also provide better explainability for recommenders. Existing methods that incorporate knowledge graphs in recommender systems can be classified into three main groups: embedding-based methods, path-based methods and unified methods. Embedding-based methods, such as CKE [14] and DKN [41], KSR [42], SHINE [43], utilize semantic representations of users and items. These methods aim to capture the underlying semantic relationships between entities in the knowledge graph, which can improve the quality of recommendations. Path-based approaches, such as Hete-MF [44], SemRec [45], RuleRec [46], EIUM [47], exploit the semantic connectivity information present in the knowledge graph to regularize the user and item representations. These methods consider the paths between users and items in the graph and leverage them to incorporate interpretability into the recommendation process. Unified methods, such as RippleNet [48], KGCN [49], KGAT [50], AKUPM [51], IntentGC [52] refine the representations of entities in the knowledge graph by leveraging embedding propagation techniques. These methods propagate the embeddings of entities through the graph structure, allowing information to flow across connected entities and refining the representations accordingly.

However, the knowledge graphs adopted in recommender systems are limited and with low usability. Reviewing the various knowledge graph datasets for recommender systems, covering the domains of movies, books, news, products, etc., these datasets are still significantly sparse compared to the vast amount of human knowledge, particularly the lack of facts, due to the expensive supervision to construct the knowledge graph. Building a comprehensive and accurate knowledge graph would be a complex and resource-intensive task, which would include data collection, integration, and cleaning to assure data quality and consistency. Limited by the expensive cost of labelling the knowledge graphs, there would usually exist missing entities or relations. The user preferences for these entities or paths may be ignored, and the recommendation performance suffers.

The ability of Large Language Models to retrieve factual knowledge as explicit knowledge bases [53–56, 56–61] has been stirred discussed, which presents an opportunity to construct more comprehensive knowledge graphs within recommender systems. Tracing back to the work [53], large language models have shown their impressive power in storing factual information, such as entities and common sense, and then commonsense knowledge can be reliably transferred to downtown tasks. **Existing methods in knowledge graphs fall short of handling incomplete KGs** [62] **and constructing KGs with text corpus** [63] and many researchers attempt to leverage the power of LLM to solve the two tasks, i.e., the knowledge completion [64] and knowledge construction [65]. **For knowledge graph completion**, which refers to the task of missing facts in the given knowledge graph, recent efforts have been made to encode text or generate facts for knowledge graphs. MTL-KGC [66] encoders the text sequences to predict the possibility of the tuples. MEMKGC [67] predicts the masked entities of the triple. StAR [68] utilizes Siamese textual encoders to separately encode the entities. GenKGC [69] uses the decoder-only language models to directly generate the tail entity. TagReal [70] generates high-quality prompts from the external text corpora. AutoKG [63] directly adopts the LLMs, such as ChatGPT and GPT-4, and designs tailored

prompts to predict the tail entity. As for the another important task, i.e., **knowledge graph construction**, which refers to creating a structured representation of knowledge, LLMs can be applied in the process of constructing knowledge graphs, including entity discovery [71, 72], coreference resolution [73, 74] and relation extraction [75, 76]. LLMs can also achieve the end-to-end construction [57, 65, 70, 77, 78] to directly build KGs from raw text. LLMs enable the knowledge distillation to construct knowledge graphs. symbolic-kg [79] distills commonsense facts from GPT3 and then finetunes the small student model to generate knowledge graphs. These models have demonstrated the capacity to store large volumes of knowledge, providing a viable option for improving the scope and depth of knowledge graphs. Furthermore, these advancements have prompted research into the direct transfer of stored knowledge from LLMs to knowledge graphs, eliminating the need for human supervision. This interesting research throws light on the possibilities of automating knowledge graph completion utilizing cutting-edge big language models.

By leveraging the capabilities of LLMs, recommender systems would benefit from a more extensive and up-to-date knowledge base. Firstly, missing faculty information can be completed to construct more extensive knowledge graphs and thus the relations between entities can be extracted for better recommenders. Secondly, in contrast to the preceding exclusively in-domain data, the large language model itself contains plenty of cross-domain information that can help achieve cross-domain recommendations, such as recommending appropriate movies based on the user's favourite music songs. To sum up, the stored knowledge can be utilized to enhance recommendation accuracy, relevance, and personalization, ultimately improving the overall performance of recommender systems. Existing work [80] prompts large language models to generate factual knowledge about movies to enhance the performance of CTR prediction models. To better utilize the factual knowledge, a *Knowledge Adaptation* module is adopted for better contextual information extraction. LLMRec [81] adopts LLM-based graph augmentation strategies to enrich the information of items and further designs denoise methods to ensure the augmentation. LLM-KRec [82] adopts the large language models to determine complementary relationships in each entity pair and construct a complementary knowledge graph, which enhances the industrial recommenders. The existing work leveraging the capabilities of knowledge base in personalization is summerized in Table 1.

It is worth noting that the **phantom** problem of large language models can be a challenge when applied to recommendation tasks. The inherent nature of large language models can introduce ambiguity or inaccurate provenance [83]. This issue can emerge as the introduction of extraneous information or even noise into the recommendation process. The large language models may generate responses that, while syntactically correct, lack informative context or relevance. According to the KoLA [84], a benchmark for evaluating word knowledge of LLMs, even the top-ranked GPT4 just achieves 0.012 in Precision and 0.013 in Recall on the task *Named Entity Recognition*, which falls far short of the performance (0.712 in Precision and 0.706 in Recall) of the task-specific models PL-Marker [85]. Such a finding suggests that common sense is still far from being sufficiently captured by LLM.

5 LLMs as content interpreter

Content-based recommenders provide an effective solution for mitigating the sparse feedback issue in recommender systems. By leveraging the attributes and characteristics of items, these systems achieve a more profound understanding of their properties, facilitating accurate

Table 1 LLMs as knowledge base

Approach	Knowledge	Task	LLM backbone	Datasets
KAR [80]	Factual knowledge	CTR prediction & Rerank	gpt-3.5-turbo	Ml-1M, Amazon Book
LLMRec [81]	User-item interactions, side information	TopK Recommendation	gpt-3.5-turbo	ML, Netflix
KRec [82]	Complementary relationship	CTR Prediction	ChatGPT 3.5, ChatGLM 2	Alipay Data

matching with user preferences. However, the content features used in content-based recommendation may also exhibit sparsity. Relying solely on the recommended supervision signal, such as clicking and browsing, might not fully exploit the potential benefits of these features. To overcome this challenge, language models emerge as powerful fundamental algorithms that act as content interpreters in processing textual features. Their utilization enhances the effectiveness of recommender systems by effectively understanding and interpreting textual content, leading to improved recommendations.

5.1 Conventional content interpreter

Conventional content interpreter includes statistical model, neural networks, and advanced NLP networks, as summarized in Figure 3. These approaches primarily focus on transforming content information, such as textual data, into feature embeddings to facilitate the recommendation process.

Statistical models like TF-IDF, Minimum Description Length (MDL) [86], and bag-of-words have been traditionally used to encode textual data such as news articles and documents into continuous value vectors. However, with the advancement of deep learning techniques, researchers have explored various neural network architectures to learn more expressive content representations. Instead of relying solely on statistical embeddings, some approaches initialize the vectors with bag-of-words representations and then employ autoencoder-based models to learn more powerful representations. For example, CDL [16] combines the latent vectors obtained from autoencoders with the original ID embeddings to enhance content representations. CRAE [87] introduces a collaborative recurrent autoencoder that captures the word order in texts, enabling the modeling of content sequences in collaborative filtering scenarios. Dong et al. [88] propose a stacked denoising autoencoder that reconstructs item/user ratings and textual information simultaneously, allowing for the joint modeling of collaborative and textual knowledge. CVAE [89] introduces a collaborative variational autoencoder that learns probabilistic textual features. While autoencoders are effective in learning low-dimensional representations from text data, they may struggle to capture semantic information effectively [90]. In some cases, approaches like doc2vec [91] are used to construct content embeddings [92, 93] and learn hidden representations. Okura et al. [94] evaluate different network architectures, including word-models and GRU networks, for representing user states.

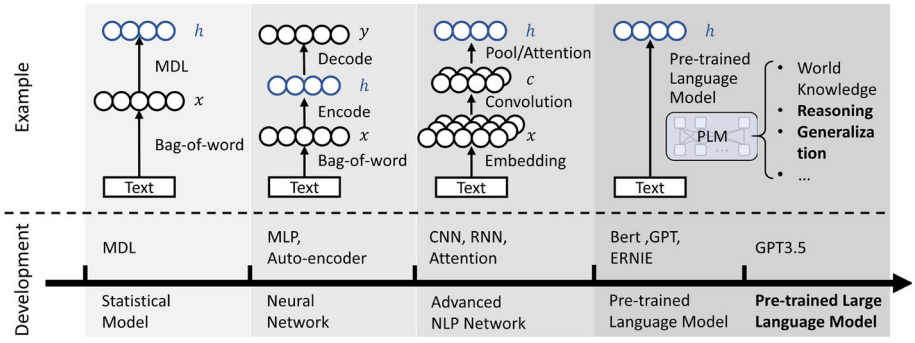

Figure 3 The development of content interpreter in recommendation

Following the advancements in neural natural language processing (NLP) models, more sophisticated architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Neural Attention models have been employed as content interpreters to extract contextual information and capture user preferences. These models take sentence inputs, such as news titles, reviews, or comments, and transform them into word embedding matrices using random initialization or word2vec embeddings [95]. Various architectures, including CNNs, attention networks, and hybrid models, are utilized to learn representations of sentences. For example, NPA [96] and LSTUR [97] incorporate attention mechanisms to determine the importance of words after CNN layers. NRMS [98] and CPRS [99] utilize multi-head self-attention networks to learn word representations. These models are effective in capturing long-context dependencies and understanding the semantic information in the text. In addition to text modeling, language models are also used as content interpreters to capture user interests based on their historical interactions. For instance, WE3CN [100] employs a 3D CNN to extract temporal features from the historical data. DKN [41] utilizes an attention mechanism to aggregate historical information related to candidate items. DAN [101] proposes an attention-based LSTM to capture richer hidden sequence features. These models leverage different neural architectures to enhance the representation of text in the context of recommendation systems. It is worth noting that these models still have limitations in terms of depth and the ability to effectively generalize semantic information.

5.2 Language model based content interpreter

In recent years, there has been a growing interest in incorporating more powerful pre-trained language models, such as BERT and GPT, into recommendation systems. These language models have shown exceptional performance in various natural language processing tasks and have sparked researchers' inspiration to leverage them for capturing deep semantic representations and incorporating world knowledge in recommendation systems. However, applying pre-trained language models to recommendation tasks presents two main challenges. Firstly, there is a misalignment of goals between general-purpose language models and the specific objectives of recommendation systems. To address this, researchers have proposed approaches that fine-tune the pre-trained models or design task-specific pre-training tasks to adapt them to recommendation tasks. For example, U-BERT [102] employs BERT as a content interpreter and introduces masked opinion token prediction and opinion rating prediction as pre-training tasks to better align BERT with recommendation objectives. Similarly, other works [103–107] have utilized pre-trained BERT to initialize the news encoder for news recommendation, enhancing the representation of textual features. The pre-trained model, ERNIE, is also utilized to enhance the representation ability of queries and documents [108, 109]. The second challenge is reducing the online inference latency caused by pre-trained language models, which can be computationally expensive. Researchers have explored techniques such as knowledge distillation and model optimization to obtain lightweight and efficient models suitable for online services. For instance, CTR-BERT [110] employs knowledge distillation to obtain a cache-friendly model for click-through rate prediction, addressing the latency issue.

Moreover, pre-trained language models have been applied beyond mainstream recommendation tasks. They have been integrated into various recommendation scenarios, including tag recommendation [111], tweet representations [112], and code example recommendation [113], to enhance the representation of textual features in those specific domains.

Additionally, some recent works [114–117] have explored using only textual features as inputs to recommendation models, leveraging pre-trained language models to alleviate cold-start problems and enable cross-domain recommendations. This paradigm offers advantages in alleviating cold-start problems and facilitating cross-domain recommendations based on the universality of natural language. ZESREC [114] uses BERT to obtain universal continuous representations of item descriptions for zero-shot recommendation. Unisrec [115] focuses on cross-domain sequential recommendation and employs a lightweight MoE-enhanced module to incorporate the fixed BERT representation into the recommendation task. VQ-Rec [116] further aligns the textual embeddings produced by pre-trained language models to the recommendation task with the help of vector quantization. Fu et al. [118] explore layerwise adaptor tuning to achieve parameter-efficient transferable recommendations.

While the pre-trained language models empower the text understanding with the benefit of capturing world knowledge first, the development of pre-trained large language models provides great emergency ability in the fields of reasoning and generalization, as shown in Table 2. TALLRe [119] explores the ability of large language models for the sequential recommendation. They observe that original language models perform poorly in zero-shot and few-shot scenarios, while recommendation-specific instruction-tuned language models demonstrate superior performance in few-shot learning and cross-domain generalization. Li et al. [123] adopt the large language models as context encoders and attempt to examine the upper limits of the large language models. Similarly, Kang et al. [120] propose a similar instruction tuning method for rating prediction recommendation tasks based on the T5 backbone. They find that the tuned language models, which leverage data efficiency, outperform traditional recommenders. PALR [121] further enhances the construction pipeline of recommendation-specific instruction tuning, which first employs large language models to generate reasoning as additional features based on the user's behavior history. Next, a small set of candidates is retrieved using any existing model based on the user profile. Finally, to adapt general-purpose language models to the recommendation task, they convert the generated reasoning features, user interaction history, and retrieved candidates into natural language instruction data and fine-tune a language model. Existing instruction tuning methods of language models for recommendation scenarios typically focus on a single type of recommendation task, limiting the full utilization of language models' strong generalization ability. InstructRec [122] addresses this limitation by formulating recommendation as an instruction-following procedure. They design various instruction templates to accommodate different recommendation tasks and employ GPT-3.5 to generate high-quality instruction data based on the historical data of users and templates. The language models fine-tuned with this instruction data can effectively handle a wide range of recommendation tasks and cater to diverse information requirements from different users.

6 LLMs as explainer

In addition to valuing the suggestions made by a recommendation model, users are also interested in the comprehensible justifications for these recommendations [124, 125]. This is crucial as most recommender systems are black boxes whose inner workings are inscrutable to human understanding [126], diminishing user trust. Taking drug recommendations, for instance, it is unacceptable to recommend drugs with good curative effects simply but fail to give reasons why they are effective. To this end, explainable recommendations aim to couple high-quality suggestions with accessible explanations. This not only helps to improve the

Table 2 LLMs for content interpreter

Approach	Task	LLM backbone	Tuning strategy	Datasets
TALLRe [119]	Sequential recommendation	LLaMA-7B	Instruct tuning & Fine tuning	MovieLens100k, BookCrossing
LLMs-Rec [120]	Rating prediction	Flan-T5-Base, Flan-T5-XXL	Fine tuning	MovieLens-1M, Amazon book
PALR [121]	Item recommendation	LLaMa-7B	Instruction tuning	MovieLens-1M, Amazon beauty
InstructRec [122]	Sequential recommendation personalized search	Flan-T5-XL	Instruction tuning	Amazon-Games, CDs

model's transparency, persuasiveness, and reliability, but also facilitates the identification and rectification of potential errors through insightful explanations. These benefits have been extensively documented in recent work [38, 127–129]. For instance, [128] conducted a study that involved addressing 40 difficult tasks and evaluating the impact of explanations on zero-shot and few-shot scenarios. Their findings demonstrated that explanations have a positive effect on model performance by establishing a connection between examples and interpretation.

Traditional approaches mainly focus on template-based explanations, which can be broadly categorized into item-based, user-based, and attribute-based explanations [130]. Item-based explainable methods relate recommendations to familiar items [131], explaining that *the recommended item bears similarity to others the user prefers*, which are prevalent on platforms like Amazon [132] and Netflix [133]. However, due to its collaboration, it may underperform in personalized recommendations requiring diversity and can struggle to identify relevant items among industrial settings with vast items efficiently. In contrast, user-based explanations [134] leverage social relationships to make recommendations by explaining that *users with similar interests also favor the recommended item*. The user's social property makes these explanations more persuasive, encouraging users to try the recommendations. However, the variance in user preferences may render this approach less impactful in gauging actual preference. Lastly, attribute-based explanations focus on highlighting the attributes of recommended items that users might find appealing, essentially conveying "*these features might interest you*". This method demands customization according to each user's interests, yielding higher accuracy and satisfaction. Thus, they are at the forefront of research [124, 135–138].

Obviously, such explanations typically employ pre-defined and formulaic formats, such as explanations based on similar items or friends. Although capable of conveying essential information, such inflexible formats may diminish the user experience and satisfaction by lacking adaptability and personalization [124]. For this reason, natural language generation approaches have received increasing attention. Early work [139–141] mainly relied on recurrent neural networks (e.g., LSTM [142], GRU [143]). Limited by the model's expressiveness, they often suffer from the issue of insufficient diversity. With the excellent performance of Transformer-based models in various natural language tasks, some work attempts to integrate Transformer-based models into explainable recommendations. Li et al. [144] use the position vectors corresponding to the user (item) IDs to predict interpreted tokens. Subsequent work [145] has shown that the generated explanation cannot justify the user's preference by synthesizing irrelevant descriptions. Therefore, Ni et al. [146] used such information as guided input to BERT to obtain a controllable justification. Considering that such auxiliary information is not always available in real-world scenarios, ExBERT [145] only requires historical explanations written by users, and utilizes a multi-head self-attention based encoder to capture the relevance between these explanations and user-item pairs. Recently, MMCT [147], EG4Rec [148], and KAER [149] have further carried out finer-grained modeling of information such as visual images, time series, and emotional tendencies to obtain high-quality interpretations. An early attempt, LLM4Vis [150], has been paid to the explainable visualization recommendation through ChatGPT. RecExplainer [151] proposes three alignment strategies for interpretability.

Due to the limited expressive power of traditional language models, natural language generation methods are prone to long-range dependence problems [145], that is, the input of long texts will appear to generate explanations that lack diversity and coherence in content. In addition, these explanation methods are tightly coupled with specific recommendation models (e.g., NETE [141]), or directly design a new recommendation model (e.g., NRT [139],

PETER [144]), and they are often powerless when faced with existing advanced recommendation models, which limits their generalizability. This is also a flaw in template-based methods. Notably, in industrial settings, recommendation algorithms frequently involve not just a single model but a cascade or integration of multiple models, and these elaborate combinations further exacerbate the difficulty of deciphering recommendations.

Thanks to LLMs' remarkable generative ability in language tasks, they are ideal for tackling the aforementioned challenges [152]. Firstly, with the leverage of extensive training data, LLMs adeptly harness human language, encompassing context, metaphors, and complex syntax. This equips them to craft customized explanations that are precise, natural, and adaptable to various user preferences [141, 144, 153], mitigating the limitations of conventional, formulaic explanations. Secondly, the unique in-context learning capabilities of LLMs, such as zero-shot prompting, few-shot prompting, and chain-of-thought prompting, enable them to garner real-time user feedback during interactions, furnish recommendation outcomes, and their corresponding interpretations, fostering bidirectional human-machine alignment. Recent study [154] has demonstrated the potential of LLMs in elucidating the intricacies of complex models, as evidenced by GPT-4 autonomously interpreting the function of GPT-2's each neuron by inputting appropriate prompts and the corresponding neuron activation. This showcases an innovative approach to interpreting deep learning-based recommendation models. It's critical to highlight that this interpretation technique is agnostic to the model's architecture, distinguishing it from traditional interpretations that are bound to specific algorithms. Thus, recommendation interpretations founded on LLMs pave the way for a versatile and scalable interpretational framework with broader applicability.

Although LLMs have inherently significant advantages in recommendation explanations, it is imperative to recognize potential issues. Firstly, akin to recommendation models, LLMs are essentially black boxes that are difficult for humans to understand. We cannot identify what concepts they give explanations based on [155]. Also, the explanation given may be insincere; that is, the explanations are inconsistent with their recommended behaviors. Some recent developments [38, 156] involve utilizing chains of thought to prompt reasoning for improved interpretability; however, the opacity of the reasoning process of each step remains a concern, and [157] has questioned the possible unfaithful explanations of chain-of-thought prompting. Secondly, the extensive data utilized by LLMs may encompass human biases and erroneous content [158]. Consequently, even if the explanation aligns with the model's recommendation behavior, both the explanation and recommendation could be flawed. Monitoring and calibrating these models to ensure fairness and accuracy in explainable recommendations is essential. Lastly, generative models exhibit varying levels of proficiency across different tasks, leading to inconsistencies in performance. Identical semantic cues could yield disparate recommendation explanations. This inconsistency has been substantiated by recent studies [159, 160] focusing on the LLMs' robustness. Addressing these issues calls for exploring techniques to mitigate or even circumvent low-reliability explanatory behavior, and investigating how LLMs can be trained to consistently generate reliable recommendation explanations, especially under adversarial conditions, is a worthwhile avenue for further research.

7 LLMs as common system reasoner

With the development of large language models, there is an observation that LLMs exhibit reasoning abilities [2, 161] when they are sufficiently large, which is fundamental for human

intelligence for decision-making and problem-solving. By providing the models with the 'chain of thoughts' [38], such as prompting with *'let us think about it step by step'*, the large language models exhibit emergent abilities for reasoning and can arrive at conclusions or judgments according to the evidence or logics. Accordingly, for recommender systems, large language models are capable of reasoning to help user interest mining, thus improving performance.

7.1 Making direct recommendations

In-context learning [167–173] is one of the emergent abilities of LLMs that differentiate LLMs from previous pre-trained language models, where, given a natural language instruction and task demonstrations, LLMs would generate the output by completing the word sequence without training or tuning [3]. As for in-context learning, the prompt follows by the task instruction and/or the several input-output pairs to demonstrate the task and a test input is added to require the LLM to make predictions. The input-output pair is called a *shot*. This emergent ability enables prediction on new cases without tuning unlike previous machine learning.

In the realm of recommender systems, numerous studies have explored the performance of zero-shot/few-shot learning using large language models, covering the common recommendation tasks such as rating prediction, and ranking prediction. These studies evaluate the ability of language models to provide recommendations without explicit tuning, as summarized in Table 3, where all methods adopt in-context learning for direct recommenders. The general process can be attached in Figure 4. Accordingly, we have the following findings:

- The aforementioned studies primarily focused on evaluating zero-shot/few-shot recommenders using open-domain datasets, predominantly in domains such as movies and books. Large language models are trained on extensive open-domain datasets, enabling them to possess a significant amount of common-sense knowledge, including information about well-known movies. However, when it comes to private domain data, such as e-commerce products or specific locations, the ability of zero-shot recommenders lacks of validation, which is expected to be challenging.
- Current testing methods necessitate the integration of additional modules to validate the performance of zero-shot recommenders for specific tasks. In particular, for ranking tasks that involve providing a list of items in order of preference, a candidate generation module is employed to narrow down the pool of items [164] and [165]. Generative-based models like gpt-3.5-turbo generate results in a generative manner rather than relying on recall from existing memories, thus requiring additional modules to implement ID-based item recommendations.
- From the perspective of recommendation performance, zero-shot recommenders exhibit some capabilities and few-shot learners perform better. However, there still exists a substantial gap when compared to traditional recommendation models, particularly fine-tuned large language models designed specifically for recommenders, such as P5 [174] and M6-Rec [175]. This highlights that large language models do not possess a significant advantage in personalized modeling.

Another important emergent ability is the *'step by step'* reasoning, where LLMs can solve complex tasks by utilizing prompts including previous intermediate reasoning steps, called the 'chain of thoughts' strategy [38]. Wang and Lim [164] design a three-step prompt, namely NIR, to capture user preferences, extract the most representative movies and rerank

Table 3 Zero/Few-shot learners of LLMs for RS

Approach	LLM backbone	Task	Metric	Datasets	ICL	COT
[162]	gpt-3.5-turbo	Rating prediction sequential recommendation direct recommendation explanation generation review summarization	RMSE,MAE HR,NDCG HR,NDCG BLUE4,ROUGE,Human Eval BLUE4,ROUGE,Human Eval	Amazon beauty	✓	
[163]	Text-davinci-002 text-davinci-003 gpt-3.5-turbo	Point-wise pairwise list-wise	NDCG,MRR	MovieLens-1M Amazon-Book Amazon-Music MIND-small	✓	
[120]	Flan-U-PALM gpt-3.5-turbo text-davinci-003	Rating prediction ranking prediction	RMSE,MAE ROC-AUC	MovieLens-1M Amazon-Books	✓	
[164]	Text-davinci-003	Reranking	NDCG,HR	MovieLens 100K	✓	✓
[165]	gpt-3.5-turbo	Reranking	NDCG	MovieLens-1M Amazon-Games	✓	
[166]	gpt-3.5-turbo	Reranking	Precision	MIND	✓	

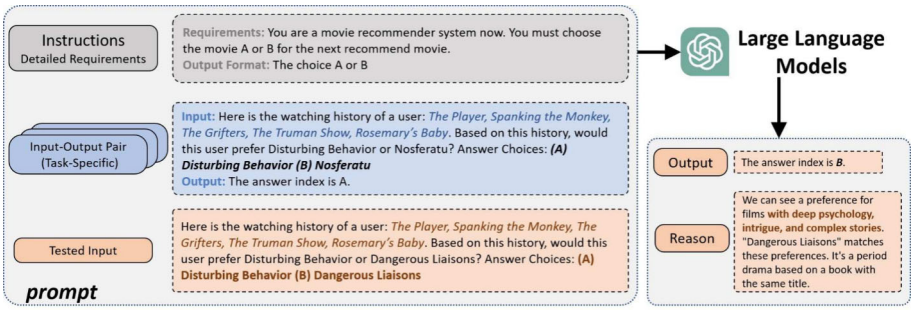

Figure 4 An example of zero/few-shot learning for direct recommenders

the items after item filtering. Such a multi-step reasoning strategy significantly improves recommendation performance.

7.2 Reasoning for automated selection

Automated Machine Learning (AutoML) is widely applied in recommender systems to eliminate the costly manual setup with trials and errors. The search space in recommender systems can be categorized in (1) Embedding size (2) Feature (3) Feature interaction (4) Model architecture. Embedding size search, such as [176–179] seeks for appropriate embedding size for each feature to avoid resources overconsumption. Searching for features consisting of raw feature search [180, 181] and synthetic feature search [182, 183], which selects a subset from the set of original or cross features to maintain informative features to reduce both computation and space cost. Feature interaction search, such as [184–188], automatically filters out feature interactions that are not helpful. Model architecture search, like [189–192], expands the search space to the integral architectures. The search strategy shifts from the discrete reinforcement learning process, which iteratively samples architectures for training and is time-consuming, into the differentiable searching, which adaptively selects architectures within one-shot learning to circumvent the computational burden, for more efficient convergence. The evaluation for each sampled architecture then acts as the signal to adjust the selections. That is, there is a decision maker who memorizes the prior results of previous architecture choices and analyzes the prior results to give the next recommended choice.

The emergent LLMs have excellent memorization and reasoning capabilities that would work for automated learning. Several works have attempted to validate the potential of automated machine learning with LLMs. Preliminarily, GPT-NAS [193] takes advantage of the generative capability of LLMs. The architecture of networks is formulated into sequential characters, and thus the generation of network architectures can be easily achieved through the generative pre-training models. NAS-Bench-101 [194] is utilized for pre-training and the state-of-the-art results are used for fine-tuning. The generative pre-training models produce reasonable architectures, which would reduce the search space for later genetic algorithms for searching optimal architectures. The relatively advanced reasoning ability is further evaluated in GENIUS [195], where GPT-4 is employed as a black-box agent to generate potential better-performing architectures according to previous trials including tried architectures with their evaluation performance. According to the results, GPT-4 can generate good architecture networks, showing the potential for more complicated tasks. Yet it is too difficult for LLMs to directly make decisions on challenging technical problems only by prompting. To balance

efficiency and interpretability, one approach is to integrate the LLMs into certain search strategies, where the genetic algorithm guides the search process and LLMs generate the candidate crossovers. LLMatic [196] and EvoPrompting [197] use code-LLMs as mutation and crossover operators for a genetic NAS algorithm. During the evolution process, each generation has a certain probability of deciding whether to perform crossover or mutation to produce new offspring. Crossover and mutation are generated by prompting LLMs. Such a solution integrates LLM into the genetic search algorithm, which would achieve better performances than direct reasoning over the whole space.

The research mentioned above brings valuable insights into the field of automated learning in recommender systems. However, several challenges need to be addressed. Firstly, the search space in recommender systems is considerably more complex, encompassing diverse types of search space and facing significant volume issues. This complexity poses a challenge in effectively exploring and optimizing the search space. Secondly, compared to the common architecture search in other domains, recommender systems lack a strong foundation of knowledge regarding the informative components within the search space, especially the effective high-order feature interactions. Unlike well-established network structures in other areas, recommender systems operate in various domains and scenarios, resulting in diverse and domain-specific components. Addressing these challenges and advancing the understanding of the search space and informative components in recommender systems will pave the way for significant improvements in automated learning approaches.

8 LLMs as conversational agent

Conversational recommender system (CRS) is a specialized type of recommendation tool that aims to uncover users' interests and preferences through dialogue, enabling personalized recommendations and real-time adjustment of recommendation strategies based on user feedback. Compared to traditional recommender systems, conversational recommender systems have the advantage of real-time understanding of user intents and the ability to adapt recommendations based on user feedback. Typically, a conversational recommender system consists of two main components: a dialogue module and a recommendation module.

In this section, we will primarily focus on discussing the dialogue module, which plays a crucial role in facilitating effective user-system interactions and understanding user preferences (Figure 5).

In a conversational recommender system, the dialogue module typically takes the form of a dialogue system. Dialogue systems can generally be classified into two main categories: chit-chat and task-oriented. The former focuses on open-domain question answering, and two major methods are commonly employed: generative and retrieval-based methods. Generative methods [198–200] utilize a sequence-to-sequence model structure to generate responses, while retrieval-based methods [201–203] transform the task of generating responses into a retrieval problem by searching for the most relevant response in a response database based on the dialogue context. In conversational recommender systems, task-oriented dialogue systems are more often required, as they are specifically designed to assist users in accomplishing specific tasks. For task-oriented dialogue systems, a common approach [24, 204] is to treat the response generation as a pipeline and handle it separately using four components: dialogue understanding [205, 206], dialogue state tracking [207–209], dialogue policy learning [24, 210], and natural language generation [211, 212]. Another approach is to employ an end-to-end method [213–215], training an encoder-decoder model to handle all the processing steps

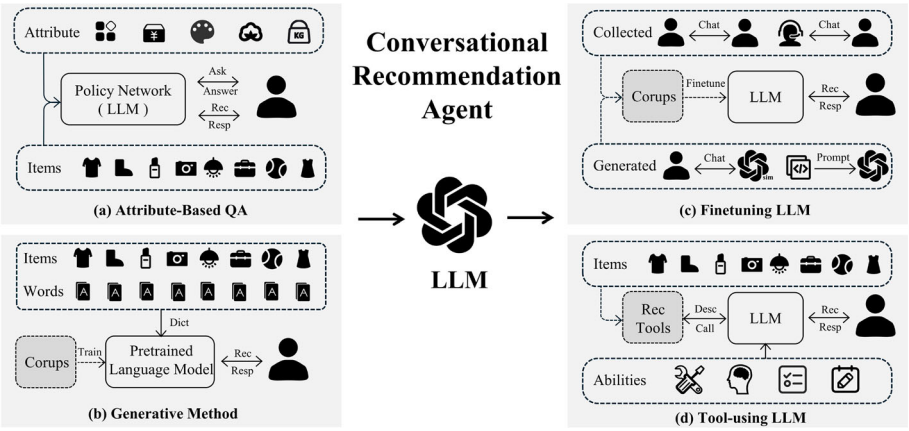

Figure 5 Important approaches in conversational agent

collectively. The first approach suffers from scalability issues and lacks synergy between the components, while the second approach requires a substantial amount of supervised data for training.

Based on the classification of dialogue systems, common approaches in conversational recommender systems can also be divided into two categories: attribute-based QA (question-answering) and generative methods. The attribute-based QA approach [24, 216–218] utilizes a pipeline method within the dialogue system. In each dialogue turn, the system needs to decide whether to ask the user a question or provide a recommendation. The decision-making process, particularly regarding which attribute to ask about, is typically handled by a policy network. On the other hand, generative methods do not explicitly model the decision-making process. Instead, they often employ an end-to-end training approach, where a sequence-to-sequence model generates output directly from a shared vocabulary of words and items. Whether the generated output is chit-chat, a question, or a recommendation is implicitly determined during the generation process. Compared to attribute-based QA methods, generative methods [219–222] appear to be simpler and more scalable. However, they require a large amount of supervised training data. With the advancement of pre-trained language models (PLMs) in the field of natural language processing, particularly models like BERT [223] and GPT [224], the capabilities of pre-trained models in language understanding and generation have become increasingly powerful. Researchers have found that fine-tuning pre-trained models with a small amount of supervised data can yield impressive results on specific tasks. This discovery has led to the application of PLMs in generative conversational recommender systems. For example, DialoGPT [215] achieved promising dialogue intelligence by fine-tuning GPT-2 on dialogue data collected from platforms like Reddit. Subsequently, BARCOR [220], RecInDial [222], and UniCRS [221] utilized DialoGPT for constructing conversational recommender systems, with variations in their action decision strategies. While PLMs reduce the dependency of generative dialogue models on extensive data, the fine-tuning process still incurs significant computational time and requires the collection of high-quality domain-specific training data due to the large parameter space.

With the increase in model parameters and training data, the intelligence and knowledge capacity of models continues to improve. OpenAI has been expanding the model parameters and training data while employing techniques such as RLHF (Reinforcement Learning from

Human Feedback) and Instruction Tuning to further fine-tune GPT-3 [3]. This has led to the emergent abilities of models like InstructGPT [37] and subsequent models like ChatGPT, which exhibit incredible intelligence and have opened the doors to new intelligent dialogue systems based on large language models (LLMs). Furthermore, Google's BARD and META's LLaMA [32] are also large language dialogue models that have been proposed and demonstrated remarkable performance in conversational abilities. The Vicuna model, for instance, utilizes dialogue corpora shared by users in using ChatGPT to fine-tune the open-source LLaMA model, with the team claiming it can achieve over 90% of ChatGPT's capability. This series of successive LLM introductions has brought new insights to conversational recommender systems. Due to the utilization of extensive open-domain corpora during the training of LLM, it possesses inherent conversational recommendation capabilities and can provide reasonable recommendations in open domains such as movies, music, news, and games.

However, there are still significant **challenges** in building an enterprise-level CRS. The first challenge is the lack of awareness of large models about private domain data. It is well known that most of the training data for LLMs, such as GPT-3, comes from publicly available sources on the internet. As a result, these models may lack visibility into the data that resides within information platforms, making their modeling and understanding capabilities of such data relatively poor. To address this challenge, there are currently two approaches being explored: fine-tuning [215] and tool learning [225, 226]. Fine-tuning involves tuning LLM using private domain-specific dialogue data. There are two major concerns in the approach. First, massive high-quality domain-specific dialogue data is required to tune the extremely large model. However, in most recommendation scenarios, data primarily consists of explicit or implicit user-item interactions, which may lack conversational context. Therefore, generating high-quality dialogue data from interaction data is a key concern in the approach. In RecLLM [226] and iEvaLM [227], researchers have proposed using LLMs to construct a user simulator for generating conversational data. Besides, the fine-tuning technique plays a crucial role in determining the ultimate quality of LLMs. A well-designed and effective fine-tuning strategy can lead to significant improvements in the model's performance and capabilities, such as instruction tuning and RLHF proposed in InstructGPT [3]. Tool learning is another approach to address this challenge, and its main idea is to treat traditional recommendation models as tools to be utilized, such as Matrix Factorization (MF) and DeepFM. For a more detailed explanation of tool learning, please refer to Section 9. Since recommendation models are domain-specific, LLM can leverage these models to obtain recommendation results and recommend them to the users in the response. In this approach, there are two main technical points: the construction of the tool model and the engineering of prompts to guide the LLM in the proper utilization of the tool. First of all, conventional recommendation models generally use id or categorical features as input, while users always give their requirements or preferences in natural language in conversations. Therefore, unstructured text features should be taken into consideration in tool construction. In Chat-Rec [225], a conventional recommendation model and a text embedding-based model(text-embedding-ada-002) are used as tools. RecLLM [226] adapted a language model enhanced dual-encoder model and several text retrieval methods as the recommendation engine. On the other hand, despite the strong intelligence and reasoning capabilities of LLMs, effectively harnessing these abilities requires well-crafted prompts for guidance. For instance, the Chain of Thought proposed by Jason [38] could trigger LLM to reason and engage in step-by-step thinking, which

benefits the tool-using capability. Subsequent studies like ToT [228], Plan-and-Solve [229] and ReAct [230] have proposed more advanced techniques for prompt design to assist in guiding LLM to engage in deeper thinking and tool planning.

The second challenge lies in the issue of memory and comprehension in long conversations. Due to the input constraints of LLMs, models like ChatGPT can support a maximum of 4096 tokens in a single call, including both input and output. In multi-turn dialogue scenarios, longer dialogue contexts often meet the risk of exceeding this token limit. The simplest approach to tackle this challenge is to trim the dialogue by discarding earlier turns. However, in conversational recommender systems, users may express a significant amount of personal information and interests in the early stages of the conversation. The omission of such information directly impacts the accuracy of recommendations. To address this issue, several relevant works have proposed solutions. MemPrompt [231] enhances the prompt by incorporating a memory module, enabling GPT-3 to possess stronger long-dialogue memory capability. Similarly, RecLLM [226] leverages LLM to extract user profiles and store them as factual statements in user memory. When processing user queries, relevant facts are retrieved based on text similarity.

9 Tool-learning and its applications in recommendation

9.1 LLM-based tool learning

Tool learning is an emerging research field that aims to enhance task-solving capabilities by combining specialized tools with foundational models, which has been understood by [232] as two perspectives:

1. **Tool-augmented learning** treats specialized tools as assistants in order for improving the quality and accuracy of tasks, or **Tool for AI**;
2. **Tool-oriented learning** focuses more on training models to effectively use tools, controlling and optimizing tool-applying processes, or **AI for Tool**.

Tool learning has found applications in various fields, and this section primarily focuses on tool learning paradigms based on large language models (LLMs). While recent works often involve a combination of these two perspectives, we do not specifically categorize each work into one type. LLMs, such as GPT, are well-suited for tool learning applications [233]. With their powerful natural language processing capabilities, LLMs can break down complex tasks into smaller sub-tasks and convert them into executable instructions. Specialized tools allow LLMs to access knowledge that is beyond their own understanding. By integrating specialized tools, LLMs can better understand and then address complex problems, offering more accurate and efficient solutions for personalized systems.

LLMs are commonly applied as controllers to select and manage various existing AI models to solve complex tasks, which rely on user input and language interfaces on making summarizations. They act as the central component, responsible for comprehending problem statements and making decisions regarding which actions to execute. Additionally, they aggregate the outcomes based on the results of the executed actions. In that case, HuggingGPT [243] leverages existing models from the Hugging Face community[3] to assist in

[3] https://huggingface.co

task-solving. Visual ChatGPT [244] combines visual foundation models like BLIP [247], Stable Diffusion [248], etc. with LangChain[4] to handle complex visual tasks, while the following TaskMatrix.AI [245] maintains a unified API Platform extending the capabilities of Visual ChatGPT, extends the capabilities of Visual ChatGPT by maintaining a unified API Platform, enabling input from multiple modalities and generating more complex task solutions. On the contrary, Auto-GPT[5] operates as an agent that autonomously understands specific targets through natural language and performs all processes in an automated loop, without requiring mandatory human input.

WebGPT [239] introduces a text-based Web browsing interactive environment, where LLMs learn to emulate the complete process of human interaction with a Web browser using behavior cloning and rejection sampling techniques. In ReAct [230], by leveraging an intuitive prompt, LLMs learn to generate both reasoning paths and task-specific actions alternately when solving a specific task. The execution of specific actions is delegated to corresponding tools, and external feedback obtained from these tools is utilized to validate and guide the reasoning process further. The motivation behind Toolformer [246] aligns closely with ReAct; however, it goes a step further by combining diverse tools within a single model. This integration provides the model with flexible decision-making abilities and improved generalization capabilities, achieved through a simple yet effective self-supervised method. In contrast to prior works, LATM [249] takes a novel approach by empowering LLMs to directly generate tools. It achieves a division of labor within the task-solving process by employing LLMs at different scales: the tool maker, tool user, and dispatcher. LATM is entirely composed of LLMs, enabling the self-generation and self-utilization of tools.

9.2 Applications in personalization scenarios

Recently, LLMs have demonstrated impressive abilities in leveraging internal world knowledge and common sense reasoning to accurately understand user intent from dialogues. Moreover, LLMs can communicate with users fluently in natural language, offering a seamless and delightful user experience. These advantages make LLMs an appealing choice as recommendation agents to enhance the personalized experience.

However, despite the impressive memory capacity of LLMs, they face challenges in memorizing specific knowledge in private and specialized domains without sufficient training. For instance, storing the item corpus and all user profiles in a recommender system can be challenging for LLMs. This limitation can result in LLMs generating inaccurate or incorrect responses and makes it difficult to control their behavior within a specific domain. Furthermore, LLMs face the challenge of the *temporal generalization problem* as external knowledge continues to evolve and change over time. To address these issues, various tools can be utilized to augment LLMs and enhance their effectiveness as recommendation agents. Table 4 shows examples of related tools.

Search engine Search engines are widely employed to provide external knowledge to LLMs, reducing LLMs' memory burden and alleviating the occurrence of hallucinations in LLMs' responses. BlenderBot 3 [250] uses specific datasets to fine-tune a series of modules, enabling LLMs to learn to invoke the search engine at the appropriate time and extract useful knowledge from the retrieval results. LaMDA [238] learns to use a toolset that includes an IR

[4] https://docs.langchain.com

[5] https://github.com/Significant-Gravitas/Auto-GPT

Table 4 LLM-based tool learning approaches

Approach	Tool usage	LLM backbone	Task
Re3 [234]	LLM	gpt3-instruct-175B gpt3-instruct-13B	Long stories generation
PEER [235]	LLM	LM-Adapted T5	Editions, citations, quotes
METALM [236]	Pretrained encoders with diverse modalities	Transformer (pretrained from scratch)	language-only tasks vision-language tasks
Atlas [237]	Dense retriever	T5	Knowledge-intensive language tasks massively-multitask language understanding question answering fact checking
LaMDA [238]	Retriever translator calculator	Decoder-only transformer	Dialog
WebGPT [239]	Web browser	gpt-3	Question answering
Mind's Eye [240]	Physics engine text-to-code LM	gpt-3 PaLM	Reasoning
PAL [241]	Python interpreter	CODEX(code-davinci-002)	Mathematical symbolic algorithmic reasoning
SayCan [242]	Robots	PaLM	Real-world robotic tasks
HuggingGPT [243]	AI models in hugging face community	gpt-3.5-turbo text-davinci-003 gpt-4	Image classification image captioning object detection etc.
Auto-GPT	Web browser	gpt-3.5-turbo text-davinci-003 gpt-4	User-specified tasks
Visual ChatGPT [244] Taskmatrix.AI [245]	Visual foundation models customized models with unified API form	text-davinci-003	Visual customized tasks
ReAct [230]	Wikipedia API	PaLM-540B	Question answering face verificaiton
Toolformer [246]	Calculator Q&A system search engine translation system calendar	GPT-J	Downstream tasks

system, a translator, and a calculator through fine-tuning to generate more factual responses. RETA-LLM [251] is a toolkit for retrieval-augmented LLMs. It disentangles IR systems and LLMs entirely, facilitating the development of in-domain LLM-based systems.Thoppilan et al. [238] shows a case of applying LaMDA to content recommendation. Preconditioned on a few role-specific dialogues, LaMDA can play the role of a music recommendation agent.

Recommendation engine Some works have attempted to alleviate the memory burden of LLMs by equipping them with a recommendation engine as a tool, enabling LLMs to offer recommendations grounded on the item corpus. The recommendation engine in Chat-REC [225] is further divided into two stages: retrieve and reranking, which aligns with typical recommendation system strategies. In the retrieval stage, LLMs utilize traditional recommendation systems as tools to retrieve 20 items from the item corpus as a candidate item set. Subsequently, LLMs employ themselves as tools to rerank the candidate item set. LLMs' commonsense reasoning ability, coupled with the internal world knowledge within them, allow them to provide explanations for the sorting results. The recommendation engine tool used in RecLLM [226] is highly similar to it in Chat-REC, and it is also divided into retrieval and reranking stages. RecLLM provides several practical solutions for large-scale retrievals, such as Generalized Dual Encoder Model and Concept Based Search, and so on.

Database Databases are also utilized as tools to supplement additional information for LLMs. In order to better cope with the cold-start problem for new items and alleviate the temporal generalization problem of LLMs, a vector database is utilized to provide information for new items that the LLMs are unaware of in Chat-REC [225]. When encountering new items, LLMs can utilize this database to access information about them based on the similarity between the user's request embedding and item embeddings in the database. User profiles can also help LLMs better understand the user's intent. RecLLM [226] employs a user profile module as a tool to deposit meaningful and enduring facts about users exposed during historical conversations in user memory and retrieve a single fact related to the current dialogue when necessary.

Although some works have applied the concept of tool learning to personalization systems, there are still interesting and promising research topics that deserve exploration. 1) **Fine-tuning models for better tool use.** In-context learning has shown promise in teaching LLMs how to effectively use tools with a small number of demonstrations, as shown in Chat-REC and RecLLM. However, LLMs often struggled to learn strategies for handling complex contexts with limited demonstrations. Fine-tuning is a viable option for improving tool use, but it requires sufficient training data and effective techniques. RecLLM further fine-tunes some modules of it using synthetic data generated by a user simulator through RLHF [37] technique. Investigating methods to obtain sufficient training data and developing tailored fine-tuning techniques for recommendation systems is a worthwhile research direction. 2) **Developing a more powerful recommendation engine.** Traditional recommendation systems often rely on collaborative filtering signals and item-to-item transition relationships for recommendations. However, with the use of LLMs as the foundation models, user preferences can be reflected through natural language and even images. Therefore, developing a recommendation engine that supports multimodal data is a crucial research direction. Additionally, the recommendation engine should also be capable of adjusting the candidate set based on user preferences or feedback (such as querying movies of a specific genre or disliking an item in the recommendation set). 3) **Building more tools.** To provide LLMs with more authentic and personalized information, the development of additional tools is crucial. For example, APIs for querying knowledge graphs [252] or accessing users' social

relationships can enhance the knowledge available to LLMs, enabling more accurate and tailored recommendations.

10 LLMs as personalized content creator

Traditional recommender systems focus on suggesting existing items based on user preferences and historical data, where displayed content is already generated for retrieval. However, with the advancements in techniques and platforms for content creators, personalized content creator has attracted more and more attention, where more appealing content is customized generated to match the user's interests and preferences, especially in the realm of online advertising [253]. The common contents contain the visual and semantic contents [254–256], such as title, abstract, description, copywritings, ad banners, thumbnail, and videos. One more widely discussed topic is text ad generation, where the ad title and ad description are generated with personalized information. Earlier works adopt the pre-defined templates [254, 257, 258] to reduce the extensive human effort, which, however, often fail to fully meet the user's interests and preferences. More recent data-driven methods have emerged, which incorporate user feedback as rewards in the reinforcement learning framework to guide the generation process [259–262]. Furthermore, the incorporation of pre-trained language models has played a significant role in improving the generation process for multiple content items [263–266]. This integration helps refine the content generation models and improve their ability to meet user preferences effectively.

As recommender systems and large language models continue to evolve, a promising technique that would bring new opportunities is the integration of AI Generated Content (AIGC). AIGC [267] involves the creation of digital content, such as images, music and natural language through AI models, with the aim of making the content creation process more efficient and accessible. Earlier efforts in this field focused on deep-learning-based generative models, including Generative Adversarial Networks (GANs) [268], Variational AutoEncoders (VAEs) [269], Normalizing Flows [270], and diffusion-based models [271] for high-quality image generation. As the generative model evolves, it eventually emerges as the transformer architecture [28], acting as the foundational blocks for BERT [29] and GPT [224] in the field of NLP, and for Vision Transformer (ViT) [272] and Swin Transformer [273] in the field of CV. Moreover, the scope of generation tasks expanded from uni-modal to multi-modal tasks, including the representative model CLIP [274], which can be used as image encoders with multi-modal prompting for generation. The multi-modal generation has become an essential aspect of AIGC, which learns the multimodal connection and interaction, typically including vision language generation [274], text audio generation [275], text graph generation [276], text Code Generation [277]. With the emergence of large language models, nowadays AIGC is achieved by extracting the human intention from instructions and generating the content according to its knowledge and intention. Representative products, including ChatGPT [278], DALL-E-2 [279], Codex [280] and Midjourney [281], have attaining significant attention from society. With the growth of data and model size, the model can learn more comprehensive information and thus leading to more realistic and high-quality content creators.

Recall to the personalized content creator, the large language models would bring opportunities from the following points (Figure 6). Large language models would further extend

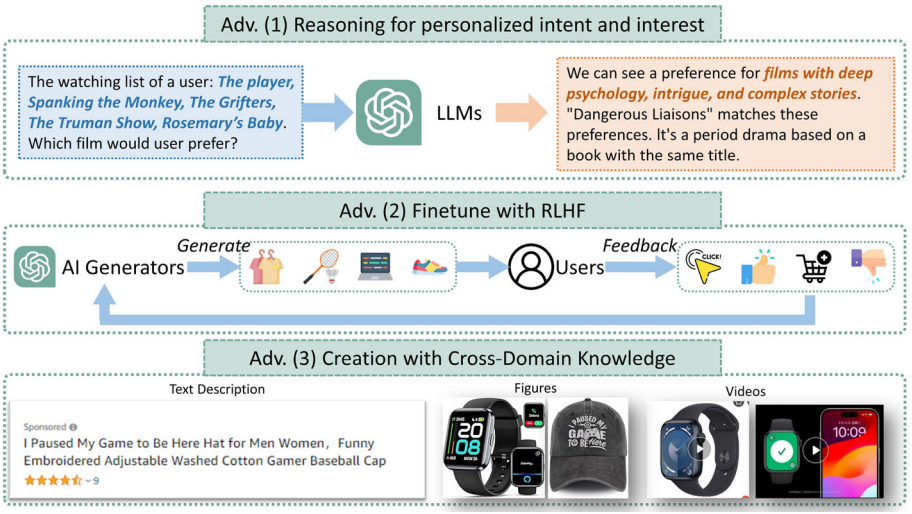

Figure 6 Advantages of content creator with LLMs

the capabilities of the pre-trained model, allowing for better reasoning of user personalized intent and interest. Previous methods [264, 265] depending on tailored pre-training models may be enhanced to better improve the reasoning abilities and few-shot prompting. Secondly, *Reinforcement Learning from Human Feedback* (RLHF) strategy can be applied to fine-tune models to better capture the user intent information, similar to existing RL-based framework [260] for text ad generation. Last but not least, the powerful generative abilities of large language models empower realistic creation thanks to the availability of sufficient cross-modal knowledge bases. The work [282] more specifically proposes a recommendation paradigm based on ChatGPT, where the generation process receives feedback and multiple rounds of conversions to better capture the user explicit preferences. Compared to previous training paradigms, more explicit expressions of user interest can be understood by the large language models and converted into corresponding instructions to guide the generation of content, significantly alleviating the problem of extremely sparse feedback. The integration of AIGC (Artificial Intelligence and Generative Content) with recommender systems offers valuable opportunities in various business scenarios. In E-commerce, large language models can power chatbots for personalized product recommendations and create captivating content to attract users. In Customer Service, AIGC enables automated responses, FAQs, and personalized assistance, improving support efficiency and customer satisfaction. Overall, integrating AIGC with recommender systems enhances user experiences and drives business growth.

However, there are two major security and privacy risks for personalized content creators. One of the concerns is the reliability of models like ChatGPT in terms of factuality, as indicated in the work [283]. While these models generate content that appears reasonable, there is a risk of distributing misleading or inaccurate information, which can weaken the truthfulness of internet content. This concern becomes particularly crucial in personalized recommendations, where the model may inadvertently promote misleading information tailored to the user's interests. The second concern revolves around data privacy, encompassing both user profiles and long-term human interaction histories. In the case of large language models, these interaction histories are collected or shared, potentially leading to the large

models memorizing sensitive user data. Previous work [284] has demonstrated that large language models, especially GPT-2 [285], memorize and leak individual training examples. This emphasizes the need for strict user approval and careful handling of annotator data to mitigate privacy risks. It is crucial to develop new techniques that prioritize privacy preservation during the training process.

11 Open challenges

11.1 Deployment difficulties

11.1.1 Industrial challenges

Personalization services, particularly with recommender systems, are complex industrial products that face numerous challenges when implemented in real-world scenarios. We will now summarize the key challenges as follows:

Scaling computational resources Existing large language models, such as BERT and GPT, demand significant computational power for training and inference. This includes high memory usage and time consumption. Fine-tuning these models to align them with personalization systems, which has shown promising results for improved personalization performance, can be computationally intensive. Several efficient finetuning strategies, e.g., option tuning in M6-Rec [175], Lora [286], QLora [287], have been developed to address this issue and pave the way for more efficient tuning.

Significant response time Achieving efficient response times is crucial for online serving and greatly impacts the personalized user experience. Response time includes both the inference phase of large language models and the concurrent user requests in large numbers. The introduction of large language models can result in considerable inference time, posing a challenge for real-world deployment. One approach is to pre-compute the embeddings of intermediate outputs from language models, storing and indexing them in a vector database, particularly for methods that utilize large language models as textual encoders. Other approaches, such as distillation and quantization, aim to strike a balance between performance and latency.

11.1.2 Laborious data collection

Large language models are widely known to leverage extensive amounts of open-domain knowledge during their training and fine-tuning processes. These knowledge sources include well-known references such as Wikipedia, books, and various websites [3]. Similarly, when applied in recommender systems, these models often rely on representative open-domain datasets such as MovieLens and Amazon Books. While this type of open-domain knowledge contains a wealth of common-sense information, personalized tasks require access to more domain-specific data that is not easily shareable. Additionally, the nature of user feedback in personalized tasks can be complex and sparse, often accompanied by noisy feedback. Collecting and filtering this data, in contrast to acquiring common-sense knowledge, presents challenges. It incurs higher labour costs and introduces additional training redundancy due to the need for extensive data processing and filtering. Furthermore, designing appropriate prompts to instruct or fine-tune large language models is crucial for aligning them with the distribution of in-domain inputs in personalization tasks. By carefully tailoring the prompts,

researchers and practitioners can guide the model to produce outputs that better cater to personalized applications, thereby maximizing performance and effectiveness.

11.2 Capacities of LLMs

11.2.1 Long text modeling

Large language models have a limitation on the maximum number of input tokens they can handle, typically constrained by the context window size, e.g., 4096 for ChatGPT. This poses challenges when dealing with long user behavior sequences, which are common in modern recommender systems. Careful design is necessary to generate effective and appropriate prompt inputs within this limited length. In the case of conversations with multiple rounds, accumulating several rounds of dialogue can easily exceed the token limit of models. The current approach in handling long conversations is to truncate the history, keeping only the most recent tokens. However, this truncation discards valuable historical information, potentially harming the performance of models. To address these challenges, several techniques can be employed. One approach is to prioritize and select the most relevant parts of the user behavior sequence or conversation history to include in the prompt. This selection can be based on various criteria such as recency, importance, or relevance to the task at hand. Another technique involves summarizing or compressing the lengthy input while preserving essential information. This can be achieved through techniques like extractive summarization or representing the long sequence in a condensed form. Moreover, architectural modifications, such as hierarchical or memory-augmented models, can be explored to better handle long sequences by incorporating mechanisms to store and retrieve relevant information efficiently.

In addition, collaborative modeling of long text data and recommendation tasks is an emerging and pressing challenge. In conventional personalization systems, item ID information along with other categorical information is commonly used for modeling feature interactions and user preferences. With the rise of large language models, there would be a growing trend toward leveraging textual information more extensively. Textual data provides unique insights about items or users, making it valuable for modeling purposes. From the perspective of modeling, dealing with long text data requires more attention and complexity compared to categorical data, not to mention the need to match the modeling of user interests. From the perspective of implementation, reforming the entire pipeline becomes necessary to accommodate the requirements of efficient latency. There lie the technical challenges of incorporating long text data into recommendation models and serving them in real time.

11.2.2 Interpretability and explainability

While large language models provide good reasoning capabilities, they are notorious for the nature of the 'black box', which is highly complex and non-linear in their enormous size and layered architecture, making it challenging to comprehend the internal workings and understand the generation process of recommendations. Without a deep understanding of how the model operates, it becomes challenging to detect and address biases or ensure fair and ethical recommendations. Once transparency about the internal mechanisms is lacking, users struggle to trust and accept the decisions made by the system. Users often desire understandable explanations for recommended choices. Addressing the challenge of model interpretability and explainability requires research involving natural language processing, explainable AI, human-computer interaction, and recommendation systems. The development of techniques

that unveil the inner workings of language models, facilitate the generation of meaningful and accurate interpretations, and enable robust evaluation methods is the main focus. By providing transparent and interpretable recommendations, users can establish trust, understand the reasoning behind the recommendations, and make informed decisions.

11.3 Evaluation

11.3.1 Tools and metrics

Conventional personalization systems typically rely on task-specific metrics such as ranking-oriented metrics, NDCG, AUC, and Recall to evaluate model performance. However, with the integration of large language models into recommender systems, the evaluation tools and metrics undergo significant changes. Traditional metrics may not sufficiently capture the performance of recommender systems powered by large language models, which introduce novel capabilities and generate recommendations in a different manner and require the development of new evaluation tools.

One crucial aspect of evaluation is considering user preferences in large language model-powered systems, which requires a user-centric approach. Metrics such as user satisfaction, engagement, and overall experience become essential considerations. For example, Liu's work [162] proposes a crowdsourcing task to assess the quality of generated explanations and review summaries, providing a way to evaluate the effectiveness of the generated content. Additionally, user satisfaction surveys and feedback questionnaires can serve as valuable options.

Another perspective to consider is the health of the system, which involves evaluating novelty and assessing factors like diversity, novelty, serendipity, and user retention rates. These metrics help evaluate the freshness of recommendations and the long-term effects of large language models. Furthermore, it is crucial to assess the interpretability and fairness of recommendations. The interpretability assessment focuses on measuring the clarity, understandability, and transparency of recommendations. Simultaneously, the fairness evaluation aims to address potential biases in personalized results. By prioritizing fairness, we strive to create personalized experiences that are equitable and inclusive for all users. Both of these evaluations are essential to enhance the overall user experience and build confidence in the personalized recommendations delivered by the system.

11.3.2 Trade-off between helpfulness, honesty, harmlessness

When large language models are employed for personalization, some of their disadvantages would be magnified. Striving for a more honest and harmless system may come at the expense of system performance.

First of all, the accuracy and factuality of the system must be ensured. Although large language models can generate seemingly reasonable content, there is a risk of disseminating misleading or inaccurate information. This becomes even more critical when incorporating user feedback, as the model may mimic user behaviors in an attempt to appear honest. However, this imitation can result in biased guidance for users, offering no real benefits.

Secondly, in terms of harmlessness, concerns regarding privacy, discrimination, and ethics arise. While large language models have the potential to provide highly personalized recommendations by leveraging user data, privacy, and data security become paramount. Unlike open-domain datasets, the privacy of individual data used for training should be rigorously

protected, with strict user permissions for sharing their personal information. For discrimination, large language models may inevitably reflect biases inherent in the training data, leading to discriminatory recommendations. Considering the biased user and item distribution, which is much more significant in recommender systems with the long-tail effect, where biased user and item distribution can lead to decisions that favor majority choices, resulting in discrimination against certain users. The final concern revolves around ethical considerations. Harmful messages, if clicked by users unconsciously, can guide large language models toward generating similar harmful content. However, when assisting in personalized decision-making, it is essential for large language models to have the capability to minimize exposure to harmful messages and guide users in a responsible manner. Approaches like constructing a Constitutional AI [288], where critiques, revisions, and supervised Learning are adopted for better training models, may offer valuable insights.

By addressing these concerns, safeguarding privacy, mitigating discrimination, and adhering to ethical guidelines, recommender systems can leverage the power of large language models while ensuring user trust, fairness, and responsible recommendations.

12 Conclusion

In conclusion, the emergence of large language models represents a significant breakthrough in the field of artificial intelligence. Their enhanced abilities in understanding, language analysis, and common-sense reasoning have opened up new possibilities for personalization. In this paper, we provide several perspectives on when large language models adapt to personalization systems. We have observed a progression from utilizing low-level capabilities of large language models to enhance performance, to leveraging their potential in complex interactions with external tools for end-to-end tasks. This evolution promises to revolutionize the way personalized services are delivered. We also acknowledge the open challenges that come with the integration of large language models into personalization systems.

Author Contributions Jin Chen organized the main writing and wrote the main manuscript text. Zheng Liu organized the entire structure of the manuscript and wrote the introduction of large language models. Xu Huang wrote the Section 8: LLMs as Conversational Agent. Chenwang Wu wrote the Section 6: LLMs as Explainer. Qi Liu and Gangwei Jiang wrote the Section 5: LLMs as Content Interpreter. Yuanhao Pu, Yuxuan Lei and Xiaolong Chen wrote Section 9: Tool-Learning and its Applications in Recommendation. Xingmei Wang wrote the Section 7.2 Reasoning for Automated Selection. Kai zheng, Defu Lian, and Enhong Chen revised the article over and over again. All authors reviewed the manuscript.

Funding Open access funding provided by Hong Kong University of Science and Technology. This work was supported by the National Natural Science Foundation of China (NSFC) No. 62022077.

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

Declarations

Competing of interest The authors declare no competing interests.

Ethical Approval This research does not involve human participants or pose potential harm to individuals. As such, formal ethical approval was not required. However, we would like to emphasize our commitment to ethical research practices and confirm that the study adheres to the ethical guidelines and principles set forth in ACM Code of Ethics.

References

1. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv:2303.18223 (2023)
2. Huang, J., Chang, K.C.-C.: Towards reasoning in large language models: a survey. arXiv:2212.10403 (2022)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
4. Salemi, A., Mysore, S., Bendersky, M., Zamani, H.: Lamp: When large language models meet personalization. arXiv:2304.11406 (2023)
5. Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al.: A survey on large language models for recommendation. arXiv:2305.19860 (2023)
6. Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Li, X., Zhu, C., Guo, H., Yu, Y., Tang, R., et al.: How can recommender systems benefit from large language models: a survey. arXiv:2306.05817 (2023)
7. Fan, W., Zhao, Z., Li, J., Liu, Y., Mei, X., Wang, Y., Tang, J., Li, Q.: Recommender systems in the era of large language models (llms). arXiv:2307.02046 (2023)
8. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186 (1994)
9. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 502–511 (2008). IEEE
10. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
11. Wang, J., De Vries, A.P., Reinders, M.J.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 501–508 (2006)
12. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The adaptive Web: Methods and Strategies of Web Personalization, pp. 325–341. Springer (2007)
13. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456 (2011)
14. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.-Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)
15. Liu, H., Wu, F., Wang, W., Wang, X., Jiao, P., Wu, C., Xie, X.: Nrpa: neural recommendation with personalized attention. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1233–1236 (2019)
16. Wang, H., Wang, N., Yeung, D.-Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244 (2015)
17. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1059–1068 (2018)
18. Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., Gai, K.: Deep interest evolution network for click-through rate prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5941–5948 (2019)

19. Wang, X., He, X., Wang, M., Feng, F., Chua, T.-S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)
20. Kang, W.-C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206 (2018). IEEE
21. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv:1511.06939 (2015)
22. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450 (2019)
23. Paschou, M., Sakkopoulos, E.: Personalized assistant apps in healthcare: a systematic review. In: 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–8 (2019). IEEE
24. Sun, Y., Zhang, Y.: Conversational recommender system. In: The 41st International Acm Sigir Conference on Research & Development in Information Retrieval, pp. 235–244 (2018)
25. Jannach, D., Manzoor, A., Cai, W., Chen, L.: A survey on conversational recommender systems. ACM Computing Surveys (CSUR) **54**(5), 1–36 (2021)
26. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Adv. Neural. Inf. Process. Syst. 13 (2000)
27. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech, vol. 2, pp. 1045–1048 (2010). Makuhari
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural. Inf. Process. Syst. 30 (2017)
29. Kenton, J.D.M.-W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
30. Shanahan, M.: Talking about large language models. Commun. ACM **67**(2), 68–79 (2024)
31. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: scaling language modeling with pathways. J. Mach. Learn. Res. **24**(240), 1–113 (2023)
32. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: open and efficient foundation language models. arXiv:2302.13971 (2023)
33. Le Scao, T., Wang, T., Hesslow, D., Bekman, S., Bari, M.S., Biderman, S., Elsahar, H., Muennighoff, N., Phang, J., Press, O., et al.: What language model to train if you have one million gpu hours?. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 765–782 (2022)
34. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv:2001.08361 (2020)
35. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D.d.L., Hendricks, L.A., Welbl, J., Clark, A., et al.: Training compute-optimal large language models. arXiv:2203.15556 (2022)
36. Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., et al.: Multitask prompted training enables zero-shot task generalization. In: International Conference on Learning Representations (2021)
37. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Adv. Neural. Inf. Process. Syst. **35**, 27730–27744 (2022)
38. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. arXiv:2201.11903 (2022)
39. Fu, Y., Peng, H., Khot, T.: How does gpt obtain its ability? tracing emergent abilities of language models to their sources. Yao Fu's Notion (2022)
40. Zhang, Y., Chen, X., Ai, Q., Yang, L., Croft, W.B.: Towards conversational search and recommendation: system ask, user respond. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 177–186 (2018)
41. Wang, H., Zhang, F., Xie, X., Guo, M.: Dkn: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1835–1844 (2018)
42. Huang, J., Zhao, W.X., Dou, H., Wen, J.-R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st international ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 505–514 (2018)

43. Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q.: Shine: signed heterogeneous information network embedding for sentiment link prediction. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 592–600 (2018)
44. Yu, X., Ren, X., Gu, Q., Sun, Y., Han, J.: Collaborative filtering with entity similarity regularization in heterogeneous information networks. IJCAI HINA 27 (2013)
45. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 453–462 (2015)
46. Ma, W., Zhang, M., Cao, Y., Jin, W., Wang, C., Liu, Y., Ma, S., Ren, X.: Jointly learning explainable rules for recommendation with knowledge graph. In: The World Wide Web Conference, pp. 1210–1221 (2019)
47. Huang, X., Fang, Q., Qian, S., Sang, J., Li, Y., Xu, C.: Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 548–556 (2019)
48. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplenet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426 (2018)
49. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, pp. 3307–3313 (2019)
50. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
51. Tang, X., Wang, T., Yang, H., Song, H.: Akupm: attention-enhanced knowledge-aware user preference model for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1891–1899 (2019)
52. Zhao, J., Zhou, Z., Guan, Z., Zhao, W., Ning, W., Qiu, G., He, X.: Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2347–2357 (2019)
53. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases?. arXiv:1909.01066 (2019)
54. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model?. arXiv:2002.08910 (2020)
55. Petroni, F., Lewis, P., Piktus, A., Rocktäschel, T., Wu, Y., Miller, A.H., Riedel, S.: How context affects language models' factual predictions. In: Automated Knowledge Base Construction
56. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Trans. Assoc. Comput. Linguist **8**, 423–438 (2020)
57. Wang, C., Liu, X., Song, D.: Language models are open knowledge graphs. arXiv:2010.11967 (2020)
58. Poerner, N., Waltinger, U., Schütze, H.: E-bert: efficient-yet-effective entity embeddings for bert. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 803–818 (2020)
59. Heinzerling, B., Inui, K.: Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1772–1791 (2021)
60. Wang, C., Liu, P., Zhang, Y.: Can generative pre-trained language models serve as knowledge bases for closed-book qa?. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 1: long papers), pp. 3241–3251 (2021)
61. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International Conference on Machine Learning, pp. 3929–3938 (2020). PMLR
62. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Adv. Neural. Inf. Process. Syst. 26 (2013)
63. Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N.: Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. arXiv:2305.13168 (2023)
64. Zhang, Z., Liu, X., Zhang, Y., Su, Q., Sun, X., He, B.: Pretrain-kge: learning knowledge representation from pretrained language models. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 259–266 (2020)
65. Kumar, A., Pandey, A., Gadia, R., Mishra, M.: Building knowledge graph using pre-trained language model for learning entity-aware relationships. In: 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), pp. 310–315 (2020). IEEE

66. Kim, B., Hong, T., Ko, Y., Seo, J.: Multi-task learning for knowledge graph completion with pre-trained language models. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1737–1743 (2020)
67. Choi, B., Jang, D., Ko, Y.: Mem-kgc: masked entity model for knowledge graph completion with pre-trained language model. IEEE Access **9**, 132025–132032 (2021)
68. Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., Chang, Y.: Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of the Web Conference 2021, pp. 1737–1748 (2021)
69. Xie, X., Zhang, N., Li, Z., Deng, S., Chen, H., Xiong, F., Chen, M., Chen, H.: From discrimination to generation: knowledge graph completion with generative transformer. In: Companion Proceedings of the Web Conference 2022, pp. 162–165 (2022)
70. Jiang, P., Agarwal, S., Jin, B., Wang, X., Sun, J., Han, J.: Text-augmented open knowledge graph completion via pre-trained language models. arXiv:2305.15597 (2023)
71. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various ner subtasks. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 1: long papers), pp. 5808–5822 (2021)
72. Li, B., Yin, W., Chen, M.: Ultra-fine entity typing with indirect supervision from natural language inference. Trans. Assoc. Comput. Linguist. **10**, 607–622 (2022)
73. Kirstain, Y., Ram, O., Levy, O.: Coreference resolution without span representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 2: short papers), pp. 14–19 (2021)
74. Cattan, A., Eirew, A., Stanovsky, G., Joshi, M., Dagan, I.: Cross-document coreference resolution over predicted mentions. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 5100–5107 (2021)
75. Lyu, S., Chen, H.: Relation classification with entity type restriction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 390–395 (2021)
76. Wang, H., Focke, C., Sylvester, R., Mishra, N., Wang, W.: Fine-tune bert for docred with two-step process. arXiv:1909.11898 (2019)
77. Han, J., Collier, N., Buntine, W., Shareghi, E.: Pive: prompting with iterative verification improving graph-based generative capability of llms. arXiv:2305.12392 (2023)
78. Trajanoska, M., Stojanov, R., Trajanov, D.: Enhancing knowledge graph construction using large language models. arXiv:2305.04676 (2023)
79. West, P., Bhagavatula, C., Hessel, J., Hwang, J., Jiang, L., Le Bras, R., Lu, X., Welleck, S., Choi, Y.: Symbolic knowledge distillation: from general language models to commonsense models. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4602–4625 (2022)
80. Xi, Y., Liu, W., Lin, J., Zhu, J., Chen, B., Tang, R., Zhang, W., Zhang, R., Yu, Y.: Towards open-world recommendation with knowledge augmentation from large language models. arXiv:2306.10933 (2023)
81. Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., Huang, C.: Llmrec: large language models with graph augmentation for recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pp. 806–815 (2024)
82. Zhao, Q., Qian, H., Liu, Z., Zhang, G.-D., Gu, L.: Breaking the barrier: utilizing large language models for industrial recommendation systems through an inferential knowledge graph. arXiv:2402.13750 (2024)
83. Razniewski, S., Yates, A., Kassner, N., Weikum, G.: Language models as or for knowledge bases. arXiv:2110.04888 (2021)
84. Yu, J., Wang, X., Tu, S., Cao, S., Zhang-Li, D., Lv, X., Peng, H., Yao, Z., Zhang, X., Li, H., et al.: Kola: Carefully benchmarking world knowledge of large language models. arXiv:2306.09296 (2023)
85. Ye, D., Lin, Y., Li, P., Sun, M.: Packed levitated marker for entity and relation extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (vol. 1: long papers), pp. 4904–4917 (2022)
86. Lang, K.: Newsweeder: Learning to filter netnews. In: Machine Learning Proceedings 1995, pp. 331–339. Elsevier (1995)
87. Wang, H., Shi, X., Yeung, D.-Y.: Collaborative recurrent autoencoder: recommend while learning to fill in the blanks. Adv. Neural. Inf. Process. Syst. 29 (2016)
88. Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., Zhang, F.: A hybrid collaborative filtering model with deep structure for recommender systems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)

89. Li, X., She, J.: Collaborative variational autoencoder for recommender systems. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 305–314 (2017)
90. Wu, C., Wu, F., Huang, Y., Xie, X.: Personalized news recommendation: methods and challenges. ACM Trans. Inf. Syst. **41**(1), 24–12450 (2023)
91. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1188–1196. JMLR.org (2014)
92. Song, Y., Elkahky, A.M., He, X.: Multi-rate deep learning for temporal recommendation. In: SIGIR, pp. 909–912. ACM (2016)
93. Kumar, V., Khattar, D., Gupta, S., Gupta, M., Varma, V.: Deep neural architecture for news recommendation. In: CLEF (Working Notes). CEUR Workshop Proceedings, vol. 1866. CEUR-WS.org (2017)
94. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1933–1942 (2017)
95. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR (Workshop Poster) (2013)
96. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: KDD, pp. 2576–2584. ACM (2019)
97. An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., Xie, X.: Neural news recommendation with long- and short-term user representations. In: ACL (1), pp. 336–345. Association for Computational Linguistics (2019)
98. Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: EMNLP/IJCNLP (1), pp. 6388–6393. Association for Computational Linguistics (2019)
99. Wu, C., Wu, F., Qi, T., Huang, Y.: User modeling with click preference and reading satisfaction for news recommendation. In: IJCAI, pp. 3023–3029. ijcai.org (2020)
100. Khattar, D., Kumar, V., Varma, V., Gupta, M.: Weave&rec: a word embedding based 3-d convolutional network for news recommendation. In: CIKM, pp. 1855–1858. ACM (2018)
101. Zhu, Q., Zhou, X., Song, Z., Tan, J., Guo, L.: DAN: deep attention neural network for news recommendation. In: AAAI, pp. 5973–5980. rAAAI Press (2019)
102. Qiu, Z., Wu, X., Gao, J., Fan, W.: U-bert: Pre-training user representations for improved recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4320–4327 (2021)
103. Zhang, Q., Li, J., Jia, Q., Wang, C., Zhu, J., Wang, Z., He, X.: Unbert: user-news matching bert for news recommendation. In: IJCAI, pp. 3356–3362 (2021)
104. Wu, C., Wu, F., Qi, T., Huang, Y.: Empowering news recommendation with pre-trained language models. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1652–1656 (2021)
105. Liu, Q., Zhu, J., Dai, Q., Wu, X.: Boosting deep ctr prediction with a plug-and-play pre-trainer for news recommendation. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 2823–2833 (2022)
106. Wu, C., Wu, F., Qi, T., Zhang, C., Huang, Y., Xu, T.: Mm-rec: visiolinguistic model empowered multimodal news recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2560–2564 (2022)
107. Yu, Y., Wu, F., Wu, C., Yi, J., Liu, Q.: Tiny-newsrec: effective and efficient plm-based news recommendation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 5478–5489 (2022)
108. Zou, L., Zhang, S., Cai, H., Ma, D., Cheng, S., Wang, S., Shi, D., Cheng, Z., Yin, D.: Pre-trained language model based ranking in baidu search. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 4014–4022 (2021)
109. Liu, Y., Lu, W., Cheng, S., Shi, D., Wang, S., Cheng, Z., Yin, D.: Pre-trained language model for web-scale retrieval in baidu search. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3365–3375 (2021)
110. Muhamed, A., Keivanloo, I., Perera, S., Mracek, J., Xu, Y., Cui, Q., Rajagopalan, S., Zeng, B., Chilimbi, T.: Ctr-bert: cost-effective knowledge distillation for billion-parameter teacher models. In: NeurIPS Efficient Natural Language and Speech Processing Workshop (2021)
111. He, J., Xu, B., Yang, Z., Han, D., Yang, C., Lo, D.: Ptm4tag: sharpening tag recommendation of stack overflow posts with pre-trained models. In: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, pp. 1–11 (2022)
112. Zhang, X., Malkov, Y., Florez, O., Park, S., McWilliams, B., Han, J., El-Kishky, A.: Twhin-bert: a socially-enriched pre-trained language model for multilingual tweet representations. arXiv:2209.07562 (2022)

113. Rahmani, S., Naghshzan, A., Guerrouj, L.: Improving code example recommendations on informal documentation using bert and query-aware lsh: a comparative study. arXiv:2305.03017 (2023)
114. Ding, H., Ma, Y., Deoras, A., Wang, Y., Wang, H.: Zero-shot recommender systems. arXiv:2105.08318 (2021)
115. Hou, Y., Mu, S., Zhao, W.X., Li, Y., Ding, B., Wen, J.-R.: Towards universal sequence representation learning for recommender systems. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 585–593 (2022)
116. Hou, Y., He, Z., McAuley, J., Zhao, W.X.: Learning vector-quantized item representation for transferable sequential recommenders. In: Proceedings of the ACM Web Conference 2023, pp. 1162–1171 (2023)
117. Yuan, Z., Yuan, F., Song, Y., Li, Y., Fu, J., Yang, F., Pan, Y., Ni, Y.: Where to go next for recommender systems? id-vs. modality-based recommender models revisited. arXiv:2303.13835 (2023)
118. Fu, J., Yuan, F., Song, Y., Yuan, Z., Cheng, M., Cheng, S., Zhang, J., Wang, J., Pan, Y.: Exploring adapter-based transfer learning for recommender systems: empirical studies and practical insights. arXiv:2305.15036 (2023)
119. Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X.: Tallrec: an effective and efficient tuning framework to align large language model with recommendation. arXiv:2305.00447 (2023)
120. Kang, W.-C., Ni, J., Mehta, N., Sathiamoorthy, M., Hong, L., Chi, E., Cheng, D.Z.: Do llms understand user preferences?. evaluating llms on user rating prediction. arXiv:2305.06474 (2023)
121. Chen, Z.: Palr: Personalization aware llms for recommendation. arXiv:2305.07622 (2023)
122. Zhang, J., Xie, R., Hou, Y., Zhao, W.X., Lin, L., Wen, J.-R.: Recommendation as instruction following: a large language model empowered recommendation approach. arXiv:2305.07001 (2023)
123. Li, R., Deng, W., Cheng, Y., Yuan, Z., Zhang, J., Yuan, F.: Exploring the upper limits of text-based collaborative filtering using large language models: discoveries and insights. arXiv:2305.11700 (2023)
124. Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., Xie, X.: A reinforcement learning framework for explainable recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 587–596 (2018). IEEE
125. Gao, J., Wang, X., Wang, Y., Xie, X.: Explainable recommendation through attentive multi-view learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3622–3629 (2019)
126. Lee, S., Wang, X., Han, S., Yi, X., Xie, X., Cha, M.: Self-explaining deep models with logic rule reasoning. Adv. Neural. Inf. Process. Syst. (2022)
127. Nye, M., Andreassen, A.J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., et al.: Show your work: scratchpads for intermediate computation with language models. arXiv:2112.00114 (2021)
128. Lampinen, A.K., Dasgupta, I., Chan, S.C., Matthewson, K., Tessler, M.H., Creswell, A., McClelland, J.L., Wang, J.X., Hill, F.: Can language models learn from explanations in context?. arXiv:2204.02329 (2022)
129. Zelikman, E., Wu, Y., Mu, J., Goodman, N.: Star: bootstrapping reasoning with reasoning. Adv. Neural. Inf. Process. Syst. **35**, 15476–15488 (2022)
130. Zhang, Y., Chen, X., et al.: Explainable recommendation: a survey and new perspectives. Found. Trends® in Inf. Retr. **14**(1), 1–101 (2020)
131. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166 (1999)
132. Linden, G., Smith, B., York, J.: Amazon. com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
133. Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: algorithms, business value, and innovation. ACM Trans. Manage. Inf. Syst. (TMIS) **6**(4), 1–19 (2015)
134. Sinha, R., Swearingen, K.: The role of transparency in recommender systems. In: CHI'02 Extended Abstracts on Human Factors in Computing Systems, pp. 830–831 (2002)
135. Xian, Y., Zhao, T., Li, J., Chan, J., Kan, A., Ma, J., Dong, X.L., Faloutsos, C., Karypis, G., Muthukrishnan, S., et al.: Ex3: explainable attribute-aware item-set recommendations. In: Proceedings of the 15th ACM Conference on Recommender Systems, pp. 484–494 (2021)
136. Wang, X., Li, Q., Yu, D., Xu, G.: Reinforced path reasoning for counterfactual explainable recommendation. arXiv:2207.06674 (2022)
137. Verma, S., Beniwal, A., Sadagopan, N., Seshadri, A.: Recxplainer: post-hoc attribute-based explanations for recommender systems. arXiv:2211.14935 (2022)
138. Zhang, W., Yan, J., Wang, Z., Wang, J.: Neuro-symbolic interpretable collaborative filtering for attribute-based recommendation. In: Proceedings of the ACM Web Conference 2022, pp. 3229–3238 (2022)
139. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 345–354 (2017)

140. Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M., Xu, K.: Learning to generate product reviews from attributes. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: volume 1, long papers, pp. 623–632 (2017)
141. Li, L., Zhang, Y., Chen, L.: Generate neural template explanations for recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 755–764 (2020)
142. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
143. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014)
144. Li, L., Zhang, Y., Chen, L.: Personalized transformer for explainable recommendation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 1: Long papers) (2021)
145. Zhan, H., Li, L., Li, S., Liu, W., Gupta, M., Kot, A.C.: Towards explainable recommendation via bert-guided explanation generator. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). IEEE
146. Ni, J., Li, J., McAuley, J.: Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 188–197 (2019)
147. Liu, Z., Ma, Y., Schubert, M., Ouyang, Y., Rong, W., Xiong, Z.: Multimodal contrastive transformer for explainable recommendation. IEEE Transactions on Computational Social Systems (2023)
148. Qu, Y., Nobuhara, H.: Explanation generated for sequential recommendation based on transformer model. In: 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), pp. 1–6 (2022). IEEE
149. Bai, P., Xia, Y., Xia, Y.: Fusing knowledge and aspect sentiment for explainable recommendation. IEEE Access **8**, 137150–137160 (2020)
150. Wang, L., Zhang, S., Wang, Y., Lim, E.-P., Wang, Y.: Llm4vis: explainable visualization recommendation using chatgpt. arXiv:2310.07652 (2023)
151. Lei, Y., Lian, J., Yao, J., Huang, X., Lian, D., Xie, X.: Recexplainer: aligning large language models for recommendation model interpretability. arXiv:2311.10947 (2023)
152. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv:2108.07258 (2021)
153. Li, L., Zhang, Y., Chen, L.: Personalized prompt learning for explainable recommendation. ACM Trans. Inf. Syst. **41**(4), 1–26 (2023)
154. Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., Saunders, W.: Language models can explain neurons in language models (2023)
155. Wu, Z., Geiger, A., Potts, C., Goodman, N.D.: Interpretability at scale: identifying causal mechanisms in alpaca. arXiv:2305.08809 (2023)
156. Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., Chen, W.: Making large language models better reasoners with step-aware verifier (2023)
157. Turpin, M., Michael, J., Perez, E., Bowman, S.R.: Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. arXiv:2305.04388 (2023)
158. Li, S., Liu, H., Dong, T., Zhao, B.Z.H., Xue, M., Zhu, H., Lu, J.: Hidden backdoors in human-centric language models. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 3123–3140 (2021)
159. Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Huang, H., Ye, W., Geng, X., et al.: On the robustness of chatgpt: an adversarial and out-of-distribution perspective. arXiv:2302.12095 (2023)
160. Han, R., Peng, T., Yang, C., Wang, B., Liu, L., Wan, X.: Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. arXiv:2305.14450 (2023)
161. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv:2206.07682 (2022)
162. Liu, J., Liu, C., Lv, R., Zhou, K., Zhang, Y.: Is chatgpt a good recommender? a preliminary study. arXiv:2304.10149 (2023)
163. Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., Xu, J.: Uncovering chatgpt's capabilities in recommender systems. arXiv:2305.02182 (2023)
164. Wang, L., Lim, E.-P.: Zero-shot next-item recommendation using large pretrained language models. arXiv:2304.03153 (2023)

165. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. arXiv:2305.08845 (2023)
166. Li, X., Zhang, Y., Malthouse, E.C.: A preliminary study of chatgpt on news recommendation: personalization, provider fairness, fake news. arXiv:2306.10702 (2023)
167. Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., Sui, Z.: A survey for in-context learning. arXiv:2301.00234 (2022)
168. Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., Wei, F.: Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. arXiv:2212.10559 (2022)
169. Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., Zettlemoyer, L.: Rethinking the role of demonstrations: What makes in-context learning work?. arXiv:2202.12837 (2022)
170. Levy, I., Bogin, B., Berant, J.: Diverse demonstrations improve in-context compositional generalization. arXiv:2212.06800 (2022)
171. Xie, S.M., Raghunathan, A., Liang, P., Ma, T.: An explanation of in-context learning as implicit bayesian inference. In: International Conference on Learning Representations
172. Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al.: In-context learning and induction heads. arXiv:2209.11895 (2022)
173. Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., Zhou, D.: What learning algorithm is in-context learning? investigations with linear models. arXiv:2211.15661 (2022)
174. Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (rlp): a unified pretrain, personalized prompt & predict paradigm (p5). In: Proceedings of the 16th ACM Conference on Recommender Systems, pp. 299–315 (2022)
175. Cui, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: M6-rec: generative pretrained language models are open-ended recommender systems. arXiv:2205.08084 (2022)
176. Liu, S., Gao, C., Chen, Y., Jin, D., Li, Y.: Learnable embedding sizes for recommender systems. arXiv:2101.07577 (2021)
177. Liu, H., Zhao, X., Wang, C., Liu, X., Tang, J.: Automated embedding size search in deep recommender systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2307–2316 (2020)
178. Deng, W., Pan, J., Zhou, T., Kong, D., Flores, A., Lin, G.: Deeplight: deep lightweight feature interactions for accelerating ctr predictions in ad serving. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 922–930 (2021)
179. Ginart, A.A., Naumov, M., Mudigere, D., Yang, J., Zou, J.: Mixed dimension embeddings with application to memory-efficient recommendation systems. In: 2021 IEEE International Symposium on Information Theory (ISIT), pp. 2786–2791 (2021). IEEE
180. Wang, Y., Zhao, X., Xu, T., Wu, X.: Autofield: automating feature selection in deep recommender systems. In: Proceedings of the ACM Web Conference 2022, pp. 1977–1986 (2022)
181. Lin, W., Zhao, X., Wang, Y., Xu, T., Wu, X.: Adafs: adaptive feature selection in deep recommender system. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3309–3317 (2022)
182. Tsang, M., Cheng, D., Liu, H., Feng, X., Zhou, E., Liu, Y.: Feature interaction interpretability: a case for explaining ad-recommendation systems via neural interaction detection. arXiv:2006.10966 (2020)
183. Yuanfei, L., Mengshuo, W., Hao, Z., Quanming, Y., WeiWei, T., Yuqiang, C., Qiang, Y., Wenyuan, D.: Autocross: automatic feature crossing for tabular data in real-world applications. arXiv:1904.12857 (2019)
184. Liu, B., Zhu, C., Li, G., Zhang, W., Lai, J., Tang, R., He, X., Li, Z., Yu, Y.: Autofis: automatic feature interaction selection in factorization models for click-through rate prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2636–2645 (2020)
185. Liu, B., Xue, N., Guo, H., Tang, R., Zafeiriou, S., He, X., Li, Z.: Autogroup: automatic feature grouping for modelling explicit high-order feature interactions in ctr prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 199–208 (2020)
186. Chen, Y., Ren, P., Wang, Y., Rijke, M.: Bayesian personalized feature interaction selection for factorization machines. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 665–674 (2019)
187. Xie, Y., Wang, Z., Li, Y., Ding, B., Gürel, N.M., Zhang, C., Huang, M., Lin, W., Zhou, J.: Fives: feature interaction via edge search for large-scale tabular data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3795–3805 (2021)
188. Su, Y., Zhang, R., Erfani, S., Xu, Z.: Detecting beneficial feature interactions for recommender systems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4357–4365 (2021)

189. Song, Q., Cheng, D., Zhou, H., Yang, J., Tian, Y., Hu, X.: Towards automated neural interaction discovery for click-through rate prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 945–955 (2020)
190. Zhao, P., Xiao, K., Zhang, Y., Bian, K., Yan, W.: Ameir: automatic behavior modeling, interaction exploration and mlp investigation in the recommender system. In: IJCAI, pp. 2104–2110 (2021)
191. Wei, Z., Wang, X., Zhu, W.: Autoias: automatic integrated architecture searcher for click-trough rate prediction. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2101–2110 (2021)
192. Cheng, M., Liu, Z., Liu, Q., Ge, S., Chen, E.: Towards automatic discovering of deep hybrid network architecture for sequential recommendation. In: Proceedings of the ACM Web Conference 2022, pp. 1923–1932 (2022)
193. Yu, C., Liu, X., Tang, C., Feng, W., Lv, J.: Gpt-nas: Neural architecture search with the generative pre-trained model. arXiv:2305.05351 (2023)
194. Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., Hutter, F.: Nas-bench-101: Towards reproducible neural architecture search. In: International Conference on Machine Learning, pp. 7105–7114 (2019). PMLR
195. Zheng, M., Su, X., You, S., Wang, F., Qian, C., Xu, C., Albanie, S.: Can gpt-4 perform neural architecture search? arXiv:2304.10970 (2023)
196. Nasir, M.U., Earle, S., Togelius, J., James, S., Cleghorn, C.: Llmatic: Neural architecture search via large language models and quality-diversity optimization. arXiv:2306.01102 (2023)
197. Chen, A., Dohan, D.M., So, D.R.: Evoprompting: language models for code-level neural architecture search. arXiv:2302.14838 (2023)
198. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. arXiv:1503.02364 (2015)
199. Vinyals, O., Le, Q.: A neural conversational model. arXiv:1506.05869 (2015)
200. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. arXiv:1506.06714 (2015)
201. Wu, W., Yan, R.: Deep chit-chat: deep learning for chatbots. In: Proceedings of the 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1413–1414 (2019)
202. Qiu, X., Huang, X.: Convolutional neural tensor network architecture for community-based question answering. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
203. Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X.: A deep architecture for semantic matching with multiple positional sentence representations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
204. Greco, C., Suglia, A., Basile, P., Semeraro, G.: Converse-et-impera: exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems. In: AI* IA 2017 Advances in Artificial Intelligence: XVIth International Conference of the Italian Association for Artificial Intelligence, Bari, Italy, November 14-17, 2017, Proceedings 16, pp. 372–386 (2017). Springer
205. Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., Yu, D.: Recurrent neural networks for language understanding. In: Interspeech, pp. 2524–2528 (2013)
206. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Interspeech, pp. 3771–3775 (2013)
207. Goddeau, D., Meng, H., Polifroni, J., Seneff, S., Busayapongchai, S.: A form-based dialogue manager for spoken language applications. In: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96, vol. 2, pp. 701–704 (1996). IEEE
208. Henderson, M., Thomson, B., Young, S.: Deep neural network approach for the dialog state tracking challenge. In: Proceedings of the SIGDIAL 2013 Conference, pp. 467–471 (2013)
209. Mrkšić, N., Séaghdha, D.O., Wen, T.-H., Thomson, B., Young, S.: Neural belief tracker: data-driven dialogue state tracking. arXiv:1606.03777 (2016)
210. Cuayáhuitl, H., Keizer, S., Lemon, O.: Strategic dialogue management via deep reinforcement learning. arXiv:1511.08099 (2015)
211. Zhou, H., Huang, M., Zhu, X.: Context-aware natural language generation for spoken dialogue systems. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2032–2041 (2016)
212. Dušek, O., Jurčíček, F.: Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. arXiv:1606.05491 (2016)
213. Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L.M., Su, P.-H., Ultes, S., Young, S.: A network-based end-to-end trainable task-oriented dialogue system. arXiv:1604.04562 (2016)

214. Bordes, A., Boureau, Y.-L., Weston, J.: Learning end-to-end goal-oriented dialog. arXiv:1605.07683 (2016)
215. Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B.: Dialogpt: large-scale generative pre-training for conversational response generation. arXiv:1911.00536 (2019)
216. Lei, W., He, X., Miao, Y., Wu, Q., Hong, R., Kan, M.-Y., Chua, T.-S.: Estimation-action-reflection: towards deep interaction between conversational and recommender systems. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 304–312 (2020)
217. Lei, W., Zhang, G., He, X., Miao, Y., Wang, X., Chen, L., Chua, T.-S.: Interactive path reasoning on graph for conversational recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2073–2083 (2020)
218. Deng, Y., Li, Y., Sun, F., Ding, B., Lam, W.: Unified conversational recommendation policy learning via graph-based reinforcement learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1431–1441 (2021)
219. Li, R., Ebrahimi Kahou, S., Schulz, H., Michalski, V., Charlin, L., Pal, C.: Towards deep conversational recommendations. Adv. Neural. Inf. Process. Syst. 31 (2018)
220. Wang, T.-C., Su, S.-Y., Chen, Y.-N.: Barcor: Towards a unified framework for conversational recommendation systems. arXiv:2203.14257 (2022)
221. Wang, X., Zhou, K., Wen, J.-R., Zhao, W.X.: Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1929–1937 (2022)
222. Wang, L., Hu, H., Sha, L., Xu, C., Jiang, D., Wong, K.-F.: 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, pp. 489–500 (2022)
223. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
224. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
225. Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., Zhang, J.: Chat-rec: towards interactive and explainable llms-augmented recommender system. arXiv:2303.14524 (2023)
226. Friedman, L., Ahuja, S., Allen, D., Tan, T., Sidahmed, H., Long, C., Xie, J., Schubiner, G., Patel, A., Lara, H., et al.: Leveraging large language models in conversational recommender systems. arXiv:2305.07961 (2023)
227. Wang, X., Tang, X., Zhao, W.X., Wang, J., Wen, J.-R.: Rethinking the evaluation for conversational recommendation in the era of large language models. arXiv:2305.13112 (2023)
228. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: deliberate problem solving with large language models. arXiv:2305.10601 (2023)
229. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.-W., Lim, E.-P.: Plan-and-solve prompting: improving zero-shot chain-of-thought reasoning by large language models. arXiv:2305.04091 (2023)
230. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: synergizing reasoning and acting in language models. arXiv:2210.03629 (2022)
231. Madaan, A., Tandon, N., Clark, P., Yang, Y.: Memory-assisted prompt editing to improve gpt-3 after deployment. arXiv:2201.06009 (2022)
232. Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Huang, Y., Xiao, C., Han, C., et al.: Tool learning with foundation models. arXiv:2304.08354 (2023)
233. Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., et al.: Augmented language models: a survey. arXiv:2302.07842 (2023)
234. Yang, K., Peng, N., Tian, Y., Klein, D.: Re3: Generating longer stories with recursive reprompting and revision. arXiv:2210.06774 (2022)
235. Schick, T., Dwivedi-Yu, J., Jiang, Z., Petroni, F., Lewis, P., Izacard, G., You, Q., Nalmpantis, C., Grave, E., Riedel, S.: Peer: a collaborative language model. arXiv:2208.11663 (2022)
236. Hao, Y., Song, H., Dong, L., Huang, S., Chi, Z., Wang, W., Ma, S., Wei, F.: Language models are general-purpose interfaces. arXiv:2206.06336 (2022)
237. Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Few-shot learning with retrieval augmented language models. arXiv:2208.03299 (2022)
238. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al.: Lamda: language models for dialog applications. arXiv:2201.08239 (2022)
239. Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al.: Webgpt: browser-assisted question-answering with human feedback. arXiv:2112.09332 (2021)

240. Liu, R., Wei, J., Gu, S.S., Wu, T.-Y., Vosoughi, S., Cui, C., Zhou, D., Dai, A.M.: Mind's eye: grounded language model reasoning through simulation. arXiv:2210.05359 (2022)
241. Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., Neubig, G.: Pal: Program-aided language models. arXiv:2211.10435 (2022)
242. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al.: Do as i can, not as i say: grounding language in robotic affordances. arXiv:2204.01691 (2022)
243. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: solving ai tasks with chatgpt and its friends in huggingface. arXiv:2303.17580 (2023)
244. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: talking, drawing and editing with visual foundation models. arXiv:2303.04671 (2023)
245. Liang, Y., Wu, C., Song, T., Wu, W., Xia, Y., Liu, Y., Ou, Y., Lu, S., Ji, L., Mao, S., et al.: Taskmatrix. ai: completing tasks by connecting foundation models with millions of apis. arXiv:2303.16434 (2023)
246. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: language models can teach themselves to use tools. arXiv:2302.04761 (2023)
247. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning, pp. 12888–12900 (2022). PMLR
248. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
249. Cai, T., Wang, X., Ma, T., Chen, X., Zhou, D.: Large language models as tool makers. arXiv:2305.17126 (2023)
250. Shuster, K., Xu, J., Komeili, M., Ju, D., Smith, E.M., Roller, S., Ung, M., Chen, M., Arora, K., Lane, J., et al.: Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. arXiv:2208.03188 (2022)
251. Liu, J., Jin, J., Wang, Z., Cheng, J., Dou, Z., Wen, J.-R.: Reta-llm: a retrieval-augmented large language model toolkit. arXiv:2306.05212 (2023)
252. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: a roadmap. arXiv:2306.08302 (2023)
253. Vempati, S., Malayil, K.T., Sruthi, V., Sandeep, R.: Enabling hyper-personalisation: automated ad creative generation and ranking for fashion e-commerce. In: Fashion Recommender Systems, pp. 25–48 (2020). Springer
254. Thomaidou, S., Lourentzou, I., Katsivelis-Perakis, P., Vazirgiannis, M.: Automated snippet generation for online advertising. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 1841–1844 (2013)
255. Zhang, X., Zou, Y., Zhang, H., Zhou, J., Diao, S., Chen, J., Ding, Z., He, Z., He, X., Xiao, Y., et al.: Automatic product copywriting for e-commerce. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 12423–12431 (2022)
256. Lei, Z., Zhang, C., Xu, X., Wu, W., Niu, Z.-Y., Wu, H., Wang, H., Yang, Y., Li, S.: Plato-ad: a unified advertisement text generation framework with multi-task prompt learning. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 512–520 (2022)
257. Bartz, K., Barr, C., Aijaz, A.: Natural language generation for sponsored-search advertisements. In: Proceedings of the 9th ACM Conference on Electronic Commerce, pp. 1–9 (2008)
258. Fujita, A., Ikushima, K., Sato, S., Kamite, R., Ishiyama, K., Tamachi, O.: Automatic generation of listing ads by reusing promotional texts. In: Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business, pp. 179–188 (2010)
259. Hughes, J.W., Chang, K.-h., Zhang, R.: Generating better search engine text advertisements with deep reinforcement learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2269–2277 (2019)
260. Wang, X., Gu, X., Cao, J., Zhao, Z., Yan, Y., Middha, B., Xie, X.: Reinforcing pretrained models for generating attractive text advertisements. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3697–3707 (2021)
261. Chen, C., Wang, X., Yi, X., Wu, F., Xie, X., Yan, R.: Personalized chit-chat generation for recommendation using external chat corpora. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2721–2731 (2022)
262. Zhang, C., Zhou, J., Zang, X., Xu, Q., Yin, L., He, X., Liu, L., Xiong, H., Dou, D.: Chase: commonsense-enriched advertising on search engine with explicit knowledge. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4352–4361 (2021)

263. Kanungo, Y.S., Negi, S., Rajan, A.: Ad headline generation using self-critical masked language model. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, pp. 263–271 (2021)
264. Wei, P., Yang, X., Liu, S., Wang, L., Zheng, B.: Creater: ctr-driven advertising text generation with controlled pre-training and contrastive fine-tuning. arXiv:2205.08943 (2022)
265. Kanungo, Y.S., Das, G., Negi, S.: Cobart: controlled, optimized, bidirectional and auto-regressive transformer for ad headline generation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3127–3136 (2022)
266. Chen, Q., Lin, J., Zhang, Y., Yang, H., Zhou, J., Tang, J.: Towards knowledge-based personalized product description generation in e-commerce. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3040–3050 (2019)
267. Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P.S., Sun, L.: A comprehensive survey of ai-generated content (aigc): a history of generative ai from gan to chatgpt. arXiv:2303.04226 (2023)
268. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural. Inf. Process. Syst. 27 (2014)
269. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv:1312.6114 (2013)
270. Dinh, L., Krueger, D., Bengio, Y.: Nice: non-linear independent components estimation. arXiv:1410.8516 (2014)
271. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
272. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations
273. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
274. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PMLR
275. Chen, M., Tan, X., Li, B., Liu, Y., Qin, T., Liu, T.-Y., et al.: Adaspeech: adaptive text to speech for custom voice. In: International Conference on Learning Representations
276. Li, X., Taheri, A., Tu, L., Gimpel, K.: Commonsense knowledge base completion. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (vol. 1: long papers), pp. 1445–1455 (2016)
277. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., et al.: Codebert: a pre-trained model for programming and natural languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1536–1547 (2020)
278. OpenAI: ChatGPT: A Large-Scale Generative Model for Conversation. OpenAI Blog (2020)
279. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning, pp. 8821–8831 (2021). PMLR
280. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.d.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al.: Evaluating large language models trained on code. arXiv:2107.03374 (2021)
281. Midjourney: Midjourney. Retrieved from. https://midjourney.com (2022)
282. Wang, W., Lin, X., Feng, F., He, X., Chua, T.-S.: Generative recommendation: towards next-generation recommender paradigm. arXiv:2304.03516 (2023)
283. Borji, A.: A categorical archive of chatgpt failures. arXiv:2302.03494 (2023)
284. Carlini, N., Tramer, F., Wallace, E., et al.: Extracting training data from large language models. (2021)
285. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners
286. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: low-rank adaptation of large language models. In: International Conference on Learning Representations (2021)
287. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. arXiv:2305.14314 (2023)
288. Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al.: Constitutional ai: Harmlessness from ai feedback. arXiv:2212.08073 (2022)

Authors and Affiliations

Jin Chen[1] · Zheng Liu[2] · Xu Huang[3] · Chenwang Wu[3] · Qi Liu[3] · Gangwei Jiang[3] · Yuanhao Pu[3] · Yuxuan Lei[3] · Xiaolong Chen[3] · Xingmei Wang[3] · Kai Zheng[4] · Defu Lian[3] · Enhong Chen[3]

✉ Jin Chen
jinchen@ust.hk

Zheng Liu
zhengliu1026@gmail.com

Xu Huang
xuhuangcs@mail.ustc.edu.cn

Chenwang Wu
wcw1996@mail.ustc.edu.cn

Qi Liu
qiliu67@mail.ustc.edu.cn

Gangwei Jiang
gwjiang@mail.ustc.edu.cn

Yuanhao Pu
puyuanhao@mail.ustc.edu.cn

Yuxuan Lei
lyx180812@mail.ustc.edu.cn

Xiaolong Chen
chenxiaolong@mail.ustc.edu.cn

Xingmei Wang
xingmeiwang@mail.ustc.edu.cn

Kai Zheng
zhengkai@uestc.edu.cn

Defu Lian
liandefu@ustc.edu.cn

Enhong Chen
cheneh@ustc.edu.cn

1 Hong Kong University of Science and Technology, Hong Kong, China

2 Beijing Academy of Artificial Intelligence, Beijing, China

3 University of Science and Technology of China, Hefei, China

4 University of Electronic Science and Technology of China, Chengdu, China